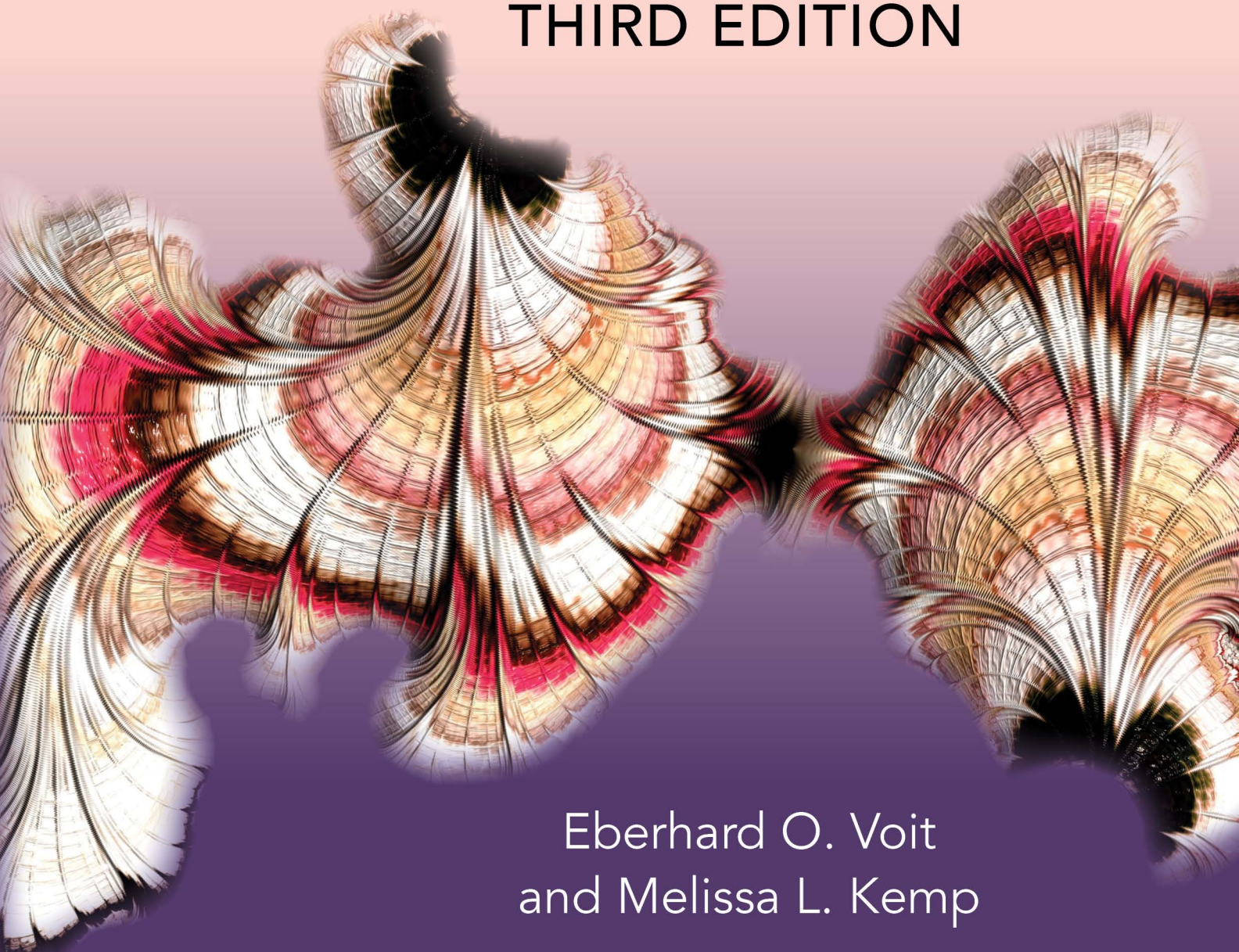
CRC Press
Taylor & Francis Group

SUPPORT MATERIAL

A FIRST COURSE IN SYSTEMS BIOLOGY

THIRD EDITION



Eberhard O. Voit
and Melissa L. Kemp

A FIRST COURSE IN SYSTEMS BIOLOGY

A First Course in Systems Biology, Third Edition is an introduction to the growing field of systems biology for advanced undergraduates and graduate students. Its focus is the design and analysis of computational models and their applications to diverse biomedical phenomena, from simple networks and kinetics to complex pathway systems, signal transduction, personalized medicine, and interacting populations. The book begins with the fundamentals of computational modeling, then reviews features of the molecular inventories that bring biological systems to life and ends with case studies that reflect some of the frontiers in systems biology. In this way, the *First Course* provides the reader with a comprehensive background and with access to methods for executing standard tasks of biomedical systems analysis, exposure to the modern literature, and a foundation for launching into specialized projects that address biomedical questions with theoretical and computational means.

This third edition has been thoroughly updated. It provides an introduction to agent-based and multiscale modeling, a deeper account of biological design principles, and the optimization of metabolic flux distributions. This edition also discusses novel topics of synthetic biology, personalized medicine, and virtual clinical trials that are just emerging on the horizon of this field.

Taylor & Francis
Taylor & Francis Group
http://taylorandfrancis.com

A FIRST COURSE IN SYSTEMS BIOLOGY

THIRD EDITION

Eberhard O. Voit and Melissa L. Kemp

CRC Press
Taylor & Francis Group
Boca Raton London New York

CRC Press is an imprint of the
Taylor & Francis Group, an **informa** business

A GARLAND SCIENCE BOOK

Designed cover image: OLaLa Merkel/Shutterstock

Third edition published 2025
by CRC Press
2385 NW Executive Center Drive, Suite 320, Boca Raton FL 33431

and by CRC Press
4 Park Square, Milton Park, Abingdon, Oxon, OX14 4RN

CRC Press is an imprint of Taylor & Francis Group, LLC

© 2025 Eberhard O. Voit and Melissa L. Kemp

Reasonable efforts have been made to publish reliable data and information, but the author and publisher cannot assume responsibility for the validity of all materials or the consequences of their use. The authors and publishers have attempted to trace the copyright holders of all material reproduced in this publication and apologize to copyright holders if permission to publish in this form has not been obtained. If any copyright material has not been acknowledged please write and let us know so we may rectify in any future reprint.

Except as permitted under U.S. Copyright Law, no part of this book may be reprinted, reproduced, transmitted, or utilized in any form by any electronic, mechanical, or other means, now known or hereafter invented, including photocopying, microfilming, and recording, or in any information storage or retrieval system, without written permission from the publishers.

For permission to photocopy or use material electronically from this work, access www.copyright.com or contact the Copyright Clearance Center, Inc. (CCC), 222 Rosewood Drive, Danvers, MA 01923, 978–750–8400. For works that are not available on CCC please contact mpkbookspermissions@tandf.co.uk

Trademark notice: Product or corporate names may be trademarks or registered trademarks and are used only for identification and explanation without intent to infringe.

Library of Congress Cataloging-in-Publication Data
Names: Voit, Eberhard O., author.
Title: A first course in systems biology / Eberhard Voit and Melissa Kemp.
Description: Third edition. | Boca Raton : CRC Press, 2025. | Includes bibliographical references and index.
Identifiers: LCCN 2024023636 (print) | LCCN 2024023637 (ebook) | ISBN 9781032515458 (hardback) | ISBN 9781032515434 (paperback) | ISBN 9781003402770 (ebook)
Subjects: LCSH: Systems biology. | Computational biology.
Classification: LCC QH324.2 .V65 2024 (print) | LCC QH324.2 (ebook) | DDC 570.285—dc23/eng/20241010
LC record available at https://lccn.loc.gov/2024023636
LC ebook record available at https://lccn.loc.gov/2024023637

ISBN: 978-1-032-51545-8 (hbk)
ISBN: 978-1-032–51543-4 (pbk)
ISBN: 978-1-003-40277-0 (ebk)

DOI: 10.1201/9781003402770

Typeset in Utopia Std
by Apex CoVantage, LLC

Access the Support Materials: https://www.routledge.com/9781032515434

To Ann and Charlie,
The Hubs of Our Support Systems

Contents

About the Authors xi
Preface xii

Chapter 1: Biological Systems 1
1.1 Reductionism and Systems Biology 6
1.2 Even Simple Systems Can Confuse Us 8
1.3 Why Now? 10
1.4 Communicating Systems Biology 13
1.5 The Task before Us 16
1.6 Exercises 17
Further Reading 18
References 18

Chapter 2: Introduction to Mathematical Modeling 21
2.1 Goals, Inputs, and Initial Exploration 26
2.1.1 Questions of Scale 26
2.1.2 Data Availability 28
2.2 Model Selection and Design 28
2.2.1 Model Structure 29
2.2.2 System Components 35
2.2.3 Model Equations 38
2.2.4 Parameter Estimation 39
2.3 Model Analysis and Diagnosis 41
2.3.1 Consistency and Robustness 42
2.3.2 Exploration and Validation of Dynamical Features 44
2.4 Model Use and Applications 46
2.4.1 Model Extensions and Refinements 47
2.4.2 Large-Scale Model Assessments 49
2.4.3 Questions of Design and Operation 51
2.4.4 Simplicity versus Complexity 53
2.5 Exercises 55
Further Reading 57
References 57

Chapter 3: Static Network Models 59
3.1 Strategies of Analysis 60
3.2 Interaction Graphs 61
3.2.1 Properties of Graphs 63
3.2.2 Small-World Networks 67
3.2.3 Stoichiometric Networks 69
3.3 Network Dynamics 71
3.3.1 Boolean Networks 72
3.3.2 Network Motifs 74
3.4 Dependencies among Network Components 77
3.4.1 Causality Analysis 77
3.4.2 Mutual Information 77
3.5 Bayesian Reconstruction of Interaction Networks 78
3.5.1 Application to Signaling Networks 81
3.5.2 Applications to Other Biological Networks 84
3.6 Exercises 85
Further Reading 89
References 89

Chapter 4: Discrete Biological Systems 93
4.1 Deterministic Recursive Models 94
4.1.1 Linear Recursions 94
4.1.2 Case Study: Red Blood Cell Dynamics 95
4.1.3 Nonlinear Recursions 97
4.1.4 Stability of a Steady State 99
4.2 Stochastic Recursive Models 102
4.2.1 Random Walks 102
4.2.2 Probability Trees 103
4.2.3 Markov Models 103
4.2.4 Randomness in Dynamical Systems 108
4.3 Exercises 109
Further Reading 111
References 111

Chapter 5: Continuous Biological Systems 113
5.1 Linear and Linearized ODE Systems 114
5.1.1 Introductory Examples 114
5.1.2 Systems of Linear ODEs 116
5.1.3 Linearized ODE Models 116
5.2 Steady-State Analysis 123
5.2.1 Determination of Steady States 123
5.2.2 Local Stability Analysis 127
5.2.3 Sensitivity Analysis 130
5.3 Dynamics of Nonlinear Models 133
5.3.1 Ad Hoc Models 133
5.3.2 Canonical Models 134
5.3.3 Lotka–Volterra Models 135
5.3.4 Generalized Mass Action (GMA) Models 136
5.3.5 S-System Models 138
5.4 Dynamic Motifs 147
5.5 Attractors of Dynamical Systems 150
5.5.1 Limit Cycles 150
5.5.2 Chaotic Attractors 154
5.6 Exercises 157
Further Reading 161
References 162

Chapter 6: Optimal Models 165
6.1 Parameter Estimation 166
6.2 Parameter Estimation for Linear Systems 167
6.2.1 Linear Regression Involving a Single Variable 167
6.2.2 Linear Regression Involving Several Variables 169
6.3 Parameter Estimation for Nonlinear Systems 172
6.3.1 Comprehensive Grid Search 174
6.3.2 Gradient Methods 176
6.3.3 Genetic Algorithms 177
6.3.4 Other Stochastic Algorithms 179
6.3.5 Typical Challenges 182
6.4 Parameter Estimation for Systems of Differential Equations 185
6.4.1 Parameter Estimation from Time-Series Data 186
6.5 Identification of the Most Appropriate Functional Representations 194
6.6 Discovery of the Optimal Structure 197
6.6.1 Design Principles 197
6.6.2 Operating Principles 200
6.7 Exercises 200
Further Reading 209
References 210

Chapter 7: Gene Systems 213
7.1 The Central Dogma 213
7.2 Key Properties of DNA and RNA 215
7.2.1 Chemical and Physical Features 215
7.2.2 Size and Organization of DNA 217
7.2.3 Genes and Noncoding DNA 218
7.2.4 Eukaryotic DNA Packing 221
7.2.5 Epigenetics 223
7.3 RNA 224
7.3.1 Messenger RNA (mRNA) 224
7.3.2 Transfer RNA (tRNA) 224
7.3.3 Ribosomal RNA (rRNA) 226
7.3.4 Small RNAs 226
7.3.5 RNA Viruses 228
7.4 Gene Regulation 228
7.4.1 The *lac* Operon 229
7.4.2 Modes of Regulation 231
7.4.3 Transcription Factors 231
7.5 Mathematical Models of Gene Expression and Gene Regulation 233
7.5.1 Generic Models 233
7.5.2 Models of Genetic Switches 236
7.5.3 Models of Genetic Response Circuits 237
7.6 Synthetic Biology 239
7.6.1 Artificial Gene Circuits 241
7.6.2 Case Study 244
7.7 Outlook 245
7.8 Exercises 246
Further Reading 249
References 250

Chapter 8: Protein Systems 255
8.1 Chemical and Physical Features of Proteins 256
8.2 Experimental Assessments of Proteins 260
8.2.1 Protein Structure 260
8.2.2 What Is There? 262

8.2.3 Localization 263
8.2.4 Interactions Involving Proteins 263
8.3 An Incomplete Survey of the Roles and Functions of Proteins 264
8.3.1 Enzymes 265
8.3.2 Transporters and Carriers 267
8.3.3 Signaling and Messenger Proteins 270
8.3.4 Proteins of the Immune System 271
8.3.5 Structure Proteins 272
8.4 Modeling Protein Activity and Dynamics 274
8.5 Exercises 278
Further Reading 281
References 281

Chapter 9: Metabolic Systems 285
9.1 Biochemical Reactions 286
9.1.1 Background 286
9.1.2 Mathematical Formulation of Elementary Reactions 288
9.1.3 Rate Laws 289
9.2 Metabolic Pathways and Pathway Systems 296
9.2.1 Biochemistry and Metabolomics 296
9.2.2 A Very Brief Synopsis of Methods for the Generation of Metabolome Data 297
9.3 Databases for Modeling Metabolic Systems 298
9.4 Control of Pathway Systems 300
9.4.1 Metabolic Control Analysis 302
9.5 Computational Analysis of Dynamic Pathway Systems 303
9.5.1 The General Stoichiometric Model 303
9.5.2 Analyses at the Steady State 306
9.5.3 Flux Balance Analysis (FBA) 307
9.5.4 Elementary Mode Analysis in Metabolic Engineering 310
9.5.5 Optimization of Dynamic Systems in Metabolic Engineering 313
9.6 Inference of Metabolic Systems 314
9.6.1 Metabolic Network Reconstruction 314
9.6.2 Extraction of Dynamic Models from Experimental Metabolic Data 316
9.7 Exercises 316
Further Reading 319
References 319

Chapter 10: Signaling Systems 323
10.1 Differential Equation Models of Signal Transduction Systems 325
10.1.1 Bistability 325
10.1.2 Hysteresis 326
10.2 Two-Component Signaling Systems 329
10.2.1 Background 329
10.2.2 Mathematical Models of TCS Systems 331
10.3 Mitogen-Activated Protein Kinase (MAPK) Cascades 333
10.3.1 Features of MAPK Cascades 333
10.3.2 Mathematical Models of MAPK Cascades 335
10.4 Adaptation 337
10.5 Other Signaling Systems 338
10.5.1 Circadian Clocks 339
10.5.2 Quorum Sensing 341
10.5.3 Cell-Free Systems 345
10.6 Exercises 346
Further Reading 349
References 350

Chapter 11: Multiscale Modeling and Agent-Based Simulations 353
11.1 The Challenge of Multiple Scales 353
11.1.1 Case Study: How Yeast Deals with Heat 354
11.1.2 Models of Some Aspects of the Heat Stress Response 357
11.1.3 Case Study: SIR Modeling Revisited 361
11.1.4 Accounting for the Host Response 362
11.1.5 Design of a Two-Scale Model 362
11.2 Agent-Based Modeling 365
11.2.1 Agent-Based Modeling in Netlogo 365
11.2.2 More Complicated Spatial Aspects in Agent-Based Models 370
11.3 Outlook for Multiscale Modeling 372
11.4 Exercises 373
Further Reading 374
References 374

Chapter 12: Physiological Modeling: The Heart as an Example 377

12.1 Hierarchy of Scales and Modeling Approaches 379
12.1.1 Basics of Heart Anatomy 379
12.1.2 Modeling Targets at the Organ Level 380
12.1.3 Modeling Targets at the Tissue Level 382
12.1.4 Modeling Targets at the Cell Level 383
12.2 Simple Models of Oscillations 385
12.2.1 Black-Box Models of Oscillations 386
12.2.2 Summary of Black-Box Oscillation Models 389
12.2.3 From a Black Box to Meaningful Models 389
12.3 Electrochemistry in Cardiomyocytes 391
12.3.1 Biophysical Description of Electrochemical Processes at the Membrane of Cardiomyocytes 394
12.3.2 Resting Potentials and Action Potentials 395
12.3.3 Models of Action Potentials 397
12.3.4 Repeated Heartbeats 400
12.4 Issues of a Failing Heart 401
12.4.1 Modeling Heart Function and Failure Based on Molecular Events 403
12.5 Exercises 408
Further Reading 411
References 411

Chapter 13: Systems Biology in Medicine and Drug Development 415

13.1 Are You Unique? 415
13.1.1 Biological Variability and Disease 415
13.1.2 Modeling Variability and Disease 416
13.2 Personalized Medicine and Predictive Health 419
13.2.1 Data Needs and Biomarkers 420
13.2.2 Personalizing Mathematical Models 421
13.3 The Drug Development Process 425
13.4 The Role of Systems Biology in Drug Development 427
13.4.1 Computational Target and Lead Identification 428
13.4.2 Receptor Dynamics 429
13.4.3 Pharmacokinetic Modeling 432
13.4.4 Pathway Screening with Dynamical Models 437
13.4.5 Virtual Clinical Trials 439
13.4.6 Emerging Roles of Systems Biology in Drug Development 440
13.5 Exercises 440
Further Reading 442
References 442

Chapter 14: Population Systems 445

14.1 Population Growth 445
14.1.1 Traditional Models of Population Growth 446
14.1.2 More Complex Growth Phenomena 448
14.2 Population Dynamics under External Perturbations 450
14.3 Analysis of Subpopulations 451
14.4 Interacting Populations 454
14.4.1 General Modeling Strategy 454
14.4.2 Phase-Plane Analysis 454
14.4.3 Mixed Communities of Populations 460
14.5 Exercises 462
Further Reading 465
References 465

Chapter 15: Emerging Topics in Systems Biology 467

15.1 Overall Long-Term Goals of Systems Biology 468
15.2 Emerging Applications 468
15.2.1 Models of Cells, Organs, and Organisms 468
15.2.2 Single-Cell Analysis 470
15.2.3 Infection, Inflammation, and Sepsis 471
15.2.4 Cancer 475
15.2.5 Other Complex Diseases 476
15.2.6 Neurons, Brains, and Neurodegenerative Disease 476
15.2.7 Quantitative Systems Pharmacology 479
15.2.8 Organoids for Personalized Medicine and Drug Screening 480

15.2.9 Organisms and Their Interactions with the Environment 480
15.3 Modeling Needs 482
15.3.1 Data-Modeling Pipelines 484
15.3.2 The Role of Machine Learning and Artificial Intelligence in Systems Biology 485
15.4 Toward a Theory of Biology . . . Or Several Theories? 488
Further Reading 490
References 490

Glossary 495
Index 509

About the Authors

Eberhard O. Voit received master's degrees in biology and mathematics and the degree of Dr. rer. nat., with emphasis in developmental and theoretical biology, from the University of Cologne, Germany. He held research and faculty positions at the University of Cologne; the University of Michigan; the Medical University of South Carolina in Charleston; the Cooperative Research Centre for Forestry in Tasmania, Australia; and he was the David D. Flanagan Chair in Systems Biology at Georgia Tech and Emory before joining the University of Texas at Dallas in 2024. Dr. Voit's research interests are in the area of complex biomedical systems. Work in his lab over the years has focused on genomic, metabolic, and signaling systems with applications reaching from microbial and plant systems to human diseases. On the methodological side, Dr. Voit has been a leader in the development and application of biochemical systems theory (BST). He has authored or co-authored over three hundred scientific articles and chapters as well as several books, some of which were translated into Chinese, Korean, and Greek. Dr. Voit is a professor emeritus at Georgia Tech and an elected fellow in the American Institute for Medical and Biological Engineering (AIMBE), the American Association for the Advancement of Science (AAAS), and the Society of Mathematical Biology (SMB). He was furthermore inducted into the Royal Society of Medicine of the United Kingdom as an overseas fellow and named a notable alumnus of the University of Cologne.

Melissa L. Kemp received her BS in nuclear engineering, minoring in biomedical engineering, from Massachusetts Institute of Technology (MIT) and her PhD in bioengineering from the University of Washington. Following postdoctoral work as a CSBi/Merck fellow in biological engineering at MIT, she joined the faculty of the Wallace H. Coulter Department of Biomedical Engineering at Georgia Institute of Technology and Emory University. As a Carol Ann and David D. Flanagan Professor and Georgia Cancer Coalition Distinguished Scholar, Dr. Kemp's research program focuses on modeling metabolic and signaling systems with applications in immunology, cancer, and regenerative medicine.

Preface

About a dozen years have passed since the first edition of this text was published. A blink of an eye from the perspective of science—but what a tremendous time span for systems biology! Twelve years ago, we marveled at the new, unprecedented methods of large-scale, high-throughput molecular biology. We were astounded by the rapid advances in the power of computers and by new, easily accessible and user-friendly software. We wondered what we might be able to do with all the new data—or whether we might be drowned by their anticipated enormity. We asked ourselves how we could effectively employ the new computing prowess in efforts to tackle some of the grand challenges facing the world, from food availability and sustainable environments to improvements in the quality of life and the treatment of dreaded diseases. At the same time, we were dismayed that systems biology was still the domain of just a few experts and that the mainstream of science was by and large ignoring the concepts and promises of the field. We almost sheepishly hoped for some miraculous emergence of systems biology—out of the shadows and right into the limelight of wider acceptance.

Fast forward, and these thoughts now seem quaint and almost irrational. How could we be so faint of heart? Why were we doubtful, despite our strong conviction that systems thinking eventually had to become a necessity in biology and medicine? So, here we are. It is hard to imagine a biomedical or bioengineering conference without a session or track dedicated to systems biology. Dozens of books have been published, including several introductions and even a few treatises for the educated lay public. Systems biology has become a core topic for a slew of old and new journals. Many colleges and universities are now embracing the need for systems biology and have created modules, courses, or even entire programs. Systems biology has become a household term.

And what about the goals and aspirations we pondered a decade ago? Not surprisingly, many of the old challenges remain, simply because they are tough; others were added, and as soon as we had solved a problem, the solution spawned new hypotheses and forays into uncharted directions. Today, we are especially intrigued by how former questions about computational capacities suddenly morphed into admiration and concern about the power and unfathomed potential of artificial intelligence. For sure, AI will affect many aspects of research, information handling, and teaching, but its disruptive power, scope, and magnitude are daunting and unforeseeable. AI is disconcerting to many, which is understandable, but it will not go away. It will be our mandate not to be afraid but to use AI boldly for the good and, within our specific realm, for the advancement of systems biology. In turn, the principles of systems modeling and analysis will be needed for the advancement of interpretable AI in biology and medicine, for the unpacking of "black box" algorithms and their results. Pattern recognition and feature extraction from data, based on machine learning and AI, will ultimately complement and enhance systems biology, not supplant it.

Today's chatbots readily write summaries of scientific concepts and even highly technical topics that are quite elegant, although they are not always

entirely correct, and they can perform relatively sophisticated mathematical tasks. Will they soon be able to design computational models from scratch when provided with a biomedical research question? Will they enable us to extract pertinent information from published literature and from databases more easily than is possible today? Will AI be able to distinguish true information from noise? Will pipeline algorithms emerge that autonomously extract and clean raw data, identify outliers, use information that has been efficiently and objectively cleansed to create novel hypotheses, and then test these automatically with the most appropriate models? How effective will machine learning be for exploring parameter spaces and selecting mechanistic models that capture complex behavior? Will AI help us design and implement biomedical simulators with the same sophistication as the gaming industry, thereby taming the steep learning curve of systems analysis and democratizing computer power among wide swaths of biologists and others?

These questions are clearly very different from those we asked a decade ago, but they are just as exciting, challenging, and thought-provoking. Progress toward answering them will add new tools to our experimental and computational arsenal, and while new answers will always lead to new questions, we will, step by step, achieve a greater understanding of the fascinating complexity of the inner workings of life. This understanding will in turn empower us to devise targeted manipulations of existing or even de novo-created organisms and bring us closer to solving some of the pressing issues our world faces.

So, let's once again embark on a journey into the unknown, hone our skills, and get ready to respond effectively to grand challenges. Obviously, this book will not reach such lofty goals per se, but it offers a solid starting point. It introduces a basis and a steppingstone for newcomers to learn and become enabled to do greater things in the future. We are enthusiastic about facilitating this first step and hope that our readers will share our excitement. We hope in particular that the presented material will inspire new generations of young, inquisitive minds to search for more, crush the boundaries of knowledge, and eventually achieve solutions that seem impossible today.

Eberhard O. Voit and Melissa L. Kemp
Georgia Institute of Technology
University of Texas at Dallas
2024

Taylor & Francis
Taylor & Francis Group
http://taylorandfrancis.com

Biological Systems

1

When you have read this chapter, you should be able to:

- Describe the generic features of biological systems
- Discuss the complexity of living systems
- Explain the goals of systems biology
- Identify the complementary roles of reductionism and systems biology
- List those challenges of systems biology that cannot be solved with intuition alone
- Discuss the importance of transdisciplinary communication
- Assemble a "to-do" list for the field of systems biology

What is cause, what is effect,
If the effect causes the cause to change?
Where effect and cause, and cause and effect,
Are entangled in a complex web,
The mind is lost
And ponders in vain
The principal cause of all.

When we think of biological systems, our minds may immediately wander to the Amazon rainforest, brimming with thousands of plants and animals that live with each other, compete with each other, and depend on each other. We might think of the incredible expanse of the world's oceans, of colorful fish swimming through coral reefs, nibbling on algae. Two-meter-high African termite mounds may come to mind, with their huge colonies of individuals that have their specific roles and whose lives are controlled by an intricate social structure (**Figure 1.1**). We may think of an algae-covered pond with tadpoles and minnows that are about to restart yet another life cycle.

These examples are indeed beautiful manifestations of some of the fascinating systems nature has evolved. However, we don't have to look that far to find biological systems. Much, much smaller systems are in own bodies and even within our cells. Kidneys are waste-disposal systems. Mitochondria are energy-production systems. Ribosomes are intracellular machines that make proteins from amino acids. Bacteria are complicated biological systems in the form of single cells that can execute all functions needed for their survival and for the survival of their species. Viruses interact with cells in a very well-controlled, systemic way. Even seemingly modest tasks within cells or organisms often involve amazingly large numbers of processes that form complicated control systems (**Figure 1.2**). The more we learn about the most basic processes of life, such as cell division or the production of a metabolite, the more we should marvel the incredible complexity of the systems

DOI: 10.1201/9781003402770-1

Figure 1.1 Biological systems abound at all size scales. Here, a termite mound in Namibia is visible evidence of a complex social system. This system is part of a larger ecological system, and it is at once the host to many systems at smaller scales. (Courtesy of Lothar Herzog under the Creative Commons Attribution 2.0 Generic license.)

that facilitate these processes. In our daily lives, we usually take these systems for granted and assume that they function adequately, and it is only when, for example, disease strikes or algal blooms kill fish, that we realize how complex biology really is and how damaging the failure of just a single component can be.

Our ancestors have been aware of biological systems since the beginning of human existence. Human birth, development, health, disease, and death have long been recognized as interwoven with those of plants and animals and with the environment. For our forebears, securing food required an understanding of seasonal changes in the ecological systems of their surroundings. Even the earliest forays into agriculture depended on detailed concepts and ideas of when and what to plant, how and where to plant it, how many seeds to eat or to save for sowing, and when to expect returns on the investment. Several thousand years ago, Egyptians

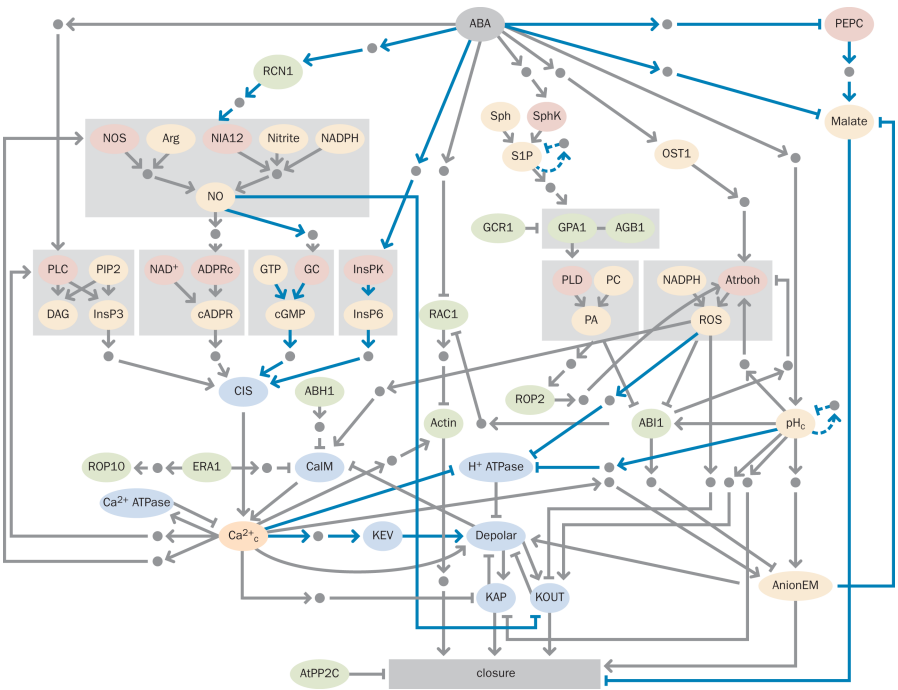


Figure 1.2 Diagram of a complicated system of molecules that coordinate the response of plants to drought. While the details are not important here, we can see that a key hormone, called abscisic acid (ABA), triggers a cascade of reactions that ultimately promote the closure of stomata and thereby reduce water evaporation [1]. Even a narrowly defined response like this closure process involves a complicated control system that contains a multitude of molecules and their interactions. In turn, this system is just one component within a much larger, physiological stress response system (cf. Figure 1.7). (From Saadatpour A, Albert I & Albert A. *J Theor Biol.* 266 [2010] 641–656. With permission from Elsevier.)

managed to ferment sugars to alcohol and used the mash to bake bread. Early pharmaceutical treatments of diseases certainly contained a good dose of superstition, and we are no longer convinced that rubbing on the spit of a toad during full moon will cure warts. Yet, the beginnings of pharmaceutical science in antiquity and the Middle Ages also demonstrate a growing recognition that particular plant products can have significant and specific effects on the well-being or malfunctioning of the systems within the human body.

Despite our long history of dealing with biological systems, our mastery of engineered systems far outstrips our capability to manipulate biological systems. We send spaceships successfully to faraway places and predict correctly when they will arrive and where they will land. We build skyscrapers exceeding the sizes of the biggest animals and plants by hundreds of times. Our airplanes are faster, bigger, and more robust against turbulence than the most skillful birds. Yet, we cannot create new human cells or tissues from basic building blocks, and we are seldom able to cure diseases except with rather primitive methods like cutting into the body or killing a lot of healthy tissue in the process, trusting that the body will heal itself afterwards. We can anticipate that our grandchildren will only shake their heads at such medieval-sounding, draconian measures. We have learned to create improved microorganisms, for instance for the bulk production of industrial alcohol or the generation of pure amino acids, but the methods for doing so rely on bacterial machinery that we do not fully understand and on artificially induced random mutations rather than targeted design strategies.

Before we discuss the root causes of the many challenges associated with understanding and manipulating biological systems in a targeted fashion—and our problems predicting what biological systems will do under yet-untested conditions—we should ask whether the goal of a deeper understanding of biological systems is even worth the effort. The answer is a resounding "Yes!" In fact, it is impossible even to imagine the potential and scope of advances that might develop from biological systems analyses. Just as nobody during the eighteenth century could have foreseen the ramifications of the Industrial Revolution or of electricity, the Biological Revolution is ushering in an entirely new world with incredible possibilities. Applications that are already emerging on the horizon are personalized medical treatments with minimal side effects, medications that will let the body regain control over a tumor that has run amok, prevention and treatment of neurodegenerative diseases, and the creation of spare organs from reprogrammed stem cells. A better understanding of ecological systems will yield pest- and drought-resistant food sources, as well as means for restoring polluted soil and water. It will help us understand why certain species are threatened and what could be done effectively to counteract their decline. Deeper insights into aquatic systems will lead to cleaner water and sustainable fisheries. Reprogrammed microbes or nonliving systems composed of biological components will dominate the production of chemical compounds from prescription drugs to large-scale industrial organics and might create energy sources without equal. Modified viruses will become standard means for supplying cells with healthy proteins or replacement genes. The rewards of discovering and characterizing the general principles and the specifics of biological systems will truly be unlimited.

If it is possible to engineer very sophisticated machines and to predict exactly what they will do, why are biological systems so different and difficult? One crucial difference is that we have full control over engineered systems but not over biological systems. As a society, we collectively know all details of all parts of engineered machines, because we made them. We know their properties and functions, and we can explain how and why some engineer put a machine together in a particular fashion. Furthermore, most engineered systems are modular, with each module being designed for a unique, specific task. While these modules interact with each other, they seldom have multiple roles in different parts of the system, in contrast to biology and medicine, where, for instance, the same lipids can be components of membranes and have complicated signaling functions and where diseases are often not restricted to a single organ or tissue but may affect the immune system and lead to changes in blood pressure and blood chemistry that secondarily cause kidney and heart problems. A chemical refinery looks overwhelmingly complicated to a layperson, but for an industrial engineer, every piece has a specific, well-defined role within the refinery, and every piece or module has properties

Figure 1.3 Analyzing a biological system resembles the task of determining the function of a complicated machine that we have never seen before. Shown here as an example is the cesium fountain laser table of the United States Naval Observatory, which is used to measure time with extreme accuracy. This atomic clock is based on transitions in cesium, which have a frequency of 9,192,631,770 Hz and are used to define the second. See also [2].

that were optimized for this role. Moreover, should something go wrong, the machines and factories will have been equipped with sensors and warning signals pinpointing problems as soon as they arise and allowing corrective action.

In stark contrast to dealing with sophisticated, well-characterized engineered systems, the analysis of biological systems requires investigations in the opposite direction. This type of investigation resembles the task of looking at an unknown machine and predicting what it does (**Figure 1.3**). Adding to this challenge, all scientists collectively know only a fraction of the components of biological systems, and the specific roles and interactions among these components are often obscure and change over time. Even more than engineered systems, biological systems are full of sensors and signals that indicate smooth running or ensuing problems, but in most cases our experiments cannot directly perceive and measure these signals and we can only indirectly deduce their existence and function. We observe organisms, cells, or intracellular structures as if from a large distance and must deduce from rather coarse observations how they might function or why they fail.

What exactly is it that makes biological systems so difficult to grasp? It is certainly not just size. **Figure 1.4** shows two networks. One shows the vast highway system of the continental United States, which covers several million miles of major highways. It is a very large system, but it is not difficult to understand its function or malfunction: if a highway is blocked, it does not take much ingenuity to figure out how to circumvent the obstacle. The other network is a comparably tiny system: the web of a diadem spider. While we can observe the process and pattern with which Ms. Spider spins her web, we do not know which neurons in her brain are responsible for different phases of the complicated web production process and how she is able to produce the right chemicals for the spider silk, which in itself is a marvel of material science, let alone how she manages to survive, multiply, and maybe even devour her husband.

Biological systems often consist of large numbers of components, but they pose an additional, formidable challenge to any analysis because the processes that govern them are not linear. This is a problem, because we are trained to think in linear ways: if an investment of $100 leads to a return of $120, then an investment of $10,000 leads to a return of $12,000. Biology is different. If we fertilize our roses with 1 tablespoon of fertilizer and the rose bushes produce 50 blossoms, a little bit more fertilizer may increase the number of blossoms, but 100 tablespoons of fertilizer will not produce 5000 blossoms but almost certainly kill the plants (**Figure 1.5**). Just a small amount of additional sun exposure turns a suntan into

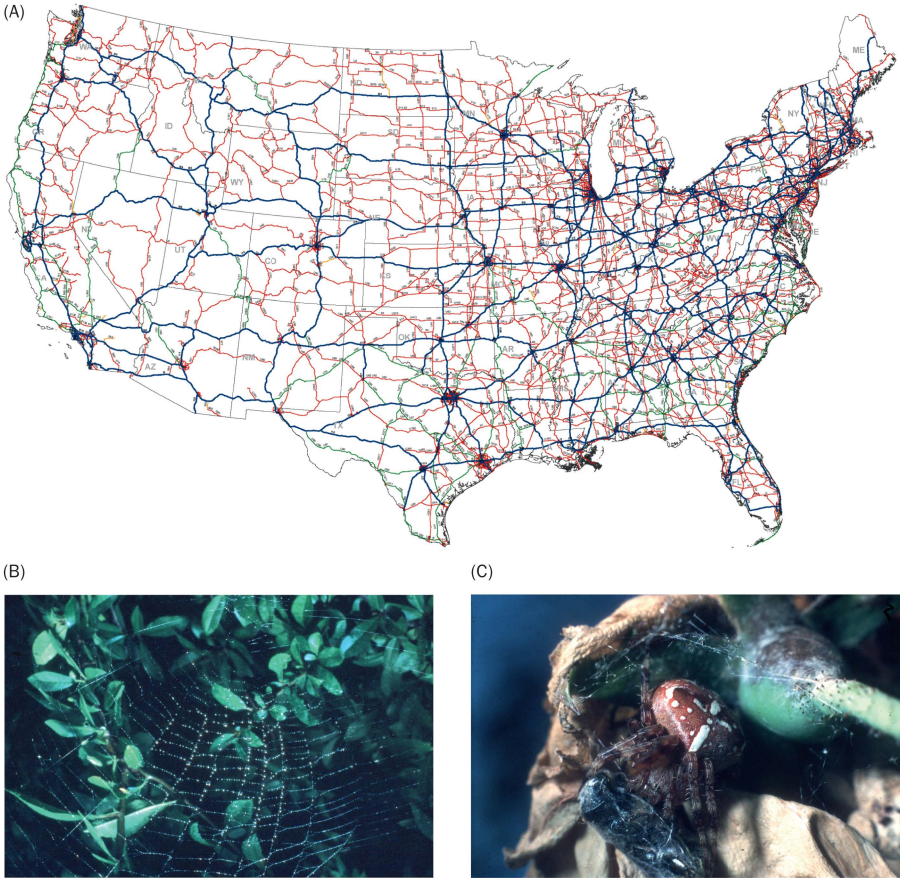


Figure 1.4 The size of a network or system is not necessarily correlated with its complexity. (A) The network of major highways in the continental United States covers over three million square miles. Nonetheless, its functionality is easy to grasp and problems with a particular road are readily ameliorated with detours. (B) The web of the European diadem spider (*Araneus diadematus*; C) is comparatively small, but the functional details of this little network are complex. Some lines are made of silk proteins that have the tensile strength of steel but can also be eaten and recycled by the spider; other lines are adhesive due to a multipurpose glue that may be sticky or rubbery depending on the situation; yet others are guide and signal lines that allow the spider to move about and sense prey. The creation of the web depends on different types of spinneret glands, whose development and function require the complex molecular machinery of the spider, and it is not yet clear how the instructions for the complicated construction, repair, and use of the web are encoded and inherited from one generation to the next. (A, From the United States Department of Transportation.)

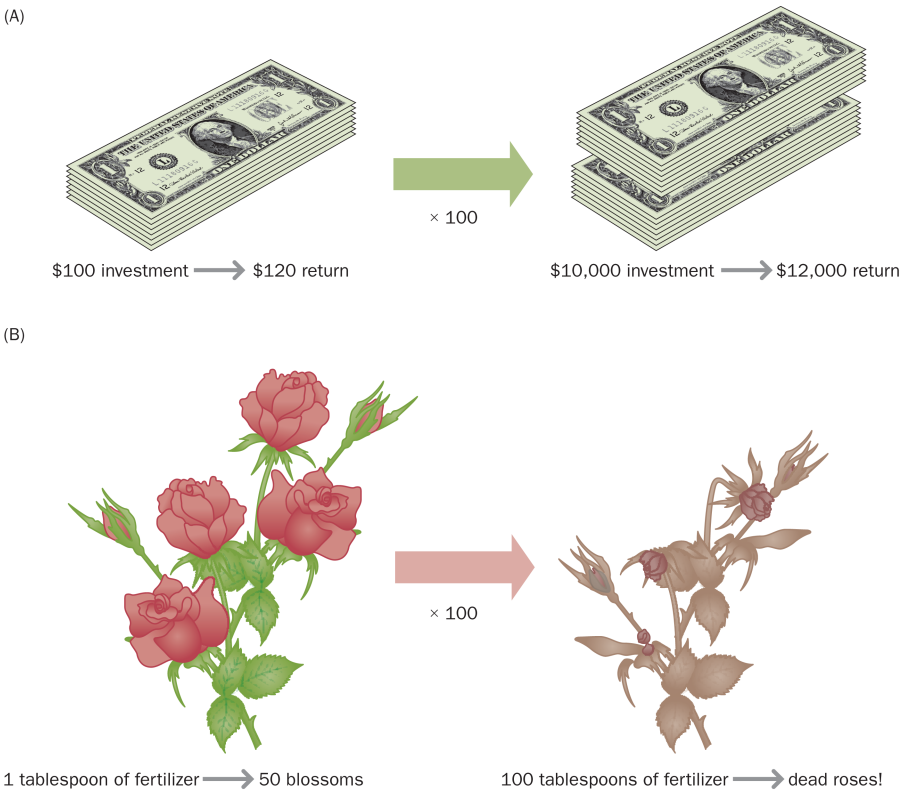


Figure 1.5 Biological phenomena are often difficult to understand, because our minds are trained to think linearly. (A) The return on an investment grows (or decreases) linearly with the amount invested. (B) In biology, more is not necessarily better. Biological responses often scale within a modest range but lead to an entirely different response if the input is increased a lot.

sunburn. Now imagine that thousands of components, many of which we do not know, respond in such a fashion, where a small input does not evoke any response, more input evokes a physiological response, and a little bit more input causes the component to fail or exhibit a totally different "stress" response. We will return to this issue later in this and other chapters with specific examples.

1.1 REDUCTIONISM AND SYSTEMS BIOLOGY

So, the situation is complicated. But because we humans are a curious species, our forebears did not give up on biological analysis and instead did what was doable, namely collecting information on whatever could be measured with the best current methods (**Figure 1.6**). By now, this long-term effort has resulted in an amazing list of biological parts and their roles. Initially, this list contained new plant and animal species, along with descriptions of their leaves, berries, and roots, or their body shapes, legs, and color patterns. These external descriptions were valuable but did not provide specific clues on how plants and animals function, why they live, and why they die. Thus, the next logical step was to look inside—even if this required stealing bodies from the cemetery under a full moon! Cutting bodies open revealed an entirely new research frontier. What were all those distinct body parts and what did they do? What were organs, muscles, and tendons composed of? Not surprisingly, this line of investigation eventually led to the grand-challenge quest of discovering and measuring *all* parts of a body, the parts of the parts (. . . of the parts), as well as their roles in the normal physiology or pathology of organisms, organs, and cells. The implicit assumption of this reductionist approach was that knowing the building blocks of life would lead us to a comprehensive understanding of how life works.

If we fast forward to the twenty-first century, have we succeeded and assembled a complete parts catalog? Do we know the building blocks of life? The answer is a combination of *Yes* and *No*. The catalog is most certainly not complete, even for relatively simple organisms. Yet, we have discovered and characterized genes, proteins, and metabolites as the major building blocks. Scientists were jubilant when the sequencing of the human genome in the early years of this Millennium was declared complete: we had identified the ultimate building blocks, our entire blueprint. It turned out to consist of roughly three billion nucleotide pairs of DNA.

The sequencing of the human genome was without any doubt an incredible achievement. Alas, there is much more to a human body than genes. So, the race for building blocks extended to proteins and metabolites, toward individual gene variations and an assortment of molecules and processes affecting gene expression, which changes in response to external and internal stimuli, during every day, throughout our lifetimes. As a direct consequence of these ongoing efforts, our parts list continues to grow at a rapid pace: a parts catalog that started with a few

Figure 1.6 Collecting information is the first step in most systems analyses. The eighteenth-century British explorer Captain James Cook sailed the Pacific Ocean and catalogued many plants and animal species that had never been seen before in Europe.

organs now contains over 20,000 human genes, many more genes from other organisms, and hundreds of thousands of proteins and metabolites along with their variants. In addition to merely looking at parts in isolation, we have begun to realize that most biological components are affected and regulated by a variety of other components. The expression of a gene may depend on several transcription factors, metabolites, and a variety of small RNAs, as well as molecular, epigenetic attachments to its DNA sequence. It is reasonable to expect that the list of processes within the body is much larger than the number of components on our parts list. Molecular biologists will not have to worry about job security any time soon!

The large number of components and processes alone, however, is not the only obstacle to understanding how cells and organisms function. After all, modern computers can execute gazillions of operations within a second. Our billions of telephones worldwide are functionally connected. We can make very accurate predictions regarding a gas in a container, even if trillions of molecules are involved. If we increase the pressure on the gas without changing the volume of the container, we know that the temperature will rise, and we can predict by how much. Not so with a cell or organism. What will happen to it if the environmental temperature goes up? Nothing much may happen; the rise in temperature may trigger a host of physiological response processes that compensate for the new conditions, or the organism may die. The outcome depends on a variety of factors that collectively constitute a complex stress response system (**Figure 1.7**). Of course, the comparison to a gas is not quite fair because, in addition to their large number, the components of a cell are not all the same, which drastically complicates matters. Furthermore, as mentioned earlier, the processes with which the components interact are nonlinear, and this permits an enormous repertoire of distinctly different behaviors with which an organism can respond to a perturbation.

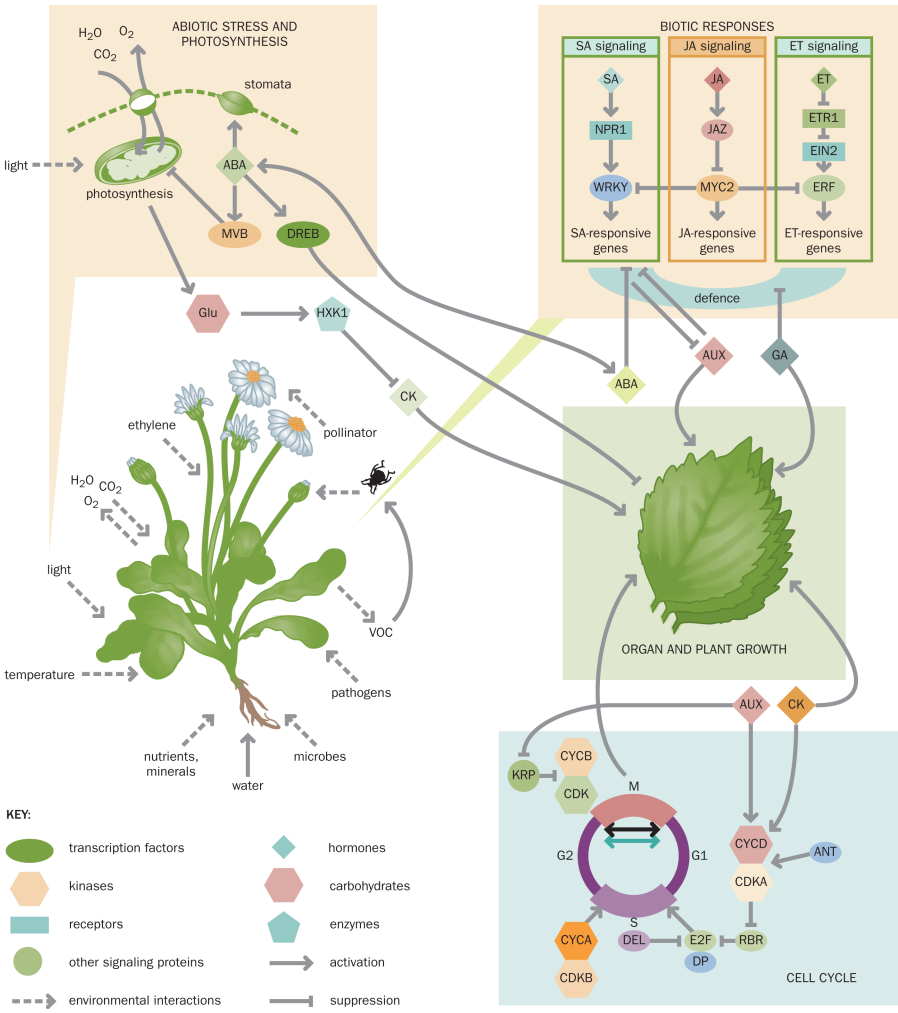


Figure 1.7 Stress responses are coordinated by systems at different levels of organization (cf. Figure 1.2). At the physiological level, the stress response system in plants includes changes at the cellular, organ, and whole-plant levels and also affects interactions of the plant with other species. (From Keurentjes JJB, Angenent GC, Dicke M, et al. *Trends Plant Sci.* 16 [2011] 183–190. With permission from Elsevier.)

1.2 EVEN SIMPLE SYSTEMS CAN CONFUSE US

It is easy to demonstrate how quickly our intuition can be overwhelmed by a few nonlinearities within a system. As an example, let's look at a simple chain of processes and compare it with a slightly more complicated chain that includes regulation [3]. The simple case merely consists of a chain of reactions, which is fed by an external input (**Figure 1.8**). It does not really matter what *X*, *Y*, and *Z* represent, but, for the sake of discussion, imagine a metabolic pathway such as glycolysis, where the input, glucose, is converted into glucose 6-phosphate, fructose 1,6-bisphosphate, and pyruvate, which is used for other purposes that are not of interest here. For illustrative purposes, let's explicitly account for an enzyme *E* that catalyzes the conversion of *X* into *Y*.

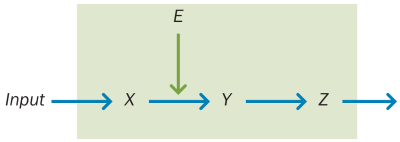


Figure 1.8 The human brain handles linear chains of causes and events very well. In this simple pathway, an external input is converted sequentially into *X*, *Y*, and *Z*, which leaves the system. The conversion of *X* into *Y* is catalyzed by an enzyme *E*. It is easy to imagine that any increase in *Input* will cause the levels of *X*, *Y*, and *Z* to rise.

We will learn in the following chapters how one can formulate a model of such a pathway system as a set of **differential equations**. And while the details are not important here, it does not hurt to show such a model, which might read

$$\begin{aligned} \dot{X} &= Input \times a\ EX^{0.5}, \\ \dot{Y} &= a\ EX^{0.5} \times bY^{0.5}, \\ \dot{Z} &= bY^{0.5} \times cZ^{0.5}. \end{aligned} \tag{1.1}$$

Here, *X*, *Y*, and *Z* are concentrations, *E* is the enzyme activity, and *a*, *b*, and *c* are rate constants that respectively represent how fast *X* is converted into *Y*, how fast *Y* is converted into *Z*, and how quickly material from the metabolite pool *Z* leaves the system. The dotted quantities on the left of the equal signs are differentials that describe the change in each variable over time, but we need not worry about them at this point. In fact, we hardly have to analyze these equations mathematically to get an idea of what will happen if we change the input, because intuition tells us that any increase in *Input* should lead to a corresponding rise in the concentrations of the intermediates *X*, *Y*, and *Z*, whereas a decrease in *Input* should result in smaller values of *X*, *Y*, and *Z*. The increases or decreases in *X*, *Y*, and *Z* will not necessarily be exactly of the same extent as the change in *Input*, but the direction of the change should be the same. The mathematical solution of the system in (1.1) confirms this intuition. For instance, if we reduce *Input* from 1 to 0.75, the levels of *X*, *Y*, and *Z* decrease, one after another, from their initial value of 1 to 0.5625 (**Figure 1.9**).

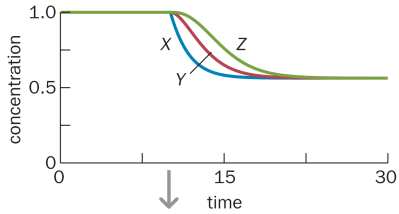


Figure 1.9 Simulations with the system in (1.1) confirm our intuition: *X*, *Y*, and *Z* reflect changes in Input. For instance, reducing *Input* in (1.1) to 75% at time 10 (arrow) leads to permanent decreases in *X*, *Y*, and *Z*.

Now suppose that *Z* is a signaling molecule, such as a hormone or a phospholipid, that activates a transcription factor *TF* that facilitates the upregulation of a gene *G* that codes for the enzyme catalyzing the conversion of *X* into *Y* (**Figure 1.10**). The simple linear pathway is now part of a functional loop. The organization of this loop is easy to grasp, but what is its effect? Intuition might lead us to believe that the positive-feedback loop should increase the level of enzyme *E*, which would result in more *Y*, more *Z*, and even more *E*, which would result in even more *Y* and *Z*. Would the concentrations in the system grow without end? Can we be sure about this prediction? Would an unending expansion be reasonable? What will happen if we increase or decrease the input as before?

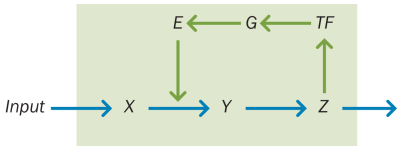


Figure 1.10 Even simple systems may not allow us to make reliable predictions regarding their responses to stimuli. Here, the linear pathway from Figure 1.8 is embedded into a functional loop consisting of a transcription factor *TF* and a gene *G* that codes for enzyme *E*. As described in the text, the responses to changes in *Input* are no longer obvious.

The overall answer will be surprising: the information given so far does not allow us to predict particular responses with any degree of reliability. Instead, the answer depends on the numerical specifications of the system. This is bad news for the unaided human mind, because we are simply not able to assess the numerical consequences of slight changes in a system, even if we can easily grasp the logic of a system as in Figure 1.10.

To get a feel for the system, one can compute a few examples with an expanded model that accounts for the new variables (for details, see [3]). Here, the results are more important than the technical details. If the effect of *Z* on *TF* is weak, the response to a decrease in *Input* is essentially the same as in Figure 1.9. This is not too surprising, because the systems in this case are very similar. However, if the effect of *Z* on *TF* is stronger, the concentrations in the system start to oscillate, and after a while these oscillations dampen away (**Figure 1.11A**). This behavior was not easy to predict. Interestingly, if the effect is further increased, the system enters a stable oscillation pattern that does not cease unless the system input is changed again (Figure 1.11B).

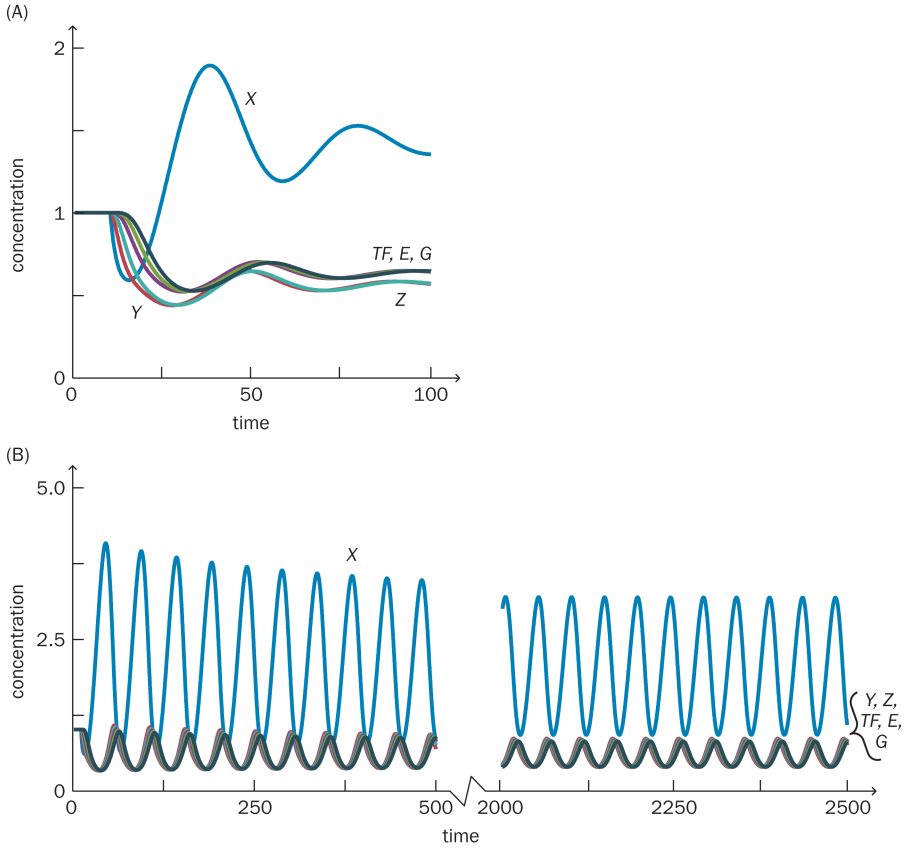


Figure 1.11 Simulation results demonstrate that the looped system in Figure 1.10 may exhibit drastically different responses. If the effect of *Z* on *TF* is very small, the response is essentially like that in Figure 1.9 (results not shown). (A) If the effect of *Z* on *TF* is relatively small, the functional feedback loop causes the system to go through damped oscillations before assuming a new stable state. (B) For stronger effects of *Z* on *TF*, the system response is a persistent oscillation.

The hand-waving explanation of these results is that the increased enzyme activity leads to a depletion of *X*. A reduced level of *X* leads to lower levels of *Y* and *Z*, which in turn lead to a reduced effect on *TF*, *G*, and ultimately *E*. Depending on the numerical characteristics, the ups and downs in *X* may not be noticeable, they may be damped and disappear, or they may persist until another change is introduced. Intriguingly, even if we know that these alternative responses are possible, the unaided human mind is not equipped to integrate the numerical features of the model in such a way that we can predict which system response will ensue for a specific setting of parameters. A computational model, by contrast, reveals the answer in a fraction of a second.

The specific details of the example are not as important as the take-home message: If a system contains regulatory signals that form functional loops, we can no longer rely on our intuition for making reliable predictions. Alas, essentially all realistic systems in biology are regulated—and not just with one but with many control loops. This leads to the direct and sobering deduction that intuition is not sufficient and that we instead need to utilize computational models to figure out how even small systems work and why they might show distinctly different responses or even fail, depending on the conditions under which they operate.

The previous sections have taught us that biological systems contain large numbers of different types of components that interact in potentially complicated ways and are controlled by regulatory signals. What else is special about biological systems? Many answers could be given, some of which are discussed throughout this book. For instance, two biological components are seldom 100% the same. They vary from one organism to the next and change over time. Sometimes these variations are inconsequential, at other times they lead to early aging and disease. In fact, most diseases do not have a single cause but are the consequence of an unfortunate combination of slight alterations in many components. Another feature that complicates intuition is the delay in many responses to stimuli. Such delays may be of the order of seconds, hours, or years, but they require the analyst to study not merely the present state of a biological system but also its history. For

instance, recovery from a severe infection depends greatly on the preconditioning of the organism, which is the collective result of earlier infections and the body's responses [4].

Finally, it should be mentioned that different parts of biological systems may simultaneously operate at different scales, with respect to both time and space. These scales make some aspects of their analysis easier and some harder. Let's begin with the temporal scale. We know that biology at the most basic level is governed by physical and chemical processes. These occur on timescales of the order of milliseconds, if not faster. Biochemical processes usually run on a scale of seconds to minutes. Under favorable conditions, bacteria divide every 20–30 minutes. Our human lifespan extends to maybe 120 years, evolution can happen at the genetic level with lightning speed, for instance, when radiation causes a mutation, while the emergence of an entirely new species may take thousands or even millions of years. On one hand, the drastically different timescales make analyses complicated, because we simply cannot account for rapid changes in all molecules of an organism over an extended period of time. As an example, it is impossible to study aging by monitoring an organism's molecular state every second or minute. On the other hand, the differences in timescales justify a very valuable modeling "trick" (Chapter 5 in [5]). If we are interested in understanding some biochemical process, such as the generation of energy in the form of adenosine triphosphate (ATP) by means of the conversion of glucose into pyruvate, we can assume that developmental and evolutionary changes are so slow in comparison that they do not change during ATP production. Similarly, if we study the phylogenetic family tree of a species, the biochemical processes in an individual organism are comparatively so fast that their details become irrelevant. Thus, by focusing on just the most relevant timescale and essentially ignoring much faster and much slower processes, any modeling effort is dramatically simplified.

Biology also happens on many spatial scales. All processes have a molecular component, and their size scale is therefore of the order of angstroms and nanometers. If we consider a cell as the basic unit of life, we are dealing with a spatial scale of micrometers to millimeters, with some exceptions such as cotton "fiber" cells reaching the length of a few centimeters [6] and the afferent axons of nerve cells in giraffes, reaching from toe to neck, extending to 5 meters ([7], p. 14). The sizes of typical cells are dwarfed by higher plants and animals and by ecosystems such as our oceans, which may cover thousands of square kilometers. As with the different temporal scales and using analogous arguments, models of biological systems often focus on one or two spatial scales at a time [8]. Nonetheless, such simplifications are not always applicable, and some processes, such as aging and algal blooms, may require the simultaneous consideration of several temporal and spatial scales. Such multiscale assessments are often very complicated and constitute a challenging frontier of current research (see Chapter 11).

1.3 WHY NOW?

Many of the features of biological systems have been known for quite a while and, similarly, many concepts and methods of systems biology have their roots in its well-established parent disciplines, including physiology, molecular biology, biochemistry, mathematics, engineering, and computer science [9–12]. In fact, it has been suggested that the nineteenth-century scientist Claude Bernard might be considered the first systems biologist, as he proclaimed that the "application of mathematics to natural phenomena is the aim of all science, because the expression of the laws of phenomena should always be mathematical" [13, 14]. A century later, Ludwig von Bertalanffy reviewed in a book his three decades of (by and large unsuccessful) attempts to convince biologists of the systemic nature of living organisms [15, 16]. At the same time, Mihajlo Mesarović used—for the first time—the term "systems biology" and declared that "real advance . . . will come about only when biologists start asking questions which are based on systems-theoretic concepts" [17]. The same year, a book review in *Science* envisioned "a field of systems biology with its own identity and in its own right" [18]. A few years later, Michael Savageau proposed an agenda for systematically studying biological phenomena with mathematical and computational means [8, 19].

In spite of these efforts, systems biology did not enter the mainstream for several more decades. Biology kept its distance from mathematics, computer science, and engineering, primarily because biological phenomena were seen as too complicated for rigorous mathematical analysis and mathematics was considered applicable only to very small systems of little biological relevance. Engineering biological systems from scratch was impossible, and the budding field of computer science did not contribute much to biology beyond rudimentary data management.

So, why has systems biology all the sudden moved to the fore? Any good detective will know the answer: motive and opportunity. The motive lies in the gradual realization that reductionist thinking and experimentation alone are not sufficient if complex systems are involved. Reductionist experiments are very good in generating detailed information regarding specific components or processes of a system, but they often lack the ability to characterize, explain, or predict systemic responses, let alone **emergent properties** that cannot be attributed to any parts of the system but only to their web of interactions. For instance, the emergence of oscillations in the example system represented by the equations in (1.1) cannot be credited to a single component of the system but is a function of its overall organization. Although we have complete knowledge of all details of this simple model pathway, it is very difficult to foresee its capacity either to saturate or oscillate in a damped or stable fashion. Biology is full of such examples.

A few years ago, Hirotada Mori's laboratory completed the assembly of a complete catalogue of single mutants in the bacterium *Escherichia coli* [20]. Yet, the scientific community is still unable to foresee with reliability which genes the bacterium will up- or down-regulate in response to new environmental conditions. Another very challenging example of emergent system properties is the central nervous system. Even though we understand quite well how action potentials are generated and propagated in individual neurons, we do not know how information flows, how memory works, and how diseases affect the normal functioning of the brain. It is not even clear how information in the brain is represented (see also Chapter 15). Thus, while reductionist biology has been extremely successful and will without any doubt continue to be a major driving force for future discovery, many biologists have come to recognize that the detailed pieces of information resulting from this approach need to be complemented with new methods of system integration and reconstruction [21].

The opportunity for systems biology is the result of the recent confluence and synergism of three scientific frontiers. The first is of course the rapid and vast accumulation of detailed biological information at the physiological, cellular, and molecular levels. These targeted investigations of specific phenomena are accompanied by large-scale, high-throughput studies that were entirely infeasible just a couple of decades ago. They include quantification of genome-wide expression patterns, simultaneous identification of large arrays of expressed proteins, comprehensive profiling of cellular metabolites, characterization of networks of molecular interactions, global assessments of immune systems, and functional scans of nervous systems and the human brain. These exciting techniques are generating unprecedented amounts of high-quality data that are awaiting systemic interpretation and integration (**Figure 1.12**).

The second frontier is the result of ingenuity and innovation in engineering, chemistry, and material sciences, which have begun to provide us with a growing array of technologies for probing, sensing, imaging, and measuring biological systems that are at once very detailed, extremely specific, and usable in vivo. Many tools supporting these methods are in the process of being miniaturized, in some cases down to the nanoscale of molecules, which allows diagnoses with minute amounts of biological materials and one day maybe biopsies of individual, living cells. Devices at this scale are beginning to permit the insertion of sensing and disease treatment devices into the human body in an essentially noninvasive and harmless fashion [22–25]. Bioengineering and robotics are beginning to render it possible to measure hundreds or thousands of biomarkers from a single drop of blood. It is even becoming feasible to use molecular structures, prefabricated by nature or mankind, for new purposes in medicine, drug delivery, and biotechnology [26] (**Figure 1.13**).

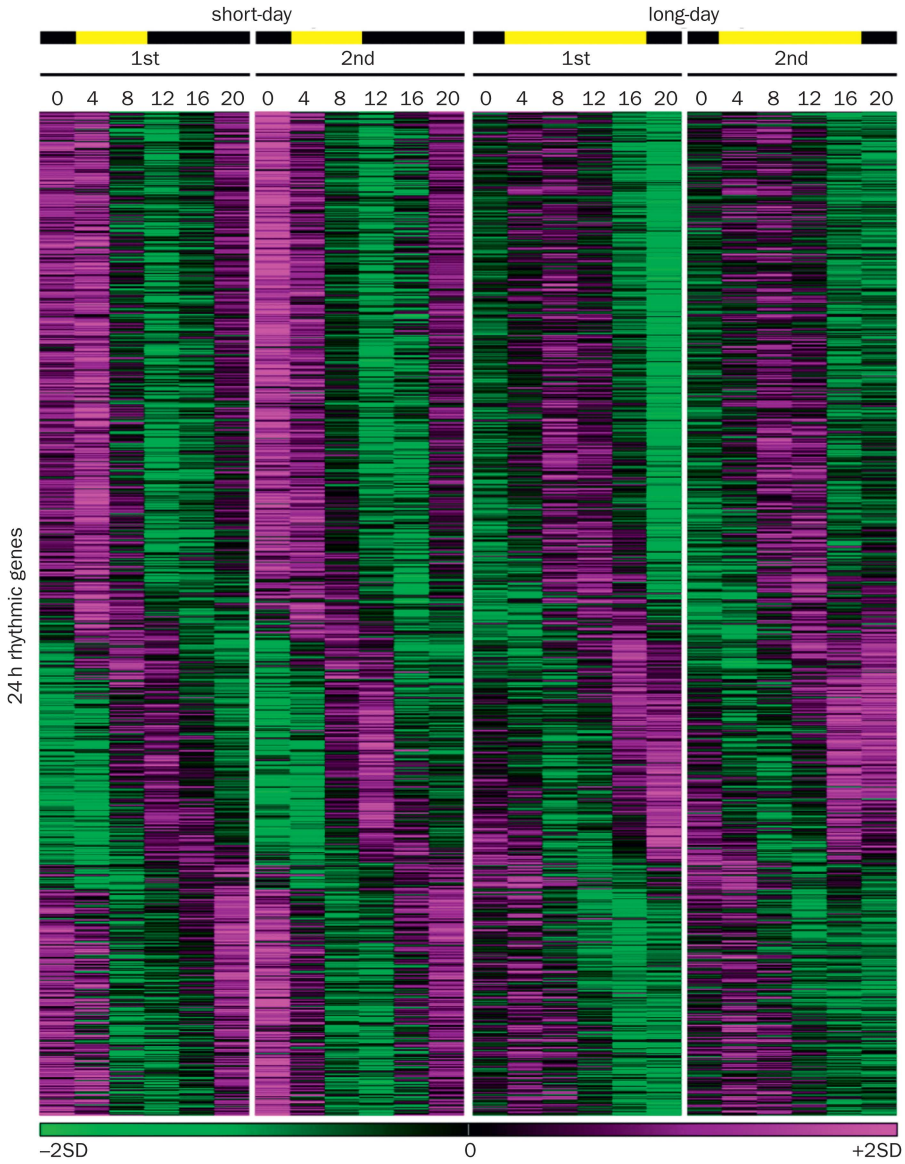


Figure 1.12 Modern high-throughput methods of molecular biology offer data in unprecedented quantity and quality. As an example, the heat map shown here represents a genome-wide expression profile of 24-hour-rhythmic genes in the mouse under chronic short-day (left two panels) and long-day (right two panels) conditions. (From Masumoto KM, Ukai-Tadenuma M, Kasukawa T, et al. *Curr Biol.* 20 [2010] 2199–2206. With permission from Elsevier.)

The third frontier is the co-evolution of mathematical, physical, and computational techniques that are more powerful and accessible to a much wider audience than ever before. Imagine that only a few decades ago computer scientists used punch cards that were read by optical card readers (**Figure 1.14**)! Now, there are even specific computing environments, including Java, R, Python, *Mathematica*®, and MATLAB®, to name but a few, as well as different types of customized mark-up languages (**XML**), such as the systems biology mark-up language **SBML** [27] and the mark-up language AGML, which was developed specifically for analyzing two-dimensional gels in proteomics [28].

Before today's much more effective computer science techniques were available, it was not even possible to keep track of the many components of biological systems, let alone analyze them. But over the past few decades, a solid theoretical and numerical foundation has been established for computational methods specifically tailored for the investigation of dynamic and adaptive systems in biology and medicine. These techniques are now on the verge of making it possible to represent and analyze large, organizationally complex systems and to study their emergent properties in a rigorous fashion. Methods of machine learning and artificial intelligence, numerical mathematics, and bioinformatics permit the efficient mining and analysis of the most useful data from within an overwhelming amount of information that is often not pertinent to a given task. Algorithmic advances permit

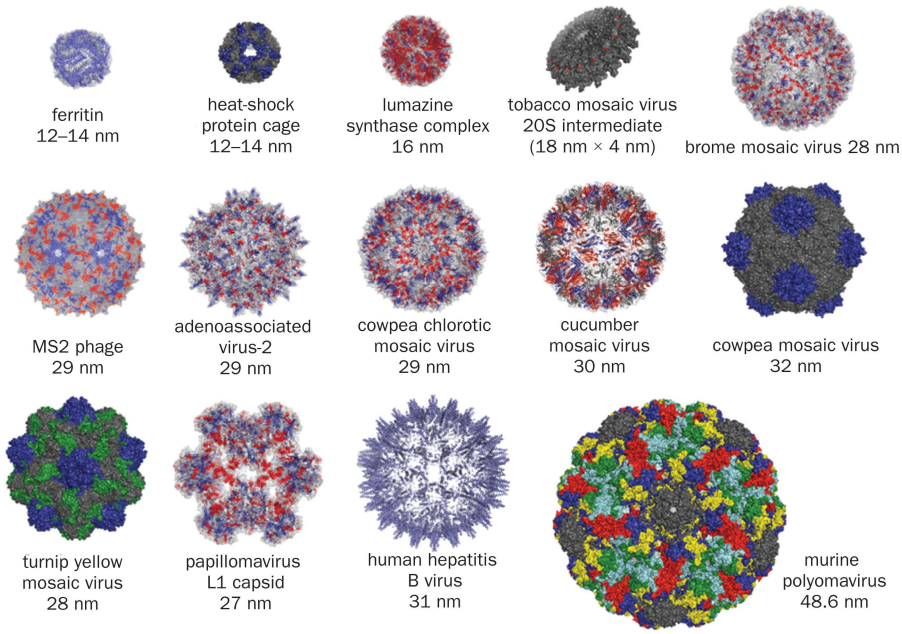


Figure 1.13 "Protein cages" are particles that have applications in bionanotechnology and nanomedicine. These particles are very interesting biological building blocks because they self-assemble into a variety of different shapes. The features of these bionanoparticles can be genetically manipulated and fine-tuned for biomedical purposes, such as drug delivery, gene therapy, tumor imaging, and vaccine development. (From Lee LA & Wang Q. *Nanomedicine* 2 [2006] 137–149. With permission from Elsevier.)

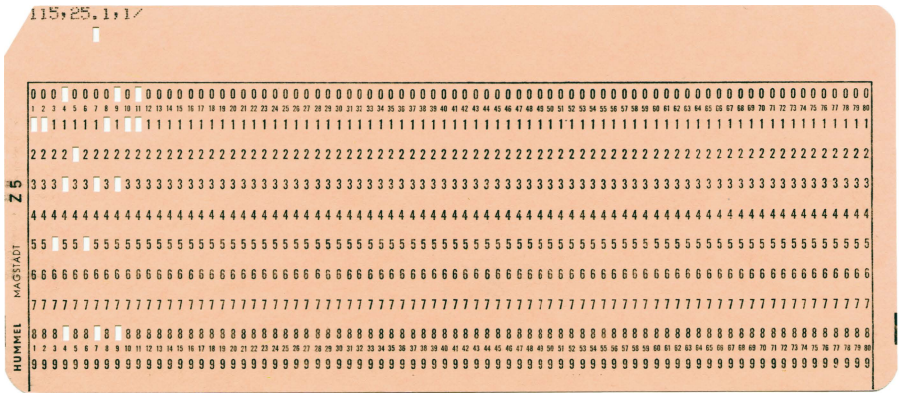


Figure 1.14 Advances in computer power, accessibility, and user-friendliness over the past 30 years have been tremendous. Not too long ago, computer code had to be fed manually into the computer with punch cards. (Courtesy of Mutatis mutandis under the Creative Commons Attribution-Share Alike 3.0 Unported license.)

the simulation and optimization of very large biological flux distribution networks. Computer-aided approximation approaches yield ever-finer insights into the dynamics of complex nonlinear systems, such as the control of blood flow in healthy and diseased hearts. New mathematical, physical, and computational methods, such as AlphaFold [29], are beginning to make it possible to predict the folding of proteins and the binding between target sites and ligands (see Chapter 8). These predictions, in turn, suggest insights into specific molecular interactions and promise the potential of targeted drug interventions that minimize toxic side effects.

Motive and opportunity have met to make systems biology attractive and feasible. It has become evident that the relevant disciplines complement each other in unique ways and that the synergism among them will revolutionize biology, medicine, and a host of other fields, including biotechnology, environmental science, food production, and drug development.

1.4 COMMUNICATING SYSTEMS BIOLOGY

It is not a trivial task to talk succinctly about 25,000 genes and their expression state or about the many processes occurring simultaneously in response to a signal that a cell receives at its outer surface. Our minds are ill equipped to characterize numerical relationships, let alone discuss complicated mathematical functions,

especially if these depend on many variables. If we did not have numbers, we would even have problems describing everyday features such as temperature. Of course, we can say that it is cold or hot, and we have a dozen or so adjectives in between. But if we need more accuracy, common language ceases to be sufficient. Discerning 37.0°C from 38.8°C is not easy without a numerical scale, but it is necessary to have the tools to describe the difference, for instance, because the former reflects normal body temperature, while the latter is a sign of fever.

We may willingly or grudgingly accept the fact that we need mathematics, which comes with its own terminology, but communication is a two-way process. If we start talking about eigenvalues and Hopf bifurcations, we are almost guaranteed to lose mainstream biologists, let alone laypeople [30]. This is a grave problem, because our results must be conveyed to biologists, who are providing us with data and to the public that pays for our research and has a right to reap the fruit of the enormous investment of resources going into science [31, 32]. The only true solution to this challenge is the bilingual education and nurturing of systems biologists who can translate biological phenomena into math and computer code and who can explain what it really means for the biological system if the real part of an eigenvalue is positive [31].

Communication is not trivial even within biology itself, because specialization has progressed so far that different fields such as molecular biology, immunology, and nanomedicine have developed their own terminology and jargon. Let's look at this issue in the form of a parable from Indian folklore that describes six blind men exploring an elephant (**Figure 1.15**). The story is quite old and usually ends in utter confusion, but it is useful to analyze it further than is usually done. The story has it that each of the blind men touched a different part of an elephant and came to a different conclusion concerning the object of his research. The man touching the side thought he was touching a wall, the one feeling the leg concluded he was touching a tree. The elephant's trunk gave the impression of a snake, the tusk that of a pointed spear, the tail felt like a rope, and the ear appeared to be like a large leaf or fan. It is not difficult to see the analogy to a complex biological system like the onset of Alzheimer's disease. The first scientist found "the Alzheimer gene," the second discovered "a strong association between the disease and former head injuries," another scientist detected "problems with fatty acid metabolism in the brain," and yet another suggested that "aluminum in cookware might be the culprit." As in the case of the elephant, the scientists were right, to some degree.

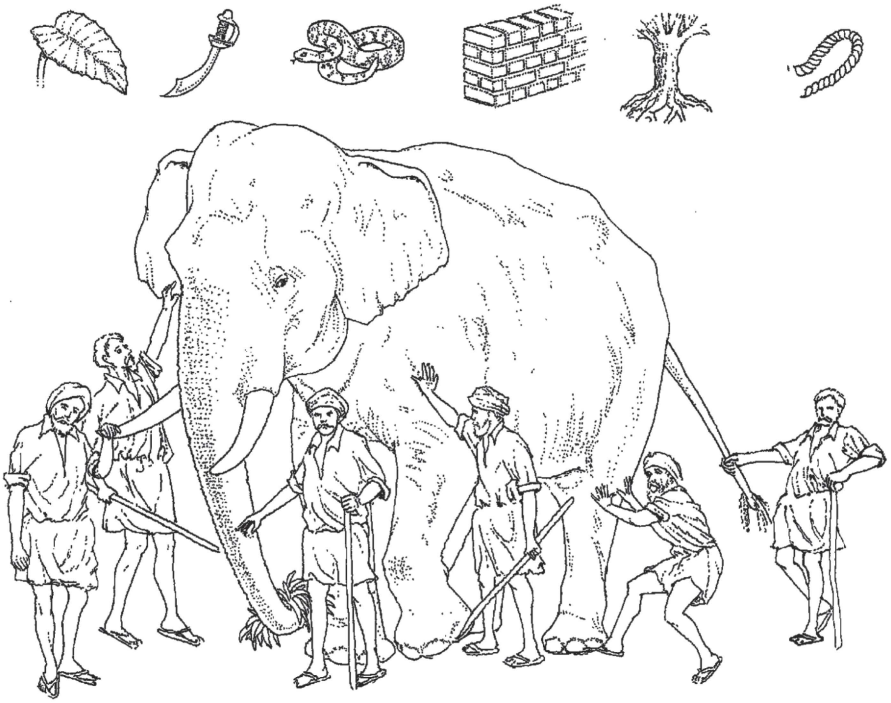


Figure 1.15 Information about isolated parts of a system alone does not always reveal the true nature of the system. An old story of six blind Indian men trying to determine what they touch is a parable for the dangers of scientific silos and the lack of good communication.

Let's analyze the elephant story a little further. The first problem among the six blind men might have been the **homogeneous** pool of researchers. Including a female or a child, or someone from a different culture, might have provided additional clues. Also, we have to feel sorry for the Indian men for being blind. However, they were apparently not mute or deaf, so that a little discussion among them might have gone a long way. While all six were blind, it is furthermore fair to assume that they had friends with working vision, who could have set them straight. They could have used not just their hands but also their other senses, such as smell. Do tree trunks really smell like elephant feet? Finally, they apparently stayed in their one spot, thereby greatly limiting their experience base.

It is again easy to translate these issues into biology, especially when we think of purely reductionist strategies. Instead of a homogeneous pool of biologists analyzing biological systems, it is without doubt more effective to have a multidisciplinary team including different varieties of biologists but also physicists, engineers, mathematicians, chemists, and smart people trained in the liberal arts or economics. Instead of only focusing on the one aspect right in front of our nose, communication with others provides context for singular findings. We don't know whether the Indian men spoke the same language, but we know that even if biologists, computer scientists, and physicists all use English to communicate, their technical languages and their views of the scientific world are often very different, so that communication may initially be superficial and ineffective. That is where multidisciplinary groups must engage in learning new terminologies and languages and include interdisciplinary translators [31]. Just as the Indian men should have called upon their seeing friends, investigators need to call in experts who master techniques that had not been applied to the biological problem at hand. Finally, it behooves the trunk analyzer to take a few steps and touch the tusk and the side. The established scientific disciplines in the past have often become silos. Sometimes without even knowing it, researchers have been keeping themselves inside these silos, unable or unwilling to break out and to see the many other silos around, as well as ample space between them.

Systems biology does not ask the six blind men to abandon their methods and instead to run circles around the elephant. By focusing on one aspect, the reductionist "elephantologists" are poised to become true experts on their one chosen body part and to know everything there is to know about it. Without these experts, systems biology would have no data to work with. Instead, what systems biology suggests is a widening of the mindset and at least rudimentary knowledge of a second language, such as math. It also suggests the addition of other researchers, assisting the "trunkologist" and the "tuskologist" by developing new tools of analysis, by telling them in their language what others had found, by closing the gap between trunks and tusks and tails.

One strategy for accomplishing this synergism is to collect the diverse pieces of data and contextual information obtained by the six blind men and to merge them into a conceptual model. What kind of "thing" could consist of parts that feel like a snake, tree trunks, large walls, two spears, two fans, and a rope? By extension, how could an aberrant gene, former head injuries, and altered brain metabolism functionally interact to result in Alzheimer's disease? Well-trained systems biologists should be able to develop strategies for merging heterogeneous information into formal models that permit the generation of testable hypotheses, such as "tree-trunk-like things are connected by wall-like things." These hypotheses may be wrong, but they can nevertheless be very valuable, because they focus the scientific process on new, specific experiments that either confirm or refute a hypothesis. An experiment could consist of walking along the "wall" as far as possible. Is there a "tree trunk" at the end? Are there "tree trunks" to the left and to the right? Is there a tusk at one or both ends? Is the "snake" connected to a "wall" or to a "tree trunk"? Does the wall reach the ground? Each answer to one of these questions constrains further and further what the unknown "thing" could possibly look like, and this is the reason that refuted hypotheses are often as valuable or even more valuable than confirmed hypotheses. "The wall does indeed not reach the ground!" Then, how is it supported? By tree trunks?

The story tells us that effective communication can solve a lot of complex questions. In systems biology, such communication is not always easy and requires not only mastering the terminology of several parent disciplines but also internalizing

the mindset of biologists and clinicians on the one hand and of mathematicians, computer scientists, and engineers on the other. So, let's learn a good bit about biology. Let's study laboratory data and information and explore the mindset of biologists. Let's also study graphs and networks with methods from computer science. Let's see how mathematicians approach a biological system, struggle with assumptions, make simplifications, and obtain solutions that are at first incomprehensible to the non-mathematician but do have real meaning once they are translated into the language of biology.

1.5 THE TASK BEFORE US

We have discussed the need to understand biological systems. But what does that really mean? Generically, it means that we should be able to explain how biological systems work and why they are constructed in the fashion as we observe them and not in a different one. Second, we should be able to make reliable predictions of responses of biological systems under yet-untested conditions. And third, we should be able to propose targeted manipulations into biological systems that change their responses to our specifications.

This level of understanding is a tall order, and we will need many years to achieve it even for a narrowly focused domain within the huge realm of biological systems. An important component of the task is the conversion of actual biological systems into computational models, because this conversion, if it is valid, allows us almost unlimited and comparatively cheap analyses. The resulting models of biological systems come in two varieties. The first type focuses on specific systems and includes all pertinent functional and numerical details; one might think of the analogy to a flight simulator. The second type of model is intended to help us understand the fundamental, generic features of the organization of biological systems; here one might think of elementary geometry, which offers us valuable insights into spatial features of the world by dealing with ideal triangles and circles that do not really exist in nature. The two model types point to two opposite ends of a large spectrum. The former models will be large and complex, while the latter will be as reduced and simple as feasible. In practice, many models will fall between these two extremes [33].

To pave the way toward these goals, this book is organized in three sections. The first introduces in six chapters various modeling tools for converting biological phenomena into mathematical and computational analogues and for diagnosing, refining, and analyzing them. The second section describes in four chapters the molecular inventories and signaling modalities that populate biological systems, as well as representative models of select aspects of the discussed topics. The remaining chapters in the third section are devoted to pertinent case studies in systems biology and a look into the future.

The modeling approaches parallel two fundamental properties of biological systems, namely their static structure and their dynamics, that is, their changes over time. For static analyses, we will characterize and rationalize how nature put particular systems together and which parts are directly or loosely connected with each other. We will see that there are distinct types of connections and interactions. One important difference is that some connections allow the flow of material from a source to a target, while others serve the sole purpose of signaling the state of some subsystem to other parts of the system. In the latter case, no material changes location. Like a billboard that is not changed whether hundreds of people look at it or nobody looks at all, a signaling component is not changed when it sends a signal. As is to be expected, some connections are crucial, while others are dispensable, but it is difficult to differentiate the two a priori. Finally, there is the very challenging question of how we can even determine the structure of a system. What types of data do we need to infer the structure of a system, and how reliable is such an inference?

The dynamics of a system is of utmost importance because all biological systems change over time. Organisms traverse a life cycle during which they undergo tremendous changes. Even a lowly yeast cell fills up with scars where it has given birth to daughter cells, and once its surface is full, the cell slides into the sunset of life. We can easily see these changes under a microscope, but an incomparably

larger number of changes remain hidden from our sight. Gene expression patterns, amounts of proteins, profiles of metabolites, these all change dramatically between birth and death. In addition to normal changes throughout its lifetime, every organism responds to fast and slow changes in the environment and adapts rather quickly to new situations. Today we may observe and characterize the gene expression network in a bacterium, but tomorrow it may already have changed in response to some environmental pressures. Indeed, bacteria evolve so quickly that the traditional term "wild type" no longer has much meaning. Even more than static aspects, dynamic aspects of biological systems mandate the use of computational models. These models help us reveal how fascinating living systems are, with respect both to their overall efficiency and to the ingenuity with which dynamic responses and adaptations are coordinated.

Whether static or dynamic, some model designs and analyses will be performed in the bottom-up and others in the top-down direction. However, since we seldom start at the very bottom, that is, with individual atoms or at the very top with models of complete organisms as they interact with their environment, most modeling strategies in systems biology are in truth "middle-out," to use Nobel Laureate Sydney Brenner's expression (cited in [34]). They begin somewhere in between the extremes, maybe with pathways or with cells. Over time, they may be incorporated into larger models or they may become more and more refined in detail.

The second section of the book addresses the molecular inventories of biological systems. Paralleling the biological organization of organisms, one chapter is devoted to gene systems, one to proteins, and one to metabolites. A further chapter discusses signal transduction systems. Clearly, these chapters are woefully incomplete and should not be thought of as substitutes for real biology books. Their purpose is merely to provide brief overviews of the main classes of biological components that are the foundation of all modeling strategies.

The third section contains case studies that in one way or another highlight aspects of biological systems that are representative for certain classes of phenomena. Two chapters describe the multiscale nature of very many biological systems and methods for grasping the simultaneous operation of these systems at different levels. The first is generic, while the second focuses on the multi-level function of the heart. The third chapter indicates how systems biology can contribute to medicine and drug development in a powerful fashion. The fourth chapter illuminates aspects of populations and their interactions. The methods we discuss are quite general and may pertain to macroscopic as well as microscopic populations. Finally, the last chapter discusses emerging and future trends in systems biology.

1.6 EXERCISES

1.1 Search the Internet, as well as different dictionaries, for definitions of a "system." Extract commonalities among these definitions and formulate your own definition.

1.2 Search the Internet for definitions of "systems biology." Extract commonalities among these definitions and formulate your own definition.

1.3 List ten systems within the human body.

1.4 Exactly what features make the system in Figure 1.10 so much more complicated than the system in Figure 1.8?

1.5 Imagine that Figure 1.10 represents a system that has become faulty owing to disease. Describe the consequences of its complexity for any medical treatment strategy.

1.6 In Figure 1.2, are there control paths along which ABA either activates or inhibits closure of stomata? If so, list at least one path each. If both paths exist in parallel, discuss what you expect the effect of an increase in ABA to be on stomata closure.

1.7 Imagine a control system like that in Figure 1.2 but much simpler. Specifically, suppose there are only one activating and one inhibiting path in parallel. Suppose further that the activating path reacts much faster than the inhibiting path. What would be the consequence with respect to the effect of an increase in input (ABA) on output (stomata closure)? How could a difference in speed be implemented in a natural cell? Does it matter how strong or weak the activation and inhibition are? Discuss!

1.8 We discussed that it is often difficult to infer the structure of a biological system from data. Is it possible that two different systems produce exactly the same input–output data? If you think it is impossible, discuss and defend your conclusion. If you think the answer is affirmative, construct a conceptual example.

1.9 List and discuss features supporting the claim that reductionism alone is not sufficient for understanding biological systems.

1.10 List those challenges of systems biology that cannot be solved with intuition alone.

1.11 Discuss why it is important to create terminology and tools for communicating systems biology.

1.12 Assemble a "to-do" list for the future of systems biology.

FURTHER READING

Alon U. *An Introduction to Systems Biology: Design Principles of Biological Circuits*. Chapman & Hall, CRC Press, New York, NY, 2006.

Covert MW. *Fundamentals of Systems Biology: From Synthetic Circuits to Whole-Cell Models*. CRC Press, New York, NY, 2017.

Feidan L. *Systems Biology: Understanding Complex Biological Systems*. Syrawood Publishing House, New York, NY, 2019.

Keneko K. *Life: An Introduction to Complex Systems Biology*. Springer Verlag, Berlin, Heidelberg, 2010.

Kitano H (ed). *Foundations of Systems Biology*. MIT Press, Cambridge, MA, 2001.

Klipp E, Herwig R, Kowald A, et al. *Systems Biology in Practice: Concepts, Implementation and Application*. Wiley-VCH, Weinheim, Germany, 2005.

Marcum J. *Conceptual Foundations of Systems Biology: An Introduction*. Nova Science Publishers, Inc., Hauppauge, NY, 2009.

Noble D. *The Music of Life: Biology Beyond Genes*. Oxford University Press, Oxford, UK, 2006.

Szallasi Z, Periwal V & Stelling J (eds). *System Modeling in Cellular Biology: From Concepts to Nuts and Bolts*. MIT Press, Cambridge, MA, 2006.

Voit EO. *Computational Analysis of Biochemical Systems: A Practical Guide for Biochemists and Molecular Biologists*. Cambridge University Press, Cambridge, UK, 2000.

Voit EO. *The Inner Workings of Life*. Cambridge University Press, Cambridge, UK, 2016.

Voit EO. *Systems Biology. A Very Short Introduction*. Oxford University Press, Oxford, UK, 2020.

REFERENCES

1. Li, S., S.M. Assmann, and R. Albert, Predicting essential components of signal transduction networks: A dynamic model of guard cell abscisic acid signaling. *PLoS Biol*, 2006. **4**(10): p. e312, https://doi.org/10.1371/journal.pbio.0040312.
2. USNO, *United States Naval Observatory Cesium Fountain*, http://tycho.usno.navy.mil/clockdev/cesium.html, 2010.
3. Voit, E.O., F. Alvarez-Vasquez, and Y.A. Hannun, Computational analysis of sphingolipid pathway systems. *Adv Exp Med Biol*, 2010. **688**: p. 264–275.
4. Vodovotz, Y., et al., Mechanistic simulations of inflammation: Current state and future prospects. *Math Biosci*, 2009. **217**(1): p. 1–10.
5. Voit, E.O., *Computational Analysis of Biochemical Systems. A Practical Guide for Biochemists and Molecular Biologists*. 2000, Cambridge, UK: Cambridge University Press, p. xii + 530.
6. Kim, H.J. and B.A. Triplett, Cotton fiber growth in planta and in vitro. Models for plant cell elongation and cell wall biogenesis. *Plant Physiol*, 2001. **127**: p. 1361–1366.
7. Kumar, A., *Understanding Physiology*. 2009, New Delhi: Discovery Publishing House PVT. LTD., p. 292.
8. Savageau, M.A., *Biochemical Systems Analysis: A Study of Function and Design in Molecular Biology*. 1976, Reading, MA: Addison-Wesley Pub. Co. Advanced Book Program, p. xvii, 379.
9. Strange, K., The end of "naive reductionism": Rise of systems biology or renaissance of physiology? *Am J Physiol Cell Physiol*, 2005. **288**(5): p. C968–C974.
10. Voit, E.O. and J.H. Schwacke, Understanding through modeling, in *Systems Biology: Principles, Methods, and Concepts*, A.K. Konopka, Editor. 2007, Taylor & Francis: New York, p. 27–82.
11. Westerhoff, H.V. and B.O. Palsson, The evolution of molecular biology into systems biology. *Nat Biotechnol*, 2004. **22**(10): p. 1249–1252.
12. Wolkenhauer, O., H. Kitano, and K.-H. Cho, An introduction to systems biology. *IEEE Contr Syst Mag*, 2003. **23**(4): p. 38–48.
13. Bernard, C., Introduction a l'étude de la médecine expérimentale, in *Collection: Texte intégral*. 1865, Paris: Éditions Garnier-Flammarion (1966), p. 318.
14. Noble, D., Claude Bernard, the first systems biologist, and the future of physiology. *Exp Physiol*, 2008. **93**(1): p. 16–26.
15. von Bertalanffy, L., Der Organismus als physikalisches System betrachtet. *Die Naturwissenschaften*, 1940. **33**: p. 521–531.
16. von Bertalanffy, L., *General system theory; foundations, development, applications*. 1969, New York: G. Braziller, p. xv, 289.
17. Mesarović, M.D., Systems theory and biology—view of a theoretician, in *Systems Theory and Biology*, M.D. Mesaroviâc, Editor. 1968, Berlin, New York: Springer, p. 59–87.
18. Rosen, R., Book review: Systems theory and biology. Proceedings of the 3rd systems symposium, Cleveland, Ohio, Oct. 1966. M.D. Mesarović, Editor. New York: Springer-Verlag, 1968, p. xii + 403. *Science*, 1968, 5 July. **161**(3836): p. 34–35.
19. Savageau, M.A., Biochemical systems analysis. I. Some mathematical properties of the rate law for the component enzymatic reactions. *J Theor Biol*, 1969. **25**(3): p. 365–369.
20. Yamamoto, N., et al., Update on the Keio collection of *Escherichia coli* single-gene deletion mutants. *Mol Syst Biol*, 2009. **5**: p. 335.
21. Savageau, M.A., The challenge of reconstruction. *The New Biol*, 1991. **3**: p. 101–102.
22. Freitas, R.A.J., *Nanomedicine, Volume I: Basic Capabilities*. 1999, Georgetown, TX: Landes Bioscience, p. 509, http://www.nanomedicine.com/NMI.htm.
23. Lee, L.A. and Q. Wang, Adaptations of nanoscale viruses and other protein cages for medical applications. *Nanomed: Nanotechnol Biol Med*, 2006. **2**(3): p. 137–149.
24. Martel, S., et al., Automatic navigation of an untethered device in the artery of a living animal using a conventional clinical magnetic resonance imaging system. *Appl Phys Lett*, 2007. **90**: p. 114105, https://doi.org/10.1063/1.2713229.
25. Deng, J., et al., Nanosensors for diagnosis of infectious diseases. *ACS Appl Bio Mater*, 2021. **4**(5): p. 3863–3879.
26. Patra, J.K., et al., Nano based drug delivery systems: Recent developments and future prospects. *J Nanobiotechnol*, 2018. **16**(1): p. 71.
27. SBML, *The Systems Biology Markup Language*, http://sbml.org/Main_Page, 2011.

28. Stanislaus, R., et al., An XML standard for the dissemination of annotated 2D gel electrophoresis data complemented with mass spectrometry results. *BMC Bioinform*, 2004. **5**: p. 9.
29. Jumper, J., et al., Highly accurate protein structure prediction with AlphaFold. *Nature*, 2021. **596**(7873): p. 583–589.
30. Voit, E.O., Perspective: Dimensions of the scientific method. *PLoS Comput Biol*, 2019. **15**(9): p. e1007279.
31. Voit, E.O., *The Inner Workings of Life*. 2016, Cambridge, UK: Cambridge University Press.
32. Voit, E.O., Perspective: Systems biology beyond biology. *Front Syst Biol*, 2022. **2**: p. 987135, https://doi.org/10.3389/fsysb.2022.987135.
33. Voit, E.O., A.M. Shah, D.V. Olivença, and Y. Vodovotz, What's next for computational systems biology? *Front Syst Biol*, 2023. **3**: p. 1250228, https://doi.org/10.3389/fsysb.2023.1250228.
34. Noble, D., *The Music of Life; Biology Beyond Genes*. 2006, Oxford, UK: Oxford University Press.

Taylor & Francis
Taylor & Francis Group
http://taylorandfrancis.com

Introduction to Mathematical Modeling

2

When you have read this chapter, you should be able to:

- Realize the need for mathematical models
- Understand the challenges of mathematical modeling
- Describe the modeling process in generic terms
- Be aware of some of the important types of mathematical models in systems biology
- Identify the ingredients needed to design a model
- Set up simple models of different types
- Perform basic diagnostics and exploratory simulations
- Implement changes in parameters and in the model structure
- Use a model for the exploration and manipulation of scenarios

At the core of any computational analysis in systems biology is a mathematical **model**. A model is an artificial construct in the language of mathematics that represents a process or phenomenon in biology. **Modeling** is the process of creating such a construct and squeezing new insights out of it. Mathematical models in systems biology can take many forms. Some are small and simple, others large and complicated. A model can be intuitive or very abstract. It may be mathematically elegant in its streamlined simplicity or account for very many details with a large number of equations and specifications that require sophisticated computer **simulations** for their analysis. The key commonality among all good models in biology is their ability to offer us insights into processes or systems that we would not be able to gain otherwise. Good models can make sense of a large number of isolated facts and observations. They explain why natural systems have evolved into the particular **motifs**, that is, recurring organizational and regulatory structures eliciting behaviors that we observe time and again; several chapters will return to questions of motifs and of natural design and operation. Models allow us to make **predictions** and **extrapolations** about experimentally untested situations and future trends. They can lead to the formulation of new hypotheses. Good models have the potential to guide new experiments and suggest specific recipes for manipulating biological systems to our advantage. Models can tell us how cell cycles are controlled and why plaques appear in certain sections of our arteries and not in others. Future models may help us understand how the brain works and prescribe specific interventions in tumor formation that will be infinitely more subtle and less crude than our current methods of radio- or chemotherapy.

DOI: 10.1201/9781003402770-2

Modeling in systems biology is really a special way of looking at the world. Whether we just conceptualize a situation in the form of components and interactions or whether we actually set up equations and analyze them, we begin to see how complicated phenomena can be conceptually simplified and dissected into manageable submodels or **modules**, and we develop a feel for how such modules interact with each other and how they might respond to changes. Be aware! Modeling can become a life-changing adventure.

An analogy for a mathematical model is a geographical map. This map is obviously very different from reality, and many details are missing. What do the houses look like in some remote village? Are the lakes clean enough for swimming? Are the people friendly? Then again, it is exactly this simplification of the complicated real world that makes the map useful. Without being distracted by secondary details, the map allows us to look at the network of major roads throughout an entire country while sitting at home on our living room floor. The map provides us with distances and highway types that let us estimate travel times. Altitudes on the map indicate whether the area we want to drive through is hilly and could be cold even during the summer months. Once we learn how to interpret the map, we can infer what the countryside might look like. Empty spaces, like the Badlands in South Dakota, suggest that living in the area might be difficult or undesirable for humans. The density of cities and villages provides us with some idea of how easy it might be to find lodging, food, or gas. Clusters of cities along the ocean probably imply that living is expensive. Mathematical models have the same generic properties: they reflect some aspects quite well but barely touch on others or ignore them altogether. They allow us to infer certain properties of reality but not others. The inclusion or exclusion of certain features in models has a direct, important implication for the modeling process. Namely, the key task in designing a suitable model is a crisp focus on the aspects of primary interest and a representation of these aspects that retains their most important properties. At the same time, extraneous or distracting information must be minimized, and the model must be limited in complexity to permit analysis with reasonable efficiency.

Modeling is a mathematical and computational endeavor that in some sense resembles the fine arts (**Figure 2.1**). Basic math problems in calculus or linear algebra have the correct solutions in the back of the book, or at least somebody, somewhere, knows these unique, correct solutions. In contrast to this crispness,

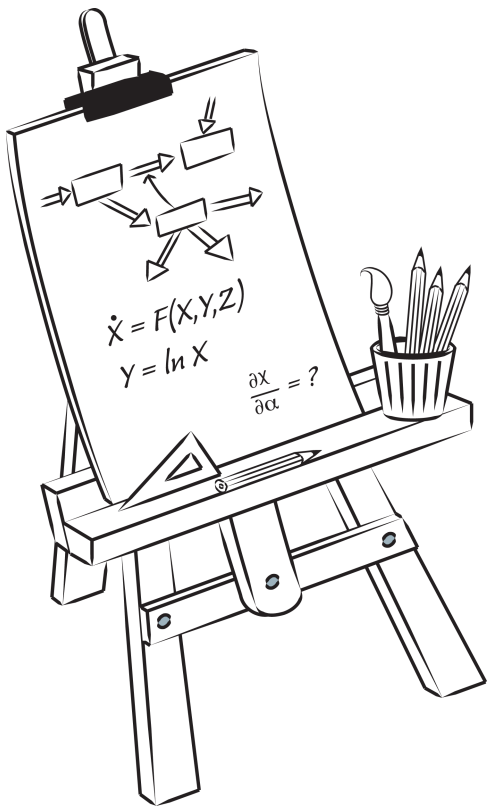


Figure 2.1 Modeling resembles a fine art. Modeling requires both mastery of techniques and creativity.

many aspects of modeling allow a lot more flexibility. They permit different approaches, and often there are multiple, quite distinct solutions that are all reasonable, valuable, and correct in their own right. Who is to say that an oil painting is better than a watercolor? Or that an opera has higher artistic value than a symphony? Similarly, a mathematical model has aspects and properties in different dimensions and its value is not always easy to judge. For instance, is a very complicated, detailed model better than a simple yet elegant model?

The answers to these questions depend critically on the goals and purposes of the model, and it is not just beneficial but indeed mandatory to specify the reasons for setting up a model and to ask a lot of questions before we embark on actually designing it. These questions must clarify the potential utilization of the model, its desired accuracy, the degree of complexity we are willing to accept, and many other details that we will touch on in this chapter. Indeed, it is prudent to invest quite a bit of time and exploration into pondering these aspects before one begins selecting a specific model structure and defining variables, parameters, and other model features, because technical challenges often distract from the big picture.

As in the fine arts, the creation of a good model requires two ingredients: technical expertise and creativity. The technical aspects can and must be learned through the acquisition of a solid repertoire of math and computer science skills. To make theoretical and numerical analyses meaningful, one also has to dig sufficiently deeply into biology in order to understand the background and terminology of the phenomenon of interest and to interpret the model results in a correct and sensible manner. Again similar to the fine arts, the creative aspect of the process is more difficult to comprehend and teach. For instance, it requires more experience than one might expect to ask the right questions and reformulate them in such a way that they become amenable to mathematical and computational analysis. For a novice, all data may look equally valuable, but they really are not. In fact, it takes time and experience to develop a feel for what is doable with a particular set of data and what is probably out of reach. It is also important to learn that some data are simply not well suited for mathematical modeling. This insight is sometimes disappointing, but it can save a lot of effort and frustration.

The shaping and refining of specific questions mandates that we learn how to match the biological problem at hand with appropriate mathematical and computational techniques. This matching requires knowledge of what techniques are available. It also means that the modeler must make an effort to understand how biologists, mathematicians, engineers, and computer scientists are trained to think because the differences in approaches are quite pronounced among these disciplines. Mathematicians are encouraged to simplify and abstract, and there is hardly anything more appealing than a tough problem that can be scribbled on the back of a napkin. Computer scientists look at problems algorithmically, that is by dissecting them into very many, very tiny steps. Engineers like to define boundaries of the investigated system, develop models with enough detail to mirror the application setting, and inform design specifications, while often making approximations outside of the scope of investigation. Contrast that with biologists, who have learned to marvel the complexity, diversity, and intricacies of even the smallest biological items, and you can imagine the mental and conceptual tensions that may develop during the modeling process.

The formation of good questions requires decisions about what is really important in the model, which features can be ignored, and what inaccuracies we are willing to tolerate when we simplify or omit details. It is this complex of decisions on the inclusion and exclusion of components, facts, and processes that constitutes the artistic aspect of modeling. As in the fine arts, a good model may give an outsider the impression "Oh sure, I could have done that," because it is well designed and does not reveal the hard work and many failed attempts that went into it. A good degree of proficiency with respect to these decision processes can be learned, but, like everything else, it requires practice with simple and then increasingly complex modeling tasks. This learning by doing can be exhilarating in success and depressing in failure, and while we have every right to enjoy our successes, it is actually often the failures that in the end are more valuable, because they point to aspects we do not truly understand.

The computational side of modeling draws from two classes of methods: mathematical analysis and computer simulation. Most practical applications use a

combination of the two, because both have their own strengths. A purely mathematical analysis leads to general solutions that are always true, if the given assumptions and prerequisites are satisfied. This is different from even a large set of computer simulations, which can never confer the same degree of certainty and generality. For instance, we can mathematically prove that every real number has a real inverse (4 has 0.25, –0.1 has –10), with the notable exception of 0. But if we were to execute a mega-simulation on a computer, randomly picking a real number and checking whether its inverse is real, we would come to the conclusion that indeed all real numbers have a real inverse. We would miss zero, because the probability of randomly picking zero from an infinite range of numbers is nil. Thus, in principle, mathematics should always be preferred. However, in practical terms, the mathematics needed to analyze biological systems becomes very complicated very quickly, and even relatively simple-looking tasks may not have an explicit analytical solution and therefore require computational analysis.

Independent of the specific biological subject area and the ultimate choice of a particular mathematical and computational framework, the generic modeling process consists of several phases. Each phase is quite distinct both in input requirements and in techniques. Broadly categorized, the phases are shown in **Figure 2.2**, together with a coarse flow chart of the modeling process. Further details are presented in **Table 2.1**. The process starts with conceptual questions about the biological problem, becomes more and more technical, and gradually returns to the biological phenomenon under study. It is very important to devote sufficient effort to each of these phases, especially the early ones, because they ultimately determine whether the modeling effort has a good chance of succeeding.

One of the early choices pertains to the explanatory character of the model. One may choose a comparatively simple regression model, which correlates two or more quantities to each other, without, however, suggesting a rationale for the correlation. As an example, one may find that cardiovascular disease is often associated with high blood pressure, but a regression model does not offer an explanation. As a stark alternative, one could develop a much more complex mechanistic model of cardiovascular disease, in which blood pressure is a component. If for-

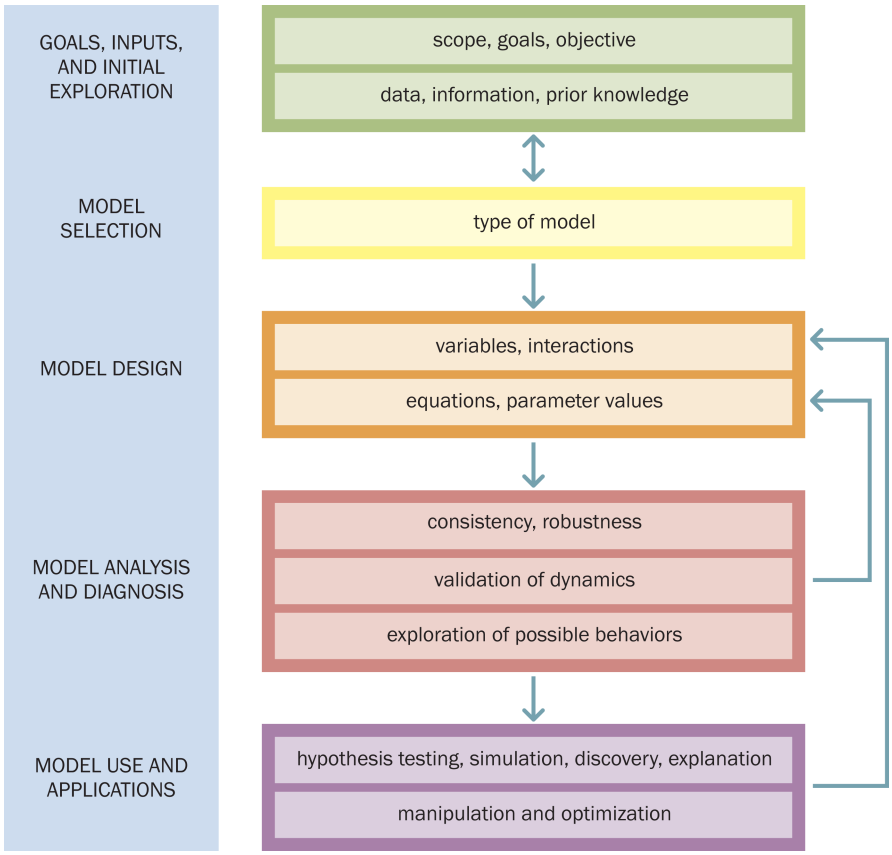


Figure 2.2 Flow chart of the modeling process. The modeling process consists of distinct phases but is often iterative and involves refinements in model design and repeated diagnosis. See also Table 2.1. (Adapted from Voit EO, Qi Z & Miller GW. *Pharmacopsychiatry* 41[Suppl. 1] [2008] S78–S84. With permission from Thieme Medical Publishers.)

TABLE 2.1: ISSUES TO PONDER DURING THE MODELING PROCESS

Goals, Inputs, and Initial Exploration

- Scope of the model, including goals, objectives, and possible applications
- Data needs and availability (types, quantity, quality)
- Other available information (non-quantitative heuristics, qualitative input from experts)
- Expected feasibility of the model
- Relevance and degree of interest within the scientific community

Model Selection

- Model structure
  - Explanatory (mechanistic) versus correlative (black box)
  - Static versus dynamic
  - Continuous versus discrete
  - Deterministic versus stochastic
  - Spatially distributed versus homogeneous
  - Open versus closed
  - Most appropriate, feasible approximations

Model Design

- Model diagram and lists of components
  - Dependent variables
  - Independent variables
  - Processes and interactions
  - Signals and process modulations
  - Parameters
- Design of symbolic equations
- Parameter estimation
  - Bottom-up versus top-down

Model Analysis and Diagnosis

- Internal consistency (e.g., conservation of mass)
- External consistency (e.g., closeness to data and expert opinion)
- Reasonableness of steady state(s)
- Stability analysis
- Sensitivity and robustness analysis
- Structural and numerical boundaries and limitations
- Range of behaviors that can or cannot be modeled (e.g., oscillations, multiple steady states)

Model Use and Applications

- Confirmation, validation of hypotheses
- Explanation of causes and effects; causes of failure, counterintuitive responses
- Best case, worst case, most likely scenarios
- Manipulation and optimization (e.g., treatment of disease, avoidance of undesired states or dynamics, yield optimization in biotechnology)
- Discovery of design principles

mulated validly, the model would explain the mechanistic connection, that is, the network of causes and effects leading from high blood pressure to the disease. Even if we disregard the difficulties in formulating a comprehensive mechanistic model, the choice of model type is not black and white. Regression models do not provide an explanation, but they often make correct, reliable predictions. An explanatory model may also be quite simple, but if it accounts for a large number of processes, it often becomes so complicated and requires so many assumptions that it may no longer describe or predict new scenarios with high reliability. In reality, most models are intermediates—explanatory in some aspects and correlative in others.

In the following sections, we discuss each phase of the modeling process and illustrate it with a variety of small, didactic "sandbox models" as well as a famous,

very old model describing the spread of infectious diseases, which was still used, with many variations, to make predictions regarding the recent COVID-19 pandemic. The process begins with an exploration of the available information and the selection of a model that can utilize this information and ultimately yield new insights (see Figure 2.2 and Table 2.1). Often, one will go back and forth between these two aspects before a suitable model design is chosen. Once an appropriate model type is determined, the technical design phase begins. Components of the model are identified and characterized. The resulting model is diagnosed, and, should troublesome aspects emerge, one returns to the model design. A model that fares well in the diagnostic phase is used for explanations, simulations, and manipulations. It often turns out that model features are not optimal after all, requiring a return to the model design phase.

2.1 GOALS, INPUTS, AND INITIAL EXPLORATION

Summarizing crucial points of the introduction, it is always prudent to reflect on one's expectations before launching a modeling project. Except as a practice exercise, there is not much relevance for developing a model without a purpose. The guiding question to be asked should be something like the following: Assuming that the planned modeling effort proceeds in an ideal fashion, what new insights would be gained from the model? Many answers are possible. A small sample is the following:

1. The model could explain some phenomenon we do not understand.
2. The model results will confirm or refute a specific hypothesis. For example, if the model analysis proves that the presumed model structure cannot generate oscillations as they are observed in actual experiments, something is wrong or missing from the alleged model structure.
3. The model allows us to study a situation that cannot be tested experimentally, because experiments would be practically impossible; unethical; or too complicated, time-consuming, or expensive.
4. The model enables predictions regarding untested situations in the future.
5. The model explains and rationalizes motifs, which are often-encountered structural or procedural patterns with which nature solves problems. Several chapters will return to this question of motifs.

In addition to asking such questions, one should try to ascertain that all (or at least most) of the needed input information can be obtained. This surveying and screening of input data and contextual information has the best chance of yielding a coarse assessment of the likelihood of success. If we have convincing answers to these questions, let's go ahead! If not, let's go back to the drawing board, ponder our modeling goals and strategies within the biological context, and possibly reframe the specific questions to be tackled.

The setting of goals, combined with an exploration of available information and the context for the model, is arguably the most crucial of all modeling phases. This assessment may sound counterintuitive, because one might think that the most effort would go into some complicated math. However, if insufficient consideration is given to this initial exploration, the modeling process may derail right away, and one may not even notice that something is not quite the way it was intended until the final phases of the modeling process.

An important side task of the input and exploration phase is to feel out whether the model may have a chance to achieve its goals, given the available data and information. "Feeling out" does not sound like rigorous mathematical terminology, but it actually describes the task quite well. It is often here where experience is needed the most: right at the beginning of the effort!

2.1.1 Questions of Scale

The first important aspects of the exploration phase are the most appropriate scales for the model, with respect to both time and space and also with respect to

the organizational level (**Figure 2.3**). These scales are typically tied to each other, because processes at the organizational level of a cell occur on a small spatial scale and usually run faster than processes at the scales of organisms or ecological systems. Suppose we are interested in trees and want to develop a mathematical model describing their dynamics. While that might sound like a fine idea, it is far too unspecific, because trees may be studied in many different ways. To model the growth of a forest or a tree plantation, the scale of organization is typically the individual tree, the spatial scale is in hectares or square kilometers, and the corresponding timescale is most likely in years. Contrast this with the photosynthesis in a leaf or needle, which in turn drives tree growth. Now the organizational level is biochemical or physiological, and the corresponding timescale is that of seconds or minutes. It is easy to imagine time and organizational scales between or outside these two scenarios.

While multiscale modeling is one of the challenging frontiers of systems biology (see Chapter 11), the attempt to accommodate too many different scales at once is very difficult and, at least at the beginning of model design, dangerous. Although there are exceptions (see, for instance, Chapter 12), it is usually not really feasible to account in the same model for the entire spectrum of processes, from the details of photosynthesis or wood formation to the long-term growth of forests. Nonetheless, the real strength of mathematical models lies in their ability to explain phenomena at one level through the analysis and integration of processes at a lower level. Thus, a good model, even of small size, may span two or maybe three levels of biological organization. For instance, it may describe the formation of a leaf in terms of genetic and biochemical processes. A model may also skip some levels. For instance, we could try to associate long-term tree growth and final size with alterations in gene expression. In this case, one might be able to translate the effects of gene modulations into the tree's growth rate, but a model of this type would not capture all the biochemical and physiological causes and effects that occur between gene expression and the total amount of wood in the plantation after 50 years.

The determination of suitable scales depends very significantly on the available data that are needed to construct, support, and validate the model. If only yearly biomass measurements of the tree stand are available, we should not even attempt to model the biochemistry that deep down controls tree growth. Conversely, if the model uses biochemical details of photosynthesis as main input data, good as they may be, this input information would hardly support a model of the growth of the tree plantation over many years.

As a different example, suppose it is our task to improve the yield of a fermentation process in which yeast cells produce alcohol. Typical experimental techniques for this task are selections of mutants and changes in the growth medium. Thus, if we intend to use a model to guide and optimize this process, the model must clearly account for gene activities and their biochemical consequences, as well as for concentrations of nutrients in the medium and their uptake by cells.

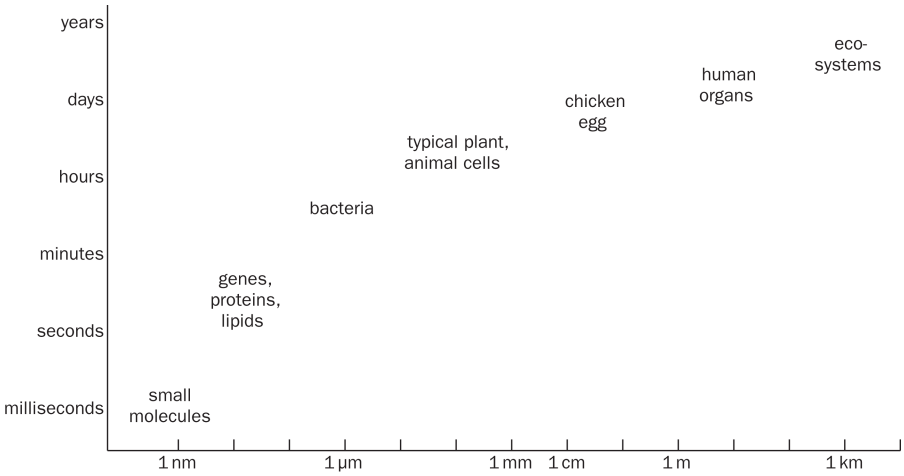


Figure 2.3 Scales in biology span several orders of magnitude in size and time. This plot shows typical sizes and time ranges. Of course, human organs and ecosystems contain small molecules and genes so that they really span multiple size and time scales, which makes multiscale modeling challenging and fascinating.

2.1.2 Data Availability

The quantity of available data provides important clues for exploring options of model selection and design. If a complicated model with many variables and parameters is supported only by very scarce data, the model may capture these data very nicely, but as soon as we try to apply it to a slightly different scenario, such as wider spacing of the trees or a richer fertilization regimen, the model predictions may become unreliable or inaccurate. This over-fitting problem (that is, using a model that is too complicated for the given data) always looms when only one or a few datasets are available. A similar problem arises if the data show a lot of **variability**—or noise, as it is often called in the field of data analysis. In this case, many models may yield a good representation of the particular data, but predictions regarding new data may be unreliable. Counterintuitive as it may sound, it is often easier to model bad data than good data, because bad data grant us a lot more latitude and potentially let us get away with an inferior model. Of course, this inferior model may turn round and bite us later when the model is expected to fit new data and simply doesn't!

While crisp, numerical data are the gold standard for mathematical modeling, we should not scoff at qualitative data or information that contains a fair amount of uncertainty. For instance, if we are told that experiments show some output increasing over time (even though the exact degree is not given), this piece of information can provide a valuable constraint that the model must satisfy. Especially in a clinical setting, this type of qualitative and semi-quantitative information is prevalent, and the more that is available, the more any model of the phenomenon is constrained. Expressed differently, this information can be extremely helpful for weeding out models that could otherwise appear plausible. In a recent study of Parkinson's disease, we asked clinicians for concentrations of dopamine and other brain metabolites that are associated with the disease. While they could not give us exact values, they provided us with rough, relative estimates of concentrations and metabolic fluxes through the system, and this information, combined with literature data, allowed us to set up a model that made surprisingly accurate predictions [1].

Finally, the exploration phase should ascertain whether experimental biologists or clinicians are really interested in the project. If so, their interest will help drive the project forward, and their direct insights and contacts with other experts can be invaluable, because much biological knowledge is not published, especially if it is semiquantitative or not yet one hundred percent supported by data and controls. It is, of course, possible to base models solely on information found in the literature, but if no experts are at hand, many pitfalls loom.

2.2 MODEL SELECTION AND DESIGN

To illustrate the process of model design, let's consider one specific example that we will carry through this chapter and revisit again in Chapter 4. This example concerns an infectious disease, such as tuberculosis or COVID-19, that easily spreads throughout a human population (**Figure 2.4**). Down the road, we would like to

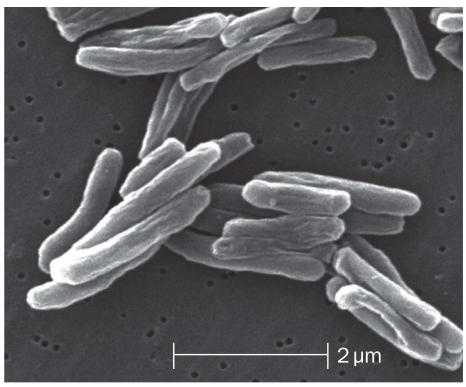


Figure 2.4 Mathematical modeling can be a powerful tool for studying the spread of bacterial diseases. Shown here is the bacterium *Mycobacterium tuberculosis*, which causes tuberculosis, one of the ongoing scourges of humankind.

determine whether it is beneficial to vaccinate all or some individuals, whether infected persons should be quarantined, or whether the disease will eventually go away by itself because people recover and acquire immunity. Immediately, many questions come to mind. Do people die from the disease? If the death rate is small, can we ignore it in the model? How does the disease get started in the first place? For how long are people sick? How infectious are they? Are there individuals who show no symptoms yet are infectious? Do all people in our target population have potential contact with all other people? Do we need to account for air travel? Should we attempt to characterize waves of infection that start at the point of the first contact with an infected person and move through certain geographical areas or the entire population? Does our disease model have to be accurate enough to predict whether a specific person will get sick, or is our target the average infection level within a community or the nation? Clearly, questions are unlimited in number and type, and some of them may be important to ponder now while others are probably less relevant, at least at the beginning.

For our purposes of studying a very complex phenomenon with as simple a model as possible, let's suppose the following:

- The infectious disease is spread exclusively by human-to-human contact
- Although bacteria or viruses are the true vectors of the disease, we ignore them
- Individuals may acquire and lose immunity
- The primary questions are how fast the disease progresses, whether it disappears without intervention due to immunity and whether vaccination, quarantine, or social distancing would be effective
- We do not intend to account for the spatial distribution of cases of infection throughout some geographical area

2.2.1 Model Structure

Once we have decided what the model is supposed to accomplish, we need to determine its structure. Some decision points are given in Table 2.1. None of these are really associated with different degrees of sophistication, usefulness, or quality of the model. We are simply required to make decisions, again based on the data and the purposes of the model.

The first decision targets the question of how much detail the model is expected to explain. As mentioned before, the two extremes in this dimension are **correlative** and **explanatory models**; most models have aspects of both. A correlative model simply and directly relates one quantity to another. For instance, data may show that the infectious disease is particularly serious among the elderly; specifically, the number of infected individuals per 100,000 may increase with age (Figure 2.5). A **linear** or **nonlinear** regression could even quantify this relationship and allow us to make predictions about the average disease risk for a person of a certain age. However, even if the prediction were quite reliable, the model would not give us a hint as to why older people are more likely to become infected. The model does not account for health status, genetic predisposition, or lifestyle. Sometimes a prediction would be wrong, but the model would not be able to explain why.

An explanatory model, by contrast, relates an observation or outcome of an experiment to biological processes and mechanisms that drive the phenomenon

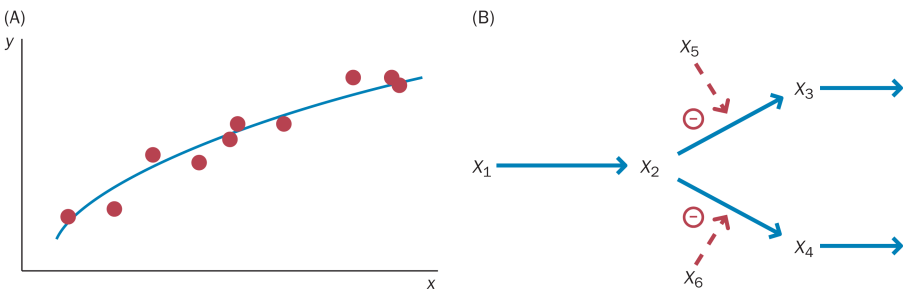


Figure 2.5 Correlative and explanatory models serve different purposes. The correlation between x and y in (A), whether linear or nonlinear, permits robust predictions of a value of y, given a new value of x, although the model does not explain the connection between x and y. An example could be disease prevalence (y) and age (x) among the elderly. The simple explanatory model in (B) could describe a metabolic pathway with two inhibitors (X_5 and X_6). It not only captures the (positive) correlation between X_3 and X_6, but also provides an explanation for why X_3 increases if the inhibitor X_6 of the other branch of the pathway is increased.

under investigation. To illustrate, imagine an explanatory model of cancer. Cancer is associated with cells that are not dying when they would in a healthy individual. Cell death normally occurs through a process called apoptosis, whose control involves the tumor suppressor protein p53, which in turn acts as a transcription factor for a number of pertinent genes. The p53 protein becomes activated under many stress conditions, such as ultraviolet radiation or oxidative stress. Pulling the pieces of information together allows us to compose a conceptual or mathematical model in the form of a chain of causes and effects: Suzie gets baked on the beach. She is exposed to enough ultraviolet radiation to activate the p53 proteins in her skin. The proteins change the normal activation of some genes. The genes cause some cells to escape apoptosis and proliferate out of control. Suzie is diagnosed with melanoma. The model explains to some degree (although not completely, of course) the development of cancer. This explanatory potential does not come for free: explanatory, mechanistic models are usually much more complicated than correlative models and require vastly more input in the form of data and assumptions. While explanatory models offer more insight, we should not discard correlative models offhand. They are much easier to construct and often make more reliable predictions than explanatory models, especially when assumptions are uncertain and data are scarce.

The second decision refers to the question of whether something associated with the phenomenon of interest changes over time. If the data show trends changing with time, we will probably prefer a **dynamic** model, which allows us to quantify these changes. By contrast, in a **static** model, such as a regression model, only the dependencies of one variable on others are explored. **Box 2.1** contrasts static and dynamic models using a specific example. Static models may appear to be insufficient, because they ignore change, but they play important roles in systems biology, where they are, for instance, used to characterize the features of large networks (see Chapter 3).

BOX 2.1: STATIC AND DYNAMIC MODELS

Suppose it is our task to compute the volume of an *Escherichia coli* cell. Studying a photograph of an *E. coli* culture (**Figure 1**), we see that each bacterium looks more or less like a hot dog with variable length, and since the volume doesn't change if the bacterium curves up, we can just study the shape when it is stretched out. Thus, a simple static model could be a cylinder of appropriate length with two "caps" at the end. The caps look a bit flatter than hemispheres and we decide to represent them mathematically by halved ellipsoids that have the same axis radius in two directions (a circular cross section) and a shorter axis radius in the third direction (**Figure 2**). The formula for the volume of a whole ellipsoid is $\frac{4}{3}\pi r_1 r_2 r_3$, where r_1, r_2, r_3 are the axis radii. Thus, if the "straight" part of the cell is x µm long and has a diameter of y µm, and if each cap has a height of z µm, we can formulate a static model for the model of an. *E. coli* cell (**Figure 3**) as

$$V(x,y,z) = \underbrace{x\pi\left(\frac{y}{2}\right)^2}_{\text{cylinder}} + \underbrace{2\left(\frac{4}{3}\pi\frac{y}{2}\frac{y}{2}\frac{z}{2}\right)}_{\text{twohalf-ellipsoildalcaps}} \quad (1)$$

$$= \left(\frac{x}{4}+\frac{z}{3}\right)\pi y^2. \quad (2)$$

Our static model allows us to pick any length, width, and cap height and to compute the corresponding volume of the bacterium.

Let's now suppose the bacterium has just divided and is growing. The size parameters x, y, and z are now variables that depend on time t, and we may write them as $x(t)$, $y(t)$, and $z(t)$. It may even happen that these variables do not develop independently from each other, but that the bacterium maintains some degree of proportionality in its shape, which would mathematically connect $x(t)$, $y(t)$, and $z(t)$ through constraint relationships. The main consequence is that the volume is no longer static, but a function of time. The model has become a **dynamic model**.

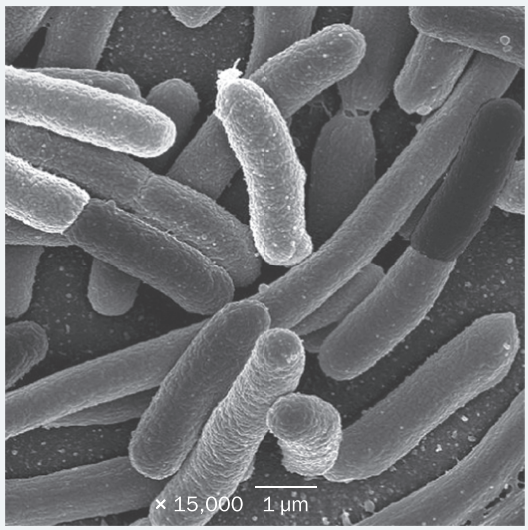


Figure 1 The shapes of *E. coli* bacteria resemble cylinders with caps. The shape may be modeled approximately with the geometrical shapes shown in Figures 2 and 3.

In the typical understanding of dynamic models, one expects that the current state of the object or system will influence its further development. In the case of a bacterium, the growth in volume very often depends on how big the bacterium already

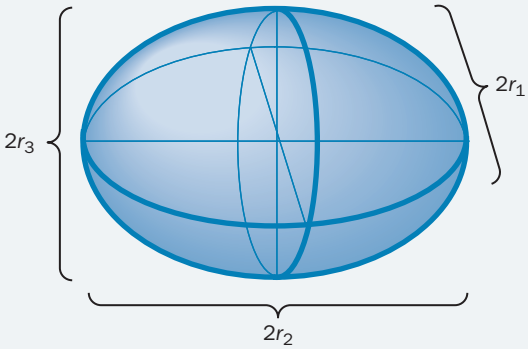


Figure 2 To model the shape of a bacterium (see Figure 1), we need a cylinder and "caps" at both ends. The caps can be modeled with flat ellipsoids. Here, such an ellipsoid is shown with two longer radii of the same length $r_1 = r_2$ and one shorter radius r_3.

is. If it is small, it grows faster, and as it reaches its final size before it divides, growth slows down. This "self-dependence" makes the mathematics more complicated. Specifically, one formulates the change in volume over time as a function of the current volume. The change in volume over time is given in mathematical terminology as its **time derivative** dV/dt. Thus, we come to the conclusion that volume-dependent growth should be formulated as

$$\frac{dV}{dt} = \text{some function of volume } V \text{ at time } t \tag{3}$$

$$= f\big(V(t)\big) = f(V). \tag{4}$$

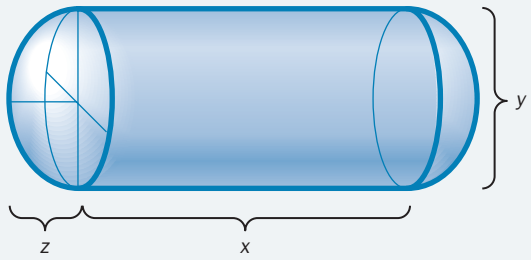


Figure 3 Static model of the shape of *E. coli*. The model is inspired by the actual cell shape (Figure 1) and composed of a cylinder with caps at both ends (Figure 2).

This type of formulation, in which the derivative of a quantity (dV/dt) is expressed as a function of the quantity (V) itself, constitutes an **ordinary differential equation**, which is also lovingly called a "diff-e-q," o.d.e., or **ODE**. The argument t is usually omitted as it is considered obvious. Very many models in systems biology are based on differential equations. The function f is not limited to a dependence on V; in fact, it most often contains other variables as well. For example, f could depend on the temperature and the substrate concentration in the medium—which will certainly affect the speed of growth—and on a lot of "internal" variables that describe the uptake mechanisms with which the bacterium internalizes the substrate. Looking more closely, many biochemical and physiological processes are involved in the conversion of substrate into bacterial volume and growth, so that a very detailed dynamic model can quickly become quite complicated. Setting up, analyzing, and interpreting these types of dynamic models is at the core of computational systems biology.

Within the realm of dynamic models, one needs to decide whether a **discrete** or **continuous timescale** is better suited. An example of the discrete case is a digital clock, which displays time only with a certain resolution. If the resolution is in minutes, the clock does not give any indication whether the minute just started or is almost over. A continuous, analog clock would, in principle, show the time exactly, although practically only to a resolution of seconds. On the one hand, the decision on a continuous or discrete timescale is very important, because the former usually leads to differential equations (see Chapter 5) and the latter to iterative or recursive maps, which require different mathematical methods of analysis and sometimes lead to different types of insights (see Chapter 4). On the other hand, if we select very small time intervals, the **behaviors** of discrete models approach those of continuous models, and differential equations, which are often easier to analyze, may replace the difference equations.

The convergence of discrete and continuous models for increasingly smaller time steps is not the only instance where one formulation is replaced with another. More generally, we need to embrace the idea of **approximations** in other dimensions as well (**Figure 2.6**). We are certainly interested in "true" descriptions of all processes, but these do not really exist. Even the famous laws of physics are not strictly "true." For instance, Newton's law of gravity is an approximation within the context of Einstein's general relativity theory. Not that there would be anything wrong with using Newton's law: apples still fall down and not up. But we should realize very clearly that we are in truth dealing with approximations and that these are both necessary and indeed very useful. The need for approximations is much more pronounced in biology, where laws corresponding to those in physics are

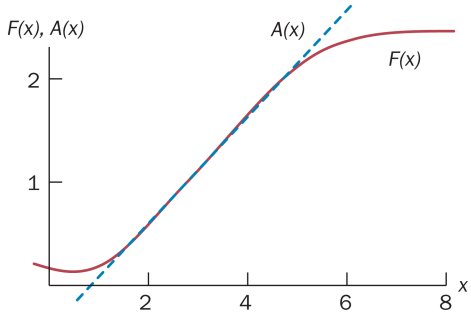


Figure 2.6 Approximations are necessary and very useful in a mathematical model for simplified representations of reality. In this illustration, $A(x)$ (dashed blue line) is a linear approximation of a more complicated nonlinear function $F(x)$ (red line). If only the range between 2 and 5 for the variable x is biologically relevant, the approximation is sufficiently accurate and easier to analyze.

almost nonexistent. Chapter 5 discusses typical approximations used in systems biological modeling and Chapter 15 discusses more philosophical issues of laws in biology.

Somewhat of an approximation is the intentional omission of spatial considerations. In the real world, almost everything occurs in three spatial dimensions. Nonetheless, if the spatial aspects are not of particular importance, ignoring them makes life much easier. With diabetes, for instance, it may be important to study the distribution of glucose and insulin throughout specific organs and the cardiovascular system of the body. But for modeling the administration of insulin in a given situation, it may also be sufficient to consider merely the overall balance between glucose and insulin. A model that does not consider spatial aspects is often called (spatially) **homogeneous**. Models accounting for spatial aspects typically require partial differential equations or entirely different modeling approaches (see Chapter 11), which we will not discuss further at this point.

Finally, a systems model may be **open** or **closed**. In the former case, the system receives input from its **environment** and/or releases material as output, while in the latter case materials neither enter nor leave the system.

The model types we have discussed so far are called **deterministic**, which simply means that all information is completely known at the beginning of the computational experiment. Once the model and its numerical settings are defined, all properties and future responses are fully determined. In particular, a deterministic model does not allow any types of **random** features, such as environmental fluctuations or noise affecting its progress. The opposite of a deterministic model is a **probabilistic** or **stochastic** model. The latter term comes from the Greek word *stochos* meaning aim, target, or guess and implies that a stochastic process or stochastic model always involves an aspect of chance. Somewhere, somebody in the system is rolling dice, and the result of each roll is fed into the model. As a consequence, the variables become **random variables** and it is not possible to predict with any degree of certainty the outcome of the next experiment or **trial** as it is often called. For instance, if we flip a coin, we cannot reliably predict whether the next flip will result in a head or tail. Nevertheless, even if we do not know exactly what will happen in future trials, we can compute the probability of each possible event (0.5 for each coin toss), and the more often we repeat the same random trial, the closer the collective outcome approaches the computed probability, even though each new trial again and again exhibits the same unpredictability.

It is no secret that biological data always contain variability, uncertainties, measurement inaccuracies, and other types of noise. Thus, as a final point to ponder when choosing a model: should we—must we—always use a model that somehow captures this randomness? Not necessarily. If we are primarily interested in average model responses, a deterministic model will generally do just fine. However, if we need to explore all possible outcomes, the worst and best cases, or the likelihood of specific outcomes from a system with a lot of uncertainty, we may be best advised to choose a stochastic model.

Because this is an important and subtle issue, let's study similarities and differences between deterministic and stochastic models in some more detail. As we discussed, each trial in a stochastic system is unpredictable, but we can make very accurate predictions about average or long-term behaviors. By comparison, deterministic models present us with situations where everything is exactly and unambiguously defined. Yet, there are strange cases where the behavior of the system

over time is so complicated that we cannot forecast it. A quite simple but very instructive example is the blue-sky catastrophe, which is discussed in **Box 2.2**. This system is completely determined and does not contain any noise or stochastic features. All information about the system is fixed in its definitions and settings, and yet the system is able to generate chaotic time courses that are impossible to anticipate. Chapter 5 discusses more cases of this nature.

BOX 2.2: AN UNPREDICTABLE DETERMINISTIC MODEL

Thompson and Stewart [2] describe a very interesting system that can be formulated as a pair of ordinary differential equations. The system, called the blue-sky catastrophe, is completely deterministic yet can produce responses that are utterly unpredictable, unless one actually solves the differential equations. The system has the form

$$\frac{dx}{dt} = y, \qquad (1)$$

$$\frac{dy}{dt} = x - x^3 - 0.25y + A\sin t.$$

If we set $A = 0.2645$ and initiate the system with $x = 0.9$, $y = 0.4$, both variables quickly enter an oscillation that is quite regular; the time course for x over the first 100 time units is shown in **Figure 1(A)**. However, if we think we understand what the system is doing, we are wrong. Continuing the same example until $t = 500$, without *any* changes to the model or its parameters, presents big surprises: out of the blue, the system "crashes" and undergoes regular or irregular oscillations at a much lower level, before it eventually returns to something like the original oscillation (Figure 1B). Continuing further in time shows irregular switches between oscillations around 1 and oscillations around −1. Intriguingly, even minute changes in the initial values of x and y or the parameter A totally change the appearance of the oscillation (Figure 1C, D). In fact, this chaotic system is so fickle that even settings in the numerical solver affect the solution. The plots in Figure 1 were computed in the software PLAS [3] with the standard local error tolerance of 10^{-6}. If one reduces the tolerance to 10^{-9} or 10^{-12}, or if one uses a different algorithm, the appearance of the figures is strikingly different.

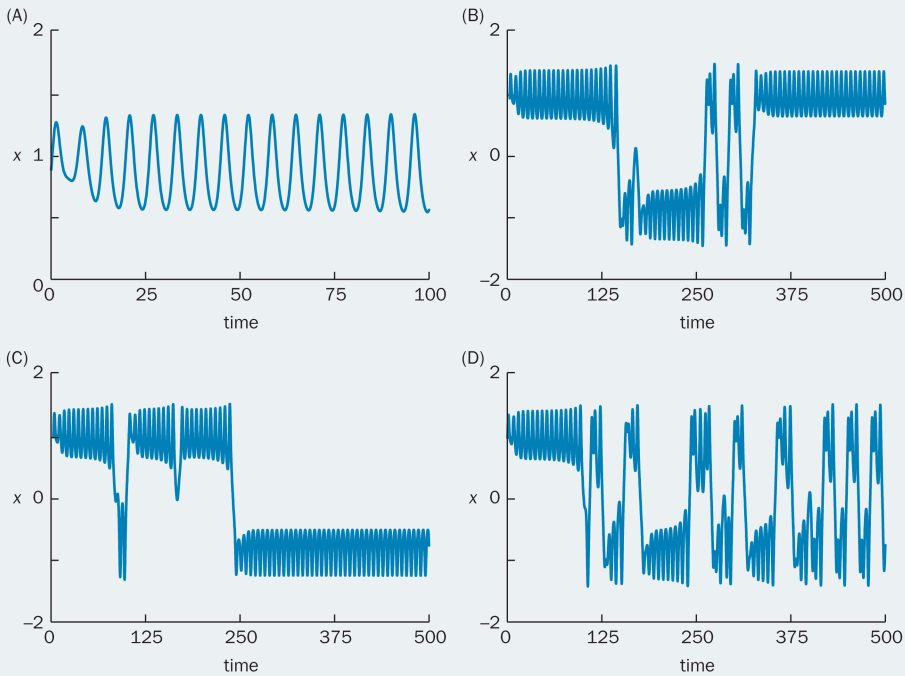


Figure 1 The intriguing model of a blue-sky catastrophe. The system of two differential equations (with $x_0 = 0.9$, $y_0 = 0.4$, and $A = 0.2645$) appears to oscillate quite regularly about a baseline value of 1 (A). However, without any changes or interventions, the oscillations later become irregular (B: note the different scales). Even very slight changes in numerical features or using a different numerical solver can change the appearance of the oscillations dramatically. (C) $A = 0.265$, $x_0 = 0.9$, $y_0 = 0.4$. (D) $A = 0.2645$, $x_0 = 0.91$, $y_0 = 0.4$. which is 2 in 100,000!

As an example of a stochastic process, suppose we irradiate a culture of microorganisms with ultraviolet light in order to introduce random mutations (see, for example, [4]). This method of *random mutagenesis* is often used to create strains that out-compete the wild type in a specific situation of interest. For instance, we may want to find a strain with improved tolerance to acid in its environment. We know from genetics that most mutations are neutral or decrease rather than

increase fitness. Nonetheless, one mutation in a million or a billion may improve the microorganism's survival or growth rate, and this particular mutation often rises to the top and multiplies in a properly designed random mutagenesis experiment.

For simplicity, let's suppose that the mutations occur randomly throughout the genome with a rate of between three and four mutations per kilobase (1,000 base pairs of DNA) [5]. A typical question in this scenario is: what is the probability of actually finding three or four mutations in a given DNA segment that is one kilobase long? Our first inclination might be to claim (wrongly) that the probability is 1, because isn't that what we just assumed? Not quite. We assumed that the *average* number of mutations is between three and four and need to realize that it could happen that there are only one or two mutations, or five or six, owing to the randomness of the mutation process. Models from probability theory allow us to assess situations of this type. For instance, we can forecast the probability of encountering one, two, or three mutations in a given piece of DNA. Or we can make statements like "with a probability of p percent, we will find at least four mutations." Would it be possible to find 200 random mutations within 1,000 base pairs? In a true stochastic process, this outcome is indeed theoretically possible, although its probability is extremely low.

Even in this clear-cut scenario with precise conditions and well-defined questions, different models are available. For instance, we could use the binomial model or the Poisson model. Both are applicable in principle. The binomial model would be better suited for small numbers (here, short segments of DNA), while it would be very cumbersome for large numbers (long segments of DNA). In fact, a pocket calculator would not be able to manage the numbers that would have to be computed for 1,000 bases. The Poisson model uses a clever approximation, which is not very accurate for small numbers (short segments of DNA) but is incomparably easier to use and misses the correct results for large numbers (long segments of DNA) by only very small errors. **Boxes 2.3 and 2.4** discuss these two models as typical representatives of simple stochastic models.

BOX 2.3: BINOMIAL PROCESS: THE ARCHETYPICAL STOCHASTIC MODEL

In the text, we asked: what is the probability of finding a certain number of mutations in a DNA segment, if one typically observes three or four mutations per 1,000 base pairs? Let's do some math to find out. We formulate the problem as a binomial process, for which we need two ingredients, namely the average mutation rate, which we assume to be 3.5 (instead of "3 or 4") mutations per kilobase, and the number of nucleotides within the DNA segment of interest. From this information, we can compute how likely it is to find 3, 4, 1, 20, 0, or however many mutations within some stretch of DNA.

In the language of probability theory, a mutation in our experiment is called a success and its (average) probability is usually termed p; in our case, p is 3.5 in 1,000 = 0.0035. A failure means that the base is not mutated; its probability is called q. Because nothing else can happen in our experiment (either a base is mutated or it is not; let's ignore deletions and insertions), we can immediately infer $p+q=1$. Put into words: "for a given nucleotide, the probability of a mutation plus the probability of no mutation together are a sure bet, because nothing else is allowed to happen."

In order to deal with more manageable numbers, let's first assume that the DNA piece contains only $n=10$ bases. Then the probability P to find k mutations is given as

$$P(k;n,p)=\frac{n!}{k!(n-k)!}p^k(1-p)^{n-k} \tag{1}$$

according to the binomial formula of probability theory, which we will not discuss further. In this formulation, the term $n!$ (pronounced "n factorial") is shorthand for the product $1\,2\,3\ldots n$. The semicolon within the function indicates that k is the independent variable, while n and p are constant parameters, which may, however, be changed from one example to the next. So, what is the probability of finding exactly one mutation in the 10-nucleotide segment? The answer results from substituting values for the parameters and for k, namely,

$$P(1;10,0.0035)=\frac{10!}{1!9!}0.0035^1(1-0.0035)^9=0.0339. \tag{2}$$

That is not a very big probability, so the result is rather unlikely, namely roughly 3%. What about two mutations? The answer is

$$P(2;10,0.0035)=\frac{10!}{2!8!}0.0035^2(1-0.0035)^8=0.000537, \tag{3}$$

which is even smaller. It is not difficult to see that the probabilities become smaller and smaller for higher numbers of mutations. So, what's the most likely event? It is that there is no mutation at all. This probability is given by the binomial **probability distribution** (1) just like the others, namely,

$$\begin{aligned}P(0;10,0.0035)&=\frac{10!}{0!10!}0.0035^0(1-0.0035)^{10}\\&=0.9655.\end{aligned} \tag{4}$$

In over 96% of all 10-nucleotide DNA segments, we expect to find no mutation at all!

From these three results, we can estimate the chance of finding more than two mutations. Because all probabilities taken together must sum to 1, the probability of more than two mutations is the same as the probability of *not* having zero, one, or two mutations. Thus, we obtain

$$P(k>2)=1-0.9655-0.0339-0.000537\approx 0.00002, \tag{5}$$

If we increase the length of the DNA piece, we should intuitively expect the probabilities of finding mutations to increase. Using the same formula, the probability of exactly one mutation in a 1,000-nucleotide segment can be written as

$$P(1;1000,0.0035)=\frac{1000!}{1!999!}0.0035^{1}(1-0.0035)^{999} \quad (6)$$

This is a perfectly fine formulation, but our pocket calculator goes on strike. 1,000! is a huge number. To ameliorate the situation, we can play some tricks. For instance, we see from the definition of factorials ($n!$ = 1 2 3 . . . n) that the first 999 terms in 1,000! and 999! are exactly the same and cancel out from the fraction, leaving us simply with 1,000 for the first term. With that, we can compute the probability of 1 mutation as 0.1054 and the probability of no mutations at all as 0.0300 (0! is defined as 1). Indeed, in contrast to the scenario with 10 bases earlier, we are now about three times more likely to encounter one mutation than no mutation at all. Is it imaginable that we would find 10, 20, or 30 mutations? The probability of finding 10 mutations is about 0.00226. Formulate the corresponding probabilities for 20 and 30. It is not hard to do, but even with the aforementioned tricks we run into problems computing the numerical results from the formulae. **Box 2.4** offers a solution!

BOX 2.4: THE POISSON PROCESS

The binomial model runs into problems for large numbers k and n. As a demonstration that modeling problems often permit different solutions, we provide here an alternative to the binomial model. The mathematician Siméon-Denis Poisson (1781–1840) developed a formula, nowadays called the Poisson distribution, that lets us compute binomial-type probabilities for large numbers k and n. The two ingredients we need are the mutation rate λ and the number k of mutations whose probability we want to compute. How could Monsieur Poisson get away with only two pieces of information instead of three? The answer is that he no longer looks at the success or failure at each nucleotide but substitutes the step-by-step checking with a fixed rate that is really only valid for reasonably long stretches or DNA. The mutation rate with respect to 1,000 nucleotides is 3.5, whereas the rate for 100 nucleotides is one-tenth of that, namely 0.35. Using a fixed rate makes life simple for large numbers of nucleotides but does not make much sense for very short DNA pieces.

Poisson's formula, which is again based on probability theory [6], is

$$P(k;\lambda)=\frac{e^{-\lambda}\lambda^{k}}{k!}. \quad (1)$$

Thus, the probability of 10 mutations (in a 1,000-nucleotide segment) is

$$P(10;3.5)=\frac{e^{-3.5}3.5^{10}}{10!}=0.00230. \quad (2)$$

The probabilities of 3 or 4 mutations are 0.216 and 0.189, respectively. The two together make up about 40% of all cases, which appears reasonable, because the overall mutation rate is 3.5 per 1,000 nucleotides. The probability of no mutation is 0.0302, which is very similar to the computation with the binomial model. With the Poisson formula, we can compute all kinds of related features. For instance, we may ask for the probability of more mutations (anywhere between 5 and 1000 mutations). Similar to the binomial case, this number is the same as the probability of *not* finding 0, 1, 2, 3, or 4 mutations. The probabilities of 1 or 2 mutations are 0.106 and 0.185, respectively; the others we have already computed. Thus, the probability of finding 5 or more mutations is approximately 1 – 0.0302 – 0.106 – 0.185 – 0.216 – 0.189 = 0.274, which corresponds to almost a third of all 1,000-nucleotide strings.

According to probability theory, the Poisson limit theorem states that the Poisson distribution becomes an increasingly better approximation of the binomial model if n becomes ever larger and p ever smaller.

2.2.2 System Components

Biological systems models contain different classes of components, which we discuss in this section. The most prominent components are variables, which represent biological entities of interest, such as genes, cells, or individuals. Because biological systems usually contain many variables, they are called **multivariate**. Variables may represent single entities or collections or pools of entities, such as all insulin molecules within a body, no matter where they are located. The second class of components consists of processes and interactions, which in some fashion involve and connect the variables. The third class contains **parameters**. These are numerical characteristics of the system, such as pH or temperature or the turnover rate of a chemical reaction, which is the number of molecules the reaction can process within a given amount of time. The particular values of the parameters depend on the system and its environment. They are constant during a given computer experiment but may assume different values for the next experiment. Finally, there are (universal) constants, such as π, e, and Avogadro's number, which never change.

In order to design a specific formulation of a model, we begin with a diagram and several lists, which are often developed in parallel and help us with our bookkeeping. The diagram contains all entities of the biological system that are of interest and indicates their relationships. These entities are represented by variables

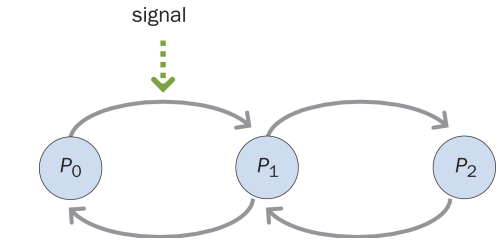


Figure 2.7 Different states of a protein in a signaling cascade. The protein may be unphosphorylated (P_0) or singly (P_1) or doubly (P_2) phosphorylated. The total amount of the protein remains the same, but, during signal transduction, material flows among the pools, driven by kinases and phosphatases, which are not shown here (see Chapter 10).

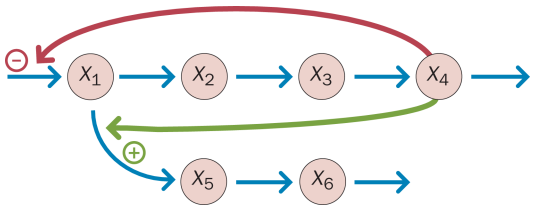


Figure 2.8 Generic pathway system, in which X_4 inhibits its own production process and activates the generation of an alternative pathway toward X_5 and X_6. The flow of material is represented with blue arrows, while modulation processes are shown with red (inhibition) and green (activation) arrows that distinguish them from flow arrows.

and drawn as nodes. Connections between nodes, called edges, represent the flow of material from one node to another. For instance, a protein in a signaling cascade may be unphosphorylated or phosphorylated in one or two positions. Within a short time span, the total amount of the protein is constant, but the distribution among the three forms changes in response to some signal. Thus, a diagram of this small protein phosphorylation system might look like Figure 2.7.

In contrast to the flow of material, a diagram may also contain the flow of information. For instance, the end product of a pathway may signal—through **feedback** inhibition—that a smaller amount of further substrate should be used. This signal is distinctly different from the flow of material, because sending out the signal does not affect the sender. It does not matter to a billboard how many people look at it. Because of this difference, we use a different type of arrow. This arrow does not connect pools but points from a pool to a flow arrow (edge). As an illustration, the end product X_4 in Figure 2.8 inhibits the process that ultimately leads to its own generation and activates an alternative branch toward X_5. These modulation processes are represented with differently colored or other types of arrows to make their distinction from edges clear.

Of course, there are many situations where we do not even know all the pertinent components or interactions in the system. Different strategies are possible to deal with such situations. In opportune cases, it is possible to infer components, interactions, or signals from experimental data. We will discuss such methods in later chapters. If such methods are not feasible, we design the model with the best information we have and explore to what degree it answers our questions and where it fails. While we understandably do not like failures, we can often learn much from them. Indeed, we typically learn more from failures than from successes, especially if the model diagnosis points toward specific parts of the model that are most likely the cause of the failure.

Parallel to drawing out the diagram, it is useful to establish four lists that help us organize the components of the model. The first list contains the key players of the system, such as metabolites in a biochemical system, genes and transcription factors in a gene regulatory network, or population sizes of different species in an ecological study. We expect that these players may change over time in their concentrations, numbers, or amounts and therefore call them **dependent variables**, because their fate depends on other components of the system. For instance, it is easy to imagine that variable X_2 in Figure 2.8 depends on X_1 and other variables.

The system may also include **independent variables**, which we collect in the second list. An independent variable has an effect on the system, but its value or concentration is not affected by the system; it is in most cases constant during a computational experiment. For example, this type of variable is often used in metabolic systems to model enzyme activities, constant substrate inputs, and cofactors. Nonetheless, it may also happen that forces outside the system cause an independent variable to change over time. For instance, a metabolic engineer

might change the amount of nutrients flowing into a bioreactor. The variable is still considered independent, if its changes are not caused by the system.

The distinctions between dependent and independent variables on the one hand and between independent variables and parameters on the other are not always clear-cut. First, a dependent variable may indeed not change during an experiment and could therefore be considered independent. This replacement would make the model simpler but might preclude analyses of other model settings where the variable does change. It actually happens more frequently that independent variables are made dependent in a refinement of the model that accounts for previously ignored effects of the system on the variable. Second, we could replace a constant independent variable with a parameter. However, it is generally beneficial to distinguish parameters and independent variables because they often have a slightly different biological meaning and the computational effort of using an independent variable or a parameter is the same.

The third list is actually closer to a table or spreadsheet. It shows which of the (dependent or independent) variables have a direct effect on any of the processes in the system. As in the example of **Figure 2.8**, an effect may be positive (enlarging or activating) or negative (diminishing or inhibiting). The list accounts for all edges (flow of material between pools), as well as all signals that affect any of the edges.

The final list contains parameters that contribute to the external or internal conditions of the system, such as pH, temperature, and other physical and chemical determinants, which will ultimately require numerical specification.

Throughout the model design process, we add to these lists quantitative information about pool sizes, magnitudes of fluxes, strengths of signals, and normal values and ranges of parameters.

The lists are the basis for establishing a diagram of the system. For the design of this diagram, it is beneficial to begin by representing each variable with one of the two core modules in **Figure 2.9**. In the simpler case (**Figure 2.9A**), each variable is displayed as a box with one process entering and one exiting. In the end, there may be more or fewer arrows, but using one influx and one efflux is a good default, as it reminds us later to include these processes when we set up equations. Indeed, a dependent variable tends to deplete without influx and to keep accumulating without efflux. Of course, there are situations where depletion or accumulation is the desired behavior, and if so, we can secondarily remove the corresponding process. However, these situations are quite rare. We may also use the default in **Figure 2.9B**, which reminds us that many processes are regulated by other variables and that the variable in question may send out signals that could affect the influxes or effluxes of other variables. Independent variables typically do not have an influx, but they have an efflux if they feed material into the system. Once all variables are defined, they need to be functionally connected: the efflux of one variable may become the influx of another, and the signal sent by one variable may affect the influxes or effluxes of other variables.

Let's illustrate this construction of a diagram with the example of a population that is experiencing the outbreak of an infection. The model is intended to answer some of the questions mentioned earlier. Following some old ideas of Kermack and McKendrick [7], we use a similar terminology and keep things as simple as possible; one should, however, note that thousands of variations on this model have been proposed since Kermack and McKendrick's days [8]. We begin by defining just three dependent variables, namely the number of individuals susceptible to the disease (S), the number of infected individuals (I) and the number of individuals that are "removed" (R) from the two pools, because they have acquired immunity. We suppose that all individuals could be in contact with each other, at least in principle, and assume that a certain percentage of the immune individuals

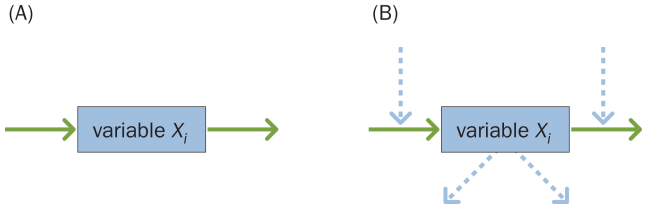


Figure 2.9 Default core modules. When designing a diagram of a system, it is beneficial to augment the display of each variable with an influx and efflux (A) and possibly with modification signals that affect the variable or are sent out by the variable (B).

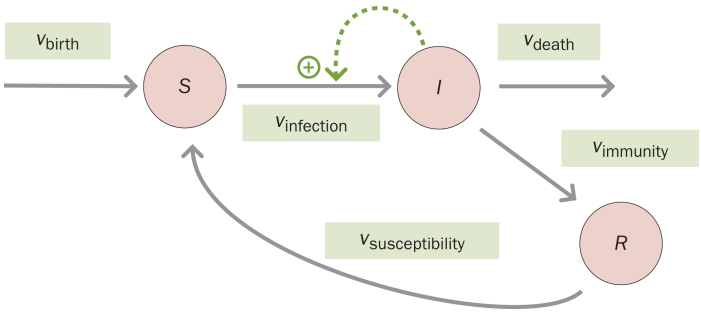


Figure 2.10 Diagram of a model for the spread of an infectious disease within a population. The model contains three dependent variables: *S*, the number of individuals susceptible to the disease; *I*, the number of infected individuals; and *R*, the number of immune ("removed" in the original terminology) individuals. Each process, in which people move from one pool to another, is shown as an edge and coded as *v* with a specific index. The process v_{birth} represents birth or immigration, $v_{infection}$ is the infection process, which requires contact between a susceptible and an infected individual. The process $v_{immunity}$ models the fact that infected individuals may acquire immunity. Infected people may also die from the disease (v_{death}). Finally, $v_{susceptibility}$ represents the rate at which individuals lose their immunity and become susceptible again. An immediately visible simplification is that *S* and *R* do not die unless they are first infected. The model does not include independent variables and signals.

(*R*) lose their immunity and become susceptible again. We also allow for the possibility that individuals are born or immigrate and that individuals may die while infected. A diagram summarizing this population dynamics is shown in Figure 2.10. Processes such as infection, recovery, and death are denoted with indexed variables v. For instance, the infection process is coded as $v_{\text{infection}}$ and involves the contact between individuals of types *S* and *I*. The label v is often used in systems models, because it characterizes the velocity of a process. Some arrows have no source (v_{birth}), which means that we do not model explicitly where the individuals (or material) come from. Similarly, arrows may not have a target (v_{death}) because we do not account for their fate; individuals or molecules simply leave the system.

2.2.3 Model Equations

Once we have completed our model diagram and established the lists of components and processes, we need to translate this information into equations that reflect the chosen model structure. At first, these equations are **symbolic**, which means that numerical values are not yet assigned to the processes that characterize the system.

For the infectious disease problem, we choose a simple differential equation model, called an **SIR** model [7], which by its nature is moderately explanatory, dynamic, continuous, deterministic, and independent of spatial aspects. More complicated choices could include random processes associated with the infection of susceptible individuals, age distributions within the population with different degrees of susceptibility and recovery, or an explicit account of the pathogen titer and the immune response (see Chapter 11).

Like Kermack and McKendrick [7], we use as a model structure the simple and intuitive process description of mass action kinetics. This structure entails that the terms in each equation are set up either as constants (birth and immigration) or as a rate constant multiplied with the dependent variables that are directly involved in the process. This strategy reflects that the number of individuals moving from one pool to another depends on the size of the source pool. The more infected individuals there are, for instance, the more people will acquire immunity.

The specific construction is implemented one differential equation at a time and requires us to introduce new quantities, namely the **rates** with which the various processes occur. The change in *S* is governed by two augmenting processes, namely birth and immigration (v_{birth}) with a collective constant rate r_B, and replenishment of the susceptible pool with individuals losing immunity ($v_{\text{susceptibility}}$) with rate r_S. At the beginning, no immune individuals are assumed to be around ($R = 0$), but the model accounts for the possibility of initially immune individuals. The pool *S* is diminished by the infection process ($v_{\text{infection}}$), which has a rate r_I and depends on both *S* and *I*, because an infection requires that a susceptible and an infected individual come into contact. Thus, the first equation reads

$$\frac{dS}{dt} = \dot{S} = r_B + r_S R - r_I SI, \quad S(t_0) = S_0. \tag{2.1}$$

Here we have snuck in a new symbol, namely a variable with dot, $\dot{S}$. This notation is shorthand for the derivative of this variable with respect to time. We have also added a second equation, or rather an assignment, namely the value of S at the beginning of the computational experiment, that is, at time t_0. Because we don't want to commit to a numerical value yet, we assign it the symbol S_0. This initial value must eventually be specified, but we do not have to do it right now. The system does not contain independent variables.

The equations for I and R are constructed in a similar fashion, and the entire model therefore reads

$$\dot{S} = r_B + r_S R - r_I SI, \quad S(t_0) = S_0 \tag{2.2}$$

$$\dot{I} = r_I SI - r_R I - r_D I, \quad I(t_0) = I_0, \tag{2.3}$$

$$\dot{R} = r_R I - r_S R, \qquad R(t_0) = R_0. \tag{2.4}$$

Here, r_R is the rate of acquiring immunity and r_D is the rate of death. The set-up implies that only infected individuals may die, which may be a matter of debate or of a later model extension.

We can see that several terms appear twice in the set of equations, once with a positive and once with a negative sign. This is a very common occurrence, because they describe in one case the number of individuals leaving a particular pool and in the other case the same number of individuals entering another pool.

The equations thus constructed are symbolic, because we have not yet committed to specific values for the various rates. Therefore, this set of symbolic equations really describes infinitely many models. Many of these will have similar characteristics, but it may also happen that one set of parameter values leads to very different responses than a different set.

2.2.4 Parameter Estimation

No matter which particular type of model we select, the model structure alone is usually not sufficient for a comprehensive assessment. For most specific analyses, we need numerical values for the model parameters. In a statistical model, such a parameter may be the arithmetic mean. In a stochastic system, it might be the average rate of an event in a binomial trial. In a deterministic model, we usually need sizes or quantities of variables, rates of processes, and characteristic features of the model components. As an example, let's consider a mathematical function that is often used in biochemistry to describe the conversion of one type of molecule, called the substrate, into a different type, called the product. This conversion is usually facilitated by an enzyme. We will discuss biochemical conversion processes, along with their context and rationale, in Chapters 5, 8, and 9. Here it is sufficient to state that such a conversion is frequently modeled either with a so-called **Michaelis–Menten function** or with a **Hill function**. Both have the mathematical form

$$v(S) = \frac{V_{\max} S^n}{K_M^n + S^n}. \tag{2.5}$$

The parameter $V_{\max}$ describes the maximal speed (velocity) of the conversion, which is attained as S becomes very large; in mathematical jargon, $V_{\max}$ is the asymptote of the function $v(S)$ as S goes to infinity. The second parameter is the Michaelis constant K_M. This describes a property of the enzyme facilitating the conversion and represents the value of S where the substrate-to-product conversion $v(S)$ runs at half-maximal speed: $v(K_M) = V_{\max}/2$. The third parameter is the Hill coefficient n. In the case of a Michaelis–Menten function, $n = 1$, while true Hill functions are typically characterized by $n = 2$ or $n = 4$; however, the Hill coefficient may also take other positive values, including non-integer values such as 1.25. Depending on the numerical values of these three parameters, the function has a different appearance (see **Figure 2.11** and Exercises).

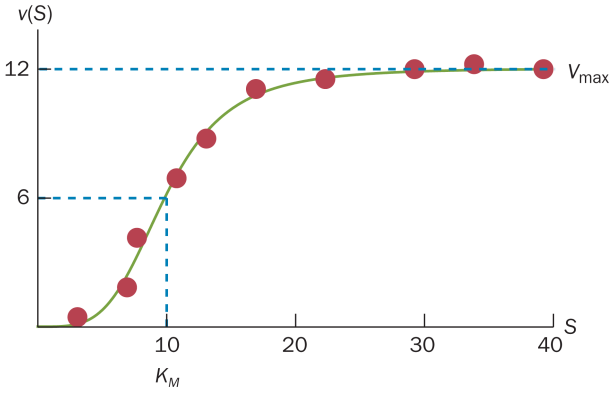


Figure 2.11 The parameters of a function determine its shape. In this case of a Hill function with $n = 4$ ($v(S) = V_{max}S^4/(K_M^4 + S^4)$; green line), the unknown parameters V_{max} and K_M can be determined from experimental data (red symbols) of $v(S)$ plotted against the substrate concentration (S). They are given, respectively, as the asymptote for large S and the value of S where $v(S)$ equals $V_{max}/2$. In the example, $V_{max} = 12$ and $K_M = 10$ (see Chapters 5, 6, and 9 for details).

Parameter values may be obtained in various ways, but their determination is almost always a complicated process (see Chapter 6). Indeed, in very many cases, parameter estimation for a systems model from actual data is the most challenging bottleneck in the entire modeling process, outside of choosing adequate functions. Two extreme classes for obtaining parameter values are called **from the bottom up** and **from the top down**; in reality, the estimation is often a mixture of the two.

In a bottom-up estimation, parameter values are collected for each system component and process. For instance, experimental methods of enzyme kinetics could be used to determine the K_M of an enzyme in the previous example (see Chapter 9). Similarly, in the SIR example, one could tally how many people die from the disease within a given period of time and translate this measurement into the parameter r_D in (2.3). Thus, parameter values are measured or collected one by one, until the model is fully specified.

Top-down estimation methods are very different. In this case, one needs experimental measurements of all dependent variables under different conditions or at successive time points. In the case of a Hill function, the top-down estimation requires data such as the red dots in Figure 2.11, which measure $v(S)$ for different values of S. In the SIR example, one would need measurements of S, I, and R at many time points, following the outbreak of the disease. In either top-down estimation, the data and the symbolic equations are entered into an optimization algorithm that simultaneously determines those values of all parameters for which the model matches the data best. Chapter 6 discusses methods of parameter estimation in detail. In our example of an infectious disease outbreak, parameter values could be derived directly or indirectly from the disease statistics established by a community health center in a bottom-up or top-down fashion.

Essentially all parameters in our case are rates (which generically describe the number of events per time) so that it is necessary to decide on a time unit. For our SIR example, we use days and set the rates correspondingly. Babies are born (or people immigrate) at a rate of 3 per day, 2% of the infected individuals actually die per day, and individuals lose immunity at a rate of 1% of R. The other rates are self-explanatory. The initial values say that, at the start of the model period, 99% of the individuals in a population of 1,000 are healthy, yet susceptible to the disease ($S = 990$), that 1% are infected ($I = 10$), for reasons we do not know, and that nobody is initially immune ($R = 0$). These settings are entered into the symbolic equations (2.1)–(2.4), and the resulting parameterized equations are thus

$$\begin{aligned} \dot{S} &= 3 + 0.01R - 0.0005SI, \quad S_0 = 990, \\ \dot{I} &= 0.0005SI - 0.05I - 0.02I, \quad I_0 = 10, \\ \dot{R} &= 0.05I - 0.01R, \quad R_0 = 0. \end{aligned} \tag{2.6}$$

In general, parameter estimation is complicated, and Chapter 6 is dedicated to the description of standard and ad hoc estimation methods. For the remainder of this chapter, we simply assume that all parameter values are available.

2.3 MODEL ANALYSIS AND DIAGNOSIS

A typical model analysis contains two phases. The first consists of model diagnostics, which attempts to ensure that nothing is obviously wrong with the model and assesses whether the model has a chance of being useful. For example, a model in which a variable disappears altogether when some physiological feature is changed by just a few percent is not very robust, and the actual natural system would not survive in the rough-and-tumble outside world for very long. After we have received green light from the diagnostics, we enter the second phase of exploring what the model is able or unable to do. Can it oscillate? Can some variable of interest reach a level of 100 units? What would it take to reduce a variable to 10% of its normal value? How long does it take until a system recovers from a **perturbation**?

Because we are dealing with mathematics, we might expect that we could directly compute all properties of a model with calculus or algebra. However, this is not necessarily so, even in apparently simple situations. As an illustration, suppose our task is to determine one or more values of x that satisfy the equation $e^x - 4x = 0$. Is there a solution? We can draw e^x and $4x$ on the same plot (**Figure 2.12**) and see immediately that the two intersect twice. Hence, the equation is satisfied at these two intersection points (see Exercise 2.5). Interestingly and maybe surprisingly, while there are two clear solutions, there is no algebraic method that would let us compute these solutions exactly. Our only systematic alternative is a numerical algorithm that (quickly) finds approximate solutions. More generally, situations in modeling where mathematical analysis must be supplemented with computer methods are the rule rather than the exception.

One very important feature of the SIR model can be assessed with purely mathematical paper-and-pencil tools. Namely, it is of obvious interest to forecast whether an infectious disease will persist or eventually die out, leading to a **disease-free equilibrium**. In the jargon of SIR modeling, the quantity characterizing this question is called the **basic reproduction number** and denoted as R^0 (pronounced R-naught). We denote it as R^0, because we already used R^0 for the initial value of R, and the two are clearly not the same.

R^0 is defined as "the average number of secondary cases produced by one infected individual introduced into a population of susceptible individuals, where an infected individual has acquired the disease, and susceptible individuals are healthy but can acquire the disease" [9]. With purely mathematical means, one can show that R^0 is related to the infection and recovery rates, as well as the initial number of susceptibles. Specifically, R^0 is defined as

$$R^0 = \beta S_0/\gamma.$$

One can prove that the infection will die out in the long run if $R^0 < 1$ and that it will spread within the population if $R^0 > 1$ [9]. The case $R^0 = 1$ is mathematically possible, but does not occur in the real world. The mathematical proof is not too hard for the simple SIR model if there are no births, no deaths, and if recovered individuals do not become susceptible again. In more complicated cases, R^0 can often be assessed mathematically as well, but the proof is more complicated.

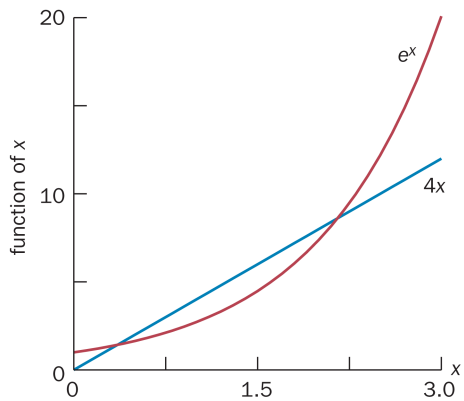


Figure 2.12 Computer simulation can be a very useful tool for model evaluation. The equation $e^x - 4x = 0$ obviously has two solutions, where the red and blue lines intersect, but we cannot compute them directly with algebraic methods. A suitable computer algorithm easily finds both solutions, although only approximately.

Values of R^0 for diseases like the flue or HIV/AIDS typically fall into a range between 2 and 8 but can reach 15 or 20, for instance, in the case of measles. Assuming a perfect vaccine, the fraction of the population to be vaccinated to prevent the sustained spread of the disease is $1 - 1/R^0$. In reality the fraction is higher, because vaccines are not 100% effective.

One should note that there are slight variations in the definition of R^0, with S_0 sometimes replaced by the total size of the population, N, or with 1, if S, I, and R are expressed as proportions rather than absolute numbers [10, 11].

2.3.1 Consistency and Robustness

Before we can rely on a model, we need to do some diagnostics. This model checking can be a lengthy process, especially if we find flaws or inconsistencies that must be ameliorated by changing or refining the model structure. The targets of the diagnostics are partly biological and partly mathematical. As a biological example, suppose that we are analyzing a metabolic network of pools and channels through which material flows. As a minimum requirement for consistency, we need to assure that all material is accounted for at every pool and every time point, that effluxes from each pool actually reach one of the other pools or correctly leave the system, and that the biochemical moieties of interest are preserved.

Many of the mathematical aspects of diagnostics follow rather strict guidelines, but the techniques are not always entirely trivial. For instance, we need to assess the robustness of the model, which essentially means that the model should be able to tolerate small to modest changes in its structure and environment. Techniques for these purposes are discussed in Chapter 5. In lieu of—or in addition to—assessing the model with purely mathematical techniques, we can learn a lot about the robustness and some of the features of the model through computational exploration. This is accomplished by using software to solve (simulate) the model under normal and slightly changed conditions. The order of items to diagnose (see Table 2.1) is not important.

For our illustration, let's start by solving the equations using a computer simulation with the baseline parameters that we set earlier. We may use for this purpose software like Java, *Mathematica*®, MATLAB®, Python, R, or the freeware PLAS [3]. **Figure 2.13** shows the result, which consists of one time course for each dependent variable. In our example, the changes within the population are quite dramatic. The subpopulation of susceptible people almost disappears, because almost everyone gets sick, but the number of immune individuals eventually exceeds half the population size. Yet, the infection does not seem to disappear within the first 100 days after the outbreak. Even after a whole year, some infectives are still present (**Figure 2.13B**); the situation is reminiscent of the persistence of diseases like tuberculosis. In general, we should always check whether the simulation results appear to make sense or are obviously wrong before we launch any comprehensive mathematical diagnostics. In our example, we cannot judge whether the model yields the correct dynamics, but at least there is nothing obviously wrong or worrisome.

Let's do some computational exploration. The first question is whether the model in (2.6) reaches a **state** where none of the variables changes anymore. The answer is *yes*: we can show such a **steady state** by solving the equations for a sufficient time range (note that the number of immune individuals still climbs after one year). After a long time, the variables eventually reach the values $S = 140$, $I = 140$, and $R = 750$, and this steady state is stable. How we assess **stability** is discussed in Chapter 5. If we compute the total population size after several years, we note that there are now more people (1,040) than at the beginning (1,000). Is that reason to worry? No, it just means that the initial values were below the "normal" steady-state size of the population, which is driven by the birth and death rates. What happens when we start with $S = 1{,}500$? Because the steady state is stable, the population will shrink and in the end approach 1,040 again (confirm this!).

Much of the remaining diagnostics characterizes the **sensitivity** of the model. The generic question here is: if one of the parameters is changed a little bit, what is the long-term effect? Sensitivity analysis can sometimes be done mathematically in one big swoop (see Chapter 5). Specifically, all changes in steady-state values, caused by a small (for example, 1%) change in each of the parameters, are com-

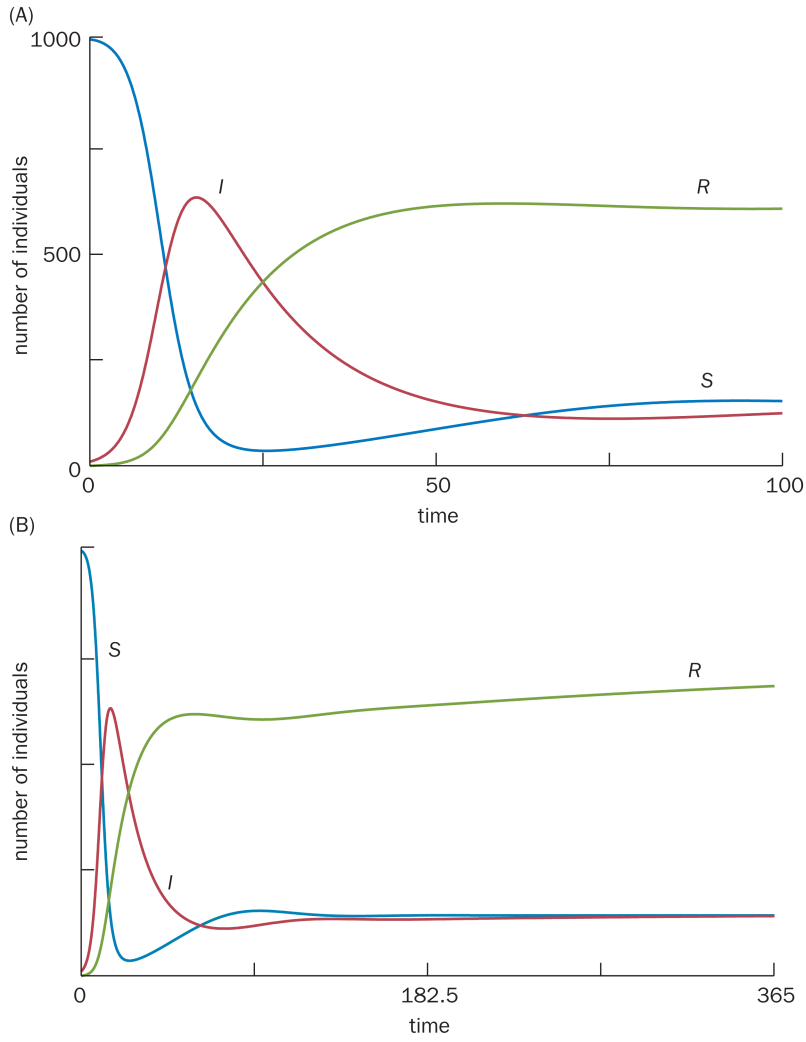


Figure 2.13 Spread of an infectious disease through a population over 100 days and 1 year. Simulations for shorter time periods (A) can provide detailed information on early transition behaviors, while simulations for extended time periods (B) can give hints regarding long-term and steady-state behaviors.

puted with methods of differentiation and linear algebra. However, we can also explore important sensitivity features by changing a parameter value manually and checking what happens per simulation (see later example). For instance, we could lower the infection rate to 20% of the present value: $r_I = 0.0001$. The consequences are dramatic, as Figure 2.14 demonstrates. Two pieces of good news are that fewer people become infected and that the total population size increases to 1,650. The bad news is that more people remain susceptible.

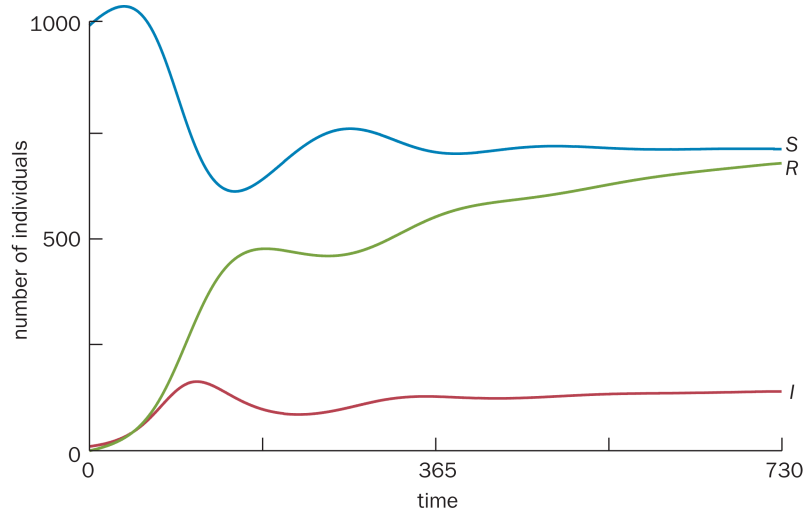


Figure 2.14 A model can be sensitive to changes in some parameters and robust against changes in others. As an example, the spread of the disease is rather different if the infection rate is smaller (here, $r_I = 0.0001$; compare with Figure 2.12).

If one has the luxury of many data, it is beneficial not to use them all for model design and parameterization and instead to keep some data for external **validation**. This validation may be accomplished intuitively or statistically. In the former (obviously simpler) approach, one compares the results of simulations with actual data and judges whether the two are sufficiently close. Often, but not always, such a simulation begins with the system at steady state and a **stimulus** that perturbs this state. Sometimes, one might be satisfied with an agreement in **qualitative behavior**, such as "If the input to the system is increased, these variables should go up, while those should decrease in value." In other cases, one may expect a model to be semi-quantitative: "If the input to the system is doubled, variable 3 should increase to between 120% and 140%." A much more stringent criterion is true numerical agreement. In addition to this type of intuitive assessment, one may use statistical methods to evaluate the quality of fit between a model prediction and observed data.

One should note that a good fit is not the only criterion of quality. A particularly prominent counterexample is over-fitting, where the model contains more parameters than can be estimated reliably from the data. While the model fit for the training data may be excellent in this case, the model tends to fail if new data are analyzed. Methods have been developed to diagnose different types of residual errors and problems of over-fitting [12, 13].

2.3.2 Exploration and Validation of Dynamical Features

Assuming that the model has survived the diagnostics of the previous phase, it is now time to explore its range of dynamical features, keeping in mind that even a comprehensive analysis might not detect all possible flaws and failures. In fact, since every model is at best a good approximation of reality, it is only a matter of time before this approximation leads to discrepancies with actual observations.

The exploration of the model consists of testing hypotheses regarding scenarios that had not been used for the model design. This process is often called a **simulation**. Some of the scenarios may correspond to situations that had already been tested biologically. In this case, a successful simulation may be used for supporting the validity of the model or for explanations of the internal operations that connect a particular input to the observed (and simulated) output. Such an explanation may lead to new hypotheses of the type "since process A is apparently more influential than process B, inhibition of A should have a stronger effect than inhibition of B." The new hypothesis may be tested mathematically, biologically, or both. It is also possible that the model makes predictions that are counterintuitive. In this case we need to check both our intuition and the model. If the model is correct, it could lead to new explanations that were not evident before.

Continuing beyond the diagnostic explorations, the scenarios of this phase are typically a little bit more complicated, and many will require changes in the model structure, that is, in the equations. These types of explorations can be used to evaluate the importance of assumptions, simplifications, and omissions. For instance, we assumed in the infectious disease model that only infected individuals die. What happens if we add mortality terms to the equations for S and R? What happens if immunity is permanent? Is it of consequence if some individuals are infected but do not show symptoms?

As a specific example, let's investigate what happens if we vaccinate before the outbreak. The first task is to figure out where to enter vaccination into the equations. Any effective vaccination before the outbreak moves susceptible individuals into the R pool. In other words, the only change occurs in the initial values, where S_0 is lowered and R_0 correspondingly elevated. The exact numbers depend on the efficacy of the vaccination campaign. It is not difficult to imagine that complete and perfect vaccination would render $S_0 = 0$, $R_0 = 1{,}000$. One could also model that vaccination would not necessarily prevent the disease altogether but only reduce the infection rate, allow faster immunity, and reduce the number of individuals in the R pool who become susceptible again.

Another scenario involves quarantine for infected individuals. Ideally, the entire pool I would immediately be subjected to quarantine until they recover, thus changing the initial value to $I_0 = 0$. It is easy to see in this extreme case that no

infection can take place, because the infection term $r_I SI$ equals zero. Therefore, no susceptibles move from S to I, and $r_I SI$ remains zero. A more interesting question is: is it really necessary to quarantine every infected person? Or is it sufficient to quarantine only a certain percentage? Let's explore this question with simulations. Suppose we continuously quarantine some of the infected individuals. Of course, actually sending infected individuals into quarantine is a **discrete** process, but we can model the process approximately by subtracting infectives with a constant rate from (2.3), which becomes

$$\dot{I} = r_I SI - r_R I - r_D I - r_Q I, \tag{2.7}$$

where r_Q is the rate of quarantining (**Box 2.5**).

BOX 2.5: THE RATE OF QUARANTINING IN THE SIR MODEL

The formulation in (2.7) is only an approximation of real-world, repeated quarantining, because the change in I is continuous, whereas the actual quarantining process is discrete. For instance, one might imagine an idealized situation where, every morning exactly at 8:00, one would send 10% of all infected individuals into quarantine. The model does not represent this daily routine but, instead, subtracts infectives at a constant rate throughout the day. Nonetheless, the approximation is sufficient for our exploration, as we can see from the following argument with a simplified situation. Suppose we start with 100 infectives, nobody becomes infected, nobody dies or becomes immune, and the only change in I is quarantining. In this simplified scenario, (2.7) becomes $\dot{I} = -r_Q I$.

Starting with $I_0 = 100$ and $r_Q = 0.1$, the number of infectives over time is reduced roughly by 10% per time unit (see Table 1). For larger values of r_Q, the correspondence is less accurate. For instance, for $r_Q = 0.5$, the number of infectives decreases from 100 only to 60.65 and 36.79 between times 0, 1, and 2, respectively, which roughly corresponds to 40% decreases per time step, rather than 50% decreases. The basis for the discrepancy is similar to continuously compounding interest.

TABLE 1: DECREASE IN THE NUMBER OF INFECTIVES IN THE SYSTEM WHEN THE CONTINUOUS QUARANTINING RATE IS *RQ* = 0.1; THE DECREASE ROUGHLY CORRESPONDS TO 10% PER TIME UNIT

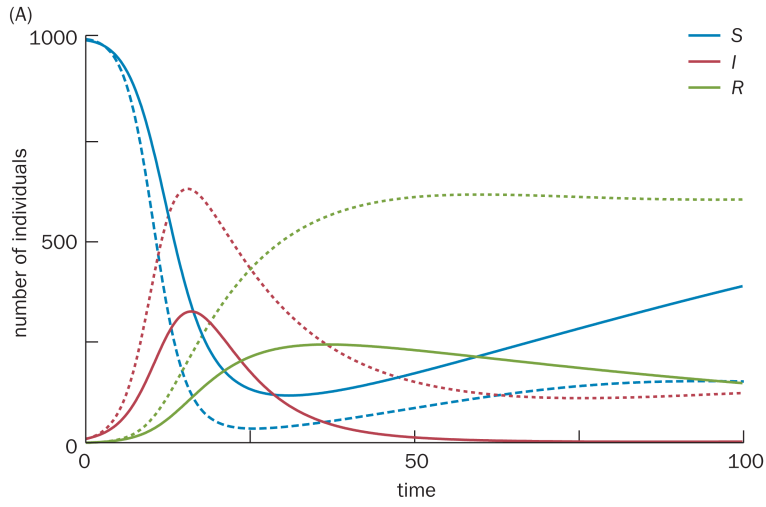
t	0	1	2	3	4	5	6	7	8	9	10
I	100	90.48	81.87	74.08	67.03	60.65	54.88	49.66	44.93	40.66	36.79

Simulating the system reveals that quarantining even at a rate of $r_Q = 0.1$ has noticeable, positive effects: the number of infected individuals is roughly halved throughout the first 100 days of disease (**Figure 2.15A**). If we quarantine with a higher rate of $r_Q = 0.4$, a few individuals still become infected, but the infection level is very low (**Figure 2.15B**). If we quarantine an even higher percentage, the number of infectives quickly goes to zero and, perhaps surprisingly at first, the susceptible pool increases. Why is this? It is due to births and immigration and to the fact that nobody in the model dies, unless she or he is first infected.

Case closed? Not quite. Something interesting happens if we let the disease process continue for a longer period of time. Because the quarantine is not perfect, some infectives remain in the population, and this can lead to repeated, smaller outbreaks. For instance, for $r_Q = 0.1$, a secondary outbreak occurs after about 165 days. Smaller outbreaks follow before the system finally settles in a state that is still not disease-free, although the infected subpopulation is smaller than in the case without quarantine (**Figure 2.16**).

Now that we have seen the positive effects of quarantine, we can explore questions such as: what happens if people are released from quarantine? Is it better to quarantine or to reduce the infection rate? Reduction of the infection rate by a certain percentage is easy to implement: we simple reduce the value of the rate r_I. **Figure 2.17** shows two results: the outcomes of quarantining and reducing the infection rate are quite different.

It is clear that the possibilities for new scenarios are endless. Thankfully, once the model is set up, many slightly altered scenarios are very easy to implement, and the process can even become addictive, if we try to find just the right parameter combinations for a desired outcome.



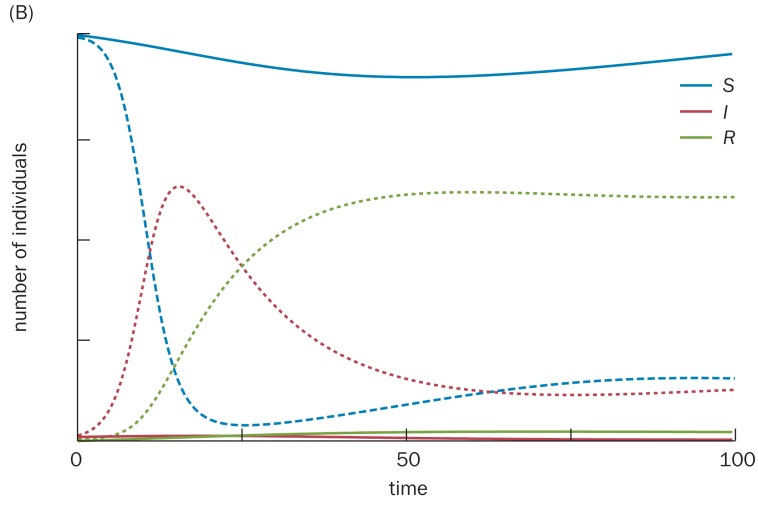


Figure 2.15 Quarantining infectives has a positive effect on the population. (A) Results of continuous quarantining at a rate of $r_Q = 0.1$ (solid lines) in comparison with no quarantine (dotted lines). (B) Corresponding result for $r_Q = 0.4$.

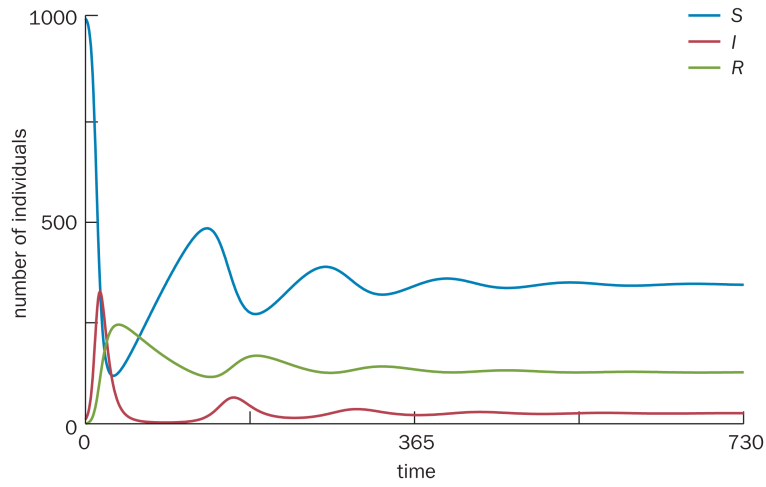


Figure 2.16 Quarantining may lead to further, milder outbreaks. For $r_Q = 0.1$, the disease seems to be under control initially, but a second outbreak occurs after about 165 days, followed by later, smaller outbreaks.

2.4 MODEL USE AND APPLICATIONS

The most common uses of a model (see Table 2.1) may be classified in two categories:

- Applications of the model as it is, including slight extension.
- Targeted manipulation and optimization of the model structure toward a specific goal.

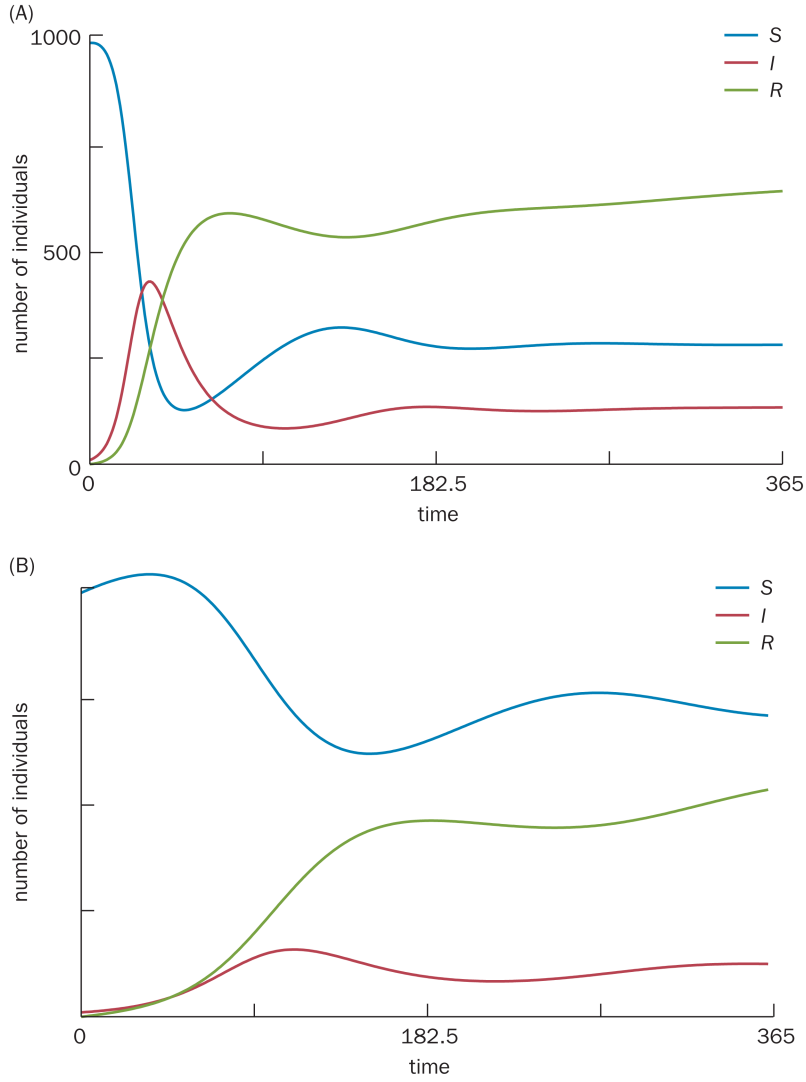


Figure 2.17 Reduction in the infection rate has a drastically different effect than quarantining. (A) The infection rate r_I has been lowered to half the original value. (B) r_I has been lowered to 20% of the original value.

2.4.1 Model Extensions and Refinements

As an example of a model extension, suppose that it becomes clear that many infected individuals do not show any serious symptoms and often do not even know that they carry the disease. In the jargon of SIR models, such individuals are called asymptomatics. How can the model account for them? The extension is really a mini version of the model design and analysis phases. First, we define a new dependent variable, A, and identify how it interacts with the other variables in the system (2.2)–(2.4). As for the initial model design, it is a good idea to sketch these interactions in a diagram. At first, we might treat A just like I, except that we should use different names for the processes, because they are indeed different (**Figure 2.18A**). However, should we also consider that the contact between S and I could lead to A, and that the contact between S and A might lead to I (**Figure 2.18B**)? The answer cannot be given with mathematical arguments and has to be based on biological or medical features of the disease process. For instance, if all A result from a weaker strain of the disease vector, an infection of S by A might always result in A and an infection by I might always lead to I; let's call this situation Case 1. But if the difference between A and I is due to an individual's general health status and his or her age and genetic predisposition, we should probably consider cases where an infection of S by A can indeed result in I, and vice versa; let's call this situation Case 2.

For Case 1, the equation for the asymptomatics is similar to that for the infectives, (2.3), namely,

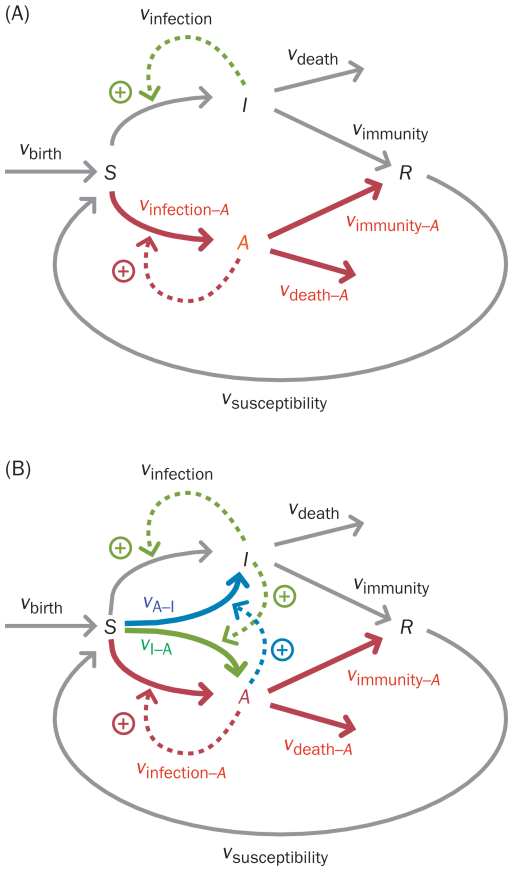


Figure 2.18 Diagrams accounting for asymptomatics. (A) The inclusion of *A* requires new processes (red) with their own names. (B) Further interactions (blue and green) are needed if an infection of *S* by *I* can result in *A* or vice versa.

$$\dot{A} = r_A SA - r_{R-A}A - r_{D-A}A, \quad A(t_0) = A_0. \tag{2.8}$$

While the structure of the equation is the same as in (2.3) for I, the parameters have different names and potentially different values. For instance, the death rate r_{D-A} might be different, because the disease in A is apparently not as bad as for I. Similar arguments might hold for r_A and r_{R-A}. Does that cover Case 1? No, for sure we also need to adjust the equations for S and R. It could even be that R is different for asymptomatics and for infectives (why?); but let's assume that that is not the case. The adjusted equations for S and R are

$$\dot{S} = r_B + r_S R - r_I SI - r_A SA, \quad S(t_0) = S_0, \tag{2.9}$$

$$\dot{R} = r_R I + r_{R-A}A - r_S R, \quad R(t_0) = R_0. \tag{2.10}$$

For Case 2, we need to add more terms: one each to the equations for A and I and two terms to the equation for S; the equation of R is not affected:

$$\dot{A} = r_A SA + r_{I-A}SI - r_{R-A}A - r_{D-A}A, \quad A(t_0) = A_0, \tag{2.11}$$

$$\dot{I} = r_I SI + r_{A-I}SA - r_R I - r_D I, \quad I(t_0) = I_0, \tag{2.12}$$

$$\dot{S} = r_B + r_S R - r_I SI - r_A SA - r_{I-A}SI - r_{A-I}SA, \quad S(t_0) = S_0. \tag{2.13}$$

As an illustration, let's simulate Case 1. For the numerical settings we will retain most values from our earlier example (2.6). Regarding A, we assume that initially half of the infected individuals are asymptomatic, so that $A(t_0) = I(t_0) = 5$. Furthermore, we assume that the infection rate for A and I is the same $(r_A = r_I = 0.0005)$, that no A die from the disease $(r_{D-A} = 0)$, and that they recover more quickly than I $(r_{R-A} = 2r_R = 0.1)$. The result of these settings is that the asymptomatics disappear quickly and the disease ultimately assumes the same profile as we saw in **Figure 2.13** (Figure 2.19A). Interestingly, the fact that none of the asymptomatics

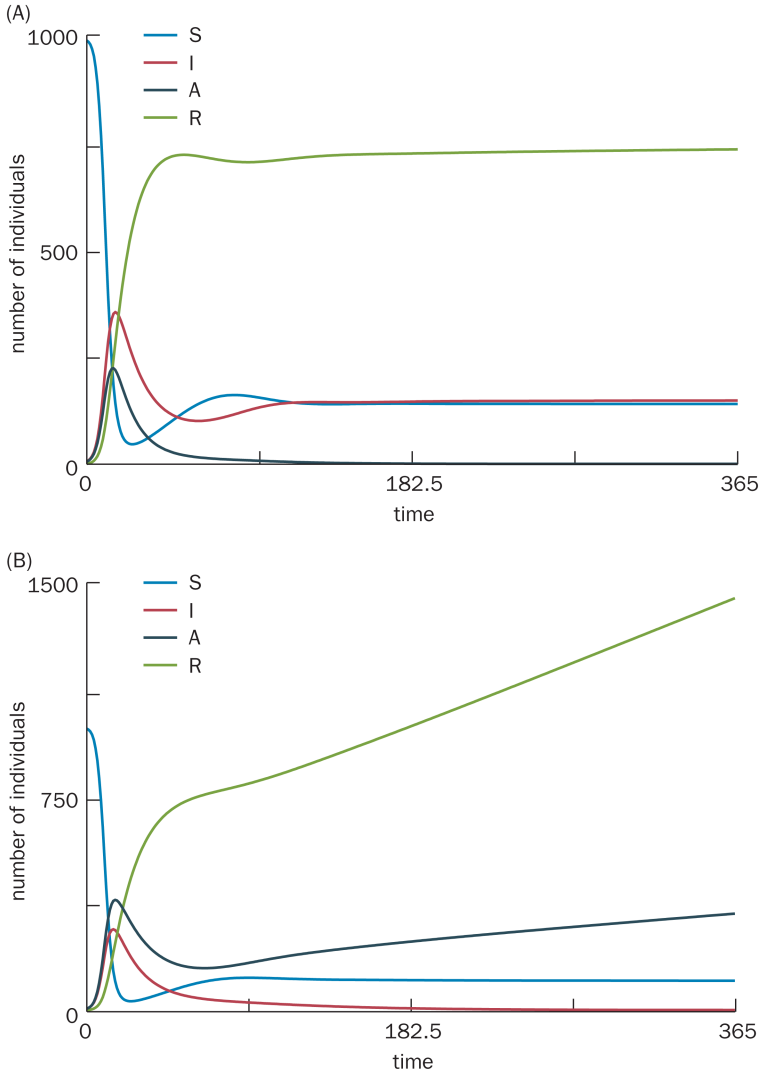


Figure 2.19 Effect on asymptomatics A on the disease progression. (A) If asymptomatics recover twice as fast as *I*, they quickly disappear from the population, whether their death rate is zero (as shown) or not. (B) If asymptomatics recover at the same rate as *I*, and if they do not die, the infectives essentially disappear, and the population grows (note different scale). However, the disease does not disappear; it is present in a growing number of asymptomatics.

die does not have much effect (results not shown). By contrast, if no A die, but the recovery rate for A and I is the same, the outcome is very different: I essentially disappears, while the R and A subpopulations grow to much higher levels (**Figure 2.19B**) The explanation is that essentially nobody in the model dies anymore. To make the model more realistic, one would need to introduce death rates for S, R, and A.

It is probably evident that uncounted scenarios may be tested depending on the effort invested. Two of the exercises discuss key aspects of the recent COVD-19 pandemic, namely social distancing and flattening the curve, that is, the attempt to avoid too many infections at once.

2.4.2 Large-Scale Model Assessments

In reality, most parameters of biological systems are not entirely fixed but can vary within certain ranges, for instance, due to inter-individual variability. If the biologically relevant ranges of all parameter values are known or can be estimated, the model may be used to evaluate best- and worst-case scenarios, along with the most likely situations. This analysis can be accomplished with targeted **stochastic simulations** in which parameters are varied, as earlier, or with large-scale, automated simulation studies, in which thousands of parameter combinations are screened. The latter method is often referred to as **Monte Carlo simulation**. The principle is the following. Suppose there are 10 parameters of interest and the range for each is more or less known. For instance, there could be information that

the infection rate in our example is between 0 and 0.002 and that the most likely value is 0.0005, as we set it in our baseline model. Suppose that this type of information is available for the other nine parameters as well. A Monte Carlo simulation consists of a series of the following **iterations**:

1. Automatically draw a random value for each parameter, according to the available information about its range.
2. Solve the equations for the given set of values and record the output values of interest.
3. Repeat this procedure thousands of times.
4. Collect all output values, along with the values of the randomly chosen parameter values.

The overall result of such a Monte Carlo simulation is a visualization or listing of a very large number of possible outcomes and their likelihoods. Furthermore, statistical analysis of the output conveys insights into which parameters or parameter combinations are most or least influential, for instance, to obtain as high an output value as possible.

As an example, let's use our infection model (2.6) again and test the effects of different infection and death rates on the total number of deaths within a year. First, we define the total number of deaths at time t as $\dot{D} = 0.02I$ (see Exercises). For an introductory demonstration, we perform a Monte Carlo simulation, consisting of just 20 random choices for the parameters r_I and r_D, which are selected randomly from the intervals [0, 0.001] and [0, 0.04], respectively. The lower bound of these intervals (0 in both cases) means no infection (death) at all, while the upper bounds (0.001 and 0.04) correspond to twice the corresponding rates in (2.6). The simulation proceeds as follows:

1. Select a random number for r_I from the interval [0, 0.001].
2. Select a random number for r_D from the interval [0, 0.04].
3. Plug these two values into (2.6).
4. Simulate the model for a time period of 365 days and record, for instance, the values of S, I, R, and D.
5. Repeat Steps 1–4, here just 20 times but typically much more often.
6. Superimpose all plots and analyze the results.

Owing to the random nature of the 20 picks of r_I and r_D, we obtain rather different 20 time courses for each variable, which are shown in **Figure 2.20**.

Now let's suppose we want to analyze the effect of the acquisition and loss of immunity. Which parameter combinations of r_R and r_S would lead to the fewest numbers of deaths? For this illustration, we run 100 simulations, where we vary the two parameters between 0 and twice the values used in the original simulation. For each iteration, we report the chosen values and also the value of D. The table of these results is sorted, for instance, in Excel, and reveals the numbers of deaths from smallest to largest, as well as the parameter combinations leading to these numbers. In this simulation, the highest total numbers of deaths is 1,894 and this number is obtained for $r_R = 0.000214$ and $r_S = 0.002778$. The lowest total numbers of deaths is 344, which corresponds to $r_R = 0.097034$ and $r_S = 0.000457$. A huge difference! Is this the best possible solution? No, it is the best solution among the 100 randomly selected combinations of parameter values. Running further simulations would likely lead to slightly better solutions, and the trend in the present results appears to be that r_R should be as large as possible and r_S should be 0. Does that make sense? Yes, the quicker infected individuals become immune and the fewer individuals lose immunity, the lower that death toll will be.

Other steady-state values, peak values, and time courses may be analyzed in a similar fashion. In a more comprehensive Monte Carlo simulation, one would simultaneously select random values for several parameters and execute so many simulations that the results would no longer change much from one run to the next. Uncounted variations on this theme are possible. As a notable example, a parameter value does not have to be chosen from its interval with equal probability, but one could introduce a distribution of values over its permitted range.

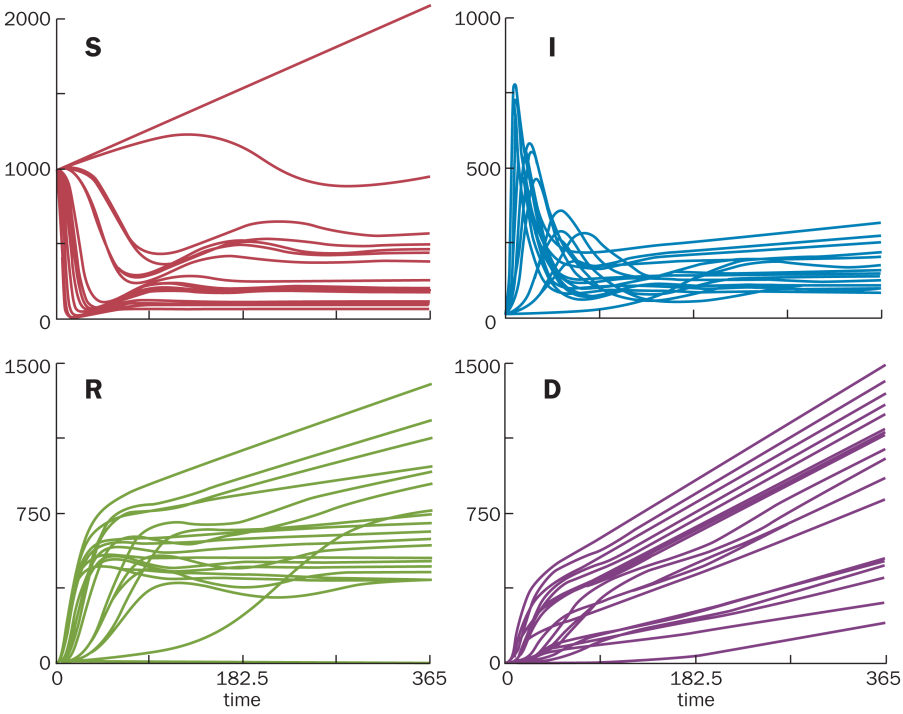


Figure 2.20 Two results of a Monte Carlo simulation. The set-up in the simulations in (A) and (B) is exactly the same; namely, (2.6) is simulated with randomly chosen values for the infection and death rates. Owing to the probabilistic nature of this process, the results are quite different in some respects, but similar in others. Notably, the simulation in (B) includes the random choice of a very low death rate, which allows S to grow. However, most other patterns in time courses are rather similar, although they differ numerically. Note the different scales on the vertical axes.

The model may also be used to screen for particularly desirable outcomes. For instance, health officials will be interested in shortening the period during which many individuals are sick, and in minimizing the chance of infection. To develop ideas for combined interventions, a Monte Carlo simulation could be a first step. It is very efficient to try out alternative strategies toward these goals with a model and, possibly, to optimize them with numerical methods from the field of operations research. Of course, predicted results always have to be considered with caution, because the model is a vast simplification of reality. Nonetheless, a good model is able to guide and correct our intuition, and the more reliable the model becomes through testing and refinements, the more competent and important its predictions will be.

2.4.3 Questions of Design and Operation

If every two biological systems had entirely different structures and modes of operation, the study of biology would have a lot of similarity with stargazing: one could only study systems one at a time and then try to classify them by some features, but one would not gain much insight into the rationale behind these features and their specific roles (Figure 2.21A). Thus, when we talk about investigating the design of a biological system, we are really trying to address frequently observed design patterns, **design principles**, or **motifs** that show up time and again with relatively

Figure 2.21 Stargazing or searching for the principles of a watch? Systems biology attempts not just to describe parts of systems but to decipher the underlying design and operating principles. (Courtesy of Horst Gunther Burkhardt III, published under CC BY-SA 3.0 license.)

minor variations. Expressed differently, instead of comparing biology with stargazing, we should study biological designs like we analyze technological and engineered designs. Good watchmakers can explain the role of every spring or cogwheel in a particular location of a mechanical watch, even if they had never seen this particular watch before (**Figure 2.21B**). The shapes of cars in a given era are far from coincidental because style and uniqueness are extremely constrained by functionality, aerodynamics, desirable features, comfort, and the drivers' perceived taste. After all, why do almost all passenger cars have four wheels, whereas most large trucks have more? What happened to three-wheeled cars that once existed? Why are brake lights red? Why are metallic paints emerging and disappearing over time?

Systems biology must ask similar questions. For instance, if we want to increase the concentration of a metabolite, should we increase the amount of substrate that is used for its production, increase the production rate, or decrease the degradation or usage of the metabolite? More generically, we should ask: why is this system organized in this particular fashion that we observe and not in an alternative fashion? This type of questioning will eventually come to a point where we understand in detail why particular genes are turned on and off in particular sequences, why cells and organisms are structured in the way they are, and what is behind the details of the function and operation of organs, tissues, cells, and intracellular components in particular situations.

The investigation of design principles is an interesting line of research that began several decades ago [14–16] but has taken off only recently (see, for example, [17–19]). It consists of two parts. First, we need to discover designs that are used over and over again in biology, and second, we have to figure out why nature apparently prefers these designs over other designs that at first seem as reasonable.

In some cases, design patterns are easy to detect, especially if they are associated with macroscopic features. Birds have wings that perfectly support their ways of flying, fish have fins in similar locations on their bodies, all healthy spiders have eight legs, and most mammals have hair. In other cases, such discoveries are much more difficult. It took quite a bit of molecular biology to figure out the almost ubiquitous structure of lipid bilayers or the genetic code that seems to be universal among essentially all earthly organisms. In yet other cases, the design patterns are so intricate that they easily escape casual observation. As a specific example, we have discovered with sophisticated computational analyses that genomes are not arbitrarily arranged in gene sequences but have much more structure. Bacterial genes are arranged in **operons**, regulons, and über-operons [20], and it appears that operons involved in multiple pathways tend to be kept in nearby genome locations [21] (**Figure 2.22**). Species show their own codon bias, and bacterial

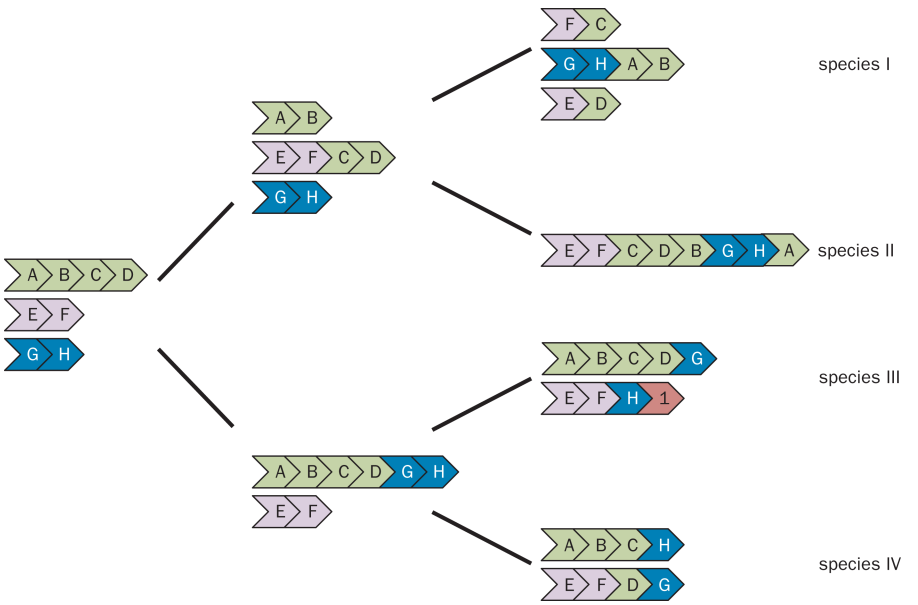


Figure 2.22 Conceptual representation of operons and über-operons. Operons are contiguous, co-transcribed, and co-regulated sets of genes. While they are usually poorly conserved during evolution, the rearrangement of genes often maintains individual genes in specific functional and regulatory contexts, called über-operons. The ancestral genome (left) is assumed to contain three clusters of similarly regulated and functionally related genes, A–H, in three separate genomic locations. During evolution, the clusters are rearranged into new clusters. (Further genome rearrangements may occur, but some clusters and individual genes are retained in the same neighborhoods.) (From Lathe WC III, Snel B & Bork P. *Trends Biochem Sci.* 25 [2000] 474–479. With permission from Elsevier.)

genomes consist of almost unique nucleotide compositions and combinations that resemble molecular fingerprints or barcodes [22] (see Chapter 7). These types of patterns were only discovered through ingenuity and with the help of very large datasets and custom-tailored computer algorithms.

It has been known for some while that the expression of a gene is correlated with the expression of other genes. But are there patterns? Microarray studies, taken at several time points after some stimulus, suggest that groups of genes may be co-expressed, while the expression of other genes is inversely correlated or correlated with some time delay. In many cases, the clusters of co-expressed genes can be explained with the coordinated function of the proteins for which they code, but in other cases they remain a puzzle (see, for example, [23]).

Like gene sequences, metabolic and signaling pathways show commonalities among different cells and organisms, some of which have been known for a long time. Feedback inhibition by the end product is usually exerted on the first step of a linear pathway, and one can show—objectively and with mathematical rigor—that this design is indeed superior to other modes of inhibition, such as inhibition of the last step—or of several steps—in a reaction sequence (cf. [24, 25]). Signaling is often accomplished with almost identical cascades of proteins that are phosphorylated and dephosphorylated at three levels (Chapter 10). Why three? We have recently learned to look for generic patterns in the connectivity patterns of natural and artificial networks (see Chapter 3). For instance, most metabolites are associated with only a few reactions, and relatively few metabolites are associated with many different production and degradation steps [26]. This same pattern of only a few highly connected hubs within a large network shows up time and again, in biology, the physical world, and society. Just study the flight routes of a major airline!

Interestingly but not really surprisingly, many networks we find in biology are very robust, often due to redundancies. In a beautiful demonstration of this feature, Isalan and collaborators randomly rewired the transcriptional network of *E. coli* and found that only about 5% of these scrambled networks showed growth defects [27]. A similar change in the connectivity of gene transcription networks seems to have occurred naturally in the evolution of yeast, and it could be that such flexibility is a general feature of gene transcription systems. It is intriguing to ponder whether it might be possible to use rewiring as a precisely targeted tool in synthetic biology [28] (see also Chapter 7).

Another important aspect of biological systems is that they employ similar components time and again in a modular and hierarchical manner. This use of the same motifs found within different networks and at different organizational levels renders complex biological networks simpler than they might otherwise have been and indeed offers hope that we might eventually understand critical aspects of the complexity of biological systems [29].

The identification and rigorous characterization of design principles requires methods of formalizing complex systems and their organization. This task may be accomplished in different ways. One school of thought has been focusing on network graphs and their properties, while another has been looking for design principles in fully regulated dynamic systems and the corresponding differential equation models. Interestingly, both approaches ultimately employ a similar strategy, namely comparisons between the observed system structures and seemingly reasonable alternatives. In the case of graphs, the properties of an observed network are compared with the corresponding properties of a randomly connected network. For instance, one might look for the number of nodes that have more links than one would statistically expect in a randomly connected network (see Chapter 3). In the case of dynamic systems, an observed design is compared with a very similar dynamic system structure that differs in just one particular feature of interest, such as a regulatory signal. We will discuss techniques for assessing the design of dynamical systems in Chapter 6.

2.4.4 Simplicity versus Complexity

This chapter has given you a flavor of what modeling means and what we can do with mathematical models in systems biology. Chapters 3, 4, and 5 describe standard techniques for setting up, diagnosing, analyzing, and using models of differ-

ent types. Before we dive into the nitty-gritty details of the mathematics of modeling, we should realize that the complexity of the mathematical representations and techniques is not necessarily correlated with the usefulness of a model. Certainly, some models must be complicated, but sometimes very simple models are "better" than more complicated counterparts. For instance, many growth models are rather simple, even if they describe very complicated biological systems, such as the human body. There is no chance that we will come up with a complete physiological model of the human body in the foreseeable future, yet the simple growth curves of children show amazingly small variation from one child to the next (**Figure 2.23**). Predictions based on these curves are much more reliable than predictions from any large physiological model that currently exists.

An example at the other end of the spectrum is the mathematical model in **Figure 2.24**. It adorns a T-shirt and the artist has even provided the correct answer, upside down below the formula, as *age = 50*. The display is certainly intimidating to mere mortals, but does that mean that we should judge this "model for age" as good? The answer is a resounding *No*, for the following (and many more) reasons. First, the model does not explain or predict anything. Most importantly, does the

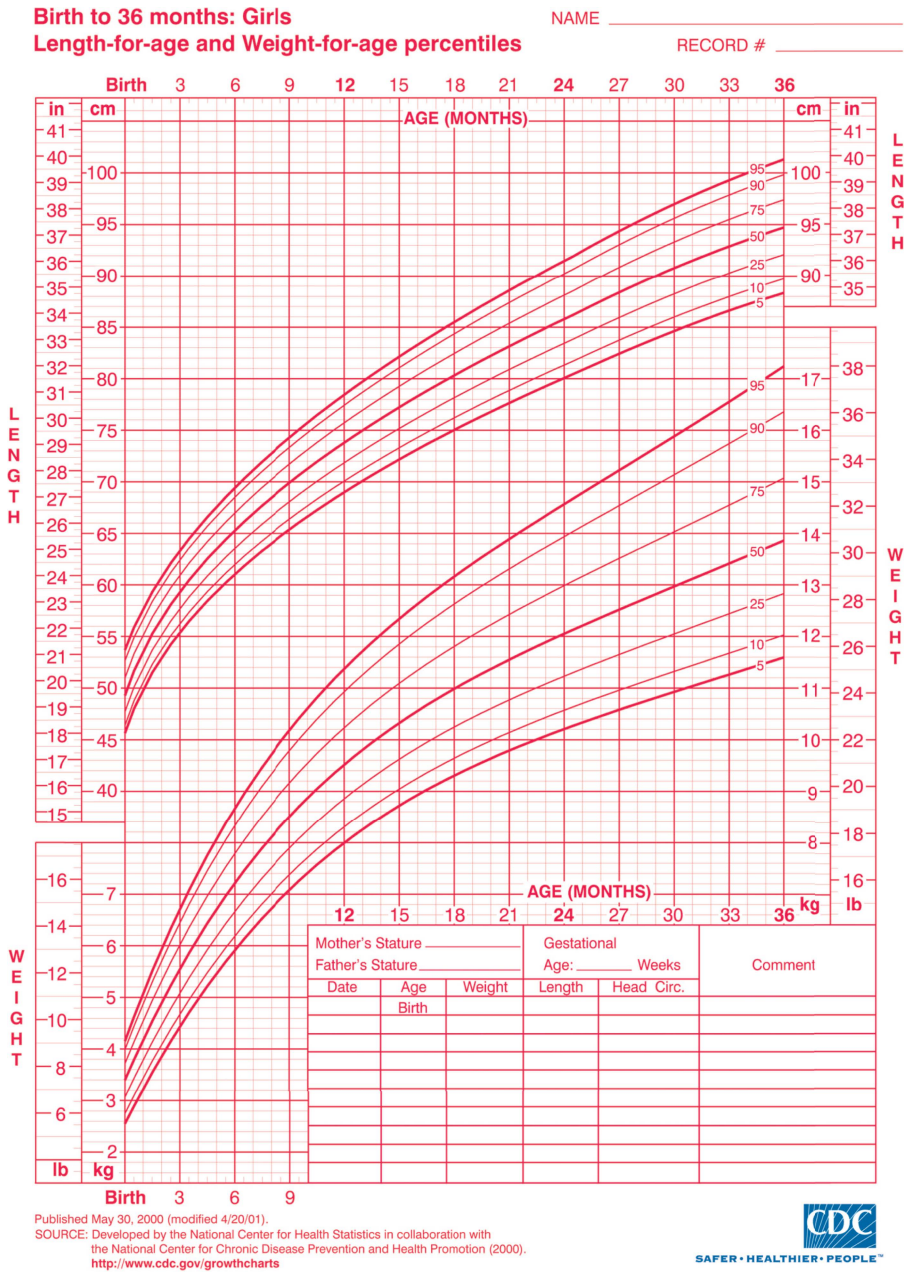


Figure 2.23 Growth curves of girls from birth to 3 years. Considering how complicated the human body is, the normal growth trends of 50% of the girls are within surprisingly narrow bounds. (From the National Center for Health Statistics.)

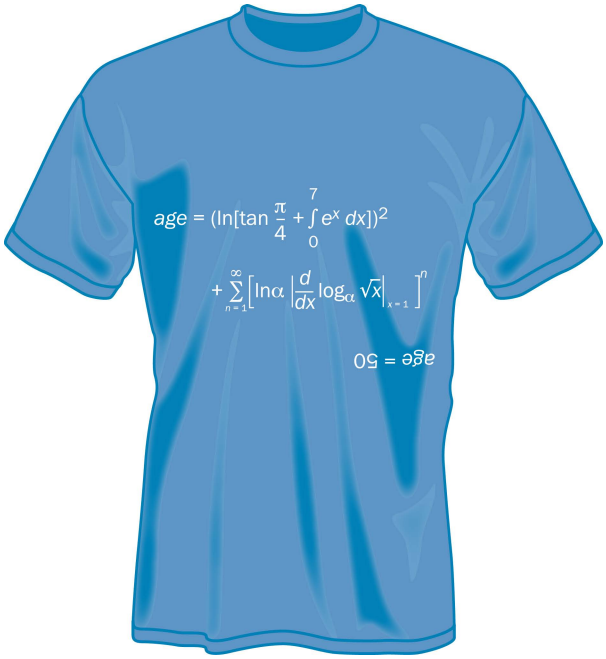


Figure 2.24 Awe-inspiring, useless model on a T-shirt. While the formula looks intimidating, it is simply a complicated way of representing the number 50. The "model" contains no real parameters and only shows the correct age if the wearer happens to be 50 years old.

T-shirt show the age of the wearer? Only if he or she happens to be 50 at the time; wearing the shirt next year, the implied age will be wrong. Age is obviously a function of time, but the T-shirt does not account for time. In fact, there is no true variable, except for the integration variable x, which is merely a placeholder and disappears as soon as the integration has been executed. There are no true parameters either, because the only candidates, α and n, disappear in the summation, and the answer is always 50. In other words, the formula does not permit any adaptation to the wearer and his or her true age. Instead, the display is a mathematically unnecessary complication of the simple statement $age = 50$ and not much more. If we are interested in a better T-shirt model of age, which correctly captures the wearer's age, this model must certainly contain time as a variable. Barring magic or serendipity, the formula must also contain a place for explicit or implicit information about the wearer's time of birth. This information may be given directly or in some convoluted fashion, but it must be there as some (potentially explicit or hidden) parameter. We also need to decide on the accuracy of the improved T-shirt model. If age in years is sufficient, the model will certainly be simpler than if age has to be computed to the day or minute.

The message from these two examples is that the simplicity or complexity of a model are not indicative of its quality. Of course, there are situations where complicated mathematics is needed, but there are also many cases where simplicity will do just fine, if not better. The key criterion of quality of a model is, instead, its usefulness. Mathematical models are tools, and tools are only good and helpful if they allow us to solve problems. Sometimes, simple tools are our best choice, and at other times the complexity of the model must match the complexity of the system we want to analyze. Because usefulness is the most important criterion, every model analysis should begin with a clear understanding of why we want to design a model and what this model is supposed to accomplish. Time and effort spent on these two questions are the most prudent investments in the realm of modeling.

2.5 EXERCISES

2.1 Explore the range of shapes that a Hill function (2.5) can attain. First, keep the settings $V_{max} = 12$ and $K_M = 10$ and use different values for n ($n = 4$, 6, 8; 3, 2, 1, 0.5, 0.2). Then, set $n = 4$ and vary V_{max} or K_M individually or both simultaneously. Discuss differences and similarities in the shapes of the resulting graphs.

2.2 Show with mathematical arguments that V_{max} is the asymptote for the Hill function (2.5) as S grows without bound.

2.3 The function $L(S) = 1/(1 + e^{10-S})$ is a shifted logistic function, whose graph is an S-shaped curve. Through trial and error, determine parameter values for the Hill function (2.5) that approximate $L(S)$ as closely as possible.

2.4 Many processes in biology can be approximated quite accurately with power-law functions of the type $P(X) = \gamma X^f$ (one variable) or $P(X_1, X_2, \ldots, X_n) = \gamma X_1^{f_1} X_2^{f_2} \ldots X_n^{f_n}$ (n variables; $n = 1, 2, 3, 4, \ldots$). The positive parameter γ is a rate constant and the real-valued parameters f are called kinetic orders. For $n = 1$ explore the shapes of these power-law functions for different values of γ and f. For γ use (as a minimum) the values 0.1, 1, 10, and 100. For the parameter f use (as a minimum) the values –2, –1, –0.5, –0.1, 0, +0.1, +0.5, +1, +2. Display your results as tables and/or graphs. Summarize the results of your analysis in a short report.

2.5 Compare the Michaelis–Menten function $V(S) = V_{\max} S/(K_M + S)$, with $V_{\max}$ 1 and $K_M = 2$, with the power-law function $P(S) = \gamma S^f P$, where $\gamma = 0.3536$ and $f = 0.5$. Discuss your findings. Choose different values for $V_{\max}$ and K_M and determine values of γ and f such that the power-law approximation matches the Michaelis–Menten function well for some range of values of S. Write a brief report about your results.

2.6 Compute the two solutions for the equation $e^x - 4x = 0$ in Figure 2.12 by trying out different values. Write down your thought processes when selecting the next value to be tried. Remember these thought processes when you study Chapter 6.

2.7 Write equations for the small protein system in the signaling cascade of Figure 2.7. Model each process as a rate constant times the concentration of the variable involved. For instance, the phosphorylation of P_0 P_0 toward P_1 should be formulated as $r_{01}P_0$. Begin with $P_0 = P_1 = P_2 = 1$ and set all rate constants to 0.5. Implement the system in a software program and simulate it. Explore the dynamics of the system by changing the initial values of P_0, P_1, and P_2. Discuss the results. Now change the rate constants one by one or in combination, until you develop a feel for the system. Discuss the results and summarize your insights into the system.

For Exercises 2.8–2.24, use the infectious disease model in (2.2)–(2.4) and (2.6) and be aware that several of these exercises are open-ended. Before you execute a particular simulation, make a prediction of what the model will do. After the simulation, check how good your predictions were. Consider a problem "done" when your predictions are consistently correct.

2.8 Change each initial value in (2.6) (up and down) and study the consequences.

2.9 What would it mean biologically if $r_I = 0$ or if $r_S = 0$? Test these scenarios with simulations.

2.10 What happens if no babies are born and nobody immigrates? Explore what happens if nobody in the model dies.

2.11 Double the death rate in the model and compare the results with the original simulation.

2.12 Alter the model in (2.2)–(2.4) so that susceptible (S) and immune (R) individuals may die from causes other than the infection. Discuss whether it is reasonable to use the same rate constant for the death of S and R individuals. Implement your proposed alterations in (2.6) and execute simulations demonstrating their effects.

2.13 Change the infection rate, starting from the value of 0.0005 in (2.6), by raising it to 0.00075, 0.001, 0.01, and 0.1, or by lowering it stepwise toward 0. Discuss trends in consequences on the dynamics of the system.

2.14 Change the infection rate or the rate of acquiring immunity to different values and solve the system until it reaches its steady state, where no variable changes anymore. For each scenario, check the number of deaths per unit time, which is given by $r_D I$. Record and discuss your findings.

2.15 Predict the consequences of partial vaccination. Implement a vaccination strategy in the model and test your predictions with simulations.

2.16 Compare the effectiveness of quarantine and vaccination.

2.17 Implement scenarios where certain percentages of asymptomatics become infectives. Test the consequences with simulations.

2.18 How could the model in (2.2)–(2.4) be expanded to allow for different levels of disease susceptibility that depend on age and gender? Without actually implementing these changes, list what data would be needed to construct such a model.

2.19 Describe in words what data would be required to develop an infectious disease model like that in (2.2)–(2.4) that accounts for the gradual spread of the disease throughout a geographical area.

2.20 Without actually implementing them in equations, propose extensions that would make the SIR model in (2.2)–(2.4) more realistic. For each extension, sketch how the equations would have to be changed.

2.21 The term $r_D I$ in the equation for I in the SIR model represents the death of some infected individuals. Use mathematical arguments to show that the variable

$$\dot{D} = r_D I$$

represents the accumulation of deaths from the infection over time. Determine a suitable initial value for this new ODE. Make a prediction regarding the trend of the new variable, then simulate the SIR model together with this variable and test your prediction.

2.22 Define a new variable that tracks the number of all individuals in this model who ever became infected. Formulate a differential equation for this variable and determine an appropriate initial value. Make a prediction regarding the trend of the new variable, then simulate the SIR model together with this variable and test your prediction.

2.23 During the COVID-19 pandemic, an important strategy was "flattening of the curve," meaning a strategy for ensuring that not too many individuals would become infected in the same time period. How could one study flattening of the curve in the SIR model?

2.24 (Project): study the effects of social distancing in the SIR model, as it was a key component of combatting COVID-19. Simulate this distancing by reducing the infection rate by some percentage (maybe 40% or 100%) reflecting partial or full compliance. For comparisons between different distancing scenarios, use as metrics the peak time of infection and the total numbers of infections and deaths after 100, 1,000, and 10,000 days (see Exercises 21 and 22). Test whether social distancing flattens the trendline of infections (see Exercise 23). Also test what happens if the requirement for social distancing ends after some time period. Note that the project allows a number of different assumptions and strategies.

2.25 Set up symbolic equations for a two-stage cancer model that consists of the following processes. Normal cells divide; some of them die. In a rare case, a normal cell becomes "initiated," which means that it is a possible progenitor for tumor formation. The initiated cells divide; some die. In a rare case, an initiated cell becomes a tumor cell. Tumor cells divide; some die.

2.26 Implement the system in Box 2.2 in computer software such as *Mathematica*®, MATLAB®, Python, Java, or PLAS [3], and compute solutions with different values for A, and with slightly changed initial values of x or y. Explore what happens if A is close to or equal to zero.

2.27 Use the binomial model in Box 2.3 to compute the probability that exactly 5 out of 10 flips of a coin are heads and 5 are tails. Compute the probability that exactly all 10 flips of a coin are heads.

2.28 Use the Poisson model in Box 2.4 to compute the probability that exactly 5 out of 10 flips of a coin are heads and 5 are tails. Compute the probability that exactly 50 out of 100 flips of a coin are heads and 50 are tails.

2.29 Use the Poisson model in Box 2.4 to compute the probability that at most 3 out of 20 flips of a coin are heads.

2.30 Implement the equation from Box 2.5 and compute the numbers of infectives over time for different rates r_Q. In parallel, design a model where a certain, fixed percentage of infectives is quarantined at $t = 0, 1, 2, \ldots$ but not in between. Compare the results of the two modeling strategies.

FURTHER READING

Adler FR. *Modeling the Dynamics of Life*, 2nd ed. Brooks/Cole, Totnes, UK, 2005.

Batschelet E. *Introduction to Mathematics for Life Scientists*, 3rd ed. Springer, New York, NY, 1979.

Britton NF. *Essential Mathematical Biology*. Springer-Verlag, New York, NY, 2004.

Edelstein-Keshet L. *Mathematical Models in Biology*. McGraw-Hill, 1988 (reprinted by SIAM, Philadelphia, PA, 2005).

Haefner JW. *Modeling Biological Systems: Principles and Applications*, 2nd ed. Springer, New York, NY, 2005.

Murray JD. *Mathematical Biology I: Introduction*, 3rd ed. Springer, New York, NY, 2002.

Murray JD. *Mathematical Biology II: Spatial Models and Biomedical Applications*. Springer, New York, NY, 2003.

Yeagers EK, Shonkwiler RW & Herod JV. *An Introduction to the Mathematics of Biology*. Birkhäuser Verlag AG, Basel, Switzerland, 1996.

REFERENCES

1. Qi, Z., G.W. Miller, and E.O. Voit, Computational systems analysis of dopamine metabolism. *PLoS One*, 2008. **3**(6): p. e2444.
2. Thompson, J.M.T. and H.B. Stewart, *Nonlinear Dynamics and Chaos*. 1986, Chichester: John Wiley and Sons.
3. Ferreira, A.E.N., *PLAS: Power Law Analysis and Simulation*, https://arquivo.pt/wayback/20230321162314/http://enzymology.fc.ul.pt/software/plas/, 2000–2024.
4. Meng, C., et al., UV induced mutations in *Acidianus brierleyi* growing in a continuous stirred tank reactor generated a strain with improved bioleaching capabilities. *Enzyme Microb Technol*, 2007. **40**(5): p. 1136–1140.
5. Fujii, R., M. Kitaoka, and K. Hayashi, One-step random mutagenesis by error-prone rolling circle amplification. *Nucleic Acids Res*, 2004. **32**(19).
6. Mood, A.M., F.A. Graybill, and D.C. Boes, *Introduction to the Theory of Statistics. McGraw-Hill Series in Probability and Statistics*. 1974, Auckland, NZ: McGraw-Hill, p. 577.
7. Kermack, W.O. and A.G. McKendrick, Contributions to the mathematical theory of epidemics. *Roy Stat Soc J*, 1927. **115**: p. 700–721.
8. Gutierrez, J.B., et al., From within host dynamics to the epidemiology of infectious disease: Scientific overview and challenges. *Math Biosci*, 2015. **270**(Pt B): p. 143–155.
9. van den Driessche, P., Reproduction numbers of infectious disease models. *Infect Dis Model*, 2017. **2**(3): p. 288–303.
10. Heesterbeek, J.A., A brief history of R0 and a recipe for its calculation. *Acta Biotheor*, 2002. **50**(3): p. 189–204.
11. Perasso, A., An introduction to the basic reproduction number in mathematical epidemiology. *ESAIM Proc Surv*, 2018. **62**: p. 123–138.
12. Raue, A., et al., Structural and practical identifiability analysis of partially observed dynamical models by exploiting the profile likelihood. *Bioinformatics,* 2009. **25**: p. 1923–1929.
13. Voit, E.O., What if the fit is unfit? Criteria for biological systems estimation beyond residual errors, in *Applied*

Statistics for Biological Networks, M. Dehmer, F. Emmert-Streib, and A. Salvador, Editors. 2011, Hoboken, NY: Wiley, p. 183–200.
14. Irvine, D.H. and M.A. Savageau, Network regulation of the immune response: Alternative control points for suppressor modulation of effector lymphocytes. *J Immunol*, 1985. **134**: p. 2100–2116.
15. Savageau, M.A., *Design of molecular control mechanisms and the demand for gene expression. Proc Natl Acad Sci USA*, 1977. **74**: p. 5647–5651.
16. Savageau, M.A., A theory of alternative designs for biochemical control systems. *Biomed Biochim Acta*, 1985. **44**(6): p. 875–880.
17. Alon, U., *An Introduction to Systems Biology: Design Principles of Biological Circuits*. *Mathematical & Computational Biology*. 2006, Boca Raton, FL: Chapman & Hall/CRC.
18. Ma'ayan, A., R.D. Blitzer, and R. Iyengar, Toward predictive models of mammalian cells. *Annu Rev Biophys Biomol Struct*, 2005. **34**: p. 319–349.
19. Shen-Orr, S.S., et al., Network motifs in the transcriptional regulation network of *Escherichia coli*. *Nat Genet*, 2002. **31**(1): p. 64–68.
20. Che, D., et al., Detecting uber-operons in prokaryotic genomes. *Nucleic Acids Res*, 2006. **34**(8): p. 2418–2427.
21. Yin, Y., et al., Genomic arrangement of bacterial operons is constrained by biological pathways encoded in the genome. *Proc Natl Acad Sci USA*, 2010. **107**(14): p. 6310–6315.
22. Zhou, F., V. Olman, and Y. Xu, Barcodes for genomes and applications. *BMC Bioinform*, 2008. **9**: p. 546.
23. Yus, E., et al., Impact of genome reduction on bacterial metabolism and its regulation. *Science*, 2009. **326**(5957): p. 1263–1268.
24. Alves, R. and M.A. Savageau, Effect of overall feedback inhibition in unbranched biosynthetic pathways. *Biophys J*, 2000. **79**(5): p. 2290–2304.
25. Savageau, M.A., *Biochemical Systems Analysis: A Study of Function and Design in Molecular Biology*. 1976, Reading, MA: Addison-Wesley Pub. Co. Advanced Book Program, p. xvii, 379.
26. Jeong, H., et al., The large-scale organization of metabolic networks. *Nature*, 2000. **407**(6804): p. 651–654.
27. Isalan, M., et al., Evolvability and hierarchy in rewired bacterial gene networks. *Nature*, 2008. **452**(7189): p. 840–845.
28. Mukherji, S. and A. van Oudenaarden, Synthetic biology: Understanding biological design from synthetic circuits. *Nat Rev Genet*, 2009. **10**(12): p. 859–871.
29. Alon, U., Simplicity in biology. *Nature*, 2007, 29 March. **446**: p. 497.

Static Network Models

3

When you have read this chapter, you should be able to:

- Identify typical components of static networks
- Understand the basic concepts of graphs and how they are applied to biological networks
- Realize different structures of networks and their features
- Describe typical examples of static networks in different fields of biology
- Discuss concepts and challenges of network inference
- Characterize the relationships between static networks and dynamic systems

All molecules in biological systems are components of **networks**. They are linked to other molecules through connections that may take a variety of forms, depending on what the network represents. Molecules may associate loosely with each other; bind to each other; interact with each other; be converted into each other; or directly or indirectly affect each other's state, dynamics, or function. Rather than focusing on one particular molecular process at a time, the computational analysis of biological networks has the goal of characterizing and interpreting comprehensive collections of connections. The methods for these analyses can be automated and applied, using computers, to very large networks. Typical results are characteristic features of the connectivity patterns within the investigated networks, which in turn provide clues regarding the functions of particular components or sub-networks. For instance, an analysis may reveal which components of a network are particularly strongly connected, and this insight may suggest important roles of these components in different situations. The analysis may also show which variables act in concert with each other and whether the connectivity pattern within the network changes, for example, during development and aging or in response to perturbations, such as disease.

Many concepts and methods of network analysis are quite simple and intuitive when studied in small networks. This simplicity is crucial for the real power of network analyses, namely, their scalability to very large assemblages of molecules and their ability to reveal patterns that the unaided human mind would simply not be able to recognize. Thus, computers can be trained to use the basic methods over and over again, establish network models from experimental data, identify their global and local features, and ultimately lead to hypotheses regarding their functionality.

In this chapter, we will study these types of analyses with small networks, because they show most clearly what features to look for and how to characterize them. Several web-tools are available for computational analyses of large biological networks. Prominent examples include the free open-source platform

DOI: 10.1201/9781003402770-3

Cytoscape [1, 2] the commercial package IPA from Ingenuity Systems [3], and other software packages, such as BiologicalNetworks [4] and Pajek [5], all of which offer a wide variety of tools for data integration, visualization, and analysis.

3.1 STRATEGIES OF ANALYSIS

The ancient Greeks had a saying that all is in flux. Heraclitus of Ephesus, to whom this wisdom is attributed, considered change as a central force in the universe and once allegedly asserted that one can never step into the same river twice. Indeed, everything in the living world is changing, rather than **static**—if not macroscopically, then at least at the molecular level. So, we must ask ourselves whether there is even a point in studying static networks, which by definition do not move or change over time. The answer is a resounding *Yes*, for a number of reasons. First, even if biological networks are dynamic, changes in interesting components are often relatively slow. For example, even if an organism is growing, its size is almost constant during a biochemical experiment that takes a few minutes. Second, much can be learned from assuming that a network is at least temporarily static. A typical example may be the activity profile of enzymes in the liver, which certainly changes throughout the life of an organism and even within a 24-hour period but which may be more or less constant within a period of a few seconds or minutes. And third, even if systems are dynamically changing, certain important features of these same systems are static. The most prominent examples are steady states, which we discuss in Chapters 2, 4, and 5, where material is moving through a system and control signals are active but, for instance, the concentrations of metabolite pools are constant. Steady states are usually easier to characterize than dynamic changes and, once established, can provide a solid starting point for looking into dynamic and stochastic features of a system [6–8].

A very practical reason for focusing on static networks or static aspects of dynamic systems is that the mathematics needed for static analyses is incomparably simpler than that required for dealing with temporally changing networks or fully regulated dynamic systems. As a consequence, even large static systems with thousands of components can be analyzed very efficiently, which is not really the case for dynamic systems of the same size. This relative simplicity is due to two aspects of static networks. First, because there is no change with time, the time derivatives in the differential equations describing the network are all equal to 0. Consequently, these equations become algebraic equations, which are easier to handle. Second, the structures of very many static networks are (at least approximately) linear, and linear algebra and computer science have blessed us with many elegant and powerful methods for analyzing linear phenomena. In some cases, these methods may even be applied to aspects of dynamic systems, if their subsystems are connected in a static fashion. A surprising example is a group of pacemaker cells in the sino-atrial node of the heart that can oscillate independently but are synchronized by their static interaction network, thereby enabling them to generate coordinated impulses that are required exactly once during each cardiac cycle (see [9] and Chapter 12).

The analysis of static networks can be subdivided in various ways. First, networks fall into distinct classes, which permit different methods of analysis. In some cases, for instance, actual material flows through the network, and accounting for this material permits accurate bookkeeping of all amounts or masses in different parts of the networks at different points in time. Metabolic pathway systems are prime examples in this category and therefore play a special role in this chapter. To illustrate, suppose there is a single reaction between a substrate and a product within a metabolic network. We immediately know that the amount of product generated in this reaction is directly related to the amount of substrate consumed; in fact, the two must be the same. This equivalence is very beneficial for mathematical analyses, because it eliminates a lot of uncertainty. By contrast, a signaling network transduces information, and whether or not this information is actually received and used is typically not relevant to the source of the information. Furthermore, the main purpose of such a signaling network is often an all-or-nothing response to an external signal, such as switching on the expression of a gene in response to some stress. There is no accounting for masses or specific amounts,

and even if the signal were much stronger, the effect would still be the same. Similarly, simply the presence of some component in a regulatory network may affect the function of the network somewhere else. An example is the inhibition of an enzyme, which may alter the steady state and dynamics of a metabolic pathway but which does not change the amount of the inhibitor itself.

A second distinguishing criterion in the treatment of static networks is the type of research activity, which can broadly be divided into analysis and construction (or reconstruction, as it is usually referred to). A typical analysis assumes that the structure of a network is known and investigates features such as the connectivity among its components and their contributions to the functioning of the network. By contrast, (re-)construction studies attempt to infer the connectivity of an unknown or ill-characterized network from experimental observations. At first glance, this distinction may appear to be splitting hairs, but the two aspects require distinctly different approaches, and the latter is incomparably more difficult than the former. It might seem logical to study the construction of a network first, before launching into its analysis, but it is easier to discuss and learn the two aspects in the opposite order. To do so, we will initially assume that we know the structure of the network and later worry about where our knowledge of that structure may have come from.

Finally, rigorous network analyses have led to the discovery of network **motifs**, which are small yet prominent connectivity patterns that are found time and again in diverse biological systems and therefore are expected to be particularly effective for solving certain tasks. Some sections of this chapter and of Chapter 6 are devoted to this fascinating topic (see also [10]).

3.2 INTERACTION GRAPHS

When we are interested in the structure of a network, the specific interpretation of its components becomes secondary, and it does not truly matter for the mathematical or computational analysis whether the interacting components are molecules, species, or subpopulations: they simply become **nodes**. Nodes are sometimes also called vertices (singular: **vertex**), components, pools, or species, or they may be termed or described in a variety of other specific or generic ways. The second set of ingredients of a network consists of the connections between nodes, which are typically called **edges** but may also be termed links, interactions, processes, reactions, arrows, or line segments. Among the edges, we actually have a bit more flexibility than with the nodes, because edges may be directed, undirected, or bidirectional. All three cases are relevant in one biological situation or another, and they may even appear within the same network (**Figure 3.1**). No features apart from nodes and edges are usually admissible.

Taken together, the nodes and edges form **graphs**, which are the principal representations of networks. Quite intuitively, the nodes of a network can only be connected by edges, and two edges can only be connected through a node between them. Two nodes connected by a single edge are called neighbors. Typical biological examples of directed graphs are transcriptional regulatory networks, where nodes represent genes and edges denote interactions between them. The choice of a directed graph seems reasonable in this case, because gene *A* might regulate the expression of gene *B*, but the opposite might not be true. Protein–protein interaction (PPI) networks, by contrast, are usually formulated as undirected graphs.

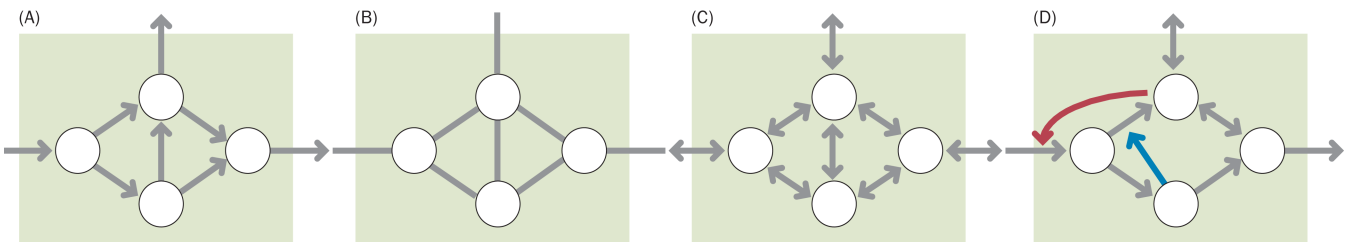


Figure 3.1 Different types of generic networks. Static networks may contain (A) directed, (B) undirected, or (C) bidirectional edges, as well as a regulation (D), such as inhibition (red) and activation (blue) signals, which require a more complex representation.

One might be tempted to surmise that all networks and systems could be represented in this fashion. However, we must be careful with such a sweeping inference. For instance, metabolic pathway systems do contain nodes (metabolite pools) and edges (chemical or enzymatic reactions and transport processes). However, they also exhibit regulatory features, with which metabolites may affect the strength, capacity, or amount of material flow through the edges of a pathway. These regulatory effects are genuinely different from metabolic conversions and therefore require other types of representations, for instance, in the form of arrows pointing from a node to an edge, rather than to another node (see Figure 3.1D) [11]. As an alternative representation, it is possible to formulate such regulated networks with bi-partite graphs, which contain two sets of different types of nodes. In the case of metabolic networks, one set represents metabolites, while the other set represents the physical or virtual locations where the enzymatic conversions between metabolites take place (**Figure 3.2**). A modeling approach following this paradigm of bi-partite graphs is **Petri net** analysis [12, 13].

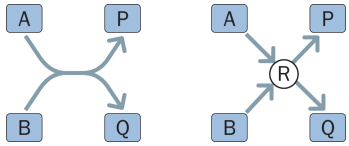


Figure 3.2 Representations of a bisubstrate reaction. Left: standard representation of a reaction involving substrates A and B that are converted into products P and Q. Right: bipartite graph representation of the same reaction, where the process R is explicitly denoted as an additional node of a different type within a Petri Net.

The subdiscipline of mathematics and computer science addressing static networks is **graph theory**, which has a long and esteemed history that reaches back to Leonard Euler in the eighteenth century. Euler's first interest in graph theory was piqued by a puzzle he posited for himself in his adopted hometown of Königsberg in Prussia. The River Pregel dissected this city in a peculiar fashion and at the time was crossed by seven bridges (**Figure 3.3**). Euler wondered whether it was possible to find a route that would cross each bridge exactly once. Of course, being a brilliant mathematician, Euler not only solved this particular puzzle but also developed an entire theoretical framework, the foundation of graph theory, to decide in which cases a route can be found, and when this is impossible.

Generically speaking, graph theory is concerned with the properties of graphs, such as their connectivity, cycles of nodes and edges, shortest paths between two nodes, paths connecting all nodes, and the most parsimonious manner of cutting edges in such a way that the graph is split into two unconnected subgraphs. Many famous mathematical problems are associated with graphs. One example, arguably the first to be proven using a computer, is the four-color theorem, which postulates that it is always possible to color a (geographical) map with four colors in such a fashion that neighboring states or countries never have the same color. Another famous problem, which combines graphs with **optimization**, is the traveling salesman problem, which asks for the determination of the shortest path through a country such that every city is visited exactly once.

Most graph analyses in biology address questions of how a network is connected and what impact this connectivity has on the functionality of the network. Thus, these analyses characterize structural features, such as the numbers and densities of connections in different parts of the graph. These types of features can be subjected to rigorous mathematical analyses, some of which lead to watertight proofs, while others are of a more statistical nature. For our purposes, we can minimize the technical details, focus on concepts and representative scenarios, and provide some key references.

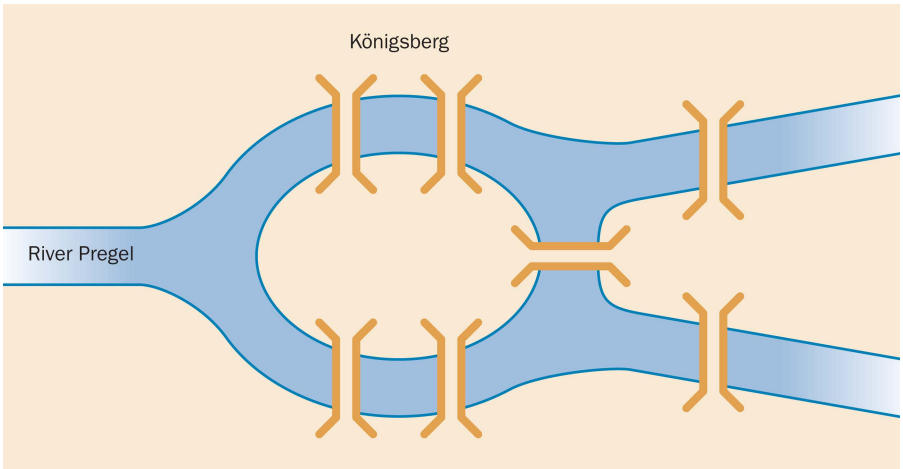


Figure 3.3 Euler's problem of the Seven Bridges of Königsberg. Euler answered the question whether it is possible to find a path crossing each bridge exactly once, and in so doing initiated the field of graph theory.

3.2.1 Properties of Graphs

Let us start with a graph G that describes a biological network and consists of nodes and edges and denote the edge connecting nodes N_i and N_j by $e(N_i, N_j)$ or e_{ij}. A first important and rather intuitive characteristic of a node is its degree, denoted by $\deg(N_i)$, which represents the number of associated edges. For an undirected graph, $\deg(N_i)$ is simply the total number of edges at N_i and this number is the same as the number of N_i's neighbors. In a directed graph, it is useful to define the in-degree $\deg_{in}(N_i)$ and out-degree $\deg_{out}(N_i)$, which refer to the numbers of edges terminating or starting at N_i, respectively (Figure 3.4).

For studies of the entire connectivity structure of a graph G with m nodes $N_1, \ldots, N_m$, one defines the adjacency matrix **A**, which indicates with 1's and 0's whether two nodes are directly connected or not. This **matrix** has the form

$$\mathbf{A} = (a_{ij}) = \begin{cases} 1 & \text{if } e(N_i, N_j) \text{ is an edge in } G, \\ 0 & \text{otherwise.} \end{cases} \tag{3.1}$$

Here, undirected edges are counted as two directed edges that represent the forward and reverse directions. Examples of **A** are given in Figure 3.5. The adjacency matrix contains all pertinent information regarding a graph in a very compact representation that permits numerous types of analyses with methods of linear algebra. Box 3.1 discusses an interesting use of **A**.

In addition to investigating degrees of nodes, it is of interest to study how edges are statistically distributed within the graph G. The principal measure of this feature is the **clustering coefficient** C_N, which characterizes the density of edges associated with a node N. Specifically, C_N quantifies how many edges exist among all neighbors of node N, divided by how many edges there could be theoretically. Let's start with the denominator. If node N has k neighbors, then, theoretically, neighbor 1 could be directly connected with neighbor 2, neighbor 3, 4, all the way to neighbor k, so, there are $k - 1$ connections because neighbor-1-to-neighbor-1 does not count. Next, neighbor 2 could be connected to neighbor 1, but we have counted this particular edge already. Thus, neighbor 2 could be connected to neighbor 3, 4, . . . , k for a total of $k - 2$ connections. Continuing to neighbor $k - 1$, the total number of new connections is 1, namely only to neighbor k; check that the sum of all these connections is $k(k - 1)/2$. It is like the number of handshakes in a room of k people, if everyone shakes hands with everybody else once.

Let's assign the symbol e to the number of edges that are actual neighbors of node N in the graph of interest. This number becomes the numerator for the computation of C_N. Therefore, C_N is defined as

$$C_N = \frac{e}{k(k-1)/2} \quad \text{if G is undirected.} \tag{3.2}$$

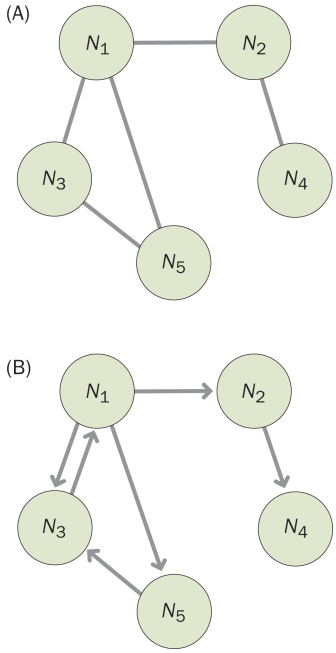
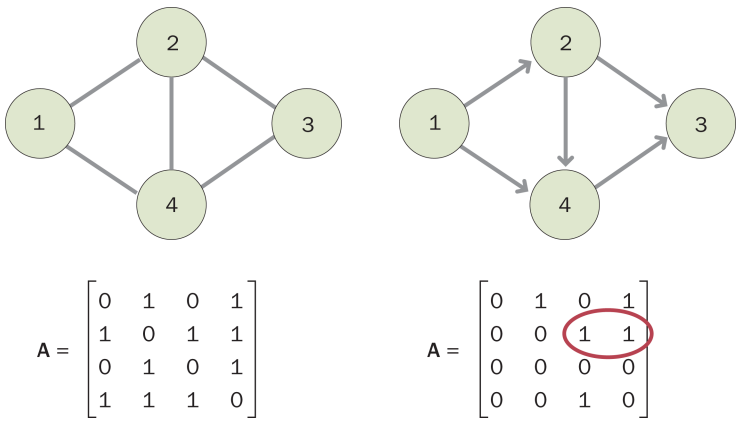


Figure 3.4 Visualization of the concept of degrees associated with nodes. (A) In this undirected graph, node N_1 has neighbors N_2, N_3, and N_5, and therefore its degree is $\deg(N_1) = 3$. Node N_4 has only one neighbor, N_2, and therefore $\deg(N_4) = 1$. (B) In this directed graph, node N_1 has an in-degree $\deg_{in}(N_1) = 1$ and an out-degree of $\deg_{out}(N_1) = 3$, while node N_4 has $\deg_{in}(N_4) = 1$ and $\deg_{out}(N_4) = 0$.



$$\mathbf{A} = \begin{bmatrix} 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 \\ 1 & 1 & 1 & 0 \end{bmatrix} \qquad \mathbf{A} = \begin{bmatrix} 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

Figure 3.5 Two graphs and their adjacency matrices. In (A), each edge is considered bidirectional, which leads to twice as many 1's as edges. (B) shows a directed graph and its adjacency matrix. As an example, the red ellipse indicates that directed edges point from node 2 to nodes 3 and 4. The third row of the matrix contains only 0's because no arrows leave node 3.

BOX 3.1: COMPUTATION OF THE NUMBER OF PATHS BETWEEN NODES

Interestingly, one can use adjacency matrices, as shown in (3.1), to compute the number of paths between two nodes that consist of a fixed number of steps. These numbers are given by powers of the matrices. For instance, to compute all paths of length 2 in the matrix on the right-hand side of Figure 3.5, we multiply the matrix by itself, which results in a new matrix **B**. According to the rules for this operation, each element B_{ij} in matrix **B** is given as the sum of the products of the elements in row i of matrix **A** and the elements in column j of matrix **A**. For instance, element B_{13} is computed as

$$B_{13} = A_{11}A_{31} + A_{12}A_{32} + A_{13}A_{31} + A_{14}A_{34} = 0.0 + 1.1 + 0.0 + 1.1 = 2. \quad (1)$$

Executing the analogous operations for all B_{ij} yields

$$\mathbf{A}^2 = \begin{bmatrix} 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 2 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} = \mathbf{B}. \quad (2)$$

According to this result, there are two alternative two-step paths from node 1 to node 3, while there is one two-step path from 1 to 4 and one from 2 to 3. These three solutions exhaust all possible two-step paths. In this simple case, the results are easily checked from the graph. The reader should confirm these results.

An example is shown in **Figure 3.6A**. The number of neighbors of N has a name that we introduced before: it is the degree of node N, $k = deg(N)$.

The computation of C_N is slightly different if the graph is directed, because the theoretical maximum of edges among neighbors is different: $k(k-1)$. This number is exactly twice the number we computed earlier, because there could be a directed edge from node N to node M and a different directed edge from node M to N. Therefore, the clustering coefficient for a directed graph is

$$C_N = \frac{e}{k(k-1)} \quad \text{(if } G \text{ is directed)}. \quad (3.3)$$

In this case, $k = deg_{in}(N) + deg_{out}(N)$ (Figure 3.6B).

It is useful to define an overall measure of the network's clustering. This is accomplished with the graph's average clustering coefficient C_G, which is defined simply as the arithmetic mean of the C_N for all nodes [14]:

$$C_G = \frac{1}{m}\sum_{N=1}^{m} C_N. \quad (3.4)$$

Intriguingly, biological systems typically have clustering coefficients that are higher than those of randomly connected graphs. Furthermore, with increasing size of the network, the clustering coefficient tends to decrease. This implies that biological networks contain many nodes with low degree but only relatively small numbers of **hubs**, which are highly connected nodes that are surrounded by dense clusters of nodes with relatively low degree [15]. For example, in both the protein interaction network and the transcription regulatory network in yeast, the hubs

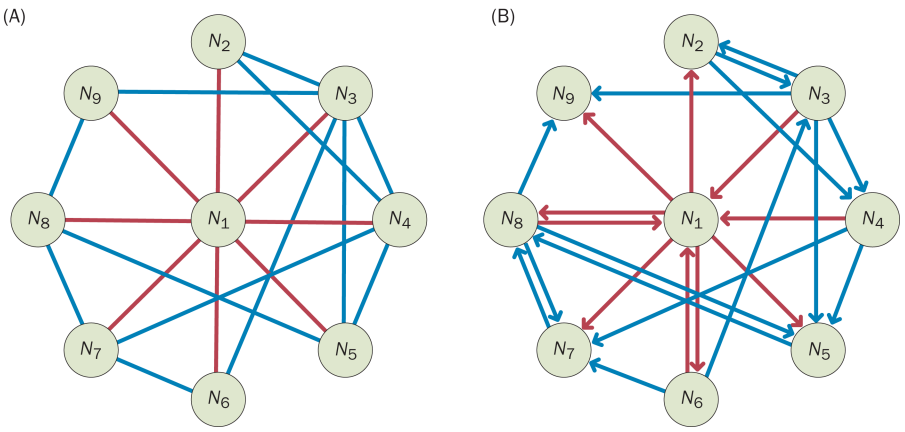


Figure 3.6 Visualization of the clustering coefficient of a node. Node N_1 has 8 neighbors. In the undirected graph (A), these neighbor nodes are connected among themselves through 12 (blue) edges. Thus, $C_{N_1} = (2{\cdot}12) / (8{\cdot}7) = 0.4286$. In the directed graph in panel (B), the neighbor nodes of N_1 are connected among themselves through 15 (blue) edges. Thus, $C_{N_1} = 15 / (8{\cdot}7) = 0.2679$.

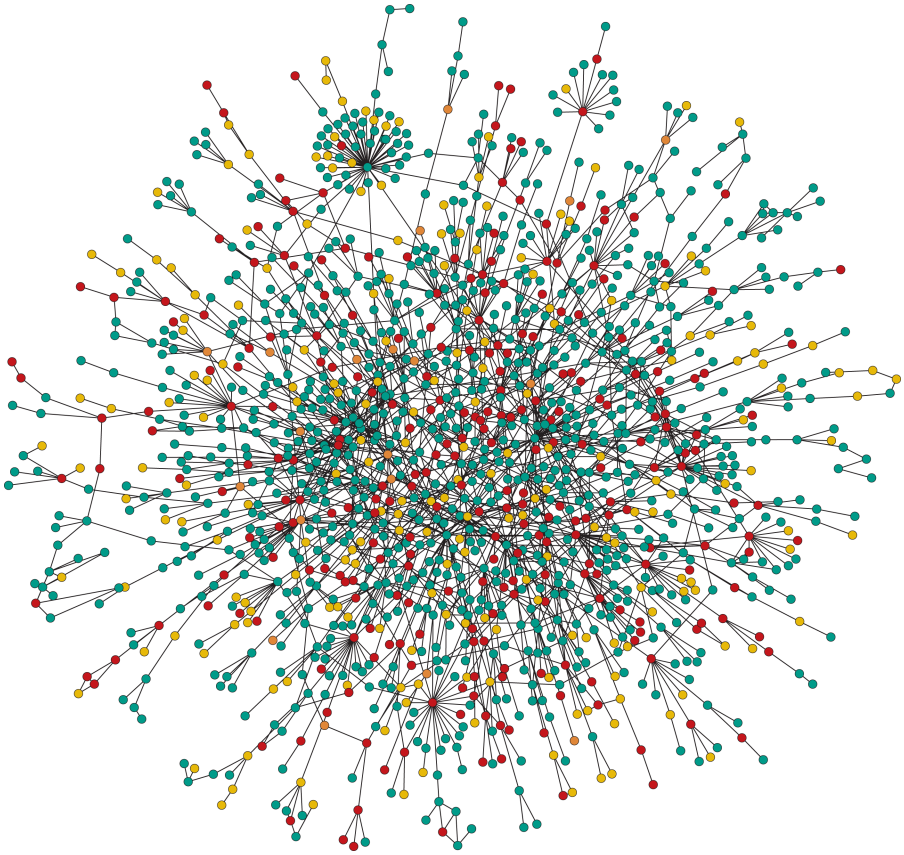


Figure 3.7 One large cluster within the protein–protein interaction network of the yeast *Saccharomyces cerevisiae*. The entire network consists of 1,870 proteins as nodes, which are connected by 2,240 direct physical interactions. The cluster presented here contains about 80% of this network. The graph clearly shows a number of hubs that are highly connected, as well as the vast majority of nodes with a low degree. Each node is colored to signify the phenotypic result if the corresponding protein is removed (red, lethal; green, nonlethal; orange, slow growth; yellow, unknown). (From Jeong JP, Mason SP, Barabási A-L & Oltvai ZN. *Nature* 411 [2001] 41–42. With permission from Springer Nature.)

are significantly more often connected to nodes of low degree, rather than to other hubs (Figure 3.7) [16, 17]. We will return to this feature later in this chapter.

A very important characteristic of a graph G as a whole is its **degree distribution** $P(k)$, which measures the proportion of nodes within G that have degree k. If m_k is the number of nodes of degree k and m is the total number of all nodes, then

$$P(k) = m_k/m. \tag{3.5}$$

Note that $P(k)$ is not a single number, but a set of numbers, with one degree value for each k ($k = 0, 1, 2, \ldots$). The division by m assures that the sum of $P(k)$ over all values of k is equal to 1. For the computation of $P(k)$, the directions of edges are typically ignored.

Many characterizations of graphs, including degree distributions, are performed by using comparisons with random graphs, which were first studied by Erdös and Rényi in the late 1950s [18]. In such an artificially created graph, a user-specified number of edges is randomly associated with nodes. One can show with mathematical rigor that the degree distribution in this case is a binomial distribution, which resembles a skinny bell curve with small variance. In other words, most nodes have a degree that is close to average.

The binomial degree distribution of random graphs is very different from what is frequently observed in biological and other real-world systems, where a few hubs are connected to disproportionally many other nodes, while most other nodes are associated with much fewer edges than in a random graph (Figure 3.8). Specifically, $P(k)$ often follows a **power-law distribution** of the form

$$P(k) \propto k^{-\gamma}, \tag{3.6}$$

where the exponent γ has a value between 2 and 3 [19]. Plotting such a power-law degree distribution in logarithmic coordinates results in a straight line with negative slope $-\gamma$ (Figure 3.8). A network whose degree distribution more or less follows

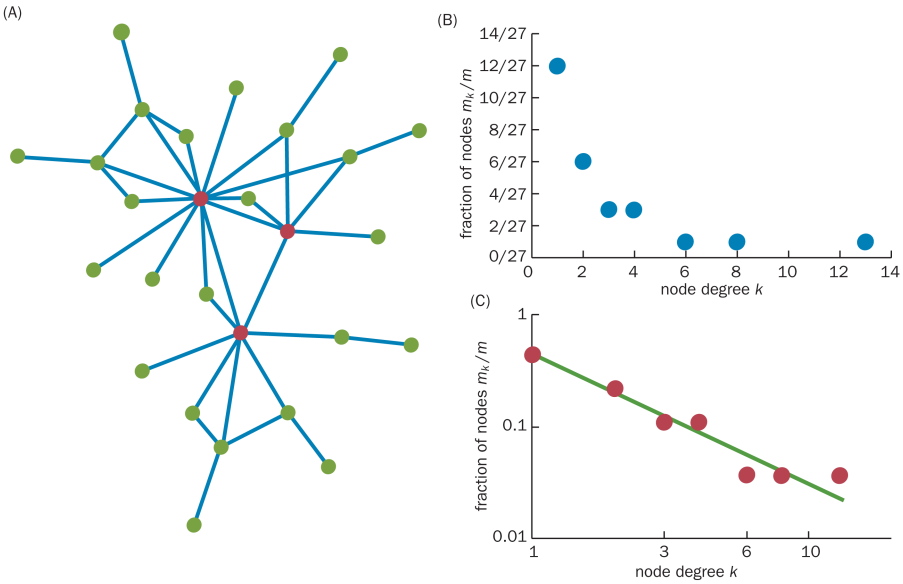


Figure 3.8 Example of a small scale-free network. (A) Among the 27 nodes, three (red) are the most highly connected hubs. (B) The relative number of links per node, plotted against the number of nodes with a given number of links, *k*, shows a decreasing power-law relationship. (C) If the relationship in (B) is plotted in logarithmic coordinates, the result is approximately linear, with negative slope *γ*. (Adapted from Barabási A-L & Oltvai ZN. *Nat Rev Genet.* 5 [2004] 101–113. With permission from Springer Nature.)

a power law is called a **scale-free** network. This terminology was chosen because the power-law property is independent of the size (or scale) of the network.

While a power-law distribution is often observed for most nodes in a large biological network, the plot is seldom truly linear for very highly and very sparsely connected nodes within such a network. Nonetheless, except for these extremes, the power-law function often holds quite nicely, which demonstrates, unsurprisingly, that most biological systems are clearly not organized randomly (see, for example, [20]).

An interesting feature of scale-free networks is that the shortest path between two randomly chosen nodes is distinctly shorter than that in a random graph. Here, a path is an alternating sequence of nodes and edges in a graph, leading from a start node to an end node, such as

$$N_6;\ e(N_6,N_1);\ N_1;\ e(N_1,N_4);\ N_4;\ e(N_4,N_5);\ N_5;\ e(N_5,N_2);\ N_2 \qquad (3.7)$$

in Figure 3.9. Of particular interest is the shortest path (or distance) between two nodes, because it contains important clues about the structure of the network. In Erdős and Rényi's random graphs [18], the average shortest distance is proportional to the logarithm of the number of nodes, log *m*. By contrast, the average shortest distance in a scale-free network with a power-law degree distribution is shorter and increases more slowly for larger networks [21]. Indeed, analyzing over 40 metabolic networks with between 200 and 500 nodes, Jeong and collaborators found that a few hubs dominated the connectivity patterns and that, as a consequence, the average shortest distance was essentially independent of the size of the network and always had an amazingly low value of about three reaction steps [22]. In a different interpretation, one may also consider the average distance as an indicator of how quickly information can be sent through the network. According to this criterion, biological networks are indeed very efficient.

Some authors refer to the average shortest distance as the diameter of the network. However, caution is needed, because other authors use the same term for the maximum number of edges that must be traversed between any two nodes along a shortest path.

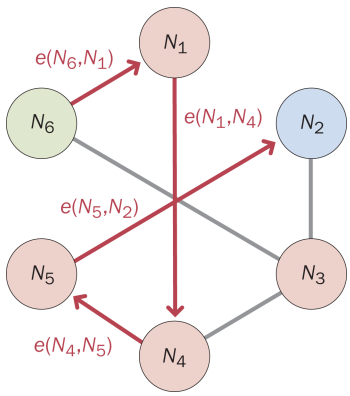


Figure 3.9 A possible path from start node N_6 to end node N_2. The path uses the edges $e(N_6,N_1)$, $e(N_1,N_4)$, $e(N_4,N_5)$, and $e(N_5,N_2)$. The path is not the shortest possible path, which would use $e(N_6, N_3)$ and $e(N_3,N_2)$. It is customary not to allow the same node or edge to appear twice in a path.

3.2.2 Small-World Networks

If a network is scale-free (i.e., if it has a power-law degree distribution and therefore a small average shortest distance) and if it furthermore has a high clustering coefficient, it is called a **small-world** network. This terminology reflects the special structure of these networks, which consists of several loosely connected clusters of nodes, each surrounding a hub. In this organization, most nodes have a low degree and are therefore not neighbors of each other, but moving from any one node to any other node in the network does not take many steps, because of the existence of the well-connected hubs with high degree. We have already mentioned metabolic networks, which have average **path lengths** in the range of 2–5 steps. Ecological studies have found the same length ranges in many food webs [23]. For other biological examples and further discussions, see [24, 25]. Outside biology, the situation is similar in commercial flight routes, which use major hubs combined with connecting flights to smaller airports. The small-world property is furthermore reminiscent of John Guare's famous play *Six Degrees of Separation*, whose premise is that essentially every human is connected to every other human by a chain of six or fewer acquaintances.

BOX 3.2: CONSTRUCTION OF REGULAR, SMALL-WORLD, AND RANDOM NETWORKS

Interesting connectivity patterns in networks can be constructed in various ways. To create a random network [18], N nodes and K (directed or undirected) edges are chosen. For each edge, a source node and a target node are assigned; two edges between the same nodes and edges from a node back to itself are not allowed. These networks have low clustering coefficients and do not have a power-law degree distribution as small-world networks do. They often also have isolated nodes, which are not connected to any other nodes [26].

In two interesting papers, Watts and Strogatz [14] and Newman and Watts [27] created interesting networks by starting with a regularly patterned ring of nodes, in which each node is associated with four edges. Namely, the node is connected to its two immediate neighbors and also to their immediate neighbors (**Figure 1**). The authors then selected a constant probability p anywhere between 0 and 1, with which each edge could be disrupted and reconnected to a new, randomly chosen, node. Of course, for the extreme case of $p = 0$, the structure of the network remained unchanged. The other extreme case, $p = 1$, was shown to result in a random network. For values of p strictly between 0 and 1, other structures were obtained. Depending on p, the graph can have a clustering coefficient and short average pathway lengths. However, all nodes are of a similar degree, and the degree distribution for $p > 0$ is not a power-law but resembles that of a random graph without isolated nodes [26].

Due to their prominence in a variety of applications, a lot of effort has been devoted to understanding small-world networks (see, for example, [26]). For instance, it was demonstrated that the small-world property emerges automatically when a network grows by preferentially adding new edges to nodes that are

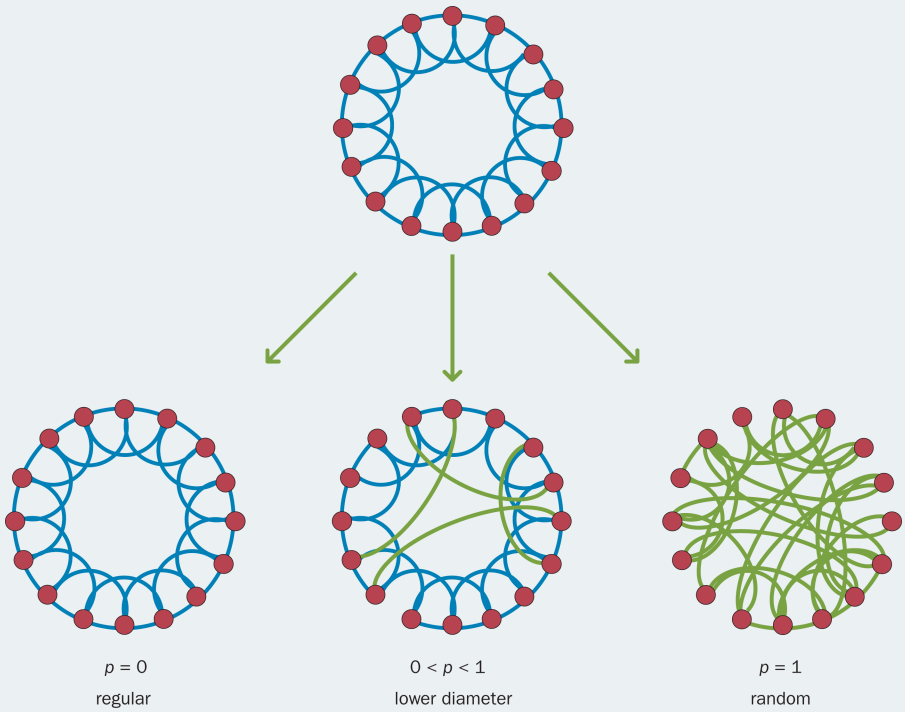


Figure 1 Construction of different types of networks, according to Watts and Strogatz [14] and Newman and Watts [27]. The construction begins with a regular network, in which each node is connected to its immediate neighbors and their immediate neighbors (top of figure). Considering one node after another, the edge to one of its nearest neighbors is removed with a probability p ($0 < p < 1$) and replaced with a new edge to a randomly selected node; double edges are not allowed. In a second round, the process is repeated for second-nearest neighbors (bottom of figure). For $p = 0$, no edges are changed and the regular network is retained; for p between 0 and 1, a few edges (green) are replaced and the result can be a small-world network; for $p = 1$, all edges are replaced and the network exhibits random connectivity.

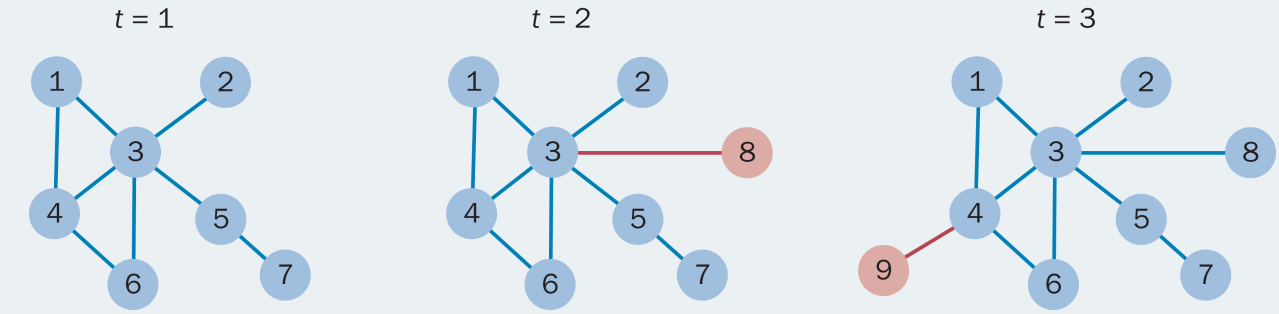


Figure 2 Growth of a graph by preferential attachment. The three graphs illustrate two steps of node addition during the evolution of a graph toward a small-world network. The degrees and attachment probabilities of the initial graph ($t = 1$) are shown in Table 1. Node 3 has the highest degree, and the probability of a new node attaching (5/16) is the highest among all nodes. Suppose a new node is indeed added to this node by chance (biased by the probabilities in Column 3). As a consequence, the degree distribution and attachment probabilities change slightly ($t = 2$), with the probabilities of lowly connected nodes decreasing. Suppose that, in the next transition, the node with the second highest degree (Node 4) is selected by chance, even though its degree is smaller than that of Node 3. The result is the graph at $t = 3$. One can prove with methods of probability theory that many steps of this type tend to yield a small-world network [28].

TABLE 1: DEGREES OF GRAPHS RESULTING FROM GROWTH BY PREFERENTIAL ATTACHMENT (*CF.* FIGURE 2)

$t = 1$			$t = 2$	$t = 3$
Node	**Degree**	**Attachment Probability for a New Edge**	**Degree**	**Degree**
1	2	2/16	2	2
2	1	1/16	1	1
3	5	5/16	6	6
4	3	3/16	3	4
5	2	2/16	2	2
6	2	2/16	2	2
7	1	1/16	1	1
8	–	–	1	1
9	–	–	–	1
	Sum = 16	Sum = 1	Sum = 18	Sum = 20

already highly connected. Specifically, a distribution with $\gamma = 3$ is the result of a growth process where, for each addition of new a node, an edge is connected from this node to an existing node N with a probability that is directly proportional to $\deg(N)$ [28]. Variations on the mechanism of preferential attachment lead to other γ values that can range anywhere between 2 and ∞. An example is shown in **Figure 2** and Table 1.

Entirely different mechanisms for the evolution of scale-free and small-world networks have been proposed. For instance, according to the duplication–divergence (DD) model, a gene or protein network may evolve owing to the occasional copying of individual genes or proteins, which is followed by mutations and pruning [29]. The DD model can lead to $\gamma < 2$. An illustration of model multiplication is given in **Figure 3.3**. For other examples, see [30].

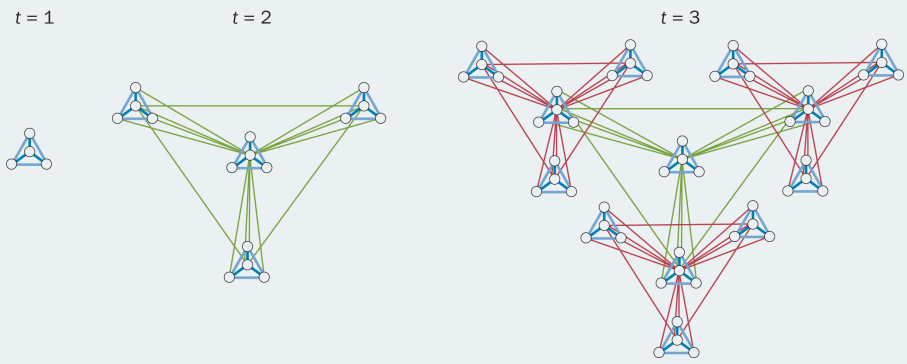


Figure 3 Evolution of a small-world network. In this case, the network grows over time through sequential multiplication of the same motif.

It is not surprising that the hubs in a gene or protein network are often among the most important components and that their destruction or mutation is frequently fatal. In fact, it seems that the degree of a protein within a PPI network may be directly correlated with the fitness of the gene that codes for it [29]. This feature correlates in an interesting fashion with the robustness of a small-world network or, specifically, its tolerance to attacks: attacks on randomly selected nodes are easily tolerated in most cases, because the likelihood of hitting a hub is low; by contrast, a single, targeted hit on a hub can break up the entire network [16, 22]. Using this rationale in a different context, the identification of hubs may be important for drug target identification. The interesting co-occurrence of robustness and fragility has been observed in many complex networks and systems, including both biological and engineered examples, and it may be a characteristic design principle of such systems [31–33].

While molecular networks have recently received the most attention, it should be mentioned that static networks have played a role in systems ecology for a long time. Beginning with the pioneering work of Joel Cohen [34], graph representations have been used to study food webs, the flow of material through them, and the robustness and reliability of their connectivity structures. It was shown, for instance, that typical food webs have maximal path lengths between 3 and 5, that marine paths tend to be longer than those in terrestrial systems, and that the reliability of a food web decreases with increasing length [23]. Food webs often exhibit loops, where A feeds on B, B on C, and C on A [35].

While biological systems are often scale-free, their sub-networks sometimes have a different structure. For instance, sets of nodes within a small-world network are highly connected around hubs and may even form **cliques**, within which each node is connected to every other node. Clearly, the connectivity patterns of a clique and the larger network are different, and caution is in order when only parts of larger networks are investigated.

One should also keep in mind that most networks are in reality dynamic. For instance, PPI networks change during development and in response to disease. They sometimes even change periodically on a shorter time scale, as is the case during the **cell cycle** in yeast [36]. Han and collaborators [37] analyzed the temporal properties of PPI hubs in yeast during the cell cycle and in response to stress and postulated two types of hubs: party hubs, which interact simultaneously with most of their neighbors and date hubs, which connect to different neighbors at different times and possibly in different cellular locations. The dynamic changes associated with hubs may indicate a control structure of organized modularity, where party hubs reside inside defined, semi-autonomous modules, which perform some biological functionality, while date hubs connect these modules to each other and organize their concerted function. However, the distinction between party and date hubs is not always clear-cut [38].

3.2.3 Stoichiometric Networks

Metabolic networks will be described in greater detail in Chapter 9, but they are intuitive enough to serve here as an illustration. Metabolic networks are uniquely suited for analyses beyond clustering coefficients and degree distributions, because they represent the flow of actual material. In contrast to proteins, whose interactions are often seen in their association and co-location, metabolic pools receive and release material, and all this material must be accounted for. Thus, if a certain amount of material leaves one pool, the same amount will be received by other pools. If glucose is converted into glucose 6-phosphate, we know that the amount of glucose used up will equal the newly added amount of glucose 6-phosphate. These precursor–product relationships tremendously constrain what may happen in a metabolic system and thereby allow a variety of interesting analyses.

For simplicity of discussion, we will focus on small systems, but it should be noted that some analyses in the literature have addressed metabolic networks of considerable size (see, for example, [15, 20, 22, 39–41]. Generically, metabolic systems describe how organic material is converted from one chemical form into another, such as the conversion of glucose into glucose 6-phosphate in the first reaction of glycolysis. As before, all compounds of interest are represented as nodes, and enzymatic reactions and transport steps are represented as edges. As noted earlier, alternatives to this type of representation are bi-partite graphs

(Figure 3.2), in which two different types of nodes represent metabolites and enzymatic reactions and which are formulated and analyzed as Petri Nets [12, 13]. In the spirit of this chapter, we study the **fluxes** through metabolic systems at steady state, and this restriction allows us to ignore many aspects, such as temporal, kinetic, and regulatory features, which we will study in Chapter 9.

In a generic network of metabolic reactions, organic compounds flow from node to node. A special but very common situation is the steady state, which in physiology is also sometimes called homeostasis. In this state, no variable changes its value over time. Variables in natural systems typically do change in quantity, but very many systems, time and again, return to this homeostatic state.

Under the assumption of a steady state, all fluxes entering a given node must collectively be equivalent in amount to all fluxes leaving this node, otherwise the quantity represented by the node would change in magnitude, which may be measured as amount or concentration. Considered purely at a steady state, metabolic systems become static networks that can be represented as graphs. For simplicity of discussion, let us assume that all edges in the metabolic network are unidirectional; that is, we have a situation as in network (A) of Figure 3.1, where material can only flow from left to right and up but not backwards. The **constraints** imposed by the material flow lead directly to mathematical relationships, which are collectively called **stoichiometry** and which we illustrate representatively with node N_3 in the generic pathway of Figure 3.10. Namely, quite obviously, the following relationship holds:

$$F_{13} + F_{23} = F_{34}. \tag{3.14}$$

In words: the same amount of material supplying the node equals the amount of material leaving this node. For a node such as N_2, we dissect the bidirectional flux F_{20} into two unidirectional, forward and reverse, fluxes F_{02} and F_{20}, which allows us to write a simple linear balance equation like (3.14) again.

Investigating fluxes at the steady state reduces the mathematical description of complex systems from a differential into an algebraic form. This simplification has led to enormous research activities on metabolic systems, which sometimes are quite large and contain dozens, if not hundreds, of metabolites and reactions. One reason for this strong interest has been a desire to manipulate the natural flux distributions toward a desired goal, such as the enhanced production of an organic compound in microorganisms, and the field of metabolic engineering has fully embraced these methods (see, for example, [42–44]). In its basic form, this type of investigation, called stoichiometric analysis, goes back several decades [45, 46]. At its core is a stoichiometric matrix, which succinctly describes the distribution of all material fluxes in a metabolic network, just as (3.14) does for a single node.

As an example, consider the small network in Figure 3.11, which contains four substrate pools $A, \ldots, D$ and seven fluxes $R_1, \ldots, R_7$. The corresponding stoichiometric matrix **N** is also shown in Figure 3.11, with entries of +1 showing material entering a pool and entries of –1 representing fluxes leaving the pool.

If two molecules are needed to create one product molecule, the matrix contains entries of –2 and 2 for the corresponding reaction and the affected metabolites.

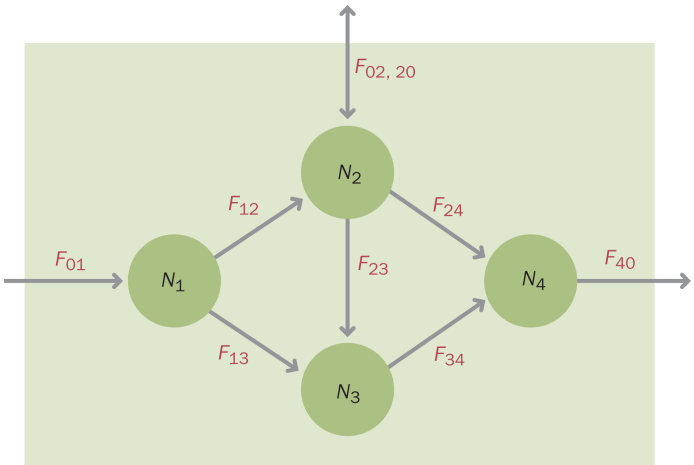
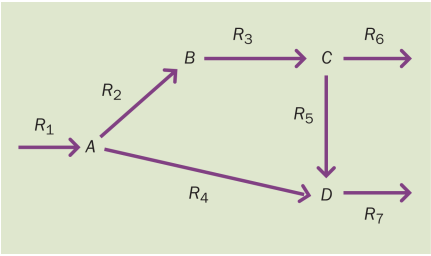


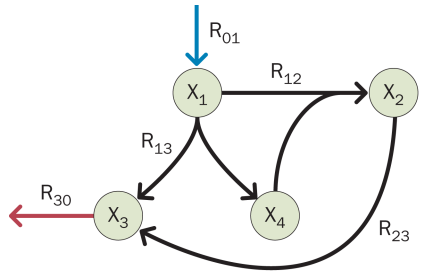
Figure 3.10 Generic network with four nodes and fluxes entering and leaving them. At any steady state of the network, the sum of all fluxes entering a node is equal in magnitude to the total efflux from this node.



$$\mathbf{N} = \begin{array}{c} \begin{matrix} R_1 & R_2 & R_3 & R_4 & R_5 & R_6 & R_7 \end{matrix} \\ \begin{bmatrix} 1 & -1 & 0 & -1 & 0 & 0 & 0 \\ 0 & 1 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & -1 & -1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 & -1 \end{bmatrix} \end{array} \begin{matrix} A \\ B \\ C \\ D \end{matrix}$$

Figure 3.11 Generic flux network (left) and its stoichiometric matrix N (right). The matrix **N** quantifies which network components are connected to each other.

Similarly, if a molecule is split into three of the same type, the corresponding reaction requires entries of –3 and 3. As a small but more complicated example, consider the artificial metabolic network in **Figure 3.12**, where the nodes represent metabolites with different numbers of carbon atoms. The system contains one influx (R_{01}) and one efflux (R_{30}), which are colored blue and red, respectively. Molecule X_1 is split unevenly into two molecules, X_3 and X_4. It also combines with X_4 to form molecule X_2. Balancing the numbers of carbon atoms of each molecule in each reaction, suppose the internal reactions perform the following conversions:



metabolite	number of carbon atoms
1	3
2	2
3	1
4	1

Figure 3.12 System of four metabolites and complicated reaction network. The indices of each reaction refer to the source and sink of the reaction, with 0 being the "outside" of the system. The molecules possess different numbers of carbon atoms (see Table Insert). The blue arrow indicates mass entering the system, while the red arrow represents an efflux of mass leaving the system.

$$R_{12}: X_1 + 3X_4 \rightarrow 3X_2$$

$$R_{13}: 2X_1 \rightarrow 5X_3 + X_4$$

$$R_{23}: X_2 \rightarrow 2X_3$$

The reaction scheme leads directly to the corresponding stoichiometric matrix:

$$\mathbf{N} = \begin{array}{c} \begin{matrix} R_{01} & R_{12} & R_{13} & R_{23} & R_{30} \end{matrix} \\ \begin{pmatrix} 1 & -1 & -2 & 0 & 0 \\ 0 & 3 & 0 & -1 & 0 \\ 0 & 0 & 5 & 2 & -1 \\ 0 & -3 & 1 & 0 & 0 \end{pmatrix} \end{array} \begin{matrix} X_1 \\ X_2 \\ X_3 \\ X_4 \end{matrix}$$

An important practical application of stoichiometric analysis is the desire to reroute the fluxes of a metabolic system in some optimal manner. For instance, the goal for the system in Figure 3.11 might be to minimize R_6 and to produce as much material through R_7 as possible. Usually, the fluxes are subject to R_7 additional constraints, such as admissible minimum and maximum values. Some optimization approaches of this type are discussed in Chapter 9.

3.3 NETWORK DYNAMICS

All discussions of networks so far were static, i.e., the networks did not change over time. It is of course possible that they do change. Chapter 4 contains a general discussion of models that capture changes in networks and systems that occur in discrete time intervals. Here, we will just look at one specific case of networks, called

Boolean networks, where the nodes can be turned on or off depending on the status of other nodes. These network models are useful in the analysis of gene regulatory systems and for some signaling phenomena.

3.3.1 Boolean Networks

Suppose a network consists of n nodes and we might imagine, as an example, that these nodes represent the expression status of each of n genes, which in the simplest implementation may just be *on* (meaning that they are expressed) or *off* (for instance, because they are presently repressed). The gene network can respond to external factors, which affect the current combination of expression. Suppose furthermore that an expressed gene ultimately leads to the production of a transcription factor or repressor that may affect its own expression as well as that of other genes in the system.

For simplicity of discussion, let's arrange the 25 genes in a 5 × 5 grid of boxes (**Figure 3.13A**) and name the genes according to their position, such that G_{11} refers to the top left corner, G_{12} lies just to the right of G_{11}, G_{21} is just below G_{11}, and so on. Of course, the genes could be arranged in many other ways. As an arbitrary assignment, let's say that a white box means the gene is *on* and that black means *off*. Because it is cumbersome to compute with colors, we translate *off* into 0 and *on* into 1. Again, this setting is arbitrary. We can now formulate statements like

$$For\ i, j = 1, \ldots, 5$$

$$G_{ij} = \begin{cases} 1, & if \quad i = j \\ 0, & if \quad i \neq j \end{cases} \tag{3.15}$$

which means that the genes on the northwest–southeast diagonal of the grid are on (Figure 3.13B).

The beauty of these types of networks lies in the simplicity of defining their dynamics. In other words, we are leaving the realm of static networks and are entering the world of systems that change over time. The following chapters focus almost exclusively on such changes.

Suppose that time moves forward in a discrete, stepwise fashion. Namely, we formulate time as $t = 0, 1, 2, \ldots$ and define the status of each $G_{ij}(t)$ as a function of all $G_{ij}(t - 1)$ (see Chapter 4 for a more general discussion of discrete systems). In **Boolean networks**, this function is defined as a specific rule that determines what happens to each state during the transition from one time point to the next. As a simple example, consider the following rule:

$$For\ i, j = 1, \ldots, 5$$

$$G_{ij}(t+1) = \begin{cases} 1, & if \quad G_{ij}(t) = 0 \\ 0, & if \quad G_{ij}(t) = 1 \end{cases} \tag{3.16}$$

Put into words, the rule says that 0 switches to 1 and 1 switches to 0 at every time transition (Figure 3.13C). It is not hard to imagine that this particular rule defines a blinking pattern switching back and forth between Figures. 3.13B and C. This dynamic is independent of the initial state of the network. In other words, we can

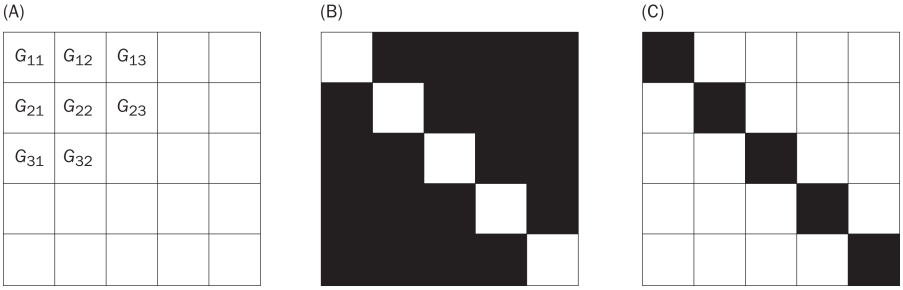


Figure 3.13 An intuitive example of a Boolean network. The network is represented by a grid with states that can either be *on* or *off*. (A) Naming of grid boxes in a systematic fashion. (B) *On–off* pattern as defined by (3.15). (C) *On–off* pattern as defined by (3.16) if the previous state was that in (B).

define the initial profile of expression, $G_{ij}(0)$, any way we want, as long as all settings are either 0 or 1.

This type of network with on/off states and transition rules is often called a Boolean network, honoring the nineteenth-century British mathematician and philosopher George Boole, who is credited with inventing the logical foundation of modern computer science. Much has been written about Boolean networks, and an excellent starting point for further exploration is Stuart Kauffman's widely recognized treatise *The Origins of Order* [47].

The rules in a Boolean network may be much more complicated than in the illustration above. For instance, a rule could demand that G_{ij} switches if the majority of its neighbors are 1. How could we formulate such a rule? Let's start with a node that is not located at one of the sides of the grid. Thus, it has eight direct neighbors, which are

$$G_{i-1,j-1}; G_{i-1,j}; G_{i-1,j+1}; G_{i,j-1}; G_{i,j+1}; G_{i+1,j-1}; G_{i+1,j}; G_{i+1,j+1}.$$

Because each neighbor can only take values of 0 or 1, we may formulate the rule as

$$G_{ij}(t+1) = \begin{cases} 1 & \text{if } G_{ij}(t)=0 \text{ and sum of all neighbors } > 4 \\ 0 & \text{if } G_{ij}(t)=0 \text{ and sum of all neighbors } \leq 4 \\ 0 & \text{if } G_{ij}(t)=1 \text{ and sum of all neighbors } > 4 \\ 1 & \text{if } G_{ij}(t)=1 \text{ and sum of all neighbors } \leq 4 \end{cases} \quad (3.17)$$

The rule is iteratively applied at the transition from any time t to time t+1, during which the values of the neighbors of each node at time t determine the value of this node at time t+1. One should note that the rule in Eq. (3.17) may be written in different, equivalent ways, some of which will certainly be more compact and elegant.

For intriguing entertainment with a more complex network that has many conceptual similarities with a Boolean network, explore the *Game of Life*, which was created by John Conway in 1970 and attracted interest from an almost Zen-like following of biologists, mathematicians, computer scientists, philosophers, and many others. The game is still available on the Internet and even has its own Wikipedia page.

Our representation of a Boolean network as a grid is a bit awkward, and we may instead locate the genes G_i as nodes in a directed graph, where incoming arrows show the influences of other genes (G_j, G_k, . . .), while outgoing arrows indicate which genes are affected by the gene G_i. A typical illustrative example is given in **Figure 3.14**. Because of evolutionary advantages with respect to robustness, gene networks appear to be only sparsely connected, and most genes have only one or two upstream regulators [48], which greatly facilitates the construction of gene regulatory network models.

The operation of this network is now defined by a specific rule for each gene G_i at the transition from time t to $t+1$, which is determined by the expression state of some or all other genes (or gene products) at time t. Under well-defined constellations of the system, G_i will "fire" (become 1), while it is silent ($G_i = 0$) otherwise. As an example, suppose that only G_1 in Figure 3.14 is on and all other genes are off at $t = 0$. The activation arrows and inhibition symbols suggest that, at $t = 1$, G_2 will still be off, while G_3 and G_4 will turn on.

To capture the full dynamics of the network, it is necessary to define for each gene whether it will be on or off at time $t+1$, given an on-off pattern of all genes at time t. These complete definitions are needed, because, for instance, the graphical representation in Figure 3.14 does not prescribe what will happen to G_3 if both G_1 and G_2 are on. Will the activation trump the inhibition or will the opposite be true? The state of G_3 at $t+1$ may also depend on its own state at time t. In general, the rules may be very complicated, but the execution of simulations is simple, which implies that rather large and complex regulatory systems may be constructed and analyzed. As an instructive example, Davidich and Bornholdt [49] modeled the cell cycle regulatory network in fission yeast with a Boolean network. A simple and a complicated example of applications of Boolean methods are included in Section 3.5, Network Motifs.

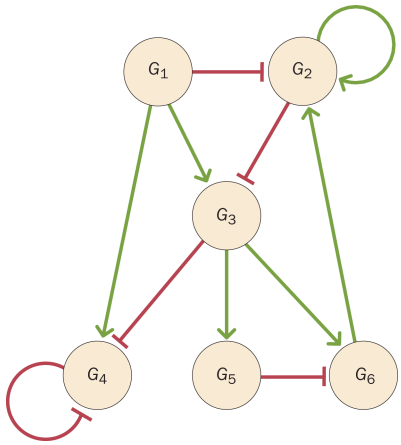


Figure 3.14 Representation of a generic gene interaction network as a graph. Arrows indicate activation while lines with terminal bars represent inhibition or repression.

Two features of Boolean networks are of particular significance. First, there is no real size limitation, and because the rules are usually defined locally, that is, for each node, it is often relatively easy to construct very large dynamic processes with complicated on/off patterns. However, even if one knows all rules, it is sometimes difficult to predict with intuition alone how a network will evolve. Second, it is conceptually easy to expand the number of possible responses per state. This expansion allows grey tones, which might correspond to several (discrete) expression levels of genes. The rules become somewhat more complicated, but the principles remain the same. It is even possible to allow for a continuous range of each expression state. With this expansion, the formal description becomes much more complicated and resembles that of an artificial neural network, except that an artificial neural network allows modifications of the strengths of connections between nodes (see [50] and Chapter 15). An expansion that is far from trivial is the permission to make the rules themselves time-dependent or adaptive in a sense that they change in response to the overall state of the network.

A limitation of traditional Boolean networks is indeed that they are typically not adaptive or allow time-dependent rules and that they have no memory: as soon as a new state is reached, the previous state of the network is forgotten. In many biological systems such memory is important. Another limitation is the strictly deterministic nature of these networks: given the initial state and the transition rules, the future of the network is completely determined. In order to overcome this particular limitation and to allow for biological variability and stochasticity, Shmulevich and colleagues proposed probabilistic Boolean networks, where the transitions between states occur with predefined probabilities [51, 52]. This set up is the basis for Markov models that will be discussed in Chapter 4 and for Agent-Based Models, which are discussed in Chapter 11.

3.3.2 Network Motifs

It may appear that every network is different. Maybe a particular network possesses the small-world property, like many other networks, but the details always seem unique. For large networks this impression may be true, but there are patterns among small networks or sub-networks of larger networks that emerge time and again. These **network motifs** consist of specific, small graphs that are encountered more often than a probability analysis would predict in a randomly connected graph [10, 53]. Large networks are often composed of many of these motifs.

The simplest network motif, containing just one node, is **auto-regulation**, which may be negative or positive (**Figure 3.15**). It should be noted that this motif covers situations with more than a single node: the arrow feeding back may consist of a chain of processes. For instance, negative auto-regulation is frequently found among transcription factors that repress their own transcription. While the transcription factor is the only node explicitly considered, the system in reality contains at least one gene, mRNA, ribosomes, and other molecular components. Negative auto-regulatory loops are very stable and reduce fluctuations in inputs. However, if the feedback is delayed, the system may also oscillate and eventually lose stability. Positive auto-regulation permits a cell to assume multiple internal states, and, in particular, toggling between two stable states. We will discuss one example in Chapter 10. Other examples have been reported for the genetic switch between lytic and lysogenic responses in the bacteriophage λ as well as in signaling cascades and cell cycle phenomena (see, for example, [54, 55]). We discuss a toggle switch in more detail in Chapter 7.

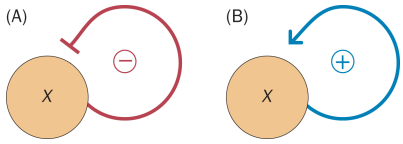


Figure 3.15 Network motifs can be very simple. Shown here are negative (A) and positive (B) auto-regulation motifs. The regulation arrows may encompass several steps.

A slightly more complicated, quite prevalent network motif is the **bifan**, which consists of two source nodes sending signals to two target nodes (**Figure 3.16A**) [53]. This design is interesting because it allows temporal regulation [56]. Applied to signal transduction, a bifan can also sort, filter, de-noise, and synchronize signals. The interactions between the two signals arriving at a target may be of different types, with examples shown in **Figure 3.16B**. In the case of an **AND gate**, a signal is only transduced if both input sources are active, while in the case of an **OR gate**, a single input source is sufficient. An OR gate confers reliability onto the system, because one source is sufficient, while the AND gate permits the detection of signal coincidence and can be employed to prevent a target from being turned on

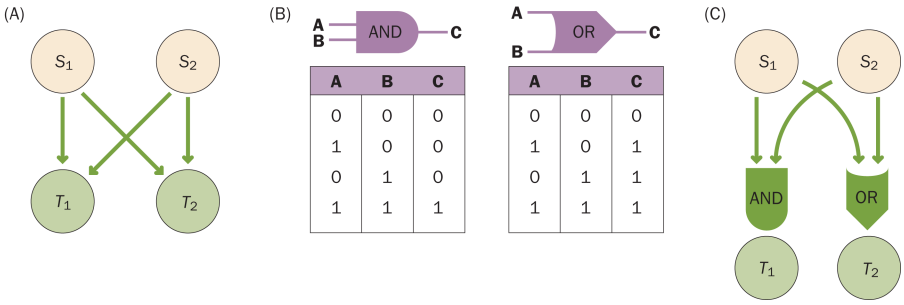


Figure 3.16 Bifans and gates. A: Typical representation of a bifan, which transduces signals from two sources (S_1, S_2) to two targets (T_1, T_2). B: AND and OR gates permit different modes of signal integration, as indicated in the Boolean tables. For an AND gate, C only fires if A and B both are active. For an OR gate, C fires if at least one of the two sources is active. C: Bifan with specific gates, causing different response patterns in T_1 and T_2.

spuriously when only one signal is present. Applied to the bifan, these gates determine which target is firing for different combinations of signals (**Figure 3.16**C).

It is not surprising that small motifs can be combined, both in nature and in theory, to form larger regulatory structures. These larger structures can be implemented in parallel or by stacking smaller motifs. They can also be complemented with additional components and links, thereby creating a potentially huge repertoire of possible responses. A well-known example of moderate complexity is the cross-talk between signaling cascades (see Chapter 10), which can create sophisticated response patterns such as in-band filters, where the system responds only to a signal within a strictly bounded range, or an *EXCLUSIVE OR* function, where the system responds only if exactly one of two sources is active but not both [57].

A complicated illustration of the power of Boolean networks consisting of numerous gates is an analysis of T cell receptor signaling [58], which is displayed in **Figure 3.17**. In this example, many signaling molecules are involved in the response of these immune cells to the presence of antigens. Some are arranged in linear pathways, some as bifans and others as signaling cascades, which we will discuss in Chapter 10.

While they are very important and instructive, small motifs often do not reveal characteristic topological features of larger networks [59]. At the same time, searching for all possible motifs of even moderate size within a large network is not a trivial matter; in fact, it is currently not computationally feasible to identify all motifs of ten or more nodes within realistically sized networks of a few thousand nodes and links. A strategy for addressing the situation is to reduce the scope of the search. To this end, Ma'ayan and collaborators [59] successfully searched large

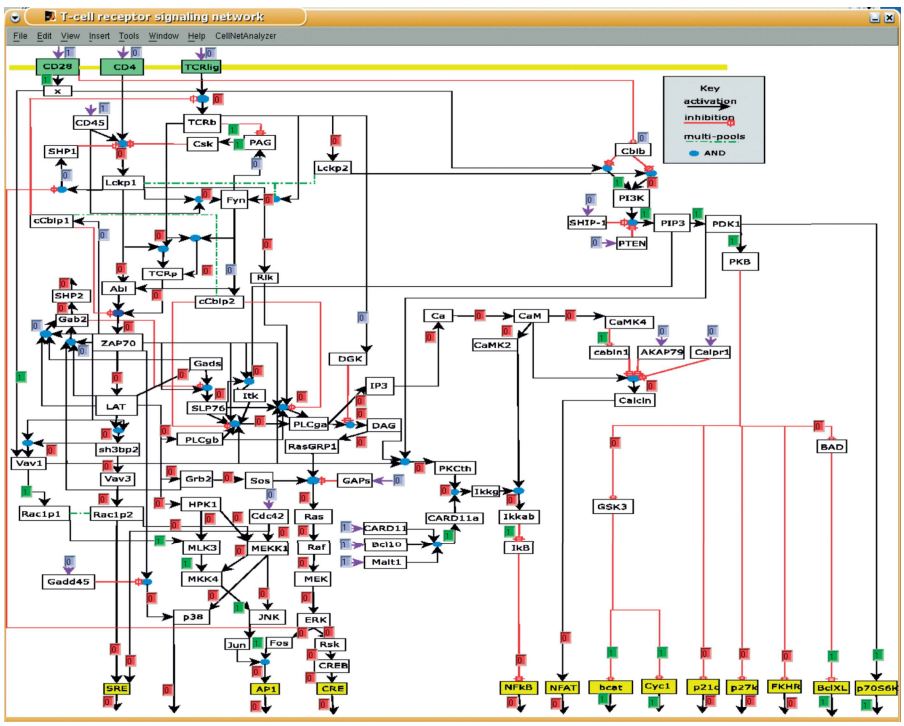


Figure 3.17 Boolean network model of T cell signaling. Each box represents a signaling molecule, which may either be on or off. The transitions are defined through gates as discussed in Figure 3.16. (From Saez-Rodriguez, J., et al. PLoS Comput Biol 3(8) [2007] e163.)

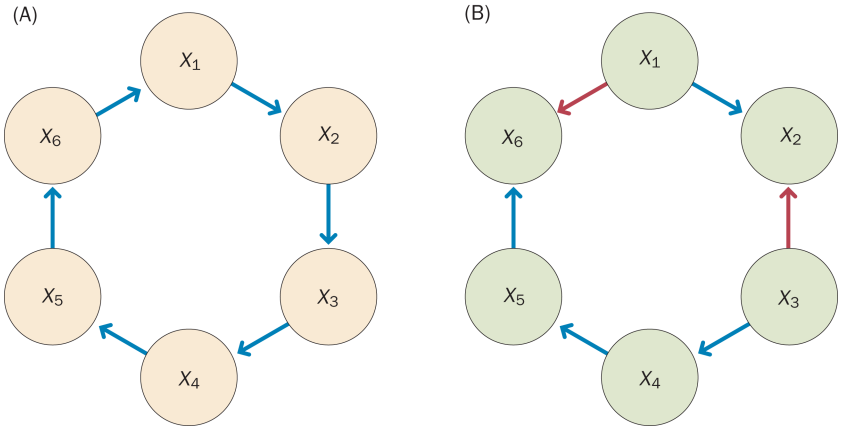


Figure 3.18 Subtle differences in network structure can have significant consequences on the stability of a network. The coherent all-pass-through feedback loop in (A) poses the risk of destabilization, while the anti-coherent network in (B) is stable, with two source nodes (X_1 and X_3), two sink nodes (X_2 and X_6), and two pass-through nodes (X_4 and X_5).

networks exclusively for cycles consisting of between 3 and 15 directional, nondirectional, or bidirectional links. Particularly important among their findings was that biological and engineered networks contain only small numbers of **feedback** loops of the types shown in Figure 3.18. Among them, all-pass-through feedback loops that are coherent in a sense that all arrows point in the same direction pose the risk of destabilizing the system, while anti-coherent cycles that contain sinks and sources are stable. Lipshtat and collaborators showed that feedback loops in signal transduction systems, if set up appropriately, can exhibit bistable behavior, oscillations, and delays [56]. Consistent with the argument of instability in coherent feedback loops, they found that hubs in natural networks are mostly either source or sink nodes but not pass-through nodes. They also observed that networks with fewer nodes involved in feedback loops are more stable than corresponding networks with more nodes involved in feedback loops. They concluded that the prudent placement of feedback loops might be a general design principle of complex biological networks.

Similar to some of the feedback loops are **feedforward** loop motifs that contain a direct and an indirect path (Figure 3.19). Each arrow may be activating or inhibitory, which for the case of three nodes leads to eight combinations with different response patterns [10]. Four of these are coherent, in the sense that the indirect path overall sends the same activating or inhibitory signal as the direct path, while the other four combinations are incoherent, because they contain opposite signals. As an example, the incoherent motif in Figure 3.19C indicates inhibition along the direct path but double inhibition, thus activation, along the indirect path. Both coherent and incoherent patterns have been observed in natural systems, and the motif with all positive signals (Figure 3.19A) is most prevalent among them.

An interesting example of a feedforward loop is a mechanism for sensing stress signals in *E. coli* and many other bacteria [60]. When the bacterium senses the

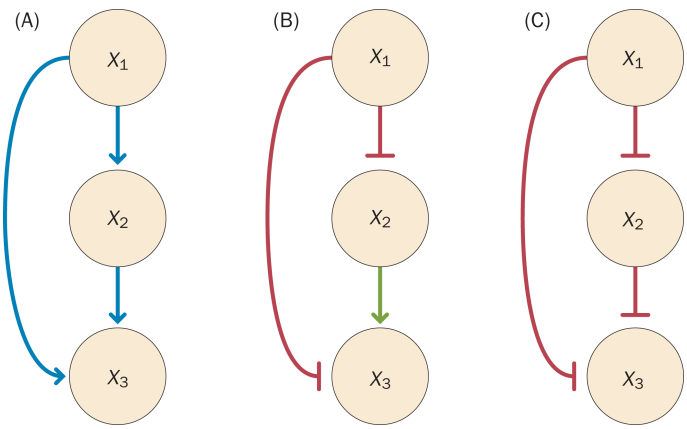


Figure 3.19 Examples of feedforward loop motifs that contain a direct and an indirect path. The motifs in (A) and (B) are coherent because the signals arriving at X_3 are both activating (A) or both inhibiting (B), while the motif in (C) is incoherent: the direct path signals inhibition, while the indirect path signals double inhibition and thus activation.

stress, the feedforward loop triggers the production of flagellum proteins (Chapter 8), which continues for about an hour, even if the signal disappears. This time period is sufficient for the bacterium to assemble the flagellum and swim away. Like other motifs, the feedforward loop in this case creates functional robustness by allowing the system to operate continuously, even if it is exposed to random fluctuations.

3.4 DEPENDENCIES AMONG NETWORK COMPONENTS

3.4.1 Causality Analysis

So far, we have discussed the connectivity among nodes in a network, as well as paths leading from a source to a target. Especially in undirected networks, the connections represent associations. However, they do not necessarily imply causation, which is more difficult to establish but yields greater insight into the functionality of the network.

The search for causes and effects is probably as old as humankind. We know that Aristotle stated that the cause has to precede an effect, while Galileo in his 1638 book *Two New Sciences* postulated that a true cause (*vera causa*) must not only precede the effect but also be both necessary and sufficient. Two hundred years later, John Stuart Mill additionally required of a true cause that other explanations for a cause–effect relationship must be eliminated, and, in the 1920s, the geneticist Sewall Wright formulated systems of linear equations and analytical tools for addressing indirect effects. Wright's studies can be considered the foundation of modern causal analysis (*cf.* [61, 62]). Nowadays, causes and effects are often studied with statistical methods. A prominent approach is Granger causality, which requires that if A causes B, then past values of both A and B should be more informative for predicting B than past information on B alone. Such a situation can be evaluated with linear regression methods applied to stochastic processes. The techniques have recently been extended to nonlinear cases, but the corresponding computations become quite complicated. While a warning is often voiced that correlation does not imply causation, techniques of path analysis, structural equation modeling, and Bayesian statistics permit to some degree the reliable inference of true causes of observed effects [62].

Typical questions are whether effects are direct or indirect and how to address the role of hidden variables, which cannot be measured. In many cases, the methods involve a cause–effect diagram, which consists of a directed graph without cycles, which means that one can never return to the same node. The problem with causal analysis in biology is that realistic systems almost always contain a considerable degree of circularity, which makes most statistical approaches to analyzing or inferring network structures and their causality troublesome. In systems with many cycles, causes may be distributed throughout the whole system, and multiple related or unrelated factors may contribute to the observed effect with different weights. As a consequence of these typical complications, causal analysis has found only limited applications in biology (for a successful exception in a relatively simple case, see [63, 64]), with the exception of Bayesian methods, which are often used for the reconstruction of genetic and signaling networks, as we will see in a later section.

3.4.2 Mutual Information

A simpler task than identifying true causality concerns the mutual dependence of two variables. As an example, suppose we would like to quantify how much more likely it is that some gene of interest is expressed if we know that certain other genes are expressed. It is not difficult to see that such an analysis of the likely co-expression of sets of genes could give important clues regarding the structure of pathways and might shed light on diseases that involve many changes in gene expression [65]. Questions of this nature may be addressed with statistical methods characterizing the mutual information between two or more quantities. These methods are derived from Claude Shannon's and Warren Weaver's ideas concerning the reliability of communication through noisy channels [66], which led to

what is now known as information theory. In this field, one studies how likely it is that one variable X is *on* (present), while another variable Y is either *on* (present) or *off* (absent). If X and Y are perfectly coupled to each other, the two will always be both *on* or both *off*, or it could be that one is always *on* if the other is *off*. However, in most cases, the coupling is not that clear-cut, and one often wonders whether two variables influence each other at all. The mutual information between X and Y characterizes their dependence in cases where the coupling is perfect, strong, loose, or absent. Expressed differently, the mutual information is a measure of the extent to which knowledge regarding one of the two variables reduces uncertainties in the other. If X and Y are totally independent of each other, the mutual information is 0, because, even if we knew everything about X, we would be unable to make reliable predictions regarding Y. By contrast, a strong dependence between X and Y corresponds to a high value of the mutual information criterion. Specific examples of the use of mutual information in molecular systems biology include inferences regarding the functional relevance of gene–gene interactions [67] and the coordination between gene expression and the organization of transcription factors into regulatory motifs [68].

A key concept for quantifying mutual information is the entropy $H(X)$, which measures the uncertainty associated with a random variable X. Here, the term random variable means that X may take different values with certain probabilities; for example, the roll of a die may result in any number between 1 and 6. Depending on the situation, the uncertainty surrounding X may depend on how much we know about another variable, Y. For instance, if two genes G_1 and G_2 are often co-expressed, and if we observe that G_2 is indeed expressed in a certain situation, we are more confident (although still not certain) that G_1 will be expressed as well. The uncertainty of X, given that we know Y, is expressed as the conditional entropy $H(X|Y)$. Furthermore, the uncertainty of a pair of random variables is quantified as the joint entropy $H(X, Y)$, which is defined as the sum of the entropy $H(X)$ and the conditional entropy $H(Y|X)$. Since $H(X, Y)$ is the same as $H(Y, X)$, we obtain the relationship

$$H(X,\,Y)=H(X)+H(Y\,|\,X)=H(Y)+H(X\,|\,Y). \tag{3.18}$$

The mutual information $I(X;\,Y)$ is the sum of the individual entropies minus the joint entropy:

$$I(X;\,Y)=\big[H(X)+H(Y)\big]-H(X,\,Y). \tag{3.19}$$

These relationships can be visualized in a Venn diagram of entropies (**Figure 3.20**). Shannon, Weaver, and others worked out the mathematical formulations of the various entropies, and these can directly be extended to uncertainties and the mutual information among more than two variables. A good introduction to the topic is [66].

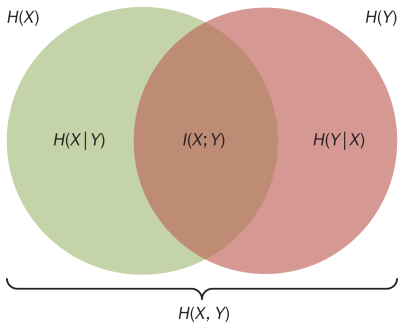


Figure 3.20 Mutual information and entropy. The Venn diagram visualizes the relationships between mutual information $I(X;\,Y)$ and different types of entropy, namely the entropies $H(X)$ and $H(Y)$ associated with the random variables X and Y, the joint entropy $H(X,\,Y)$, and the conditional entropies $H(X|Y)$ and $H(Y|X)$.

3.5 BAYESIAN RECONSTRUCTION OF INTERACTION NETWORKS

While it is relatively easy to characterize graphs using measures such as their degree distribution or clustering coefficients, it is much more difficult to find the connectivity within an unknown graph. This section introduces the basic concepts of **reverse engineering** ill-characterized graphs. The important message of this section is the confirmation that it is indeed possible to infer the structure of unknown static networks with some reliability, at least under favorable conditions. The technical details, which are not always simple and are unfortunately difficult to avoid, should not cloud this message.

It is easy to imagine that one can infer the structure of a network only if one has suitable data—lots of them. It is also clear that such data in reality are essentially always corrupted by noise, which may come from genuine randomness or from inaccuracies during the experimental measurements. For instance, the

components of a signaling system that are active at a particular point in time may depend on all kinds of factors, including environmental perturbations, in-built redundancies, and the recent history of the system. Overlaid on these perturbations are the experimental difficulties of obtaining accurate characterizations of active and inactive states. Given such enormous challenges, is it possible—and valid—to infer an active, causative role of two components A and B in the stimulation of C, if C is usually active when A and B are both turned on? It is not difficult to imagine that the reliability of such an inference grows the more often we observe the same pattern of C being on if both A and B are on, and C being off if A or B or both are off. In other words, it is easy to imagine that network reconstruction tasks require many data that permit the application of statistical tools.

Useful data can come in two forms. In the first case, one has many independent observations of co-occurrences of component activities, as in the case of A, B, and C. In the second case, one may have measurements of many or all components of the network, which were obtained as a sequence of many time points, following some stimulus. For example, one could expose a cell culture to an environmental stress and study the expression profile of a set of genes over many minutes, hours, or days afterwards. In this chapter, we discuss only the former case of many co-occurrences. We will return to the latter case in Chapter 6, which addresses inverse modeling tasks for dynamic systems.

The strategy for attempting to infer the structure of a network from co-occurrence data is of a statistical nature and goes back almost 250 years to Thomas Bayes [69]; it is still called **Bayesian network inference**. If the data are plentiful and reasonably good, the strategy is rather robust and addresses direct and indirect causal effects. However, it also has its limitations, as we will discuss later.

Before we can outline the principles of Bayesian network inference, it is necessary to review the very basics of probabilities. We do this in a somewhat nonchalant, conceptual fashion, but there are many books that treat these ideas in a rigorous mathematical fashion (for example, [70]).

By convention, probabilities are real numbers between 0 and 1 (or 100%), where 0 signifies the impossibility of an event and 1 denotes the certainty of an event. Suppose we have nine new $1 bills and one new $20 bill in our wallet and they are not sorted in any particular way. Because the bills are new and of the same size, weight, and material, we cannot feel differences between them, and, pulling a bill blindly out of the wallet, the probability of a $1 bill is 90% or 0.9, whereas the probability of the $20 bill is 10% or 0.1. The probability of pulling a $5, $100, or $7.50 bill is 0, and the probability of pulling either a $1 or a $20 bill is 1. All this makes intuitive sense and can be proven with mathematical rigor.

Imagine now that we are repeating the experiment many times, always returning the bill to the wallet, and recording every time what dollar amount we pulled. In the previous example, if we pull a bill often enough, we expect to pick the $20 bill roughly every tenth time. Now suppose we do not know how many $1 or $20 bills are among the ten bills are in our wallet. Pulling only one bill does not really answer the question, but if we repeat the experiment many times, the numbers of $1 or $20 pulls begin to give us an increasingly clearer picture. The argument is as follows: If we blindly pull a bill 50 times, and if there is only one $20 bill, we would expect a total of five $20 pulls. It might be four, it might be six, but five would be our best estimate. By contrast, if there were six $20 bills among the ten bills, we would expect a much higher return, namely something like 30 pulls of $20 bills, corresponding to the 60% chance for each pull. Thus, just by observing the outcome of many experiments, we can infer something about the unknown internal structure of the "system." **Bayesian inference** is a sophisticated formalization of such a strategy. We are not 100% sure about our results, because they are probabilistic, but our confidence grows with the number of observations.

Continuing with the example, suppose we are in a game show where we have to choose one of three wallets and then blindly pull one bill out of it. The wallets have different colors, and, for whatever reason, we have a slight preference for gray. To be specific, let's say the probability to select the gray wallet is 40% and the other two probabilities are 30% each. Of course, we don't know what each wallet contains. Suppose the brown wallet contains two $100 bills and eight $1 bills, the gray wallet has six $20 bills and four $1 bills, and the green wallet contains ten $10 bills (**Figure 3.21**).

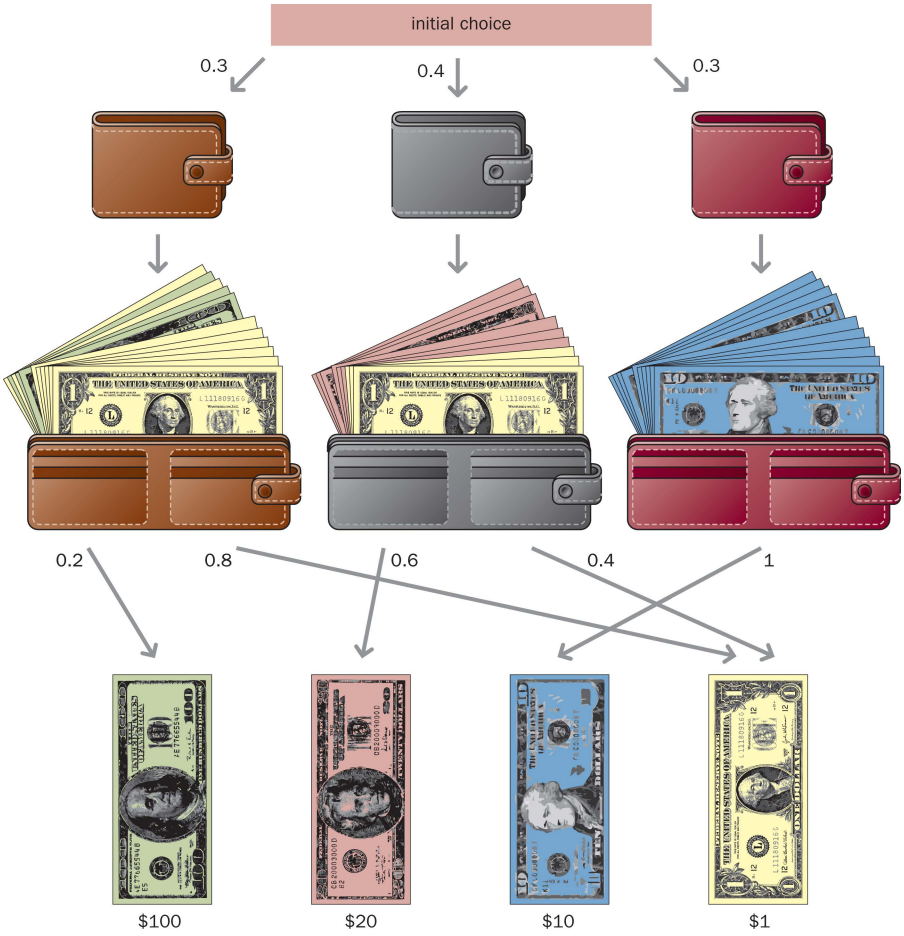


Figure 3.21 Conditional probabilities can be explored intuitively with playing cards, dice, and games. Shown here is the schematic of a fictitious game show, where the contestant selects a wallet and pulls a bill out of it.

Clearly, our personal outcome of the game will ultimately be \$1, \$10, \$20, *or* \$100, but what is the likelihood of each? It is easy to see that the probability of pulling a particular dollar amount out of a wallet depends on which wallet we select in the first place. In statistical terms, the probability is conditional, because it depends (is conditioned) on the selection of a particular wallet. Figure 3.21 illustrates the situation.

In general terms, the **conditional probability** $P(B\,|\,A)$ (that is, the probability of B given A) of an event B, given that event A is true (is occurring or has happened before) is formally given as

$$P(B\,|\,A) = \frac{P(B \text{ and } A)}{P(A)}. \tag{3.20}$$

Here $P(B \text{ and } A)$ denotes the (joint) probability that both A and B are simultaneously true, and it is tacitly assumed that the probability of A, $P(A)$, is not zero. Because $P(B \text{ and } A)$ is the same as $P(A \text{ and } B)$, we can formally rearrange the terms and write a sequence of equations as

$$P(B|A)P(A) = P(B \text{ and } A) = P(A \text{ and } B) = P(A|B)P(B). \tag{3.21}$$

The equivalence of the outer expressions here is known as Bayes' theorem, Bayes' formula, or the Bayes rule, and is typically written as

$$P(A|B) = \frac{P(A \text{ and } B)}{P(B)} = \frac{P(B|A)P(A)}{P(B)}. \tag{3.22}$$

The formula says that we can make a statement of the (unknown) occurrence of event A given an (observed) outcome B, assuming that $P(B)$ is not zero.

It is not difficult to make various forward computations with the formulae for conditional probabilities, for instance, for our game show example. Because the choice of the wallet and the pulling of a bill from it are independent events, their probabilities simply multiply. Thus, the probability of choosing the brown wallet and securing the grand prize of \$100 is $0.3 \times 0.2 = 0.06$, which corresponds to a measly 6% probability. The probability of choosing the gray wallet and ending up with \$100 is nil. In this fashion, we can exhaust all possibilities and, for instance, compute what the probability is of ending up with \$20. Namely, the probability of \$20 from the brown or red wallets is zero. The gray wallet is chosen with probability 0.4 and choosing a \$20 bill among the bills is 0.6, so that the overall probability is 0.24. To compute the probability of the meager outcome of \$1, we need to consider the brown and the gray wallets, and the probability is $0.3 \times 0.8 + 0.4 \times 0.4 = 0.4$. Finally, the probability of an overall outcome of \$10 is 0.3 or 30%. As should be expected, the probabilities for \$1, \$10, \$20, and \$100 sum up to 1.

Another question we can answer quite easily is the following: if we knew the contents of the three wallets, would the choice of the gray wallet be optimal? The answer lies in the **expected value**, which we may imagine as the average over very many runs of the game. The easiest case for computing the expected values is the red wallet, which, when chosen, will always reward us with \$10. For the brown wallet, the expected value is \$20.80, because in 100 runs of the game we would expect to pull about 20 \$100s and 80 \$1s. The total dollar amount of these is \$2080 which corresponds to an average of \$2080, per run (even though we can never actually get \$2080 in any given run!). For the gray wallet, the analogous computation yields $(60 \times 20 + 40 \times \$1)/100$, which is only \$12.40. Gray might be a nice color, but it is a bad choice in this game!

All these types of computations are straightforward because they follow well-established probability theory and, in particular, the law of total probabilities. This law says that the probability of an event B that is conditioned on distinct events A_1, A_2, . . . , An can be computed as

$$P(B) = \sum_{i=1}^{n} P(B \mid A_i) P(A_i). \tag{3.23}$$

We can see that this result is directly related to Bayes' formula, if we consider all possible and distinct events A_i and compute all conditional probabilities $P(B \mid A_i)$. The probability of event B is then the sum of all these $P(B \mid A_i)$, given that A_i actually happens, which it does with probability $P(A_i)$. A direct example is the preceding computation of the least desirable outcome of \$1 in the game show example. Each $P(A_i)$ is called a prior probability, and $P(B \mid A_i)$ is again the conditional probability of B, given A_i.

3.5.1 Application to Signaling Networks

Many of the concepts of Bayesian analysis are applicable to genomic, proteomic, signaling, and other networks. Here we use signaling as the prime example for illustration, but it should be clear that gene and protein networks can be analyzed in a similar fashion.

Signal transduction is a key element for the proper function of cells and organisms. Simplistically speaking, a cell receives a chemical, electrical, mechanical, or other signal at the outside of its membrane. This signal is internalized in some fashion, for instance with the help of G-protein-coupled receptors, and typically triggers an amplification mechanism that is often referred to as a signaling cascade. Ultimately, the received and amplified signal leads to some cellular response, such as the expression of appropriate genes. Fundamental aspects of these fascinating systems are discussed in Chapter 10.

For an illustration here, let us apply our insights regarding conditional probabilities to a signaling network like the one in **Figure 3.22**. Suppose two receptors, R and S, are anchored in the cell membrane and respond independently to appropriate ligands. The response to the binding of L consists of sending a signal to a signaling cascade C, whose details are unimportant here, but which ultimately triggers some genomic response G. For the cascade to be activated, both R and S

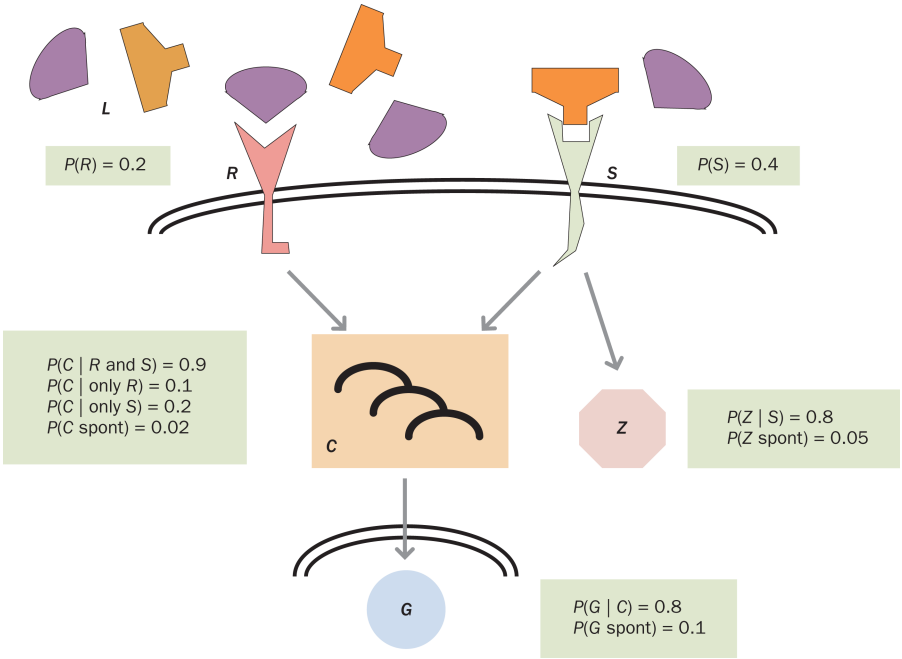


Figure 3.22 A generic, simplified signaling system responds to the binding of two different ligands. The expected numbers of correct or faulty responses can be computed with conditional probabilities. See the text for details.

should fire. Furthermore, if a ligand binds only to S, the system triggers a different response Z. In reality, the interactions are corrupted by noise, such as the binding of nonspecific ligands, so that the cascade may sometimes be triggered even if only R or S is active, or it may not be triggered, even if both are occupied by ligands. The terminology of noise is commonly used, even though signaling cascades don't make audible sounds.

The connectivity of the system may be sketched out in a diagram, as shown in Figure 3.22, which also shows probability tables of various events. For instance, the probability that receptor R is active (occupied by a specific ligand L) is 0.2. The probability that cascade C is triggered if both R and S are occupied by ligands, is $P(C \mid R \text{ and } S) = 0.9$, whereas spontaneous activation of the cascade has a very low probability of $P(C \text{ spont}) = 0.02$. The probability tables allow us to compute all kinds of probabilities, such as the probability that a ligand binds to S and triggers Z. Or we might compute the probability that (1) ligands bind to R and S; (2) the cascade is activated; and (3) the result is a genomic response G. This latter scenario is modeled as

$$\begin{aligned} P(G, C, R, S \text{ are "on"}) &= P(G|C)P(C|R \text{ and } S)\, P(R)P(S) \\ &= 0.8 \times 0.9 \times 0.2 \times 0.4 = 0.0576 \end{aligned} \tag{3.24}$$

The roughly 6% probability of the response may seem low, but it is mainly dictated by the relatively low binding probabilities of ligands of R and S, which both must happen for a normal response.

A typical situation for Bayesian inference is the following. If we observe that gene G is expressed, can we make inferences on whether the cascade is activated and/or whether R and/or S are present? Knowing the probability tables, we can infer with high probability that C is active, because the probability that G is on, given that C is active, is high (0.8), and spontaneous gene expression is not overly likely (0.1). It is not difficult to imagine that the inferences become better and better as more observations on the on–off states of the nodes in the graph become available.

Bayesian inference really becomes interesting if we do not know the probability tables but must deduce them from observations on the states of the system. Not surprisingly, the execution of such an inference is rather complicated and requires sophisticated computer algorithms. Nonetheless, it is useful to understand the underlying conceptual ideas. In general, a Bayesian network is formulated as a **directed acyclic graph** (**DAG**), which means that its nodes Xi are connected by

unidirectional arrows heading and that there is no possibility of returning to any of the nodes, once one leaves this node through any of its outgoing arrows; Figure 3.22 is an example. The parents of a node X_i, denoted by $\mathrm{Par}_j(X_i)$, are defined as those nodes that directly send a signal to X_i through a single connecting arrow. The state of a parent may again be *on* or *off*, 1 or 0, or take other values, just as X_i. If the graph consists of n nodes, then the probability P that the state of the system has particular values (x_1, x_2, . . . , xn) can be computed, with a dependence on the state of the parents, as

$$P(X_1 = x_1 \wedge X_2 = x_2 \wedge \ldots \wedge X_n = x_n) = \prod_{i=1}^{n} P\left(X_i = x_i \,|\, \left\langle \mathrm{Par}_j(X_i) = x_j \right\rangle\right). \tag{3.25}$$

Here, the notation $\wedge$ is the logical "and." Thus, the expression on the left-hand side formalizes the probability P that all the nodes X_i simultaneously have the particular values x_i. The notation <$\mathrm{Par}_j(X_i) = x_j$> on the right-hand side refers to the set of all parents P_j of X_i and their particular states x_j. The probabilities in the product are therefore probabilities that X_i has the value x_i, conditioned on the states of all its parents. The states of the parents can in turn be expressed in terms of their own parents, and so on, as we exemplified above. This stepwise dependence is possible because the graph is acyclic; if it had loops, we would be in trouble. With this formalization, we can compute any conditional probability in the system. These computations are often very tedious, but they are straightforward and can be executed with customized software.

The computation of conditional probabilities requires that we know (a) the connectivity of the network and (b) the probability tables at each node, as it was the case in Figure 3.22. For many actual signal transduction networks in biology, we may have a rough idea about their connectivity, but we usually do not know the probabilities. Sometimes, we have information on whether the probabilities are high or rather low, but we seldom know precise numerical values. That's where Bayesian inference comes in. Instead of being able to compute numerically forward, as we did so far, one needs a large dataset of observations consisting of *on* or *off* states of all or at least many of the nodes. The focus of the analysis is now on correlations of *on* or *off* values among the nodes. For instance, two observations regarding the signal transduction system in Figure 3.22 could be

$$(R = \mathrm{off};\ S = \mathrm{on};\ C = \mathrm{off};\ G = \mathrm{off};\ Z = \mathrm{on})$$

and

$$(R = \mathrm{on};\ S = \mathrm{on};\ C = \mathrm{on};\ G = \mathrm{on};\ Z = \mathrm{off}).$$

If C and G are strongly correlated, as they are in the above example, many observations will have them either both *on* or both *off*. By contrast, such a correlation would not be expected between R and Z.

The inference strategy begins with composing a network graph that contains all known, alleged, or hypothesized connections, the direction of causality between any two nodes, and, if possible, likely ranges of probabilities. All probabilities are represented by symbolic parameters p_1, p_2 ,. . . , p_m rather than numerical values, and the goal is to determine the unknown values for these parameters in such a manner that the parameterized model reflects all (or at least most) observed combinations of states as accurately as possible. Needless to say, figuring out such a well-matching configuration of many parameters constitutes an optimization task that is anything but trivial, especially if there are many nodes. Several methods have been developed for this purpose in recent times, including Markov chain Monte Carlo (MCMC) algorithms such as the Gibbs sampler and the Metropolis–Hastings method, and they are all quite involved [8]. The basic concept of these optimization methods consists of randomly choosing probabilities (first blindly or within known or assumed ranges, and later within shrinking ranges that are determined by the algorithm) and computing the corresponding joint distributions of

all nodes (recall the Monte Carlo simulations in Chapter 2). This forward computation with putative probabilities is relatively simple and executed thousands of times. A statistical comparison between the computed joint distributions and the observed distributions is then used to determine probability values that slowly improve during the algorithmic iterations and, if successful, ultimately yield a constellation of probability values with which the model matches the observations best. For further details, the interested reader may consult [8] and [71]. The same type of analysis is sometimes used to decide which set of network connections among several candidates is most likely, given observations on input and output nodes.

One theoretical and practical limitation of the Bayesian inference strategy is that it does not permit graphs that contain any types of loops. For instance, chains of arrows starting and ending at the same node, as well as feedback signals, are not permitted. Furthermore, graphs or probability tables that change over time make inferences much more complicated. Current research efforts are focusing on these challenges.

It may seem that Bayesian inferences of actual signal transduction networks might run into insurmountable problems. However, several success stories have been documented. A premier example is the identification of a signaling network in $CD4^+$ T cells of the human immune system [72]. The data consisted of thousands of single-cell flow-cytometric measurements of multiply phosphorylated protein and phospholipid components in individual cells that followed external molecular perturbations. Specifically, the data consisted of knowledge of the molecular stimulus (inhibition or stimulation of a specific component) and the corresponding state or condition of each of 11 phosphorylated molecules within a given cell, 15 minutes after the stimulus. The Bayesian inference successfully confirmed known relationships between some of the signaling components and revealed additional, previously unknown, causal relationships. Not surprising, the analysis missed three causal relationships that corresponded to known loops in the signal transduction network.

3.5.2 Applications to Other Biological Networks

Outside of signaling systems, graph methods and Bayesian approaches have been used to reconstruct protein–protein interaction networks, where the main purpose was to identify which proteins are interacting with each other, and which of these interactions are functionally meaningful [73, 74]. As just one example, Rhodes and collaborators [75] proposed a model of the human protein–protein interaction network that integrated information from protein domain data with gene expression and functional annotation data. Using Bayesian methods, they constructed and partially validated a network consisting of nearly 40,000 protein–protein interactions. Beyond the presence or absence of particular interactions, one is also often interested in the structure of these networks and the features of the underlying graphs, such as types of hubs, cliques, and small-world or random structure, as we discussed earlier in this chapter. Since large-scale protein–protein interaction networks often contain hundreds of nodes, their visualization is a great challenge [76].

Bayesian methods have also been applied to gene regulatory networks and the control of transcription [77, 78]. In genomic networks, one asks which genes are functionally co-regulated and whether the expression of one set of genes might depend on the expression of another set of genes. Answers can be found in microarray data from perturbation experiments, from the interactions between transcription factors and promoters, or from time series data, in which gene expression is measured multiple times following a stimulus (see, for example, [79, 80]). A sophisticated example is the analysis of genetic variations among individuals and their influence on the occurrence and severity of complex diseases such as obesity, diabetes, and cancer [81].

Bayesian inference methods are seldom applied to metabolic networks. The main reason is that the interactions in these networks are strictly dictated by biochemistry so that it is usually more or less clear how these networks are connected. Because of this strong biochemical basis, metabolic networks allow a whole class of other analyses, which we will discuss in Chapter 9 and straddle the fence between static networks and dynamic systems.

3.6 EXERCISES

3.1 Try to figure out Euler's problem of the Seven Bridges of Königsberg (see Figure 3.3): Is there a path that uses every bridge exactly once? Does the solution change if any one of the bridges is removed? If so, does it matter which bridge is removed?

3.2 Are there connected paths in the graphs in Figure 3.23 that use every edge exactly once? Does it matter where one starts? Given a successful starting point, does the direction matter?

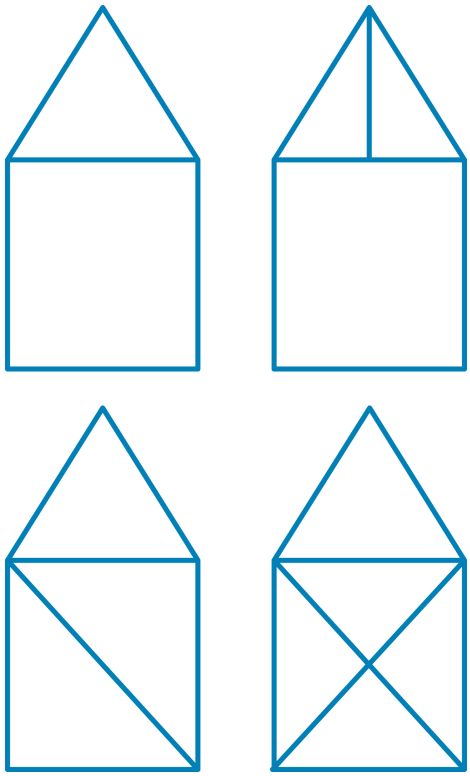


Figure 3.23 Simple graphs with different connectivities. Is it possible to find paths that use every edge exactly once?

3.3 Consider the directed graph in Figure 3.24.

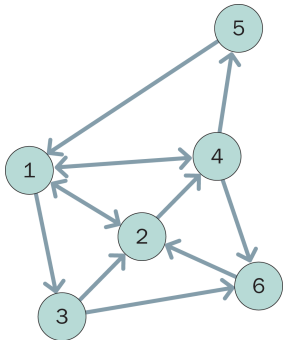


Figure 3.24 Directed network with six nodes. The graph is to be analyzed in Exercise 3.3.

3.3.1 Construct the adjacency matrix.

3.3.2 Identify the in- and out-degree of each node.

3.3.3 Find the clustering coefficient for each node.

3.3.4 Determine the clustering coefficient for the graph.

3.3.5 Construct a matrix (s_{ij}) with each entry being the length of the shortest path between nodes i and j.

3.3.6 What is the diameter of the graph? How does is relate to the matrix in 3.3.5?

3.3.7 Using the adjacency matrix, find the matrix (m_{ij}) with each entry being the number of 3-step paths that connect the nodes i and j.

3.4 Construct ten undirected random graphs with m nodes and a total of $m{\cdot}k$ edges, using different combinations of m and k. As a simple start, study a graph with six nodes and determine random edges with two dice, which identify the nodes to be connected. Record the degree distributions. As a more efficient alternative, write a computer algorithm and compute the degree distribution for each graph.

3.5 Construct ten directed graphs with five nodes that in each case form a single loop, such as $N_1 \rightarrow N_5 \rightarrow N_3 \rightarrow N_4 \rightarrow N_2 \rightarrow N_1$ and contain no other edges. How many different graphs of this type are there? Study similarities in their adjacency matrices and summarize them in a brief report.

3.6 How are hubs reflected in the adjacency matrix of an undirected network?

3.7 How are cliques reflected in the adjacency matrix of an undirected network?

3.8 What happens to the adjacency matrix of a graph if this graph consists of two subgraphs that are not connected to each other? Can you make a general statement? Does this statement cover both directed and undirected graphs?

3.9 Using the method in Box 3.1, compute all three-step paths of the matrix on the left of Figure 3.5.

3.10 Confirm that in an undirected graph the number of edges between all neighbors of a node is at most $k(k-1)/2$.

3.11 Compute the clustering coefficients of the red nodes in the undirected graph of Figure 3.25.

3.12 Consider the following undirected graph in Figure 3.26.

3.12.1 Create the degree distribution table (degree | frequency) for the graph.

3.12.2 Plot the degree distribution histogram (frequency vs. degree).

3.12.3 Plot the degree distribution in the log-log scale. Does the degree distribution seem to have power-law dependence? Briefly explain your answer.

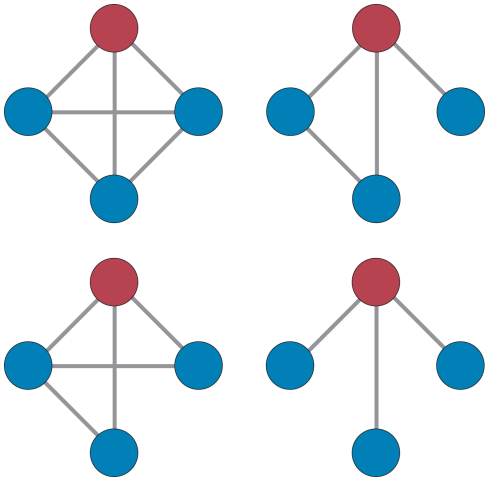


Figure 3.25 Graphs with four nodes. The connectivity patterns determine their clustering coefficients.

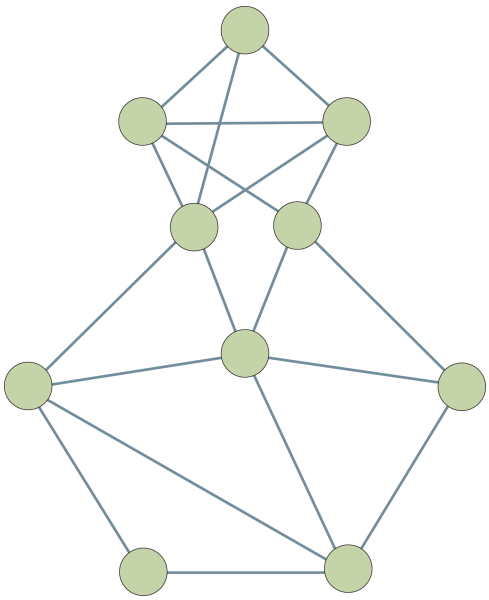


Figure 3.26 Example of an undirected graph. The graph is to be used for assessments of connectivity and other features.

3.13 What are the minimum and maximum values of C_N in a network without multiple edges?

3.14 What is the average clustering coefficient of a clique?

3.15 Provide rationale for the two different definitions of C_N for directed and undirected graphs.

3.16 Construct five random graphs with six nodes and 12 directed edges. Write a computer program or use two dice to determine source and end nodes; do not allow double edges. Compute for each case all shortest path lengths and the average shortest distance.

3.17 Search the Internet for actual biological networks with documented power-law degree distributions. Study γ and discuss what happens to the power-law relationship for very high and very low k.

3.18 Search the Internet for two studies that discuss party hubs and date hubs. Write a short report about the goals, methods, and results of each study.

3.19 Box 3.2 discusses the construction of different connectivity pattern from an initial ring-like structure. Execute this construction yourself with different probabilities p ($0 \le p \le 1$). Specifically, study a ring with 12 nodes, which is connected as shown in Box 3.2, and perform the analysis with $p = 1/6, 2/6, \ldots, 5/6$. Start with the first node and go around the ring in one direction. For each node, roll a die to determine whether one of its edges should be replaced or not. For instance, for $p = 2/6$, replace an edge only if you roll a 1 or 2. If an edge is to be replaced, use a coin to determine whether to remove the node's connection to its immediate right or left neighbor. Use a coin and a die to determine the node to which a new edge is to be connected (head: nodes between one and six; tail: nodes between seven and 12). In cases where you identify the source node itself or where the edge already exists, roll again. After one lap, repeat the procedure, this time using second-nearest neighbors. Once you have finished the second lap, report the degree distribution and the network's clustering coefficient. As a more efficient alternative, write an algorithm that implements the above procedure with m nodes and a probability $p (0 \le p \le 1)$.

3.20 Determine the degree distribution of the Watts-Strogatz model in Box 3.2. for $p = 0$.

3.21 Discuss the tolerance of the graphs in Box 3.2 to random attacks and to targeted attacks, where one node is eliminated in each case.

3.22 Compute the probability that the cascade in Figure 3.22 is active even though R and S are not active. Interpret $P(C \text{ spont})$ as the probability that C is on, although neither R nor S is active. Compute the probabilities that R or S is not active as $1 - P(R)$ and $1 - P(S)$, respectively.

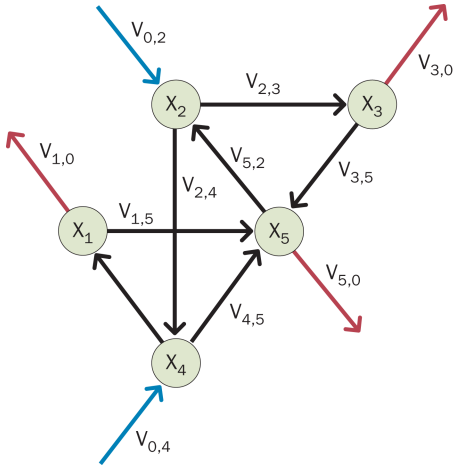
3.23 Compute the probability that there is a genomic response in the system in Figure 3.22 even though R and S are not active. Interpret $P(G \text{ spont})$ as the probability that G is on, although C is not active.

3.24 Search the Internet for an investigation of a protein–protein interaction network. Write a short report about the goals, methods, and results of the study. Pay particular attention to the visualization of the network.

3.25 Search the Internet for an investigation of a gene interaction network. Write a short report about the goals, methods, and results of the study.

3.26 Search the Internet for an example of a stoichiometric or flux balance analysis. Write a short report about the goals, methods, and results of the study.

3.27 Consider the metabolic network in **Figure 3.27**, which consists of five metabolites and 12 fluxes



metabolite	number of carbon atoms
1	2
2	2
3	1
4	2
5	3

Figure 3.27 Metabolic Network with five metabolites. The table on the right shows the numbers of carbon atoms of each metabolite.

corresponding to seven metabolic reactions (gray), two input processes (blue), and three output processes (red). The carbon composition of each metabolite is given in the table insert.

Arrange all fluxes in the flux vector:

$$\vec{V} = \left(V_{1,5}, V_{1,0}, V_{2,3}, V_{2,4}, V_{3,5}, V_{3,0}, V_{4,1}, V_{4,5}, V_{5,2}, V_{5,0}, V_{0,2}, V_{0.4}\right)^{tr}$$

3.27.1 List all metabolic reactions making sure that the number of carbons is conserved. For example: $V_{1,5}: 3X_1 \rightarrow 2X_5$, because 3·(2 carbons) = 2·(3 carbons).

3.27.2 Construct the stoichiometric matrix, N. Make sure that the order of columns is consistent with the flux vector given above, and the order of rows is consistent with the numerical order of the metabolites.

3.27.3 At the steady state, your stoichiometric matrix must satisfy flux balance according to the matrix equation $\mathbf{N}\cdot\vec{V} = 0$. This linear system has 5 equations with 12 unknowns. Let's assume that 7 of the fluxes are actually known:

$$V_{1,5} = 2, V_{2,3} = 3, V_{3,5} = 1, V_{4,1} = 8,$$
$$V_{5,2} = 2, V_{0,2} = 2, V_{0,4} = 6$$

Use this information to determine all remaining fluxes.

3.27.4 Let's assume that the values of incoming and outgoing fluxes represent the change in the number of metabolite molecules, say per second. For example, the flux balance equation for the metabolite 1 (the first equation in $\mathbf{N}\cdot\vec{V} = 0$) should be. From 1.3 you know that $-3V_{1,5} - V_{1,0} + V_{4,1} = 8$, which means that eight molecules of metabolite 1 are incoming, while six and two molecules are outgoing. Furthermore, in terms of carbons, you see that the pool of metabolite 1 receives 8 × 2 = 16 and loses 6 × 2 = 12 and 2 × 2 = 4 carbons per second. Continue this line of thinking, and determine the number of carbons involved in each process of the metabolic network. Indicate your result schematically on the graph.

3.28 Set up a stoichiometric model for the pentose pathway shown in **Figure 3.28**. This pathway exchanges carbohydrates with different numbers of carbon atoms, as shown in **Table 3.1**. Thus, two units of X_1 are needed to produce one unit of X_2 and X_3, and each reaction $V_{5,3}$ uses one unit of X_5 to generate two units of X_3. Note that X_3 appears in two locations but represents the same pool.

3.29 For the same pathway shown in Figure 3.28, assume that the value for the input flux $V_{7,1}$ is 20 units and try different values for $V_{3,0}$. Compute the corresponding flux distributions. How much CO_2 is produced?

3.30 If the influx in the pathway systems in Figure 3.28 were doubled, would doubling all other fluxes in the system lead to a valid solution? Argue from a mathematical and a biological point of view.

3.31 Consider a metabolic system described by the stoichiometric matrix *N* in Table 3.2. Draw a diagram of the metabolic system. Some of the species of the system are composed of three carbons ("trioses"), while others are composed of six carbons ("hexoses"). Which of the species of the system are composed of three carbons and which of six carbons?

3.32 Formulate the rules in (3.17) in a different manner.

3.33 Formulate rules, corresponding to (3.17), for nodes with fewer than eight neighbors (at the top, bottom, and sides of the grid).

3.34 Create a 5 × 5 grid with 25 nodes. Starting with an arbitrary combination of *on*/*off* states, we established a simple rule producing a blinking dynamics in which every state switched from *on* to *off* or from *off* to *on* at every transition. Is this rule unique or can other rule sets produce blinking? Is it possible for each state to stay *on* (*off*) for two time units, then switch to *off* (*on*) and stay *off* for two time

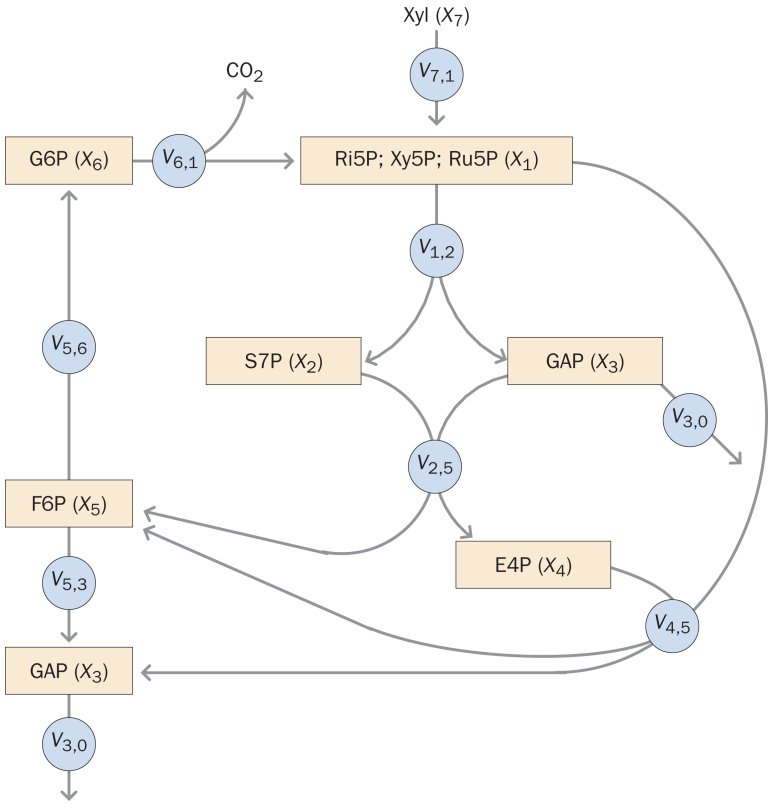


Figure 3.28 Diagram of the pentose phosphate pathway. Given influxes and effluxes, internal fluxes can be computed. Table 3.1 shows the number of carbon atoms in each metabolite. (From Wiechert W & de Graaf AA. *Biotechnol Bioeng.* 55 [1997] 101–117. With permission from John Wiley & Sons.)

TABLE 3.1: NUMBER OF CARBON ATOMS IN EACH METABOLITE IN THE PATHWAY IN FIGURE 3.28

Compound Pool	*Number of Carbon Atoms*
X_1	5
X_2	7
X_3	3
X_4	4
X_5	6
X_6	6
X_7	5

units, and then to switch back and forth every two time units? Explain your answer.

3.35 For a 5 × 5 grid with 25 nodes and initially arbitrary *on*/*off* states, create rules such that eventually all states are zero except for the center state, which should be constantly on. Are the rules unique?

3.36 For a 5 × 5 grid with 25 nodes, define rules and initial conditions such that an *on*-column moves from left to right, then wraps around by starting on the left again.

3.37 Establish rules that make an *on*-diagonal move in southwest direction and then start again in the northeast corner for the next diagonal wave.

3.38 For a 5 × 5 grid with 25 nodes and initially arbitrary *on*/*off* states, is it possible to create rules such that no *on*/*off* pattern is ever repeated? Either create such rules or prove that that is not possible.

3.39 Suppose each node may have one of three states, such as red, blue, or green. Construct a blinking pattern, in which each state cycles through the three colors.

3.40 Search the literature or internet for at least three biological examples that were modeled with Boolean methods. Write a brief summary report.

TABLE 3.2: STOICHIOMETRIC MATRIX N ASSOCIATED WITH EXERCISE 3.1

	F_1	F_2	F_3	F_4	F_5	F_6	F_7	F_8	F_9	F_{10}	F_{11}	F_{12}
A	—1	0	1	0	0	1	0	2	0	0	0	0
B	0	—1	0	0	0	0	0	0	1	0	0	1
C	0	2	—1	—2	—1	0	0	0	0	0	0	0
D	0	0	0	0	0	—1	—1	0	0	2	0	0
E	0	0	0	1	0	0	0	—1	—1	—1	1	0

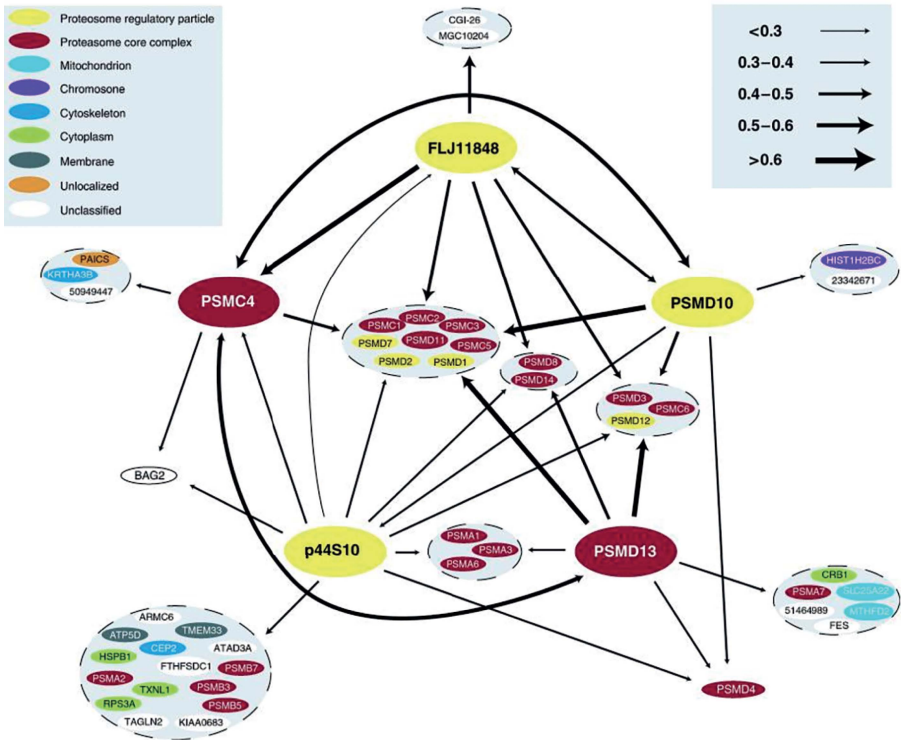


Figure 3.29 Protein-protein interaction network of the human proteasome. The graph is used in Exercise 3.45 to assess connectivity features. From [81].

3.41 How could one set up a Boolean network model that remembers the two previous states? Sketch out a plan and implement a small network.

3.42 Consider the protein-protein interaction network in Figure 3.29.

3.42.1 Which bait protein has the highest clustering coefficient? Consider the directionality of the interactions.

3.42.2 What is the clustering coefficient of the entire network?

3.42.3 What is the degree distribution of the network?

3.42.4 How many paths of length 3 exist from PSMD10 to the KRTHA3B module?

FURTHER READING

Alon U. *An Introduction to Systems Biology: Design Principles of Biological Circuits*. Chapman & Hall, CRC Press, Boca Raton, FL, 2006.

Barabási A-L. *Linked: The New Science of Networks*. Perseus, Cambridge, MA, 2002.

Davidson EH. *The Regulatory Genome: Gene Regulatory Networks in Development and Evolution*. Academic Press, Burlington, MA, 2006.

Menczer F, Fortunato S & Davis CA. *A First Course in Network Science*. Cambridge University Press, Cambridge, UK, 2020.

Newman M. *Networks*, 2nd ed. Oxford University Press, Oxford, UK, 2018.

Raman K. *An Introduction to Computational Systems Biology: Systems-level Modelling of Cellular Networks*. CRC Press, Taylor & Francis Group, New York, NY, 2021.

Zhang A. *Protein Interaction Networks: Computational Analysis*. Cambridge University Press, Cambridge, UK, 2009.

REFERENCES

1. Cytoscape, http://www.cytoscape.org/.
2. Shannon, P., et al., Cytoscape: A software environment for integrated models of biomolecular interaction networks. *Genome Res*, 2003. **13**(11): p. 2498–2504.
3. Ingenuity, http://www.ingenuity.com/index.html.
4. BiologicalNetworks, http://biologicalnetworks.net/index.php.
5. Mrvar, A. and V. Batagelji, Analysis and visualization of large networks with program package Pajek. *Complex Adapt Syst Model*, 2016. **4**: p. 6.
6. Purvis, J.E., R. Radhakrishnan, and S.L. Diamond, Steady-state kinetic modeling constrains cellular resting states and dynamic behavior. *PLoS Comput Biol*, 2009. **5**(3): p. e1000298.

7. Lee, Y. and E.O. Voit, Mathematical modeling of monolignol biosynthesis in *Populus* xylem. *Math Biosci*, 2010. **228**(1): p. 78–89.
8. Wilkinson, D.J., *Stochastic Modelling for Systems Biology*. *Mathematical and Computational Biology Series*. 2006, Boca Raton: CRC Press, Taylor and Francis.
9. Keener, J. and J. Sneyd, *Mathematical Physiology*. *Interdisciplinary Applied Mathematics*. 1998, New York: Springer-Verlag, Inc.
10. Alon, U., *An Introduction to Systems Biology: Design Principles of Biological Circuits*. *Mathematical & Computational Biology*. 2006, Boca Raton, FL: Chapman & Hall/CRC.
11. Voit, E.O., *Computational Analysis of Biochemical Systems: A Practical Guide for Biochemists and Molecular Biologists*. 2000, New York: Cambridge University Press, p. xii, 531.
12. Chaouiya, C., Petri net modelling of biological networks. *Brief Bioinform*, 2007. **8**(4): p. 210–219.
13. Hardy, S. and P.N. Robillard, Modeling and simulation of molecular biology systems using Petri nets: Modeling goals of various approaches. *J Bioinform Comput Biol*, 2004. **2**(4): p. 619–637.
14. Watts, D.J. and S.H. Strogatz, Collective dynamics of 'small-world' networks. *Nature*, 1998. **393**(6684): p. 440–442.
15. Ravasz, E., et al., Hierarchical organization of modularity in metabolic networks. *Science*, 2002. **297**: p. 1551–1555.
16. Jeong, H., et al., Lethality and centrality in protein networks. *Nature*, 2001. **411**(6833): p. 41–42.
17. Maslov, S. and K. Sneppen, Specificity and stability in topology of protein networks. *Science*, 2002. **296**: p. 910–913.
18. Erdős, P. and A. Rényi, On random graphs. *Publicationes Mathematicae*, 1959. **6**: p. 290–297.
19. Barabási, A.-L. and Z.N. Oltvai, Network biology: Understanding the cell's functional organization. *Nat Rev Genet*, 2004. **5**: p. 101–113.
20. Wagner, A. and D.A. Fell, The small world inside large metabolic networks. *Proc Roy Soc London B*, 2001. **268**: p. 1803–1810.
21. Chung, F. and L. Lu, The average distance in random graphs with given expected degrees. *Proc Natl Acad Sci USA*, 2002. **99**(25): p. 15879–15882.
22. Jeong, H., et al., The large-scale organization of metabolic networks. *Nature*, 2000. **407**(6804): p. 651–654.
23. Jordán, F. and I. Molnár, Reliable flows and preferred patterns in food webs. *Evol Ecol Res*, 1999. **1**: p. 591–609.
24. Lima-Mendez, G. and J. van Helden, The powerful law of the power law and other myths in network biology. *Mol Biosyst*, 2009. **5**(12): p. 1482–1493.
25. Nacher, J.C. and T. Akutsu, Recent progress on the analysis of power-law features in complex cellular networks. *Cell Biochem Biophys*, 2007. **49**(1): p. 37–47.
26. Barabási, A.-L., *Linked: The New Science of Networks*. 2002, Cambridge, MA: Perseus.
27. Newman, M.E. and D.J. Watts, Scaling and percolation in the small-world network model. *Phys Rev E Stat Phys Plasmas Fluids Relat Interdiscip Topics*, 1999. **60**(6 Pt B): p. 7332–7342.
28. Albert, R. and A.-L. Barabási, The statistical mechanics of complex networks. *Rev Mod Phys*, 2002. **74**: p. 47–97.
29. Fraser, H.B., A.E. Hirsh, L.M. Steinmetz, et al., Evolutionary rate in the protein interaction network. *Science*, 2002. **296**: p. 750–752.
30. Barabasi, A.L., *Network Science*. 2013, Cambridge, UK: Cambridge University Press.
31. Carlson, J.M. and J. Doyle, Complexity and robustness. *Proc Natl Acad Sci USA*, 2002. **99**(Suppl. 1): p. 2538–2545.
32. Csete, M.E. and J.C. Doyle, Reverse engineering of biological complexity. *Science*, 2002. **295**(5560): p. 1664–1669.
33. Doyle, J.C., et al., The "robust yet fragile" nature of the internet. *Proc Natl Acad Sci USA*, 2005. **102**(41): p. 14497–14502.
34. Cohen, J.E., Graph theoretic models of food webs. *Rocky Mountain J Math*, 1979. **9**(1): p. 29–30.
35. Pimm, S.L., *Food Webs*. 2002, Chicago: University of Chicago Press.
36. de Lichtenberg, U., et al., Dynamic complex formation during the yeast cell cycle. *Science*, 2005. **307**(5710): p. 724–727.
37. Han, J.D., et al., Evidence for dynamically organized modularity in the yeast protein-protein interaction network. *Nature*, 2004. **430**(6995): p. 88–93.
38. Agarwal, S., et al., Revisiting date and party hubs: Novel approaches to role assignment in protein interaction networks. *PLoS Comput Biol*, 2010. **6**(6): p. e1000817. doi:10.1371/journal.pcbi.1000817.
39. Reed, J.L., et al., An expanded genome-scale model of *Escherichia coli* K-12 (iJR904 GSM/GPR). *Genome Biol*, 2003. **4**(9): p. R54.
40. Schryer, D.W., M. Vendelin, and P. Peterson, Symbolic flux analysis for genome-scale metabolic networks. *BMC Syst Biol* 2011. **5**: p. 81, doi:10.1186/1752-0509-5-81.
41. Wagner, A., Evolutionary constraints permeate large metabolic networks. *BMC Evol Biol*, 2009. **9**: p. 231, doi:10.1186/1471-2148-9-231.
42. Palsson, B.Ø., *Systems Biology: Properties of Reconstructed Networks*. 2006, New York: Cambridge University Press.
43. Stephanopoulos, G.N., A.A. Aristidou, and J. Nielsen, *Metabolic Engineering. Principles and Methodologies*. 1998, San Diego, CA: Academic Press.
44. Torres, N.V. and E.O. Voit, *Pathway Analysis and Optimization in Metabolic Engineering*. 2002, Cambridge, UK: Cambridge University Press, p. 305.
45. Clarke, B.L., Complete set of steady states for the general stoichiometric dynamical system. *J Chem Phys*, 1981. **75**: p. 4970–4979.
46. Gavalas, G.R., *Nonlinear Differential Equations of Chemically Reacting Systems*. 1968, Berlin: Springer-Verlag.
47. Kauffman, S.A., *Origins of Order: Self-Organization and Selection in Evolution*. 1993, Oxford: Oxford University Press.
48. Leclerc, R., Survival of the sparsest: Robust gene networks are parsimonious. *Mol Syst Biol*, 2008. **4**.
49. Davidich, M.I. and S. Bornholdt, Boolean network model predicts cell cycle sequence of fission yeast. *PLoS One*, 2008. **3**(2).
50. Russell, S. and P. Norvig, *Artificial Intelligence: A Modern Approach*. 2002, Upper Saddle River, NJ: Prentice Hall.
51. Shmulevich, I., et al., Probabilistic Boolean networks: A rule-based uncertainty model for gene regulatory networks. *Bioinformatics*, 2002. **18**(2): p. 261–274.
52. Shmulevich, I., E.R. Dougherty, and W. Zhang, From Boolean to probabilistic Boolean networks as models of genetic regulatory networks. *Proc IEEE*, 2002. **90**(11): p. 1778–1792.
53. Milo, R., et al., Network motifs: Simple building blocks of complex networks. *Science*, 2002, 25 October. **298**: p. 824–827.
54. Ptashne, M., *A Genetic Switch: Phage Lambda Revisited*. 3rd ed. 2004, New York, NY: Cold Spring Harbor Laboratory Press.
55. Weitz, J.S., et al., Collective decision making in bacterial viruses. *Biophys J*, 2008. **95**(6): p. 2673–2680.
56. Lipshtat, A., et al., Functions of bifans in context of multiple regulatory motifs in signaling networks. *Biophys J*, 2008. **94**: p. 2566–2579.
57. Schwacke, J.H. and E.O. Voit, The potential for signal integration and processing in interacting MAP kinase cascades. *J Theor Biol*, 2007. **246**(4): p. 604–620.

58. Saez-Rodriguez, J., et al., A logical model provides insights into T cell receptor signaling. *PLoS Comput Biol*, 2007. **3**(8): p. e163.
59. Ma'ayan, A., et al., Ordered cyclic motifs contribute to dynamic stability in biological and engineered networks. *Proc Natl Acad Sci USA*, 2008. **105**(49): p. 19235–19249.
60. Alon, U., Simplicity in biology. *Nature*, 2007, 29 March. **446**: p. 497.
61. Olobatuyi, M.E., *A User's Guide to Path Analysis*. 2006, Lanham, MD: University Press of America, Inc.
62. Shipley, B., *Cause and Correlation in Biology: A User's Guide to Path Analysis, Structural Equations and Causal Inference*. 2000, Cambridge: Cambridge University Press.
63. Torralba, A.S., et al., Experimental test of a method for determining causal connectivities of species in reactions. *Proc Natl Acad Sci USA*, 2003. **100**: p. 1494–1498.
64. Vance, W., A.P. Arkin, and J. Ross, Determination of causal connectivities of species in reaction networks. *Proc Natl Acad Sci USA*, 2002. **99**: p. 5816–5821.
65. Anastassiou, D., Computational analysis of the synergy among multiple interacting genes. *Mol Syst Biol*, 2007. **3**: p. 83, https://doi.org/10.1038/msb4100124.
66. Shannon, C.E. and W. Weaver, *The Mathematical Theory of Communication*. 1949, Urbana, IL: University of Illinois Press.
67. Butte, A.J. and I.S. Kohane, Mutual information relevance networks: functional genomic clustering using pairwise entropy measurements. *Pac Symp Biocomput*, 2000. **5**: p. 415–426.
68. Komili, S. and P.A. Silver, Coupling and coordination in gene expression processes: A systems biology view. *Nat Rev Genet*, 2008. **9**: p. 38–48.
69. Bayes, T., An essay towards solving a problem in the doctrine of chances. *Phil Trans Royal Soc London*, 1763. **53**: p. 370–418.
70. Pearl, J., *Causality, Models, Reasoning, and Inference*. 2000, Cambridge, UK: Cambridge University Press.
71. Mitra, S., et al., *Introduction to Machine Learning and Bioinformatics*. *Computer Science and Data Analysis Series*. 2008, Boca Raton: CRC Press, Taylor and Francis.
72. Sachs, K., et al., Causal protein-signaling networks derived from multiparameter single-cell data. *Science*, 2005. **308**: p. 523–529.
73. Bork, P., et al., Protein interaction networks from yeast to human. *Curr Opin Struct Biol*, 2004. **14**(3): p. 292–299.
74. Zhang, A., *Protein Interaction Networks: Computational Analysis*. 2009, Cambridge, UK: Cambridge University Press. 292.
75. Rhodes, D.R., et al., Probabilistic model of the human protein-protein interaction network. *Nat Biotechnol*, 2005. **23**(8): p. 951–959.
76. Ewing, R.M., et al., Large-scale mapping of human protein-protein interactions by mass spectrometry. *Mol Syst Biol*, 2007. **3**: p. 89.
77. Libby, E., T.J. Perkins, and P.S. Swain, Noisy information processing through transcriptional regulation. *Proc Natl Acad Sci USA*, 2007. **104**: p. 7151–7156.
78. Zhu, J., et al., Increasing the power to detect causal associations by combining genotypic and expression data in segregating populations. *PLoS Comput Biol*, 2007. **3**(4): p. e69.
79. Gasch, A.P., et al., Genomic expression programs in the response of yeast cells to environmental changes. *Mol Biol Cell*, 2000. **11**(12): p. 4241–4257.
80. Hecker, M., et al., Gene regulatory network inference: Data integration in dynamic models-a review. *Biosystems*, 2009. **96**(1): p. 86–103.
81. Sieberts, S.K. and E.E. Schadt, Moving toward a system genetics view of disease. *Mamm Genome*, 2007. **18**(6–7): p. 389–401.

Taylor & Francis
Taylor & Francis Group
http://taylorandfrancis.com

Discrete Biological Systems

4

When you have read this chapter, you should be able to:

- Discuss recursive systems with memory 1 and 2
- Execute basic modeling approaches for discrete systems
- Understand the use of matrix equations for recursions
- Describe the differences and relationships between linear and nonlinear models
- Understand concepts of steady states and their stability
- Discuss the differences between deterministic and stochastic discrete models
- Design and analyze Markov models
- Discuss deterministic and stochastic aspects of Markov models
- Execute the steps of a typical analysis of discrete biological systems

Chapter 2 introduced some general concepts for modeling biological systems in an intuitive fashion, while skipping a lot of technical detail. Chapter 3 continued the discussion by addressing static networks, which do not change over time. This chapter and the next present mathematical and computational methods for analyzing dynamic systems, which do have the potential of changing over time. The range of such methods is almost unlimited, and not even experts master them all in detail. Nonetheless, there are various standard methods that are sufficient for setting up and analyzing basic models in a variety of formats, even if they may not have all the bells and whistles of the most modern mathematical techniques and software packages.

Chapter 2 provided a bird's eye view. Here and in Chapter 5, we "look under the hood," design models from scratch, introduce common methods of analysis, and introduce the thrills of exploring biology on a computer, especially in cases where the time trends turn out to be "weird." The material will also provide you with a solid platform from which to dive into the rich literature of dynamic modeling, which spans a vast range from relatively easy introductions to very specialized texts and a growing number of journal articles.

As discussed in the introduction to modeling in Chapter 2, there are numerous types and implementations of mathematical and computational models. Here, we focus on systems that are studied at **discrete time points**. Like a digital clock that shows hours and minutes, but not seconds, these systems are observed and modeled at time points 1, 2, 3, . . . but not in between.

The chapter discusses linear and nonlinear models. The former are mathematically very convenient, while the latter are often more realistic in biological applications. Discrete linear representations allow us to use **matrices** and **vectors** and

DOI: 10.1201/9781003402770-4

therewith the enormous repertoire of tools from linear algebra. They can effectively describe deterministic processes, as well as some stochastic processes that capture the probabilities with which a system develops over time. An important example is the Markov chain model that has numerous applications not just in biology but in a variety of fields. Matrix models are at the heart of many techniques in traditional statistics, such as regression and the analysis of variance, as well as optimization and operations research. We will not discuss these far-reaching applications though, as they are outside our scope.

The chapter requires, here and there, basic working knowledge of vectors, matrices, and determinants, which you might want to review. The main goal of the chapter is to formalize concepts of model design, diagnostics, and analysis of discrete systems and to provide the skills to set up and analyze models from scratch. In contrast to a typical math book, which uses the famous lemma–theorem–proof–corollary–example style, this chapter introduces concepts and definitions on the fly, uses examples as motivation, and stops before things get too complicated. It is subdivided into discussions of **deterministic** and **stochastic**, as well as **linear** and **nonlinear** discrete models.

4.1 DETERMINISTIC RECURSIVE MODELS

4.1.1 Linear Recursions

A linear recursion model describes the state of a system as a linear function of the state one time unit earlier. As a simple example, imagine a cell that divides into two, then four, then eight daughter cells, so that the assembly of cells doubles in number at every division step. Discrete linear models may take on a variety of formats, but generically just require a discrete time scale and exclusively linear functions. This chapter focuses primarily on discrete models that are constructed **recursively**. This terminology means that time is only considered in discrete steps and that time periods in between are ignored (as on the display of a digital clock). In the case of linear recursions, the state of the system at time t is represented as a linear function of the state of the system at time $t - 1$. In the simplest, single-variable, case, the state is characterized by a single feature, whereas a multivariate state is characterized by several, simultaneous features. For instance, the blood pressure "state" of a person consists of two variables, namely, a systolic and a diastolic pressure reading. The state of an ecosystem might be describable in terms of the numbers of organisms within each species at a given time, and it could also include temperature, pH, and other factors. A metabolic profile describes the state of a cell's metabolism as a vector of hundreds or thousands of metabolites. During each step of a recursion, some of the metabolites might be converted into other metabolites.

A typical single-variable example is the size P_t of a small bacterial population at time points $t = 0, 1, 2, \ldots$ The time points may refer to minutes, hours, days, years, or any other fixed time periods, such as 20-minute intervals. The index notation P_t is commonly used for discrete models, whereas the notation $P(t)$ usually refers to a continuous model, where any values of t are permitted (see Chapter 5).

For simplicity, let us assume that the bacteria are synchronized and all divide simultaneously every τ minutes; without these assumptions, the situation quickly becomes complicated [1, 2]. Given synchronous doubling, we can formulate a simple recursive model of the type

$$P_{t+\tau} = 2 \cdot P_t. \tag{4.1}$$

The equation simply says that at time point $t + \tau$ there are twice as many bacteria as at time t, and this is true for any time point t and no matter how many bacteria are actually present at t. Note that, in this formulation, t may progress from 1 to 2 to 3, or maybe from 60 to 120 to 180, and that τ remains fixed at 1 or 60, respectively, throughout the computational experiment. Also note that the equation is called a linear equation, but that the process really describes **exponential growth**, because starting with $P_t = 1$, we obtain the series 1, 2, 4, 8, 16, ... for the size of the population.

Suppose the population starts with ten cells at time $t = 0$ and the division time is $\tau = 30$ minutes. Then we can easily project the growth of this population, according

TABLE 4.1: GROWTH OF A BACTERIAL POPULATION ACCORDING TO (4.1)

Time (min)	0	30	60	90	120	150	180
Population size	10	20	40	80	160	320	640

to Table 4.1: every 30 minutes, the population size doubles. Using (4.1) for the second doubling step, we immediately see that

$$P_{(t+\tau)+\tau} = 2P_{t+\tau} = 2\cdot 2\cdot P_t = 2^2\cdot P_t, \tag{4.2}$$

and this relationship is true no matter how far t has progressed in its discrete steps. By extension, if we move forward n doubling steps of 30 minutes each, we find

$$P_{t+n\tau} = 2^n P_t. \tag{4.3}$$

This formulation contains two components: the initial population size at some time point t, which is P_t, and the actual doubling process, which is represented by the term 2^n. Because the **power function** 2^n is mathematically defined as $\exp(n\ln 2)$, this process is called **exponential growth**. Much has been written about this process, and further exploration is left for the Exercises. A feature to ponder is what happens (biologically and mathematically) at time points between t and $t+\tau$ or between $t+3\tau$ and $t+4\tau$. Of course the bacterial population does not cease to exist, but the model simply does not address these time points. The reader should think about how the model could be altered conceptually or mathematically to accommodate these missing time points.

A slightly more complicated recursive growth model, which has found wide application in population studies, was proposed by Leslie [3]. The key difference from the bacterial growth process in (4.1) is that the population in this model is stratified by age, which means that it consists of individuals with different ages that typically have different proliferation and survival rates. Thus, two processes overlap: individuals age and also bear children, with a fecundity rate that depends on their age. Furthermore, a certain portion of each age group dies. We will discuss this model in Chapter 14.

4.1.2 Case Study: Red Blood Cell Dynamics

Leah Edelstein-Keshet introduced a small, yet informative example of a recursive model [4]. It describes, in a very simplified form, the dynamics of red blood cells. These cells are constantly formed by the bone marrow and destroyed by the spleen, but in a healthy body, their number remains more or less constant from day to day. In the simple model, the loss is modeled as proportional to the current number of cells in circulation, and the bone marrow produces a number of new cells that is proportional to the number of cells lost on the previous day. Thus, there are two critical numbers, namely,

R_n = number of red blood cells in circulation on day n

and

M_n = number of red blood cells produced by the bone marrow on day n.

These numbers depend recursively on cell numbers during the previous day. Specifically, R_n is given as the number of red blood cells in circulation on the previous day, and this number is diminished by losses through degradation in the spleen and increased by cells that were newly produced by the bone marrow on the previous day. If the fraction of cells removed by the spleen is f, which typically is a small number between 0 and 1, and if the production rate is denoted by g, which is defined as the number of cells produced per number of cells lost, one can formulate the following model equations:

$$R_n = (1-f)\cdot R_{n-1} + M_{n-1}, \tag{4.4a}$$

$$M_n = g\cdot f\cdot R_{n-1}. \tag{4.4b}$$

Similar to the time counter in the previous example, n is simply a counter representing the day under investigation. We can write these recursive equations also with shifted indices as

$$R_{n+1} = (1-f)\cdot R_n + M_n, \tag{4.5a}$$

$$M_{n+1} = g\cdot f\cdot R_n, \tag{4.5b}$$

and the two formulations (4.4) and (4.5) are exactly equivalent.

We can now ask interesting questions. For instance, given starting values R_0 and M_0, as well as f and g, how do the numbers R_n and M_n develop over time? Will they grow? Will they decrease? Will they settle down at some value? The problem is simple enough to allow an analysis in Excel. As a specific example, suppose $R_0 = 1000$, $M_0 = 10$, $f = 0.01$, and $g = 2$. Solving the equations from $n = 0$ to $n = 1$, $n = 2$, and so forth shows that both R_n and M_n continue to increase (**Figure 4.1**). Resetting $g = 1$, the plot is different: both data trends become flat after just a few of steps (not shown) and the trends settle down at $R_n \approx 962$ and $M_n \approx 48$.. Thus, the overall number of cells remains the same but the initial 1010 cells become differently distributed over time (the reader should confirm these results).

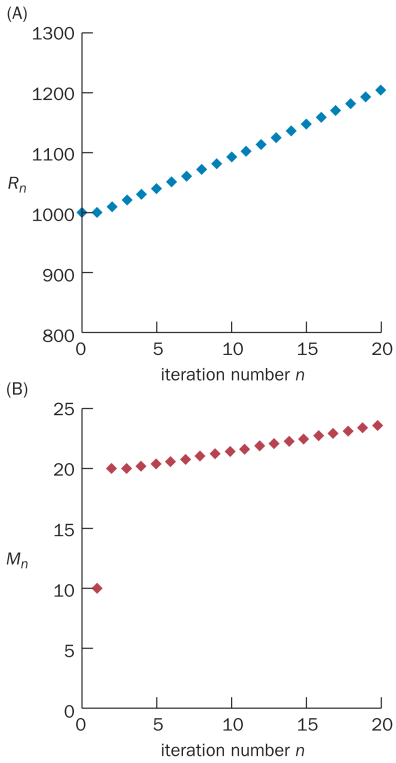


Figure 4.1 Trends in red blood cells over time. (A) The number of cells in circulation, R_n, and (B) the number of cells produced by the bone marrow, M_n, versus number of iterations, n. The model settings were $R_0 = 1{,}000$, $M_0 = 10$, $f = 0.01$, and $g = 2$.

The model allows us to compute under what conditions the system exhibits a **steady state**. This special state, in biology sometimes called **homeostasis**, is defined by the situation where neither R_n nor M_n changes in value from one iteration to the next. Expressed mathematically, the condition of a steady state requires

$$R_{n+1} = R_n \tag{4.6a}$$

$$M_{n+1} = M_n \tag{4.6b}$$

If this condition is satisfied, then $R_{n+2} = R_{n+1}$, $R_{n+3} = R_{n+2}$, and so on, and the same is true for M_n. Expressed in words, neither R nor M will change ever again.

An interesting way to assess these steady-state conditions is to recall that the progression in R and M from $n - 1$ to n is exactly the same as the progression from n to $n + 1$. Therefore, we are allowed to call up the recursion formula for M_n at time point n (4.4b), instead of $n + 1$, and plug the result into the equation for R_{n+1} at time point $n + 1$ (4.6a). The result is

$$\begin{aligned} R_{n+1} &= (1-f)\cdot R_n + M_n \\ &= (1-f)R_n + g\cdot f\cdot R_{n-1} \end{aligned} \tag{4.7}$$

The notable feature of (4.7) is that it contains only R but now at three time points. For R (and M) to be in a steady state on days $n - 1$, n, and $n + 1$, we must demand that $R_{n+1} = R_n = R_{n-1}$, because if this condition is given, R_{n+2}, R_{n+3} and all later values of R will remain the same, and the same is true for M. This pair of equations, $R_{n+1} = R_n = R_{n-1}$, has two solutions, namely $R_{n+1} = R_n = R_{n-1} = 0$, which is not very interesting because there would be no cells at all, and $1 = (1 - f) + g \cdot f$, which is satisfied exclusively for $g = 1$. What does that mean biologically? The condition mandates that the bone marrow produces exactly as many cells as are lost, which in retrospect makes a lot of sense. This is true for every positive value of f.

So, what are the actual steady-state values for R and M? We know that g must be equal to 1; so, let us assume that that is true. Furthermore, let us pick some (yet unknown) value for f and try to compute the steady-state values R_n and M_n from (4.6) by again requiring $R_{n+1} = R_n$ and $M_{n+1} = M_n$. The second equation yields $M_n = M_{n+1} = f R_n$; thus, the number of red blood cells produced by the bone marrow on day n must equal the cells lost on day n. Plugging this result into the first equation of the set just yields $R_n = R_{n+1} = (1-f)R_n + f R_n = R_n$, which means that the system is at a steady state for *any* values of f and R_n, as long as $g = 1$ and $M_n = f R_n$. Confirm this result with numerical examples and ponder whether it makes sense biologically.

It is beneficial for many analyses to rewrite these types of linear recursive equations as single matrix equations. In our case, this equation is

$$\begin{pmatrix} R \\ M \end{pmatrix}_{n+1} = \begin{pmatrix} 1-f & 1 \\ g & f & 0 \end{pmatrix} \cdot \begin{pmatrix} R \\ M \end{pmatrix}_n \tag{4.8}$$

According to the rules for multiplying a vector by a matrix, we obtain the value for R_{n+1} by multiplying the first row of the matrix with the vector $\begin{pmatrix} R \\ M \end{pmatrix}_n$, which yields $R_{n+1} = (1 - f)R_n + M_n$, and we obtain M_{n+1} by multiplying the second row of the matrix with the same vector, which yields $M_{n+1} = gfR_n$. Thus, the matrix equation is a compact equivalent of the recursive equations (4.5a, b). This representation is particularly useful for large systems.

The matrix formulation makes it easy to compute future states of the system, such as R_{n+2} or M_{n+4}. First, we know that

$$\begin{pmatrix} R \\ M \end{pmatrix}_{n+2} = \begin{pmatrix} 1-f & 1 \\ gf & 0 \end{pmatrix} \cdot \begin{pmatrix} R \\ M \end{pmatrix}_{n+1} \tag{4.9}$$

because n is only an index. Second, we can express the vector $\begin{pmatrix} R \\ M \end{pmatrix}_{n+1}$ according to the matrix equation (4.8). Combining these two aspects, the result is

$$\begin{aligned} \begin{pmatrix} R \\ M \end{pmatrix}_{n+2} = \begin{pmatrix} 1-f & 1 \\ gf & 0 \end{pmatrix} \cdot \begin{pmatrix} R \\ M \end{pmatrix}_{n+1} &= \begin{pmatrix} 1-f & 1 \\ gf & 0 \end{pmatrix} \cdot \begin{pmatrix} 1-f & 1 \\ gf & 0 \end{pmatrix} \cdot \begin{pmatrix} R \\ M \end{pmatrix}_n \\ &= \begin{pmatrix} 1-f & 1 \\ gf & 0 \end{pmatrix}^2 \cdot \begin{pmatrix} R \\ M \end{pmatrix}_n \end{aligned} \tag{4.10}$$

According to the rules of matrix multiplication,

$$\begin{pmatrix} A & B \\ C & D \end{pmatrix} \cdot \begin{pmatrix} a & b \\ c & d \end{pmatrix} = \begin{pmatrix} Aa + Bc & Ab + Bd \\ Ca + Dc & Cb + Dd \end{pmatrix} \tag{4.11}$$

we obtain

$$\begin{pmatrix} 1-f & 1 \\ gf & 0 \end{pmatrix}^2 = \begin{pmatrix} (1-f)^2 + gf & 1-f \\ gf(1-f) & gf \end{pmatrix} \tag{4.12}$$

The matrix equation is not limited to two steps, and indeed, any higher values of R_{n+k} and M_{n+k} can be computed directly from R_n and M_n, when we use the corresponding power of the matrix:

$$\begin{pmatrix} R \\ M \end{pmatrix}_{n+k} = \begin{pmatrix} 1-f & 1 \\ gf & 0 \end{pmatrix}^k \cdot \begin{pmatrix} R \\ M \end{pmatrix}_n \tag{4.13}$$

Matrix methods for recursive systems can also be used to analyze the stability of steady states or to make statements regarding the long-term trends of the system, which we will discuss in a later section. The red blood cell model is of course extremely simplified. Nonetheless, the same structure of a recursive model has been used for realistic models of the dynamics of red blood cells in health and disease, containing thousands of variables [5, 6].

4.1.3 Nonlinear Recursions

In contrast to linear recursions, nonlinear models require us to make an additional, rather challenging choice. Namely, we need to determine or select the most suitable function(s) for the right-hand side(s) of the **difference equations** that characterize these models. Linear models always have the same generic format, but in the case of nonlinear models, infinitely many alternatives are available.

Nonlinear representations are often more realistic for biological systems than their linear counterparts. For instance, the linear model in (4.1) may be a good representation for the growth of small bacterial cultures, but it is quite evident that the model eventually becomes unrealistic, because it predicts that the entire world

would ultimately be covered with a thick (and ever-growing) layer of this particular bacterium. In reality, all biological systems have mechanisms that slow down growth or production, for instance, through feedback. These mechanisms are usually nonlinear and lead to time trends that can be distinctly different from the time courses we observe in linear difference equations.

To illustrate how quickly nonlinearities may become drastically different from linear models, let us consider the **logistic map**, which the mathematical ecologist Robert May introduced in 1976 to describe the dynamics of populations [7]. This recursive model is not all that different from the bacterial growth equation (4.1) but contains an additional term that makes the model nonlinear:

$$P_{t+1} = r \cdot P_t \cdot (1 - P_t). \tag{4.14}$$

The starting point P_1 is somewhere between 0 and 1, and r is a parameter between 0 and 4. It is easy to compute P_{t+1} from P_t and r, but it is very difficult to make general predictions of what this little model will do if we let it run. Let's look at a few cases and leave the rest as an exercise. As the first illustration, suppose $r = 1.5$. Starting with $P_1 = 0.5$, the model quickly approaches its steady state, which mathematicians like to call **fixed point**, of 1/3. Indeed, if we plug $r = 1.5$ and $P_t = 1/3$ into (4.14), we can easily check that P_{t+1} also equals 1/3 (**Figure 4.2A**), which implies that all further P's are also 1/3; convince yourself that this is true. Turning this argument around, we can compute the steady state of a logistic model if we know r: namely, we require $P_{t+1} = P_t$ and numerically solve for the steady state P_{FP}. Thus, for $r = 2$, we obtain

$$P_{FP} = 2 \cdot P_{FP} \cdot (1 - P_{FP}). \tag{4.15}$$

Simple algebra shows that 0 is a fixed point, and so is 0.5.

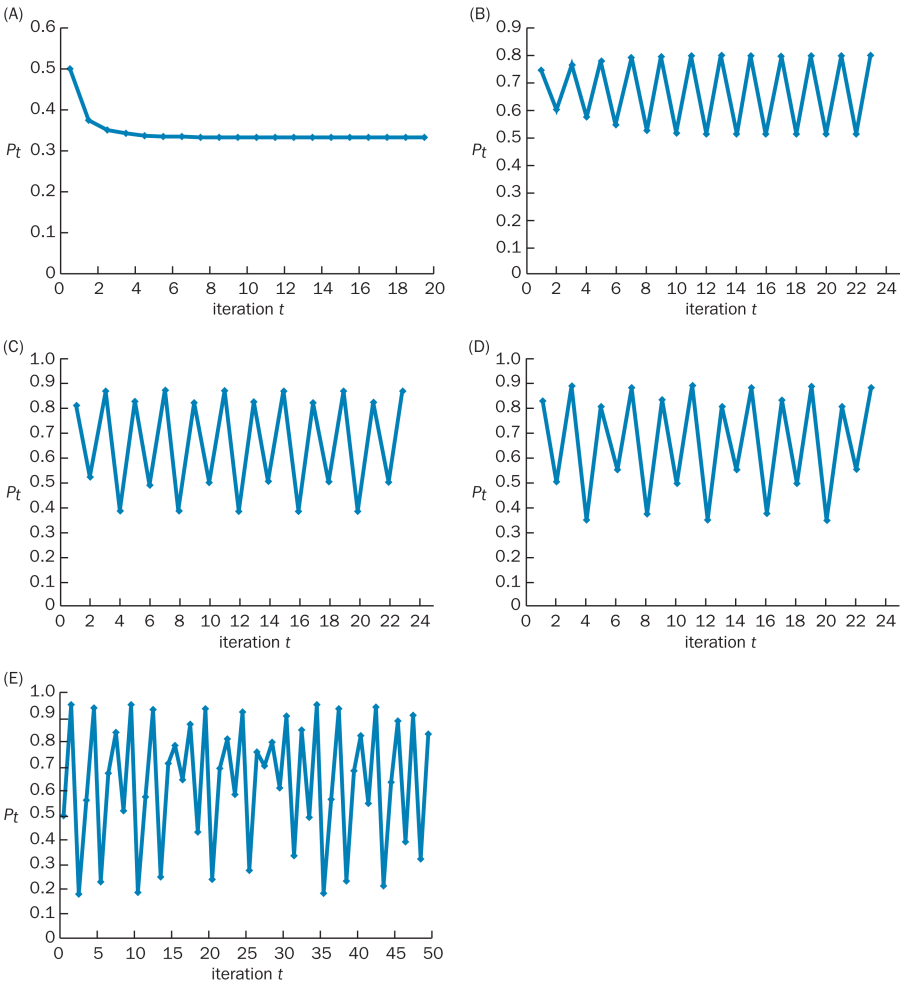


Figure 4.2 Values of the logistic map (Eqn. 4.15) over time. Iterations of this map can lead to amazingly complicated behaviors. The transition from a simple steady state to chaos leads through phases of period doubling: Panel A: $r = 1.5$. Panel B: $r = 3.2$ (Period 2). Panel C: $r = 3.5$ (Period 4). Panel D: $r = 3.56$ (Period 8). Panel E: $r = 3.8$ (Chaos).

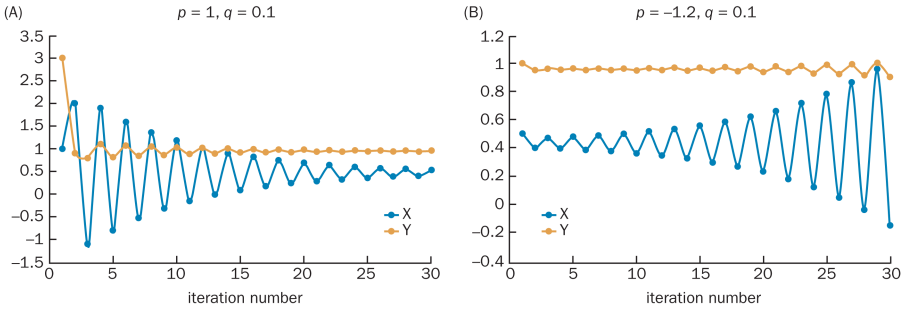


Figure 4.3 Simulation results for different recursions with the system in (4.16). A: Results for $p = -1$, $q = -0.1$; the steady state is stable. B: Results for $p = -1.2$, $q = -0.1$; the steady state is unstable. Only the dots represent the solution; the lines are added for better visibility of the trends in X and Y.

Now let's consider the setting $r = 3.8$. Using the same strategy, we compute the fixed point as 0.7368 and confirm that we obtain $P_{t+2} = P_{t+1} = 0.7368$ if $P_t = 0.7368$, and the same value holds for all future iterations. However, if we start the recursion with $r = 3.8$ and a value of P_1 that is not exactly 0.7368, the model jumps around erratically, and one can show that its fluctuations are so unpredictable that the field of dynamical systems calls them **chaotic** (Figure 4.3E). The transition from an "orderly" approach of a steady state to chaos exhibits several identifiable intermediates, where the periodicity of the oscillation changes. This so-called *period doubling* happens for $r \approx 3.2$ (period 2), $r \approx 3.5$ (period 4), $r \approx 3.56$ (period 8) (see Figure 4.3B–D). Further doublings become more difficult to decipher. We will discuss other chaotic processes toward the end of the chapter and in Chapter 5.

The logistic map has attracted a lot of attention in the literature, and features such as the stability of its fixed point and the progression toward chaos have been studied intensely (see, for example, [8]). Considering the surprising dynamics of the simple logistic map, it is easy to imagine how it is possible for iterative maps with many variables and more complicated nonlinearities to become extremely complicated. Indeed, the cover of this book contains a nonlinear iterative map based on a complex polynomial. Depending on the details of such a map, the results are called fractals, Mandelbrot sets, or Julia sets. They can be quite diverse and are often stunningly beautiful [9].

4.1.4 Stability of a Steady State

In the red blood cell example, we discussed the concept of a steady state, in which the variables of a system do not change from one iteration to another. Very many biological systems operate close to such a state of homeostasis. Alas, environmental perturbations always tend to move a biological system away from this state, and the question arises whether the system returns to this state or moves away from it. If a system returns to a steady state after a small perturbation, this steady state is called **stable**. If a steady state is unstable, the system will not return but deviate, often, but not always, moving toward ∞ or $-\infty$, sometimes through oscillations with increasing amplitude. A recursive system may have one, several, or no steady states. In the case of several steady states, some are usually stable and some unstable.

The stability of a steady state point can be assessed with **eigenvalue** analysis of the recursion equations. If the right-hand sides of the recursion are linear, the eigenvalues are computed for the system matrix. If the system is nonlinear, the system must first be linearized at the steady state point.

As an illustration, consider the following linear recursive system with two variables:

$$X_{n+1} = p\,X_n + Y_n \tag{4.16a}$$

$$Y_{n+1} = q\,X_n + 1, \tag{4.16b}$$

where p and q are parameters that still are to be specified. The first order of business is the computation of all steady states. As we discussed before (4.6), the conditions for a steady state are

$$X_{n+1} = X_n$$

$$Y_{n+1} = Y_n \tag{4.17}$$

Computing the steady states allows the same trick we used for the red blood cell example. Namely, we may use Y_n and Y_{n-1} instead of Y_{n+1} and Y_n in (4.16b) and then substitute Y_n in (4.16a). The result is a single equation that determines whether the system has one or more steady states and what they are:

$$X_{n+1} = p\,X_n + q\,X_{n-1} + 1 \tag{4.18}$$

Furthermore, because $X_{n+1} = X_n = X_{n-1}$ at the steady state, we can replace the indices and require for any steady-state value X_{SS}:

$$X_{SS} = p\,X_{SS} + q\,X_{SS} + 1 \tag{4.19}$$

Straightforward algebra yields answer: Any steady state must satisfy

$$X_{SS} = \frac{1}{(1-p-q)},\quad Y_{SS} = \frac{q}{(1-p-q)} + 1 \tag{4.20}$$

where the second condition is derived from the first and from (4.16b). Thus, unless $p + q = 1$, the system has a steady state. We can still simulate the system even for $p + q = 1$, but the system does not have a steady state.

Let's now study the **stability** of a steady state. As an example, suppose, $p = -1$ and $q = -0.1$. The steady state then is $X_{SS} = \frac{1}{2.1} \approx 0.47619, Y_{SS} = \frac{-0.1}{2.1} + 1 \approx 0.95238$. We may put the system with these settings into a program like Excel and confirm that the values do not change if we start with $X_1 = X_{SS}$, $Y_1 = Y_{SS}$. The question now is whether this steady state is stable. Starting away from the steady state, for instance, $X_1 = 1$, $Y_1 = 3$ confirms stability: The recursion yields a damped oscillation toward the steady state (**Figure 4.3A**).

Let's change the parameter value of p slightly by setting $p = -1.2$. The steady state now is $X_{SS} = \frac{1}{2.3} = 0.434783, Y_{SS} = -\frac{0.1}{2.3} + 1 = 0.956522$. Simulation confirms that this state is a steady state. However, the stability test, for example, starting close to the steady state with $X_1 = 0.5$, $Y_1 = 1$ yields a very different result than the previous situation: The values of X and Y keep growing. In words, the recursion with $p = -1.2$ and $q = -0.1$ has a steady state, but it is unstable (Figure 4.3B).

Why are the results so different? We can use eigenvalue analysis to find an answer. Let's formulate our linear system in matrix format. Eqn. (4.16) then reads:

$$\begin{pmatrix} X \\ Y \end{pmatrix}_{n+1} = \begin{pmatrix} p & 1 \\ q & 0 \end{pmatrix} \begin{pmatrix} X \\ Y \end{pmatrix}_n + \begin{pmatrix} 0 \\ 1 \end{pmatrix} \tag{4.21}$$

The eigenvalues of the system matrix are computed from the determinant of the system matrix where λ is subtracted from the elements of the main diagonal [10]:

$$det \begin{pmatrix} p-\lambda & 1 \\ q & 0-\lambda \end{pmatrix} = (p-\lambda)\cdot(-\lambda) - q = \lambda^2 - p\,\lambda - q = 0 \tag{4.22}$$

The two solutions of this equation, λ_1 and λ_2, are the eigenvalues of the system. They are

$$\lambda_{1,2} = \frac{p \pm \sqrt{p^2 + 4q}}{2} \tag{4.23}$$

The rule for stability, which we will rationalize later, says: if both of these two eigenvalues are less than 1 in magnitude, the steady state is stable; otherwise, it is not. With this rule, we can test all scenarios of this type of recursion. For the two previous situations, $q = -0.1$ in both cases. For $p = -1$, the eigenvalues are

$$\lambda_1 = \frac{-1 + \sqrt{1-0.4}}{2},\ \lambda_2 = \frac{-1 - \sqrt{1-0.4}}{2} = -0.112. \tag{4.24}$$

By contrast, for $p = -1.2$, the eigenvalues are

$$\lambda_1 = \frac{-1.2 + \sqrt{1.44 - 0.4}}{2},\ \lambda_2 = \frac{-1 - \sqrt{1 - 0.4}}{2} = -1.11. \tag{4.25}$$

Thus, if $p = -1$, both eigenvalues are less than 1 in magnitude and the steady state is stable, whereas for $p = -1.2$, one eigenvalue exceeds 1 in magnitude, and that is sufficient to render the steady state unstable. Recall that the magnitude of the complex number $a + i\,b$ is $\sqrt{(a^2 + b^2)}$.

What is the rationale for the stability cut-off of 1? The reasons reside again in the field of linear algebra. For a two-variable recursion

$$\begin{pmatrix} X \\ Y \end{pmatrix}_{n+1} = A \begin{pmatrix} X \\ Y \end{pmatrix}_n \tag{4.26}$$

the initial vector may be written as a linear combination of the two eigenvectors of the system matrix; let's call them V_1 and V_2:

$$\begin{pmatrix} X \\ Y \end{pmatrix}_0 = \alpha_1 V_1 + \alpha_2 V_2 \tag{4.27}$$

For the first recursive update, one may write the solution of the linear system in terms of eigenvalues, which yields:

$$\begin{pmatrix} X \\ Y \end{pmatrix}_1 = A \begin{pmatrix} X \\ Y \end{pmatrix}_0 = A\left(\alpha_1 V_1 + \alpha_2 V_2\right) = \alpha_1 \lambda_1 V_1 + \alpha_2 \lambda_2 V_2 \tag{4.28}$$

Similarly, higher updates are

$$\begin{pmatrix} X \\ Y \end{pmatrix}_2 = A \begin{pmatrix} X \\ Y \end{pmatrix}_1 = A\left(\alpha_1 \lambda_1 V_1 + \alpha_2 \lambda_2 V_2\right) = \alpha_1 \lambda_1^2 V_1 + \alpha_2 \lambda_2^2 V_2 \tag{4.29}$$

and, ultimately

$$\begin{pmatrix} X \\ Y \end{pmatrix}_n = \alpha_1 \lambda_1^n V_1 + \alpha_2 \lambda_2^n V_2 \tag{4.30}$$

The last equation makes it evident that any eigenvalues with magnitude greater than 1 lead to values for the variables of the recursion that increase in magnitude. They may all be positive, all negative, or switch back and forth. The same arguments apply to recursions with more than two variables.

Nonlinear systems do not have eigenvalues and eigenvectors per se. However, the stability of a steady state of a nonlinear system can be assessed with a linear system that approximates the nonlinear system at a steady-state point. Thus, we **linearize** the nonlinear system at the steady state and then proceed as before.

As an illustration, let's modify the earlier example (4.16) by defining the system as

$$X_{n+1} = p\sqrt{X_n} + Y_n \tag{4.31a}$$

$$Y_{n+1} = q\,X_n^2 + r\,Y_n^{0.1} \tag{4.31b}$$

The steady state is not easy to compute now, but it can be revealed in a simulation. Let's start with $p = -1$, $q = -0.3$, $r = 2$ and initiate the system at $X_1 = Y_1 = 1$. The recursion oscillates, but converges toward a steady state (X_{SS}, Y_{SS}) = (0.91857, 1.87688) (**Figure 4.4A**). Changing r to a higher value, such as $r = 4.5$, destabilizes the steady state, and even initiating the recursion close to this value leads to oscillations increasing in amplitude, and the system never returns to the steady state (Figure 4.4B).

To assess the stability of the steady state in our nonlinear recursion (4.31), we linearize the system at the steady state and use the eigenvalues of the linearized

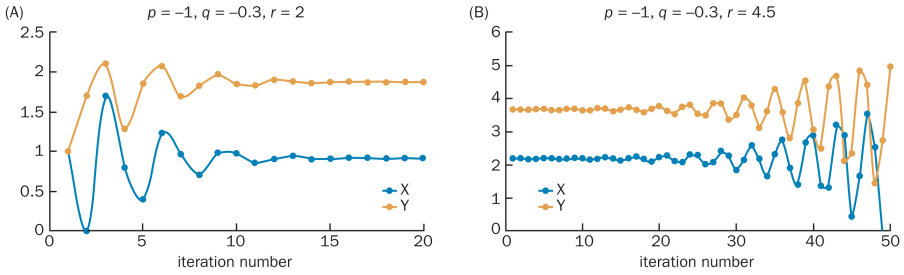


Figure 4.4 Results of simulations with the nonlinear recursion in (4.31). A: the parameter combination $p = -1$, $q = -0.3$, $r = 2$ yields a stable steady state. B: Changing r to 4.5 changes the steady state to different values. This steady state is unstable.

system for our analysis. This linearization is accomplished with the **Jacobian matrix**, which contains as elements the partial derivatives of the right-hand sides of (4.31a) and (4.31b) with respect to X and Y, respectively, evaluated at the steady state. This Jacobian for our example is

$$J = \begin{pmatrix} 0.5pX_{SS}^{-0.5} & 1 \\ 2qX_{SS} & 0.1rY_{SS}^{-0.9} \end{pmatrix} \tag{4.32}$$

The eigenvalues of this matrix are computed, as for the linear case, by solving the determinant equation

$$det = \begin{pmatrix} 0.5pX_{SS}^{-0.5} - \lambda & 1 \\ 2qX_{SS} & 0.1rY_{SS}^{-0.9} - \lambda \end{pmatrix} = \mathbf{0}. \tag{4.33}$$

Substituting the first set of numerical steady-state and parameter values yields $\lambda_1 = 0.56583$, $\lambda_2 = -0.974038$, indicating stability of the steady state, and $\lambda_1 = -1.1755$, $\lambda_2 = 1.12102$ for the second set, confirming that the steady state in this case is unstable.

4.2 STOCHASTIC RECURSIVE MODELS

The crucial difference between deterministic and stochastic recursive models is that the latter have probabilistic features, which means that they account for random occurrences. We describe three frequent types: random walks, probability trees, and Markov models.

4.2.1 Random Walks

Chapter 2 introduced binomial and Poisson processes, in which we considered sequences of the same random events, such as flipping a coin. Here we extend the concepts of stochastic modeling a bit further. A prominent example is the so-called **random walk**, in which an "agent" moves on a regular grid in one or more dimensions. A typical question is: what is the probability that the agent reaches some point of interest on that grid [11]?

In one dimension, a random walk is also called the gambler's ruin: a gambler with a certain amount of money, say \$500, repeatedly plays a game, every time either losing a fixed amount, say \$10, or winning this same amount (Figure 4.5). Before starting to play, the gambler decides to stop when s/he is broke or when his/her

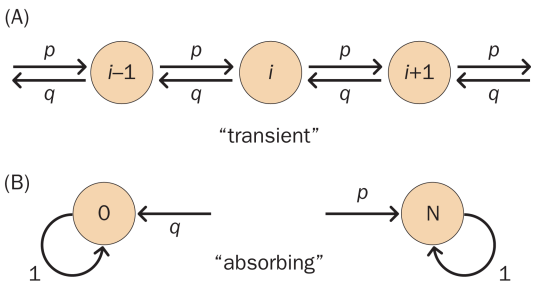


Figure 4.5 The gambler's ruin. In this one-dimensional random walk, a gambler wins \$1 with probability p and loses \$1 with probability $q = 1 - p$. What are the chances of ending up with a certain amount of dollars after a fixed number of games? Will the gambler go bankrupt and, if so, how long will that take on average? Panel A lets the gambler play without limits, and even go into debt. In Panel B, the gambler stops playing when s/he reached N dollars or is bankrupt (state 0). How are these chances affected if p and q are not 0.5 (fair game) but slightly or very different?

money has increased to say \$750. How long will the game last, and will s/he reach within a predefined number of games or go broke? The exercises discuss this application in several variations. This application may sound quite silly, but it can lead to surprising results. For instance, it can be used to derive the typical diffusion equation that drives all kinds of processes related to the spread of a liquid [4] or even the distribution of fire ants over time (see Chapter 14, [12], and Exercises).

4.2.2 Probability Trees

A second interesting stochastic model is a probability tree, which permits the computation of the probability of an outcome in a branched tree of events. These trees are quite easy to understand because they transition only forward and do not return to states encountered before. As an example, consider a fictitious two-step diagnostic test for some type of cancer. Suppose that a patient has a hidden cancerous lesion and suspects does not know it with certainty. The first test is cheap and noninvasive. It indicates that there is a lesion, but the probability of being correct is known to be only 70% on average. Because the test is not sufficiently reliable, a second test is performed that measures a different cancer biomarker. This test is more expensive, but also more discerning. Let's assume that the probability of Test 2 detecting the lesion is 90% if the first test had indicated it already, but that it refutes the earlier result with 10% probability. Furthermore, the second test still detects a lesion with 80% probability, even if the first test did not suggest it. The reason for the lower performance of the second test in the latter case might be that the lesion is in truth difficult to detect. The probabilities of different outcomes can be summarized in the probability tree in **Figure 4.6**. The overall result is that the second test increases the probability of a correct diagnosis from 70% to almost 90% (63% + 24%).

Very different examples of the use of probability trees in biology have included investigations of inheritance patterns [13] and the classification of cell types [14]. Similar trees have been used as event trees and fault trees in risk assessment [15, 16], which allow the identification of weak links in complex networks.

4.2.3 Markov Models

A slightly more complicated, and very widely applicable probabilistic model is a **Markov model** or Markov chain model, which superficially has a format similar to a larger version of the matrix equation in (4.8); a good introduction to these models is [17]. Generically, a Markov model describes a system that can assume m different states. At each (discrete) time point, the system is in exactly one of these states, from where it transitions with some probability into a different state or stays where it is.

An intuitive example is a pinball machine, where the possible locations of the ball are at the flippers and bumpers. These locations are the states of the model. An adequate time step in this case is the time for the ball to move from one state to the next. For simplicity, let's assume that the travel times between states are all the same and call the time points $0, 1, \ldots, t, t+1, \ldots$ At each time point, the one and

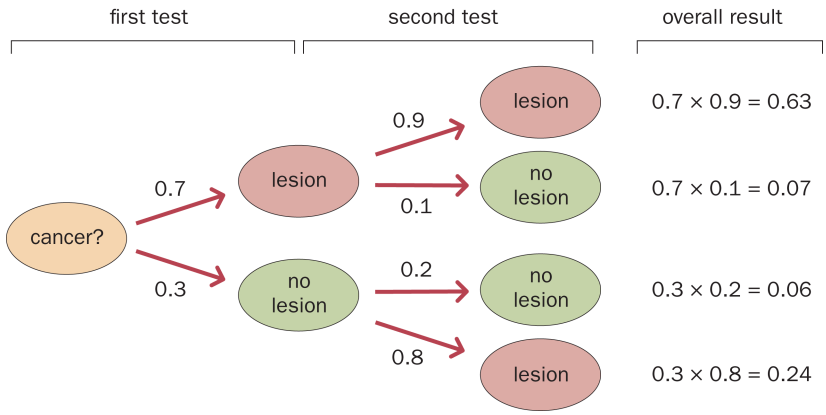


Figure 4.6 Probability tree for cancer diagnosis. The probability of a correct diagnosis is increased by the second test from 70% to 87%.

only ball is in one and only one state, from where it transitions, with some probability, to another state or remains in the same state.

A Markov model is constructed by looking collectively at very many games with the same pinball machine. Imagine we could record all movements of the ball from all games. If so, we would find balls and movements all over the place. However, some transitions between states would certainly be observed more often than others and, focusing on some given state i, each transition to some other state k would be observed with some specific frequency. The idea behind a Markov model is to use this observed frequency and interpret it as the probability of future transitions from state i to state k. Thus, the Markov process is defined by a fixed set of probabilities of transitions between all pairs of states: if a ball is in state i, the probability of moving to state j in one step is defined as p_{ij}, and this is true no matter where the ball was previously. Expressed differently, the Markov model has no memory. For a process with m states, we can put all probabilities into an $m \times m$ matrix, which is called the Markov matrix:

$$M = \begin{pmatrix} p_{11} & p_{12} & \dots & p_{1,m-1} & p_{1m} \\ p_{21} & p_{22} & \dots & p_{2,m-1} & p_{2m} \\ p_{31} & p_{32} & \dots & p_{3,m-1} & p_{3m} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ p_{m1} & p_{m2} & \dots & p_{m,m-1} & p_{mm} \end{pmatrix} \tag{4.34}$$

We can use M to compute the dynamics of the discrete Markov process. Let's assume that at the beginning of each game ($t = 0$) the ball is in State 1, which could be interpreted as the start position. Thus, $X_1 = 1$ and all other $X_i = 0$. At the next observation time point ($t = 1$) in game 1, the ball has moved to another state (or stayed in its start position). In the next game, the ball might have moved from the start position to a different state, and, in a third game, it may have moved to yet another position. This process is random for each game, but collectively governed by the probabilities in M: p_{13} is the probability of moving from state 1 to state 3 within one time unit, and p_{16} is the probability of moving to state 6. Thus, the first row of the matrix describes what percentages of games are expected to move the ball from state 1 into any of the states i, namely p_{1i}. With the next move, the ball moves again, according to the probabilities in (4.34). Tracking all these expectations and transitions amounts to a big bookkeeping problem, which the matrix equation (4.34) solves beautifully.

What do we know about the matrix elements p_{ij}? They describe the probabilities of moving from state i to state j, where j must be one of the m states in the equation; there is no other option, the ball cannot leave the system or hover between states. As always, every probability has a value between 0 and 1. Indeed it is possible that, for instance, $p_{13} = 0$, which means that it is impossible to move from state 1 to state 3 in one step. The ball could get to 3 in several steps, but not in one. If a probability p_{ij} is equal to 1, it means that the ball, if in state i, has no choice but to move to state j.

Suppose a ball is in state 4. Because it must move to one of the other states (or stay in state 4), all available probabilities, namely p_{4j} ($j = 1, \dots, m$), summed together must equal 1. Generally, we find, for each state i,

$$\sum_{j=1}^{m} p_{ij} = 1 \tag{4.35}$$

It is possible that one of the probabilities for leaving state i is 1 and all others are 0. For instance, if $p_{i2} = 1$, then it is certain that every ball in state i in every pinball game of this type moves to state 2 next. In particular, if $p_{ii} = 1$, the ball can never again leave state i. Game over! A state that cannot be left is called an **absorbing state**. Note that the summation over the first index (that is, a column), $p_{1j} + p_{2j} + \dots + p_{mj}$, does not necessarily yield 1 (explain why!).

Markov models can be found in diverse corners of systems biology. For instance, they may describe the dynamics of networks, they can be used to compute the probability of extinction of a population, and they appear in the form of hidden Markov models in bioinformatics [18]. The adjective "hidden" in this case refers to the situation that one cannot tell whether the process is in one hidden state or another.

BOX 4.1: MARKOV MATRIX VERSUS ADJACENCY MATRIX

The astute reader might notice similarities between the structure of a Markov transition matrix **M** and the adjacency matrix **A** described in Chapter 3. There are indeed strong conceptual similarities but also important differences. Both matrices capture the connectivity structure among nodes or states. Constructing **A** consists of indexing connections between nodes as existing (1) or absent (0). As a detail, **A** does not account for connections of nodes to themselves, which eliminates all diagonal elements with indices $i = j$. Conceptually, the network structure captures the *possibility* of direct interactions, mass flow, or information transmission, depending on the application domain of the network. Computing the n^{th} power of **A** allows an assessment of which node is reachable from any other node in exactly n steps. By contrast, **M** defines the *probabilities* of the system moving from one state to any other in one time step, including remaining in the current state. Computing the n^{th} power of **M** allows assessing the probability that a node is reached from any other node in exactly n steps.

For an illustrative example, consider an adjacency matrix **A** describing direct airline connections between cities. Powers of **A** permit figuring out how often one must change flights to reach the target, which Chapter 3 called the path length. Suppose cancellation statistics are available for all flights, e.g., under wintery conditions. The Markov matrix **M** uses the adjacency matrix as a starting point for capturing all direct connections between cities, that is, to determine all elements with non-zero value, but ultimately predicts to probability of arrival in spite of inclement weather.

As an example, consider the transition diagram in Figure 4.7, which corresponds to the matrix

$$\mathbf{M} = \begin{pmatrix} 0.1 & 0.5 & 0.4 \\ 0.2 & 0.3 & 0.5 \\ 0.8 & 0.2 & 0 \end{pmatrix} \tag{4.36}$$

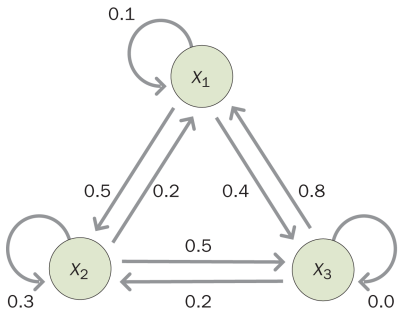


Figure 4.7 Diagram of a Markov process. The process is in one state, X_1, X_2, or X_3, at each time point and moves among these states with the probabilities shown on the figure.

Suppose the process starts in state X_2 at time $t = 1$, which we denote as $X_1(1) = 0$, $X_2(1) = 1$, $X_3(1) = 0$. Linear algebra allows us to determine the probabilities of finding the process in each of the three states at time $t = 2$. This computation is accomplished by multiplying and adding the appropriate probabilities, namely:

The probability to be in a particular state, say State 1, at time $t = 2$ is computed as:

(probability of being in State 1 at time $t = 1$) × (probability of staying in State 1, p_{11})
+ (probability of being in State 2 at time $t = 1$) × (probability of moving from 2 to 1, p_{21})
+ (probability of being in State 3 at time $t = 1$) × (probability of moving from 3 to 1, p_{31}).

The analogous computation is done for States 2 and 3. Matrix algebra elegantly puts all computations for all states into a succinct format:

$$\begin{pmatrix} X_1 & X_2 & X_3 \end{pmatrix}_{t=1} \cdot \mathbf{M} = \begin{pmatrix} X_1 & X_2 & X_3 \end{pmatrix}_{t=2}. \tag{4.37}$$

Let's put (4.37) in words. The probabilities of being in each of the states at $t = 2$ are computed by multiplying M to the row vector of the probabilities of being in each of the states at $t = 1$.

Specific to our example, this means

$$\begin{pmatrix} 0 & 1 & 0 \end{pmatrix} \cdot \begin{pmatrix} 0.1 & 0.5 & 0.4 \\ 0.2 & 0.3 & 0.5 \\ 0.8 & 0.2 & 0 \end{pmatrix} = \begin{pmatrix} 0.2 & 0.3 & 0.5 \end{pmatrix} \tag{4.38}$$

Exactly the same procedure allows us to compute the probabilities at later time points. For instance, for $t = 3$, we obtain

$$\begin{pmatrix} X_1 & X_2 & X_3 \end{pmatrix}_{t=2} \cdot \mathbf{M} = \begin{pmatrix} X_1 & X_2 & X_3 \end{pmatrix}_{t=3} \tag{4.39}$$

and therefore

$$\begin{pmatrix} 0.2 & 0.3 & 0.5 \end{pmatrix} \cdot \begin{pmatrix} 0.1 & 0.5 & 0.4 \\ 0.2 & 0.3 & 0.5 \\ 0.8 & 0.2 & 0 \end{pmatrix} = \begin{pmatrix} 0.48 & 0.29 & 0.23 \end{pmatrix} \tag{4.40}$$

In both results, (4.38) and (4.40), the three probabilities in each row vector add up to 1, which is necessarily so because the process has only these three states as possible options.

As a side remark: most books compute this progression by multiplying a column vector from the right to the transpose of **M**, which is mathematically equivalent with (4.37), but it appears that (4.37) is more intuitive.

Because all equations are linear, they scale up to a different interpretation. This fact means that we could ask what would happen to a population of "agents" moving among states according to the same Markov model. To answer the question, we may simply use the proportions of this population, whose members currently reside in any of the various states, to compute where they are likely to be found at the next time point. Using the same example and supposing that all agents are in State 2 at time $t = n$, we obtain

$$\begin{pmatrix} 0 & 1000 & 0 \end{pmatrix}_n \cdot \begin{pmatrix} 0.1 & 0.5 & 0.4 \\ 0.2 & 0.3 & 0.5 \\ 0.8 & 0.2 & 0 \end{pmatrix} = \begin{pmatrix} 200 & 300 & 500 \end{pmatrix}_{n+1}$$

and (4.41)

$$\begin{pmatrix} 200 & 300 & 500 \end{pmatrix}_{n+1} \cdot \begin{pmatrix} 0.1 & 0.5 & 0.4 \\ 0.2 & 0.3 & 0.5 \\ 0.8 & 0.2 & 0 \end{pmatrix} = \begin{pmatrix} 480 & 290 & 230 \end{pmatrix}_{n+2}$$

The numbers in each row sum to 1,000 because we started with 1,000 agents and no agents are lost or added during the progression.

Similar to the discrete red blood cell model, the state vector at any later time point can be computed in a single step from the vector at time 1, by using the corresponding power of the matrix **M**:

$$\begin{pmatrix} X_1 & X_2 & X_3 \end{pmatrix}_{t=1} \cdot \mathbf{M}^2 = \begin{pmatrix} X_1 & X_2 & X_3 \end{pmatrix}_{t=3} \tag{4.42}$$

and

$$\begin{pmatrix} X_1 & X_2 & X_3 \end{pmatrix}_{t=1} \cdot \mathbf{M}^k = \begin{pmatrix} X_1 & X_2 & X_3 \end{pmatrix}_{t=k+1} \tag{4.43}$$

The second power of the Markov matrix in (4.42), $\mathbf{M}^2$, contains so-called two-step probabilities: For instance, the element (1, 2) expresses the probability of moving from State 1 to State 2 in exactly two steps. For our example,

$$\mathbf{M}^2 = \begin{pmatrix} 0.43 & 0.28 & 0.29 \\ 0.48 & 0.29 & 0.23 \\ 0.12 & 0.46 & 0.42 \end{pmatrix}. \tag{4.44}$$

Thus, the probability to move from State 1 to State 2 in exactly two steps (element (1, 2)) is 0.28. Analogously, the k^{th} power of the matrix contains k-step probabilities.

Under certain mathematical conditions, which are satisfied here [17], continuing the same type of multiplication as in (4.43) eventually leads to something like a steady state of this Markov process, which is called the **limiting distribution** among the states and which does not change with further multiplications. During the stepwise approach of this distribution, the columns of the powers of **M** eventually become identical. For instance, the sixteenth power of **M** is

$$\mathbf{M}^{16} = \begin{pmatrix} 0.35088 & 0.33918 & 0.30994 \\ 0.35088 & 0.33918 & 0.30994 \\ 0.35088 & 0.33918 & 0.30994 \end{pmatrix} \tag{4.45}$$

and higher powers are the same within the numerical accuracy of the solution.

A first comment regarding this result is that the rows still sum to 1, which makes sense because they still consist of the transition probabilities out of a particular state. Second, each column contains the same elements. What does that mean?

Let's look at the element (1, 2) in this matrix $\mathbf{M}^{16}$ again. It quantifies the probability of reaching State 2 from State 1 in exactly k steps. Furthermore, it has the same value as elements (2, 2) and (3, 2). This equivalence is no coincidence and can be interpreted as follows: after sufficiently many steps, it does no longer matter (much) where one had started. In the world of Markov models, history becomes less and less important for the present and the future the further it lies in the past.

Ultimately, each row of $\mathbf{M}^{\infty}$ consists of the components of the limiting distribution, which is typically denoted as a vector of π_i's. In our example,

$$\begin{pmatrix}\pi_1 & \pi_2 & \pi_3\end{pmatrix}=\begin{pmatrix}0.35088 & 0.33918 & 0.30994\end{pmatrix} \tag{4.46}$$

This limiting distribution is equivalent to the non-negative solution of the following matrix equation:

$$\begin{pmatrix}\pi_1 & \pi_2 & \pi_3\end{pmatrix}\cdot\mathbf{M}=\begin{pmatrix}\pi_1 & \pi_2 & \pi_3\end{pmatrix} \tag{4.47}$$

where the sum of the π's equals 1. The values of π_i may be interpreted as the probabilities of finding the Markov process in States 1, 2, or 3, respectively, once the process had enough time to converge. They also reflect the average proportions of time that the process spends in these states over a long period of time. Equation (4.47) furthermore expresses that this distribution does not change if the Markov matrix is applied to it, which is the same as the steady-state condition for recursive systems discussed earlier (e.g., 4.6). Despite this apparent determinism, the Markov process remains to be stochastic, and the next state can only be predicted in a probabilistic manner (see Box 4.2).

BOX 4.2: STOCHASTIC OR DETERMINISTIC?

The simulation project *Rat Race* illustrates the apparent duality of a Markov model between being stochastic and deterministic. We consider a rat in a maze (**Figure 1A**). In contrast to a real rat, this virtual rat is really dumb and never learns. Put into the start box (State 1), it moves randomly, in search of food. Because of the randomness (and the failure to learn), the rat runs into each wall of the box with equal probability 1/6. If the wall contains an exit, the rat runs into the neighboring hexagon. Furthermore, if the rat reaches the food, it stays there for at 40 time units, and if it gets shocked, it is incapacitated for 30 time units.

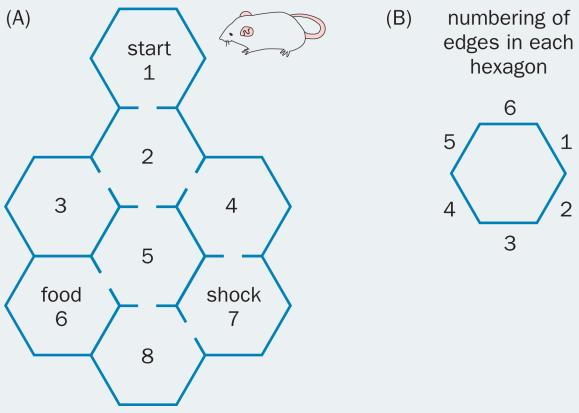
The project consists of the following:

- Using a die, simulate the Rat Race multiple times. Specifically, if you roll a one, the rat moves northeast, if you roll a three, it moves straight down, and so on, according to Figure 1B. During each simulation, record where the rat is after six, 12, 18, 24, and 30 steps.
- Also record the average number of steps it takes to reach one of the two absorbing states (food, shock), if the rate actually reaches one of them.
- Set up a Markov model for this scenario and analyze the process.

Once you have simulated this truly stochastic die-rolling process many times, you have data for five distributions. These quantify how often the rat was in a particular hexagon after six, 12, 18, 24, and 30 steps. Results from a class of 24 students executing the project are shown in **Figure 2** for times 6 and 30.

One sees immediately that the two histograms are rather different. They are also quite ragged. The raggedness would smooth out if we simulated the Rat Race hundreds of times. The left histogram indicates that even after six steps many rats (9/24) are still (or again) in the start box, whereas the right histogram shows that many rats are incapacitated and that no rat had made it to hexagon 8 or that it had reached the hexagon and then left again.

How does all this relate to the Markov model? First, we set up the Markov matrix. It has the form



Box 4.2 Figure 1 Setup for the Rat Race. A virtual rat is put into start box 1 and randomly searches for food. A. Numbering of compartments. B. Numbering of edges in each compartment.

$$\mathbf{M}=\begin{pmatrix} 5/6 & 1/6 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1/6 & 2/6 & 1/6 & 1/6 & 1/6 & 0 & 0 & 0 \\ 0 & 1/6 & 5/6 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1/6 & 0 & 4/6 & 0 & 0 & 1/6 & 0 \\ 0 & 1/6 & 0 & 0 & 3/6 & 1/6 & 0 & 1/6 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1/6 & 1/6 & 1/6 & 3/6 \end{pmatrix}.$$

Note that we set the transition probabilities p_{66} and p_{77} to 1 because the rats cannot leave the food and shock compartments within the time period of the experiment (see Exercises).

Second, the initial vector is

$$\begin{pmatrix} X_1 & X_2 & \cdots & X_8 \end{pmatrix}_{t=1} = \begin{pmatrix} 1 & 0 & \cdots & 0 \end{pmatrix}.$$

Computing the sixth and thirtieth powers of M and multiplying the results to the initial vector produces the distributions of locations of the rats after six and 30 steps, respectively, averaged over infinitely many simulations. These distributions are (with rounded numbers)

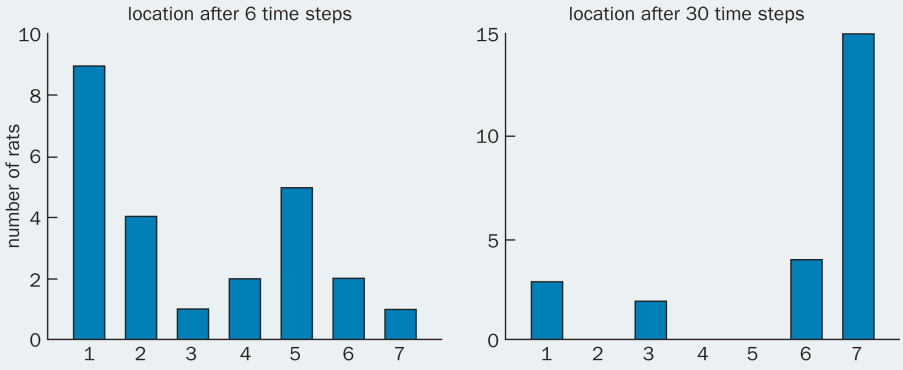
$$\begin{pmatrix} X_1 & X_2 & \cdots & X_8 \end{pmatrix}_{t=6} = \begin{pmatrix} 0.450 & 0.188 & 0.115 & 0.084 \\ 0.066 & 0.037 & 0.042 & 0.018 \end{pmatrix}$$

and

$$\begin{pmatrix} X_1 & X_2 & \cdots & X_8 \end{pmatrix}_{t=30} = \begin{pmatrix} 0.092 & 0.066 & 0.088 & 0.038 \\ 0.028 & 0.303 & 0.376 & 0.010 \end{pmatrix}.$$

Of course, these numbers are different from our experimental results (Figure 2), but if we redid the experiment very many times, the experimental results would be increasingly more similar to these numbers. Even the results of 24 experiments already indicate the same trends as the averages computed by the Markov model: after six steps, most rats are in the top hexagons, whereas after 30 steps most rats are either in the food or the shock compartment. The probability of being in hexagon 8 after 30 steps is only 1% in the Markov computation; in the experiment, we did not observe this outcome at all. The exercises ask you to answer additional questions.

To summarize this project: the Markov process is stochastic, but the matrix computations we executed previously are entirely deterministic and reflect *average* behaviors of the process for very many simulations. This duality is similar to rolling a die, where the next roll may produce any number between one and six but the expected outcome of each roll is always exactly the same: each number has the same probability of 1/6.



Box 4.2 Figure 2 Partial results of maze simulations ran by a group of 24 students. The histograms after six and 30 iterations are distinctly different and reflect, on average, the results of the corresponding Markov model. No rat was found in hexagon 8.

4.2.4 Randomness in Dynamical Systems

Random walks, probability trees, and Markov models, as well as binomial and Poisson processes, fall into the huge field of stochastic processes, which represent phenomena that contain random components [17]. The possibilities of combining randomness with deterministic systems analysis are unlimited, and every time we develop a deterministic model, we should ask whether we should account for stochasticity, which almost always affects biological systems. There are no general guidelines or answers to this question, and sound judgment is the best advice. As discussed in Chapter 2, the goals and purposes of the model, as well as the technical difficulties associated with each candidate method, should govern this thought process. In favor of stochastic models is the observation that experimental data always contain noise, which corresponds to some randomness and that some (many? or indeed all?) biological processes have features that make them unpredictable. However, there is also much to be said in favor of deterministic models. First and foremost, fully deterministic systems are usually much easier to analyze. Second, if noise is superimposed on some otherwise rather smooth dynamical process, subtraction of the noise may eliminate mathematical distraction and allow us to home in on the truly important determinants of the process.

Toward the end of Chapter 5, we will discuss systems that are chaotic and appear to be driven by randomness, although they are entirely deterministic. In these cases, the immediate future is predictable, but details of the long-term dynamics cannot be foreseen. The stochastic systems have the opposite property: we can compute heir long-term average behavior but not the next step.

4.3 EXERCISES

4.1 Describe in words, as well as mathematically, what happens if the initial size P_t in the recursive model $P_{t+\tau} = 2P_t$ in (4.1) is different from ten.

4.2 What happens (biologically and mathematically) in (4.1) at time points between t and $t + \tau$ or between $t + 3\tau$ and $t + 4\tau$?

4.3 Is it important to distinguish whether or not the bacterial culture in the text (Table 4.1) is synchronized? What happens if τ is slightly different for each bacterium?

4.4 Is anything special about the number 2 in (4.1)–(4.3)? What does it mean if 2 is replaced by some other integer, such as 3 or 4? What about 1?

4.5 Is it mathematically and/or biologically meaningful if the number 2 in (4.1)–(4.3) is replaced by –2, –1, or 0?

4.6 What is the final size of the population in (4.1)–(4.3)? What is the final size if 2 is replaced with 0.5, 1, or –1? What effect does the initial population size P_1 have on the final size in each case?

4.7 Construct a recursive model that approaches a constant final value (steady state) that is not 0, $+\infty$, or $-\infty$. Is this possible with a linear model? If so, construct an example. If not, explain why. Secondly, allow the model to be nonlinear.

4.8 For the example in (4.4), compute R_{n+1} from R_n and then R_{n+2} from R_{n+1} for different values of f, g, R_n, and M_n.

4.9 For the example in (4.4), compute R_{n+2} directly from R_n, using the matrix formulation, and compare the results with those in Exercise 4.8. Compute M_{n+4}.

4.10 What happens with the logistic map (4.14) if r is greater than 4? What happens if r is less than 0?

4.11 Simulate the so-called Gauss map $P_{t+1} = \exp(a \cdot P_t^{\,2}) + b$. Start with the values $a = -6.2$, $b = -1$, $P_0 = 0.2$. Change P_0 and record what happens. Increase b in small steps (e.g., 0.1) toward 1. Does the system have a steady state? Can it be computed algebraically?

4.12 Explore the discrete strange attractor proposed by Hénon, which is defined recursively as

$$x_{n+1} = y_n - 1.4x_n^2 + 1$$
$$y_{n+1} = 0.3x_n$$

Typical starting values are $(x_0, y_0) = (1, 1)$. To see the true shape of the chaotic attractor, execute several hundred iterations, for instance, in Excel, and plot y_n against x_n for all iterations.

4.13 Explore the discrete attractor proposed by Hénon, which is defined in Exercise 4.29 after replacing the factor 1.4 with 1.25.

4.14 Compute the steady state(s) of the Hénon map in Exercises 4.12 and 4.13.

4.15 Compute the steady state(s) of the following two recursive systems.

$$\text{A:}\quad \begin{aligned} X_{n+1} &= (1-p)X_n + Y_n \\ Y_{n+1} &= qX_n + 1 \end{aligned} \qquad \text{B:}\quad \begin{aligned} X_{n+1} &= (1-p)X_n + Y_n \\ Y_{n+1} &= qX_n \end{aligned}$$

Discuss steady states and their stability. Confirm your results with simulations.

4.16 Stability of a recursive system

4.16.1 Determine the steady state of the following recursive system and its stability

$$X_{n+1} = \frac{a_1 Y_n}{b_1 + Y_n} - c_1 X_n$$

$$Y_{n+1} = \frac{a_2 X_n}{b_2 + X_n} - c_2 Y_n$$

With parameter values $a_1 = 4$, $b_1 = 3$, $c_1 = 0.4$, $a_2 = 10$, $b_2 = 3$, $c_2 = 0.1$.

4.16.2 Change c_2 to 0.3 and test for stability again.

4.17 Discuss the steady state(s) of the so-called Fibonacci series, which is given by

$a_{n+2} = a_{n+1} + a_n$ and is usually started with $a_1 = 1$ and $a_2 = 1$. Also discuss the steady states of the associated series of ratios $b_{n+1} = a_{n+1}/a_n$.

4.18 Suppose a gambler starts with \$500, each game adds or subtracts \$10 with equal probability, and the gambler decides to stop after winning an additional \$250 or when broke. Implement the game with 100 iterations in Excel or some other software and simulate it at least 20 times. How often does the gambler win or lose? How often does the gambler reach neither \$750 nor \$0?

4.19 How do the results of the simulation in Exercise 4.18 change if \$50 are wagered in each game? Make predictions before you run the simulations.

4.20 How would you interpret a situation where p and q are different? How does the outcome of the gambler's ruin model change in this case, for instance if $p = 0.49$ or $p = 0.3$?

4.21 What is the behavior of a Markov chain model in which all matrix elements are the same? What is the behavior if all matrix elements on the diagonal are 1?

4.22 Describe the dynamics of a Markov model with the matrix

$$M = \begin{pmatrix} 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 \end{pmatrix}.$$

4.23 Do the matrix elements in each column of a Markov matrix necessarily sum to 1? Why or why not?

4.24 We assumed that the travel times between states in the pinball example are all the same. Discuss the implications of this assumption for setting up a Markov model.

4.25 Compute $\begin{pmatrix} X_1 & X_2 & X_3 \end{pmatrix}_4$ for the example in Eqns. (4.36)–(4.39) in two ways. First, multiply $\begin{pmatrix} X_1 & X_2 & X_3 \end{pmatrix}_3$ to **M**. Second, multiply $\begin{pmatrix} X_1 & X_2 & X_3 \end{pmatrix}_2$ to the square of **M**. Are the two results identical? Explain why or why not.

4.26 Compute the limiting distribution for the example in (4.36)–(4.39) from equation (4.47) with methods of linear algebra.

4.27 Does the limiting distribution contain the same amount of information as **M**? Explain.

4.28 Formulate the gambler's ruin model with absorbing states as a Markov matrix. What would be a complication of this task if the random walk did not have absorbing states?

4.29 Formulate as a Markov matrix a gambler's ruin model with absorbing states, in which the gambler at each iteration wins with a probability of p, loses with a probability of q, or stays at the same amount with probability r. What constraints do p, q, and r have to satisfy?

4.30 How would the setup of the rat race example in Box 4.1 change if the rat did not explore the walls without exits? Instead, it would look around and in the next step use one of the exits.

4.31 Which observations in the maze project are most closely related to the results of computing the Markov process with the Markov matrix?

4.32 What issues would you have to address if you ran the simulation in Box 4.2 for 60 iterations? How could you potentially address these issues?

4.33 How would the model setup change if the rat learned features about the maze and did not move randomly anymore?

4.34 The stochastic movement of molecular motor proteins such as kinesin may be described as a Markov process. These proteins bind to microtubules and hydrolyze ATP in order to move vesicular cargo throughout the cell. The process is nicely captured by Markov models because the conformational changes that occur are largely discrete. For example, Figure 4.8 below depicts the "hand-over-hand" paradigm of kinesin movement. The model has four states (unattached, attached, ATP-bound, second-head-bound) and five transitions among them, of which the last is the conformational shift that occurs with ATP hydrolysis and allows the kinesin to "walk" along a microtubule with 8 nm steps for each ATP molecule used.

Use the following probabilities to develop a Markov matrix for this system:

$P(\text{Attachment}) = 0.0001$
$P(\text{ATP binding}) = 0.055$
$P(\text{ATP hydrolysis}) = 0.25$
$P(\text{Conformational change}) = 0.6$
$P(\text{Detachment}) = 0.0001$

4.34.1 Initializing the system as entirely detached, what is the stationary state of the model? How do you interpret this state biologically?

4.34.2 Does the stationary distribution change if you start with the assumption of equal proportions in each state? Why or why not?

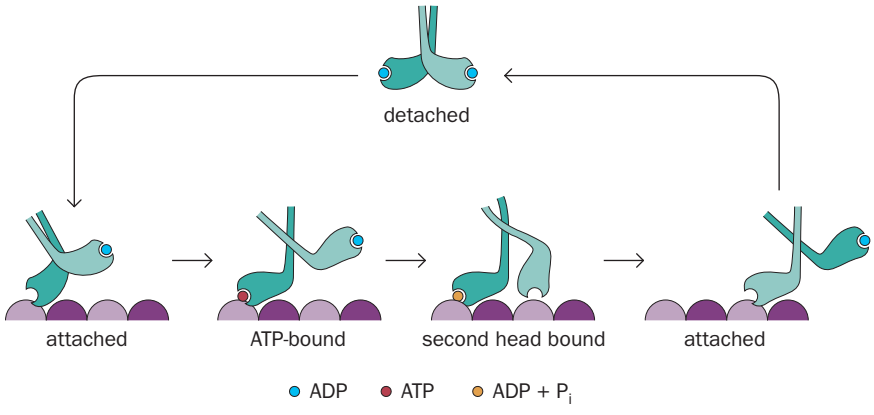


Figure 4.8 Processes involved in the stochastic movement of molecular motor proteins. (Adapted from Schief WR & Howard J. *Curr Opin Cell Biol.* 13 [2001] 19–28. With permission from Elsevier.)

FURTHER READING

Easy Introductions

Batschelet E. *Introduction to Mathematics for Life Scientists*, 3rd ed. Springer, New York, NY, 1979.
Edelstein-Keshet L. *Mathematical Models in Biology*. SIAM, Philadelphia, PA, 2005.
Voit EO. *The Inner Workings of Life. Vignettes in Systems Biology,* xii + 210 pp. Cambridge University Press, Cambridge, UK, 2016.
Voit EO. *Systems Biology*. *A Very Short Introduction*. Oxford University Press, Oxford, UK, 2020.

General Texts on Mathematical Biology

Adler FR. *Modeling the Dynamics of Life: Calculus and Probability for Life Scientists*, 3rd ed. Brooks/Cole, Totnes, UK, 2012.
Beltrami E. *Mathematics for Dynamic Modeling*, 2nd ed. Academic Press, Cambridge, MA, 1998.
Britton NF. *Essential Mathematical Biology*. Springer, New York, NY, 2004.
Haefner JW. *Modeling Biological Systems: Principles and Applications*, 2nd ed. Chapman & Hall, New York, NY, 2005.
Keener J & Sneyd J. *Mathematical Physiology*. Springer, New York, NY, 1998.
Klipp E, Herwig R, Kowald A, Wierling C & Lehrach H. *Systems Biology in Practice: Concepts, Implementation and Application*. Wiley-VCH, Weinheim, Germany, 2005.
Kremling A. *Systems Biology. Mathematical Modeling and Model Analysis*. Chapman & Hall, CRC Press, New York, NY, 2013.
Kreyszig E. *Advanced Engineering Mathematics*, 10th ed. Wiley, Hoboken, NJ, 2011.
Murray JD. *Mathematical Biology I: Introduction*, 3rd ed. Springer, New York, NY, 2002.
Murray JD. *Mathematical Biology II: Spatial Models and Biomedical Applications*. Springer, New York, NY, 2003.
Strang G. *Introduction to Applied Mathematics*. Wellesley-Cambridge Press, Wellesley, MA, 1986.
Strogatz SH. *Nonlinear Dynamics and Chaos*. Perseus Books, New York, NY, 1994.
Yeargers EK, Shonkwiler RW & Herod JV. *An Introduction to the Mathematics of Biology*. Birkhäuser, Basel, Switzerland, 1996.

Stochastic Processes and Techniques

Pinsky M & Karlin S. *An Introduction to Stochastic Modeling*, 4th ed. Academic Press, Cambridge, MA, 2011.
Ross SM. *Introduction to Probability Models*, 10th ed. Academic Press, Cambridge, MA, 2010.
Wilkinson DJ. *Stochastic Modelling for Systems Biology. Mathematical and Computational Biology*, 2nd ed. Chapman & Hall, CRC Press, New York, NY, 2011.

REFERENCES

1. Voit, E.O. and G. Dick, Growth of cell populations with arbitrarily distributed cycle durations; I. Basic model. *Mathem Biosci*, 1983. **66**: p. 229–246.
2. Voit, E.O. and G. Dick, Growth of cell populations with arbitrarily distributed cycle durations; II. Extended model for correlated cycle durations of mother and daughter cells. *Mathem Biosci*, 1983. **66**: p. 247–262.
3. Leslie, P.H., On the use of matrices in certain population mathematics. *Biometrika*, 1945. **33**: p. 183–212.
4. Edelstein-Keshet, L., *Mathematical Models in Biology*. 1988, New York: McGraw Hill.
5. Fonseca, L.L., et al., Quantifying the removal of red blood cells in Macaca mulatta during a *Plasmodium coatneyi* infection. *Malaria J*, 15, 2016. **15**: p. 410.
6. Fonseca, L.L. and E.O. Voit, Comparison of mathematical frameworks for modeling erythropoiesis in the context of malaria infection. *Math Biosc*, 2015. **270**: p. 224–236.
7. May, R.M., Simple mathematical models with very complicated dynamics. *Nature*, 1976. **261**(5560): p. 459–467.
8. Thompson, J.M.T. and H.B. Stewart, *Nonlinear Dynamics and Chaos* 1986, Chichester: John Wiley and Sons.
9. Fractal-Foundation, *Fractal Pack 1: Educators' Guide*. Fractals are smart: Science, Math & Art!, 2009, https://fractalfoundation.org/fractivities/FractalPacks-EducatorsGuide.pdf.
10. Strang, G., *Introduction to Applied Mathematics*. 1986, Wellesley, MA: Wellesley-Cambridge Press.
11. Pólya, G., Über eine Aufgabe der Wahrscheinlichkeitsrechnung betreffend die Irrfahrt im Strassennetz. *Math Ann*, 1921. **84**(1): p. 149–160.
12. Lockley, T.C., *IPM World Textbook*, https://ipmworld.umn.edu/lockley, 2004–2007.
13. Garimella, U. and N.S., Inheritance patterns: Probability rules & probability trees. *The American Biology Teacher*, 2022. **84**(1): p. 22–27.
14. Nijenhuis-Voogt, J., S. Yeni, and E. Barendsen. Integrating CT into biology: Using decision tree models to classify cell types, in *CTE-STEM 2022 Conference*, 2022, Technical University Delft: Delft, the Netherlands, https://doi.org/10.34641/ctestem.2022.460.
15. 6-Sigma_Study_Guide, *What is Fault Tree Analysis?* https://sixsigmastudyguide.com/fault-tree-analysis/, accessed February 2023.
16. Ruijters, E. and S. Mariëlle, Fault tree analysis: A survey of the state-of-the-art in modeling, analysis and tools. *Comp Sci Rev*, 2015. **15–16**: p. 29–62.
17. Ross, S.M., *Introduction to Probability Models*. 3rd ed. 1985, Orlando, FL: Academic Press Inc.
18. Durbin, R., et al., *Biological Sequence Analysis*. 1998, Cambridge, UK: Cambridge University Press.

Taylor & Francis
Taylor & Francis Group
http://taylorandfrancis.com

Continuous Biological Systems

5

When you have read this chapter, you should be able to:

- Explain the difference between discrete and continuous systems
- Understand the reasons for using ordinary differential equations (ODEs)
- Describe the components of ODE systems
- Rationalize the need for approximations
- Linearize nonlinear systems of ODEs
- Approximate nonlinear systems with power-law functions
- Design and analyze basic models with one or more variables in the formats of:
  - Continuous linear systems
  - Continuous nonlinear systems
- Execute the steps of a typical biological systems analysis
- Perform local stability analysis
- Understand the concepts of sensitivity analysis
- Describe different types of attractors and repellers

Chapter 4 described general concepts and analyses for modeling biological systems that operate on a discrete time scale. This chapter presents mathematical and computational methods for analyzing dynamic systems that are defined on a continuous time scale. All analyses shown in this chapter can be executed with a wide variety of computer languages, as well as higher-level software like *Mathematica*® and MATLAB®. A particularly easy-to-learn software designed especially for analyzing **ordinary differential equation (ODE)** systems is PLAS [1]. On occasion, the material in this chapter requires basic working knowledge of ordinary and partial **differentiation**, **vectors**, and **matrices**.

Like the previous chapter, this chapter first discusses **linear** and then **nonlinear systems**. The latter often capture biomedical phenomena much better than the former. However, the former are of utmost importance for two reasons. First, they permit more mathematical analyses than any other format and, second, it may be surprising, but it is possible to analyze many features of nonlinear systems with methods of linear analysis.

A very interesting feature germane to linear systems is the principle of **superposition**. This principle says the following. If we analyze a system by giving it an input I_1, to which it responds with output O_1, and if we do a second experiment

DOI: 10.1201/9781003402770-5

with input I_2 and obtain output O_2, then we know that the output response to a combined input $I_1 + I_2$ will be $O_1 + O_2$. Maybe this superposition sounds obvious, trivial, or unassuming, but it has enormous consequences. Only if this principle holds is it possible to analyze the behavior of a system by studying each component separately and then validly summing the individual results. As a trivial example, suppose we determine the weight of 30 pennies to be 75 grams. We also weigh 20 dimes, obtaining 45.4 grams. We can now predict with great confidence what chump change of 10 pennies and 10 dimes will weigh, namely (25 + 22.7) grams = 47.7 grams. The reason why we have not chosen a biological example here is that most biological systems are indeed not linear, thereby violating the principle of superposition. For instance, the growth of a plant depends on genetics as well as environmental conditions, including nutrients. However, the two are clearly not independent of each other, and separate experiments with different strains and different experimental conditions may not allow us to make valid predictions of how a particular strain will grow under untested conditions.

If we can validly represent a system in a linear format, an incredible repertoire of mathematical and computational tools becomes available. In particular, the entire field of linear algebra addresses important aspects of these systems and concepts such as vectors and matrices have become crucial tools for the analysis of ODEs. For instance, if the right-hand sides of an ODE system are linear, the computation of the steady states of this system becomes a matter of solving a system of linear algebraic equations, for which there are many methods.

Linear representations are crucially important for biology, even though most systems in biology are nonlinear. The reason is that many nonlinear systems analyses can be performed once a system is **linearized**, which essentially means that one "flattens" the nonlinear system close to a point of interest. The justification for this strategy comes from a powerful theorem established by Hartman and Grobman [2–6]. It says that, under most pertinent conditions, important clues for the behavior of a nonlinear system can be deduced from the linearized system. We will make use of this insight when we discuss stability and sensitivities.

5.1 LINEAR AND LINEARIZED ODE SYSTEMS

5.1.1 Introductory Examples

ODE models may be linear or nonlinear. As in the case of discrete systems, one should note that even if the right-hand sides of a set of differential equations are linear, the solutions to these differential equations are combinations of linear as well as exponential and trigonometric functions, which are clearly nonlinear.

The simplest example of a linear differential equation is

$$\frac{dX}{dt} = \dot{X} = a \cdot X. \tag{5.1}$$

Expressed in words from left to right, the equation says that change, as expressed by the derivative on the left-hand side, is directly proportional to the current size or amount of the quantity X, with the proportionality constant a. The notation $\dot{X}$ is short-hand and exactly the same as $\frac{dX}{dt}$. Note that we write X rather than $X(t)$, even though X depends on time, and that time t is no longer visible in $\dot{X}$ and on the right-hand side. This omission is a convention of simplicity because it is implicitly understood that X is a function of time. If X and a are positive, the change is positive, and X continues to grow. If a is negative, X keeps decreasing. What is the output of the equation (5.1) if $a = 0$?

The simple example of (5.1) is used frequently in biology. If a is positive, it describes **exponential growth**, and if a is negative, it represents **exponential decay**. In addition, the equation is used for transport processes in biochemical and physiological systems and for physical phenomena such as heating and cooling. We will return to this important equation in Chapter 9.

The theory of ODEs tells us that it takes infinitely long for an exponential decay process to reach its final value: zero. Therefore, processes where X_0 is positive and a is negative are often characterized by their **half-life**, which is the time it takes for half of the initial amount of material to be lost (for an application, see Chapter 13).

Because of the simplicity of (5.1), we can compute the solution of the differential equation by hand. Here, "solution" means that we figure out explicitly which function describes the dependence of X on time. In this particular case, the solution consists of the exponential function $X(t) = X_0 e^{at}$. We can confirm this statement if we differentiate this exponential function, which yields $\frac{dX}{dt} = a\, X_0\, e^{at}$. Clearly, the result equals $a \cdot X(t)$ and thus the right-hand side of (5.1), which proves that (5.1) has as its solution an exponential function. Note that the quantity X_0 is part of the time-dependent solution, but not of the differential equation. Nevertheless, X_0 has a clearly defined meaning for the ODE as the **initial value** of X at time $t = 0$. This is again easy to confirm: We just substitute t with 0 in $X(t) = X_0 e^{at}$, which directly returns the result $X(0) = X_0$. This last statement is often added to the definition of the differential equation in (5.1):

$$\frac{dX}{dt} = \dot{X} = a \cdot X,\ X(0) = X_0. \tag{5.2}$$

Like the recursive growth process in Chapter 4, the differential equation with a positive initial value X_0 either goes to ∞ (positive a) or zero (negative a). If X_0 is negative, the solution can go to $-\infty$ or zero, depending on a. Again, what happens for $a = 0$ if the initial value X_0 is positive, negative, or zero?

A well-known example of a linear "system" is the sine-cosine oscillator. It can be represented as a pair of linear differential equations, namely

$$\begin{aligned} \frac{dx}{dt} &= y & x(0) &= 0 \\ \frac{dy}{dt} &= -x & y(0) &= 1. \end{aligned} \tag{5.3}$$

First, let's convince ourselves that these two equations really correspond to a sine oscillator. We know that x and y are functions of time and try $x(t) = \sin(t)$ and $y(t) = \cos(t)$. The derivative of $\sin(t)$ is $\cos(t)$, thus satisfying the first equation, while the derivative of $\cos(t)$ is $-\sin(t)$, thereby satisfying the second equation. Furthermore, for $t = 0$ the sine function is zero and the cosine is 1. Put the ODE system (5.3) into a numerical solver and, voilà, sine and cosine oscillations emerge. We will return to this system later.

Even for slightly more complicated ODEs, an explicit solution like the exponential or sine functions can often not be obtained by hand and indeed may not even exist. A solution can only be obtained with a computer algorithm. But how can a computer determine something that does not exist? The answer is that the algorithm computes a very good numerical approximation. Specifically, it computes values of $X(t)$ that quite accurately satisfy the ODE, but this collection of values does not have a name like "exponential function" or "sine function"; it is just a collection of numerical values. Many of such "integration" algorithms are available for solving ODEs [7]. The oldest and simplest is the Euler method. For example, to solve (5.3), it would start with the initial value $X(0)$, and compute $a \cdot X(0)$. Exactly at $t = 0$, this term is the correct slope of the unknown function to be computed. Thus, the computer inches forward with this slope, but the farther it moves away from 0, the more $a \cdot X(0)$ differs from the true value $a \cdot X(t)$. Therefore, the computer takes very small steps, of a length Δt, which may be 10^{-8} or so. At the end of the first step, it obtains an approximate value $\tilde{X}(\Delta t)$, which is slightly different from the true $X(\Delta t)$. This difference is the cost of using an approximation, but we have no real alternative. Now the computer computes $a \cdot \tilde{X}(\Delta t)$ and inches forward again. After thousands, if not millions of steps of this type, it has completed the collection of values that (approximately) satisfy the ODE in integrated form.

5.1.2 Systems of Linear ODEs

Systems of linear differential equations are natural extensions of (5.1)–(5.3). They have the same general form as (5.2), but **X** is now a vector and **A** is a matrix with **coefficients** *Aij*:

$$\frac{d\mathbf{X}}{dt} = \dot{\mathbf{X}} = \mathbf{A}\cdot\mathbf{X}, \quad \mathbf{X}(0) = \mathbf{X}_0. \tag{5.4}$$

Uncounted examples of this type of linear system may be found in about all branches of engineering. An important biological example is a physiological **compartment model**, in which nutrients, drugs, or toxic substances can move between the bloodstream and the organs and tissues in the body (for details see, for example, [8] and Chapter 13).

The main feature of systems like (5.4) is that they allow us to compute explicit solutions in the form of time-dependent functions, as for the simple examples (5.1) and (5.3). This fact is a stark difference to nonlinear systems, where solutions almost always require numerical simulations (see later). If **X** in (5.4) has n elements, the solutions consist of n time-dependent functions $X_i(t)$. These solutions are determined by the **eigenvalues** of the matrix **A**, which may be real or complex numbers that characterize **A**. For small systems they can be computed algebraically, and for larger systems there are numerous computational methods and software programs.

Assuming that the eigenvalues of **A** are all different, the solution of (5.4) consists of a sum of exponential functions whose **rate constants** equal the eigenvalues of the system matrix. However, because these exponential functions may have complex arguments, the possible solutions also include trigonometric functions, such as sine and cosine.

For a linear system with three equations of the format

$$\frac{d}{dt}\begin{bmatrix} X_1 \\ X_2 \\ X_3 \end{bmatrix} = \boldsymbol{A}\begin{bmatrix} X_1 \\ X_2 \\ X_3 \end{bmatrix} \qquad \begin{bmatrix} X_1 \\ X_2 \\ X_3 \end{bmatrix}(t_0) = \begin{bmatrix} X_{10} \\ X_{20} \\ X_{30} \end{bmatrix} \tag{5.5}$$

the solution is given as

$$\begin{bmatrix} X_1(t) \\ X_2(t) \\ X_3(t) \end{bmatrix} = c_1 e^{\lambda_1 t}\, EV_1 + c_2 e^{\lambda_2 t}\, EV_2 + c_3 e^{\lambda_3 t}\, EV_3. \tag{5.6}$$

The exponents contain the quantities λ_1, λ_2, and λ_3, which are the eigenvalues of **A**; these may be real or complex numbers. EV_1, EV_2, and EV_3 are the corresponding eigenvectors, and the coefficients c_1, c_2, and c_3 are real numbers. Eqns. (5.1)–(5.3) were simple special cases with one or two variables, respectively. The single eigenvalue for (5.1) equals a, whereas the eigenvalues for (5.3) are purely imaginary numbers, namely $\lambda_1 = +i$ and $\lambda_2 = -i$, where $i = \sqrt{-1}$.

Linear ODE systems may contain additional input functions, which make the solutions more complicated. Nonetheless, many methods are available for studying linear differential equation systems, including the so-called Laplace transform, which permits analyses without the need to integrate the differential equations [9]. Fourier transforms may be used to solve certain types of differential equations [10].

5.1.3 Linearized ODE Models

Someone once said that the distinction between linear and nonlinear models is like a classification of the animal kingdom into elephants and non-elephants. It is

true: there is essentially only one linear structure, but there are uncounted—and in fact infinitely many—nonlinear structures. Nevertheless, **linearization** is of great importance because a nonlinear system and its linear counterpart behave very similarly close to a chosen **operating point (OP)** in many realistic situations. Linearization yields a linear system exactly like (5.4), and all methods applicable to linear systems hold for linearized nonlinear systems as well. In particular, linearization allows us to assess important features of a nonlinear system, such as the stability of a steady state and parameter sensitivities (see later section).

The overall strategy of linearization, which is a type of approximation, is to flatten the nonlinear system within a small region around the operating point. This point can be chosen arbitrarily, as long as it is possible to compute derivatives of the nonlinear system at this point. In practical investigations, *OP* is usually selected as a steady state, and we will not consider any other points here.

The image in **Figure 5.1** visualizes the linearization of a curved surface at some point *P*. The linearized system operates on the tangent plane and has exactly the same value as the curved surface and also the same slopes at *P*. At other points, the two diverge. Importantly, all paths C_i on the colored surface that go through *P* have the same local directions T_i in the tangent plane, and many analyses can be validly done in this tangent plane rather than on the curved surface. These features hold for any dimension but are of course difficult to visualize for more than two variables.

As an illustration, suppose a system has the form

$$\frac{dX}{dt} = f(X,Y)$$

$$\frac{dY}{dt} = g(X,Y), \tag{5.7}$$

where *f* and *g* are nonlinear functions. The first order of business is the selection of an operating point, where we anchor the linearization. As stated before, we do have choices but, in this book, always take this point as a steady-state point of the system. The main ingredient of the linearization process is differentiation, and because the system has two independent variables, *X* and *Y*, the derivatives must be partial derivatives. These are computed like regular derivatives under the assumption that the other variable is kept constant. The linearization is based on a famous theorem of the English mathematician Brook Taylor (1685–1731), who discovered that a function may be approximated with a polynomial, which in the simplest case is a linear function (see **Box 5.1**).

For our demonstration, we approximate *f* and *g* in the vicinity of a steady-state operating point, (X_{OP}, Y_{OP}). We denote a solution in the vicinity of the steady-state operating point as $[X_{OP} + X'(t), Y_{OP} + Y'(t)]$, so that $X'(t)$ and $Y'(t)$ are deviations in the *X* and *Y* directions. Because we intend to stay close to the operating point, this solution is called a **perturbation**. The strategy is now to formulate this

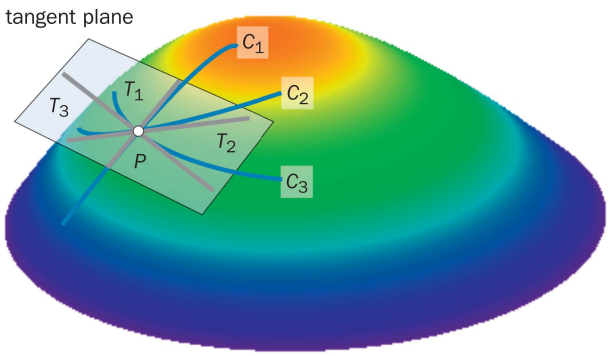


Figure 5.1 Illustration of the linearization of a curved surface. At *P*, the tangent plane and the curved surface touch and have the same slopes.

perturbation as a new system of differential equations. Plugging the perturbed values into (5.7) directly yields:

$$\begin{aligned}\frac{d}{dt}\left(X_{OP}+X'\right)&=f\left(X_{OP}+X',Y_{OP}+Y'\right)\\ \frac{d}{dt}\left(Y_{OP}+Y'\right)&=g\left(X_{OP}+X',Y_{OP}+Y'\right)\end{aligned}\tag{5.8}$$

where the argument (t) is again not shown.

To obtain the linearization, we apply Taylor's insight (**Boxes 5.1** and **5.2**) to the right-hand sides of this equation and obtain the linear approximation

$$f\left(X_{OP}+X',Y_{OP}+Y'\right)\approx f\left(X_{OP},Y_{OP}\right)+\left.\frac{\partial f}{\partial X}\right|_{(X_{OP},Y_{OP})}\cdot X'+\left.\frac{\partial f}{\partial Y}\right|_{(X_{OP},Y_{OP})}\cdot Y'\tag{5.9a}$$

$$g\left(X_{OP}+X',Y_{OP}+Y'\right)\approx g\left(X_{OP},Y_{OP}\right)+\left.\frac{\partial g}{\partial X}\right|_{(X_{OP},Y_{OP})}\cdot X'+\left.\frac{\partial g}{\partial Y}\right|_{(X_{OP},Y_{OP})}\cdot Y'\tag{5.9b}$$

Note that the right-hand sides are approximations of the left-hand sides because all higher derivatives are omitted. The curly ∂ signifies partial differentiation and the vertical lines with subscripts indicate that this differentiation is evaluated at the operating point. Because the operating point is a steady state, we know that $dX/dt = dY/dt = 0$. Thus, the terms $f(X_{OP}, Y_{OP})$ and $g(X_{OP}, Y_{OP})$ equal 0 and drop out. This result can be seen directly from (5.7). because at the steady state we obtain

$$\begin{aligned}0&=\frac{dX}{dt}=f(X,Y),\\ 0&=\frac{dY}{dt}=g(X,Y).\end{aligned}\tag{5.10}$$

We can gain some intuition about equations (5.9a,b) by dissecting them into recognizable components (**Figure 5.2**). Expressed in words, the function f (or g) can be approximated at some point by using the value at this point and adding to it deviations in the directions of the dependent variables. Each deviation consists of the slope in that direction (expressed by the partial derivative) and the distance of the deviation along the corresponding axis (here X' and Y', respectively). This method works for any number of dependent variables.

The left-hand sides of (5.9a, b) can be simplified as well. Namely, we are allowed to split the differential of each sum into a sum of two differentials. Furthermore, $dX_{OP}/dt = dY_{OP}/dt = 0$, because X_{OP} and Y_{OP} are constant values. Thus, only dX'/dt

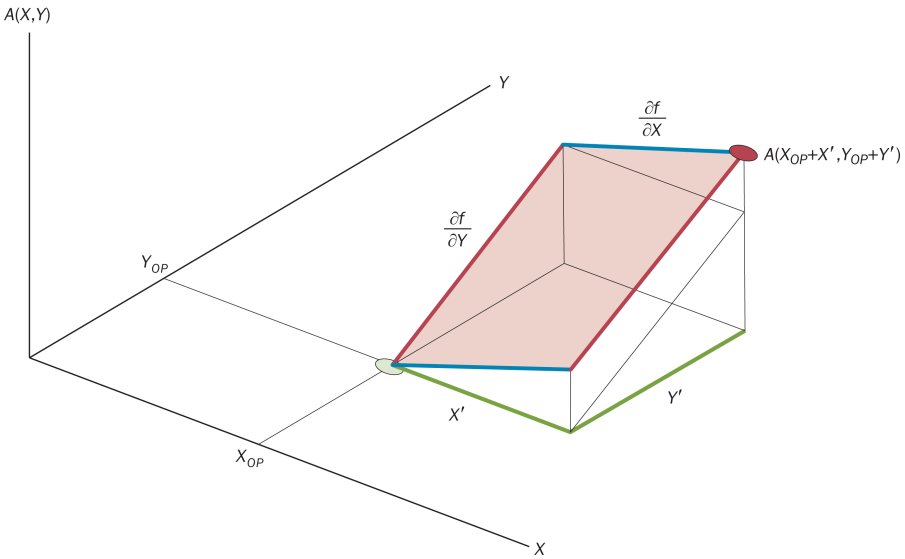


Figure 5.2 Visualization of part of a plane A(X,Y) (pink) approximating some function f(X,Y), which is not shown, at some operating point (X_{OP} and Y_{OP}) (green). The plane is constructed according to (5.9a). The blue lines have the slope of f at the operating point in the X-direction, while the red lines have the slope of f at the operating point in the Y-direction. The green lines indicate distances in the two directions from the operating point (X' and Y', respectively). The point $A(X_{OP} + X', Y_{OP} + Y')$ (red) lies on the plane but is only approximately equal to $f(X_{OP} + X', Y_{OP} + Y')$ (not shown). The two are equivalent for $X' = Y' = 0$ and similar for small values of X' and Y'.

and dY'/dt are left. Taking the simplifications of the two sides together, the statements in (5.8) and (5.9) reduce to

$$\frac{d}{dt}X' \approx \left.\frac{\partial f}{\partial X}\right|_{(X_{OP},Y_{OP})} \cdot X' + \left.\frac{\partial f}{\partial Y}\right|_{(X_{OP},Y_{OP})} \cdot Y', \tag{5.11a}$$

$$\frac{d}{dt}Y' \approx \left.\frac{\partial g}{\partial X}\right|_{(X_{OP},Y_{OP})} \cdot X' + \left.\frac{\partial g}{\partial Y}\right|_{(X_{OP},Y_{OP})} \cdot Y'. \tag{5.11b}$$

Note that this linearization may be a rather crude approximation if f and g are strongly curved. But the approximation may also be very good for moderate deviations from the operating point. Without further information about f and g, we just don't know, even for rather large perturbations. Theory assures us though that the linearized system is a good representation of the original nonlinear system if the perturbation is sufficiently small (whatever that means in a specific situation). A second important result is that the right-hand sides are linear.

Let's rename the partial derivatives in (5.11), evaluated at the operating point, as

$$\begin{aligned} a_{11} &= \left.\frac{\partial f}{\partial X}\right|_{(X_{OP},Y_{OP})} \qquad a_{12} = \left.\frac{\partial f}{\partial Y}\right|_{(X_{OP},Y_{OP})} \\ a_{21} &= \left.\frac{\partial g}{\partial X}\right|_{(X_{OP},Y_{OP})} \qquad a_{22} = \left.\frac{\partial g}{\partial Y}\right|_{(X_{OP},Y_{OP})} \end{aligned} \tag{5.12}$$

which presents us the linearization of (5.11) as

$$\begin{aligned} \frac{dX'}{dt} &= a_{11}X' + a_{12}Y', \\ \frac{dY'}{dt} &= a_{21}X' + a_{22}Y'. \end{aligned} \tag{5.13}$$

In general, we may write the linearization of a system with n variables in matrix form as

$$d\begin{pmatrix} X_1' \\ X_2' \\ \vdots \\ X_n' \end{pmatrix} / dt = \left.\begin{pmatrix} \frac{\partial f_1}{\partial X_1} & \frac{\partial f_1}{\partial X_2} & \cdots & \frac{\partial f_1}{\partial X_n} \\ \frac{\partial f_2}{\partial X_1} & \frac{\partial f_2}{\partial X_2} & \cdots & \frac{\partial f_2}{\partial X_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial f_n}{\partial X_1} & \frac{\partial f_n}{\partial X_2} & \cdots & \frac{\partial f_n}{\partial X_n} \end{pmatrix}\right|_{OP} \cdot \begin{pmatrix} X_1' \\ X_2' \\ \vdots \\ X_n' \end{pmatrix} \tag{5.14}$$

The matrix on the right-hand side is again the **Jacobian** of the system (evaluated at the operating point OP) that we had already encountered in Chapter 4.

The linearized system describes approximately the distance of the nonlinear system from its steady-state operating point. Specifically, in our case,

$$\begin{aligned} X'(t) &\approx X(t) - X_{OP}, \\ Y'(t) &\approx Y(t) - Y_{OP}. \end{aligned} \tag{5.15}$$

Close to the operating point, the linear representation is very good, but moving away from this point, the approximation error typically becomes larger. Important investigations into the stability and sensitivity of linear and nonlinear systems are executed in the immediate vicinity of the steady state, so that the accuracy of the approximation actually becomes secondary for these purposes (see later).

BOX 5.1: APPROXIMATION AND LINEARIZATION

Many approximations in mathematics are based on an old theorem by the English mathematician Brook Taylor (1685–1731), who showed that any smooth function (one that has sufficiently many continuous derivatives) can be represented as a power series. If this series has infinitely many terms, it is an exact representation of the original function within some region of convergence. However, if the series is truncated, i.e., if one ignores all terms beyond a certain number, the resulting Taylor polynomial approximates the function. This approximation is still exact at a chosen operating point and very similar to the original function close to this operating point. However, for values distant from the operating point, the approximation error may be high or still acceptable; it depends on the particularities of the function. The coefficients of the Taylor polynomial are computed through differentiation of the original function.

If the real-valued function $f(x)$ has at least n continuous derivatives, it is approximated at the operating point p by

$$f(x)\approx f(p)+f^{(1)}(p)(x-p)+\frac{1}{2!}f^{(2)}(p)(x-p)^2 +\frac{1}{3!}f^{(3)}(p)(x-p)^3+\ldots+\frac{1}{n!}f^{(n)}(p)(x-p)^n \quad (1)$$

The expressions 2!, 3!, $n!$ are factorials, which we encountered already in Box 2.3, and $f^{(1)}$, $f^{(2)}$, $f^{(3)}$, and $f^{(n)}$ denote the first, second, third, and n^{th} derivatives of $f(x)$, respectively. It is not difficult to see that the original function f and the Taylor approximation are exactly the same at the operating point p: just substitute p for x.

A very important special case is linearization, where $n = 1$. That is, one keeps merely the constant term and the term with the first derivative and ignores everything else. This strategy sounds overly radical, but it is extremely useful, because the linearity of the result permits many simplified analyses close to the operating point. The result of this linearization is

$$L(x)=f(p)+f^{(1)}(p)(x-p) \quad (2)$$

which has the familiar format $y = mx + b$ of a straight line, with slope m and intercept b (**Figure 1**). Here, m corresponds to the derivative $f^{(1)}(p)$ and the intersect with the vertical axis is $b = f(p) - p\,f^{(1)}(p)$.

As an example, consider the shifted exponential function

$$f(x)=e^x+1 \quad (3)$$

and choose $x_0 = 0$ as the operating point; in this illustration, the operating point is not a steady state. The linearization includes the value of f at x_0, which is 2, and the first derivative, evaluated at the operating point, which is

$$f^{(1)}(0)=e^x=1. \quad (4)$$

Thus, the linearization of f at (x_0) is

$$L(x)=2+1\cdot(x-0)=2+x \quad (5)$$

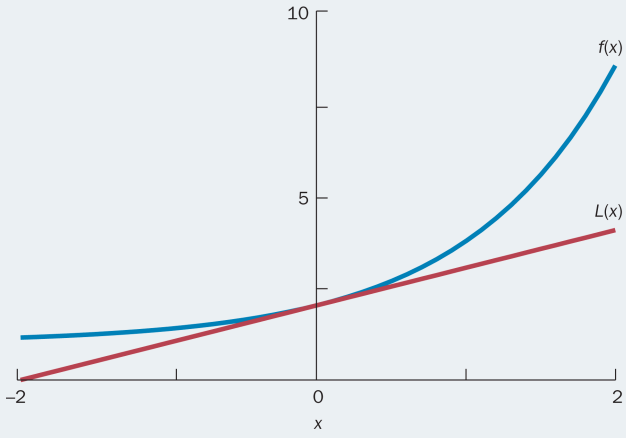


Figure 1 Taylor approximation can be used to linearize a nonlinear function. Here, $L(x)$ is the linear approximation of the shifted exponential function $f(x)$ in (3), developed at the operating point $x_0 = 0$

BOX 5.2: LINEARIZATION OF NONLINEAR FUNCTIONS IN TWO VARIABLES

Intriguingly, an approximation analogous to the Taylor representation in Box 5.1 can be performed in higher dimensions. For the case of two dependent variables, one obtains the approximation

$$A(x,y)=f(x_0,y_0)+(x-x_0)\cdot f_x(x_0,y_0)+(y-y_0)\cdot f_y(x_0,y_0) +\frac{1}{2!}[(x-x_0)^2\cdot f_{xx}(x_0,y_0)+2\cdot(x-x_0)\cdot(y-y_0)\cdot f_{xy}(x_0,y_0)+(y-y_0)^2\cdot f_{yy}(x_0,y_0)]+\ldots. \quad (1)$$

The first term on the right-hand side is the value of f at the chosen operating point (x_0, y_0), the expressions f_x and f_y are the first derivatives of f with respect to x and y, respectively, and f_{xx}, f_{xy}, and f_{yy} are the corresponding second derivatives of f, which are all evaluated at the operating point. Note that $f_{xy} = f_{yx}$. The dots . . . indicate that all higher derivatives would follow but are omitted.

Ignoring the second and all higher derivatives, we obtain the linearization $L(x, y)$ of $f(x, y)$. Specifically, we can write

$$L(x,y)=f(x_0,y_0)+(x-x_0)\cdot f_x(x_0,y_0)+(y-y_0)\cdot f_y(x_0,y_0). \quad (2)$$

At the operating point, L has the same value and the same slopes as f in both the x- and the y-direction.

As an example, consider the function $f(x, y) = y \cdot ex$ and choose the operating point $(x_0, y_0) = (0, 0)$. The linearization includes the value of f at (x_0, y_0), which is 0, and the first derivatives, evaluated at the operating point. These are

$$f_x(0,0)=y\cdot e^x=0\cdot 1=0 \text{ and } f_y(0,0)=e^x=1. \quad (3)$$

Thus, the linearization of f at (x_0, y_0) is

$$L(x,y) = 0 + (x-0)\cdot 0 + (y-0)\cdot 1 = y. \tag{4}$$

The linearization is flat with respect to the x-direction and has a slope of 1 in the y-direction. Some values of f and L are given in **Table 1**. The approximation for small x-values is good, even if y is large, because f is already linear with respect to y. By contrast, large x-values lead to much higher approximation errors, because L does not even depend on x, owing to the choice of $x_0 = 0$ for the operating point, while f increases exponentially with respect to x.

TABLE 1: SELECTED VALUES OF $F(x, y)$ AND $L(x, y)$ FOR DIFFERENT COMBINATIONS OF x AND y.

(x, y)	(0, 0)	(0, 1)	(0.1, 0.1)	(0.1, 1)	(0.2, 0.2)	(0.1, 0.5)	(0.5, 0.1)	(0.5, 0.5)	(1, 0.1)
$f(x, y)$	0	1	0.115	1.105	0.244	0.553	0.165	.824	0.272
$L(x, y)$	0	1	0.1	1	0.2	0.5	0.1	0.5	0.1

As a numerical demonstration of the linearization process, consider the system

$$\begin{aligned} \dot{X} &= f(X,Y) = 2Y - 3X^4 \\ \dot{Y} &= g(X,Y) = 0.5\exp(X)\cdot Y - 2Y^2. \end{aligned} \tag{5.16}$$

The system has the steady state ($X_{ss} \approx 0.776$, $Y_{ss} \approx 0.543$), which we use as our operating point (OP). The values of the partial derivatives, evaluated at OP, are

$$\begin{aligned} a_{11} &= \left.\frac{\partial f}{\partial X}\right|_{(X_{OP},Y_{OP})} = \left. -12X^3\right|_{(X_{OP},Y_{OP})} = -5.608 \\ a_{12} &= \left.\frac{\partial f}{\partial Y}\right|_{(X_{OP},Y_{OP})} = 2 \\ a_{21} &= \left.\frac{\partial g}{\partial X}\right|_{(X_{OP},Y_{OP})} = \left. 0.5\exp(X)\cdot Y\right|_{(X_{OP},Y_{OP})} = 0.590 \\ a_{22} &= \left.\frac{\partial g}{\partial Y}\right|_{(X_{OP},Y_{OP})} = \left. 0.5\exp(X) - 4Y\right|_{(X_{OP},Y_{OP})} = -1.086 \end{aligned} \tag{5.17}$$

The linearization is therefore

$$\begin{aligned} \frac{dX'}{dt} &= -5.608X' + 2Y', \\ \frac{dY'}{dt} &= 0.590X' - 1.086Y'. \end{aligned} \tag{5.18}$$

The variables X' and Y' are functions of t that describe approximately how far $X(t)$ and $Y(t)$ deviate from the steady state ($X_{SS} = 0.776$, $Y_{SS} = 0543$).

Suppose the original nonlinear system uses as initial values $X_0 = X_{SS} + 0.75$ and $Y_0 = Y_{SS} - 0.2$. The linearized system (5.18) may start with any initial values, of course, but for a direct comparison with the original nonlinear system we choose the differences between the initial values and the steady state of the nonlinear system, namely 0.75 and –0.2. We solve this ODE system and then shift the solution to the operating point: $\tilde{X}(t) = X'(t) + X_{SS}$, $\tilde{Y}(t) = Y'(t) + Y_{SS}$. **Figure 5.3** shows that $\tilde{X}$ and $\tilde{Y}$ approximate X and Y very well close to the operating point, which equals the steady state. At the beginning of the simulation, where X and Y are further away from the steady state, the approximation of X is not quite as good.

As a second example, recall the SIR model (Eqn. (2.6)) that we analyzed in Chapter 2 for the spread of infectious diseases. It has the form

$$\begin{aligned} \dot{S} &= 3 + 0.01\cdot R - 0.0005\cdot S\cdot I & S_0 &= 170 = S_{SS} + 30 \\ \dot{I} &= 0.0005\cdot S\cdot I - 0.05\cdot I - 0.02\cdot I & I_0 &= 120 = I_{SS} - 30 \\ \dot{R} &= 0.05\cdot I - 0.01\cdot R & R_0 &= 700 = R_{SS} - 50 \end{aligned} \tag{5.19}$$

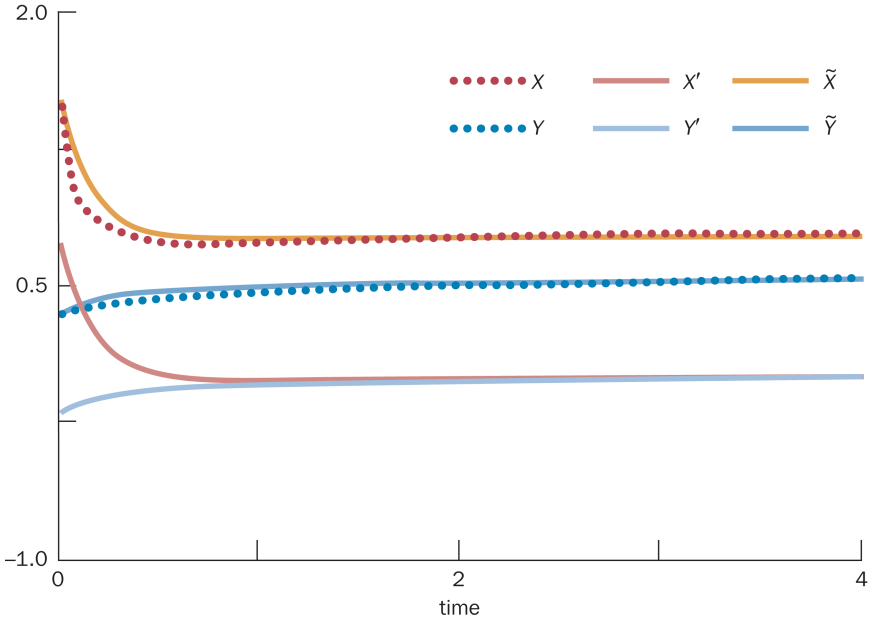


Figure 5.3 Linearization of the system in (5.16). Solution points of the nonlinear system are shown as dots. X' and Y' are the solutions of the linearized system (5.18), and $\tilde{X} = X' + X_{ss}$ $\tilde{Y} = Y' + Y_{ss}$ constitute the linearization of the nonlinear system at the operating point, which here is the steady state.

Note that, for convenience, the initial values are expressed here as deviations from the steady state, which is (S_{ss}, I_{ss}, R_{ss}) = (140, 150, 750) and which we use as our operating point (*OP*). Without going through the theoretical derivation, we can immediately compute the Jacobian, which is

$$J = \begin{pmatrix} \frac{\partial f_1}{\partial S} & \frac{\partial f_1}{\partial I} & \frac{\partial f_1}{\partial R} \\ \frac{\partial f_2}{\partial S} & \frac{\partial f_2}{\partial I} & \frac{\partial f_2}{\partial R} \\ \frac{\partial f_3}{\partial S} & \frac{\partial f_3}{\partial I} & \frac{\partial f_3}{\partial R} \end{pmatrix}_{OP} = \begin{pmatrix} -0.075 & -0.07 & 0.01 \\ 0.075 & 0 & 0 \\ 0 & 0.05 & -0.01 \end{pmatrix}, \tag{5.20}$$

where f_1, f_2, and f_3 are the right-hand sides of the system in (5.19), and the derivatives are evaluated at the steady-state operating point *OP*. Therefore, the linearized equations are given as

$$\begin{pmatrix} \frac{dS'}{dt} \\ \frac{dI'}{dt} \\ \frac{dR'}{dt} \end{pmatrix} = \begin{pmatrix} -0.075 & -0.07 & 0.01 \\ 0.075 & 0 & 0 \\ 0 & 0.05 & -0.01 \end{pmatrix} \begin{pmatrix} S' \\ I' \\ R' \end{pmatrix}. \tag{5.21}$$

We solve these equations with initial values that correspond to the deviations of the nonlinear system from the steady state, namely (–30, 30, –50) (see Eqn. (5.19) and **Figure 5.4**). Once computed, the solution is shifted to the operating point *OP*, which corresponds to the steady state: $\tilde{S}(t) = S'(t) + S_{ss}$, $\tilde{I}(t) = I'(t) + I_{ss}$, $\tilde{R}(t) = R'(t) + R_{ss}$. Figure 5.4 shows the result: the linearization reflects the nonlinear system very well, because the deviation from the steady state is relatively minor.

In summary, the steps for performing linearization of a multivariate ODE system are as follows:

1. Pick an operating point OP (here, always a steady state).
2. Calculate the Jacobian and evaluate it at this OP.

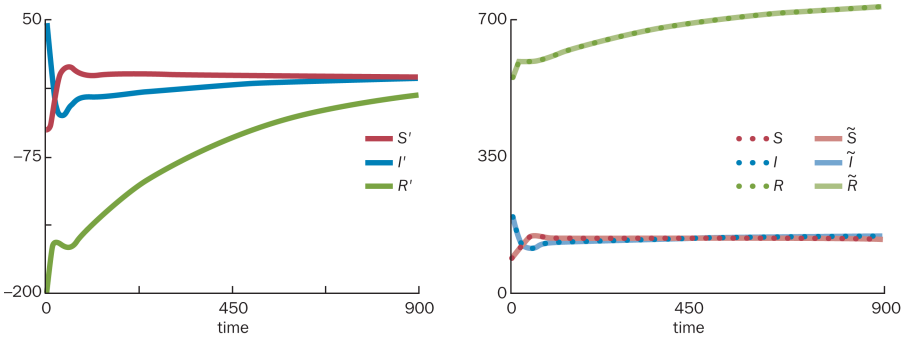


Figure 5.4 Linearization of the SIR system in (5.19). (A) The solution to the linearization in (5.21). (B) The same solution, shifted to the operating point *OP*. The solution points of the nonlinear system are shown as dots. $\tilde{S}$, $\tilde{I}$, and $\tilde{R}$ constitute the linearization of the nonlinear system at the steady-state operating point. The linearization is excellent, because the nonlinear system does not deviate very far from the steady state.

3. Use the Jacobian to set up the linearized equations that approximate the deviation from the OP. This system allows us to capture features such as the stability of a steady state.
4. To approximate the nonlinear system, shift the linearized form back by amounts that are equivalent to the deviations of the initial conditions from the OP.

5.2 STEADY-STATE ANALYSIS

Although uncounted representations of a biological system are possible, even with a restriction to continuous models, there are general strategies of analysis. Indeed, much of this analysis is conceptually straightforward and follows guidelines that are independent of the particular system under investigation. The most pertinent standard methods for analyzing biological systems fall into two broad classes. One pertains to dynamic features, such as the exploration of what types of model responses are possible if one introduces different perturbations. The other class focuses on static features of a dynamical systems model and is based on analyses of the system within the neighborhood of a steady state. Such a steady state is of enormous importance, both mathematically and biologically. The mathematical appeal is that all ODEs of a system at this point equal zero (see below), which clearly is an exceptional situation. Biologically, the steady state is important because it represents **homeostasis**, where all variables are in balance. Most biological systems are assumed to operate close to homeostasis.

5.2.1 Determination of Steady States

We have mentioned steady states several times, quasi on the fly, and it is time now to be a bit more formal. A steady state is a condition of a system in which none of the variables changes in number, amount, or concentration:

$$X_i \text{ at a steady state} = \text{constant, for all variables } X_i \text{ of a system.} \tag{5.22}$$

For ODE models, this condition translates into

$$\dot{X}_i \text{ at a steady state} = 0, \text{ for all variables } X_i \text{ of a system.} \tag{5.23}$$

The lack of change does not imply that nothing is happening in the system. It simply means that all influxes and effluxes at all variables are in perfect balance. As an example, consider the concentrations of chemical components in the blood. There is ongoing metabolism, and blood cells take up and deliver oxygen. Nevertheless, many of the components of the serum remain within very close ranges about their normal states of homeostasis.

Steady-state analyses related to dynamical models address four basic questions, and these are similar to what we discussed in Chapter 4. First, does the system have no, a single, or more than one steady state? Second, can we compute them? Third, is the system stable at a given steady state? And fourth, how sensitive to perturbations is the system at a given steady state?

In addition to these diagnostic topics, one might be interested in controlling, manipulating, or optimizing the steady state of the system. We already hinted at the fact that most steady-state analyses are executed with the use of linear mathematics, motivated and supported by the insight of Hartman and Grobman that many behaviors of a nonlinear system can be studied with the corresponding linear system [2–6]. In fact, now that we have gained some experience with approximations, we can say more exactly what that means. It refers to the (univariate or multivariate) linearization of the nonlinear system, with the steady state chosen as the operating point.

In a linear system, the steady state is relatively easy to assess, because if the derivatives are zero, the system of differential equations becomes a system of linear algebraic equations. Recalling (5.4), but allowing for an additional constant input **u**, and restricting the analysis to the steady state yields

$$\frac{d\boldsymbol{X}}{dt} = \dot{\boldsymbol{X}} = \boldsymbol{A}\cdot\boldsymbol{X} + \boldsymbol{u} = 0\,. \tag{5.24}$$

Linear algebra tells us that there are three possibilities: The system has

- no steady-state solution
- exactly one solution or
- whole lines, planes, or higher-dimensional analogs of planes satisfying (5.24).

As simple but illustrative two-variable example, let us look at five systems S1–S5, which appear to be quite similar.

$$\textbf{S1}: \begin{array}{l} \dot{X}_1 = 2X_2 - 2X_1 \\ \dot{X}_2 = X_1 - 2X_2 \end{array}$$

In this example without input the steady-state computation requires $2X_2 - 2X_1 = 0 = X_1 - 2X_2$, which reduces to $X_1 - 2X_1 = 0$ and is only possible for $X_1 = 0$. Plugging this value into one of the equations yields $X_2 = 0$. Thus, the solution ($X_1 = X_2 = 1$) is unique and not particularly exciting. If we put the system into an ODE solver and start, say at ($X_1 = X_2 = 1$), we find that both variables decrease toward the only steady-state solution $X_1 = X_2 = 0$. This type of solution is often called a **trivial steady state**, because one or more variables are zero.

$$\textbf{S2}: \begin{array}{l} \dot{X}_1 = 2X_2 - 0.5X_1 \\ \dot{X}_2 = X_1 - 2X_2 \end{array}$$

This system is very similar to S1, except that the coefficient of X_1 in the first equation is –0.5 instead of –2. The system again satisfies that trivial solution $X_1 = X_2 = 0$, but, in this case, the dynamical solution does not approach this steady-state solution. In fact, if we start a simulation anywhere but at (0, 0), the system leads to unbounded growth.

$$\textbf{S3}: \begin{array}{l} \dot{X}_1 = 2X_2 - X_1 \\ \dot{X}_2 = X_1 - 2X_2 \end{array}$$

In this case the coefficient of X_1 in the first equation is –1. Setting the derivatives in both equations equal to zero and multiplying one of them with –1 shows immediately that they are equal. Both require that X_1 is twice as big as X_2, but there are no other conditions. A bit of analysis (or computer simulation) confirms that any combination consisting of an arbitrary value of X_1 and half that value for X_2 satisfies the steady state. The system has infinitely many solutions, including (2, 1), (8, 4) and (0, 0) (**Figure 5.5**); all these solutions fill a straight line, defined by $X_1 = 2X_2$. Exercise 5.60 asks you to draw a pathway diagram of the system.

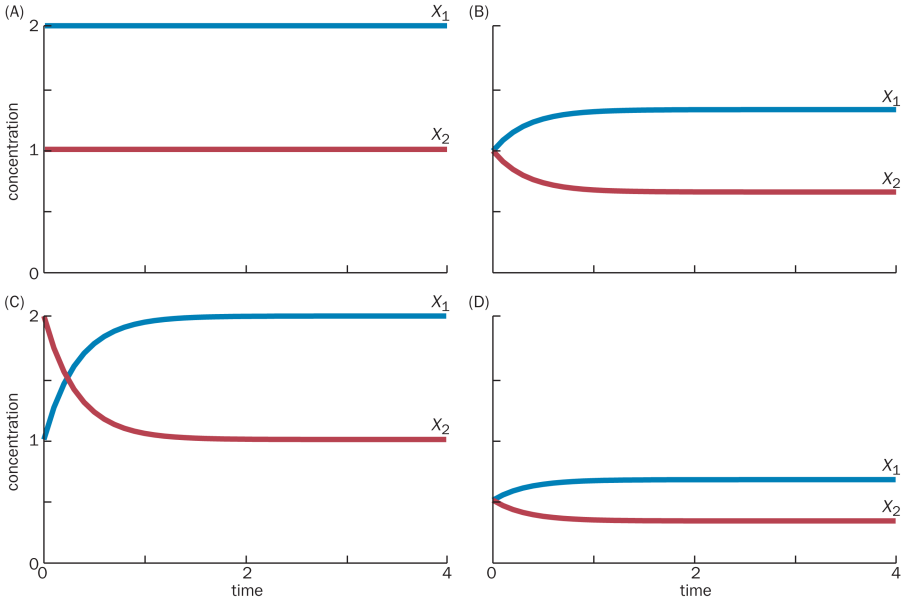


Figure 5.5 Four solutions of the system S3, which differ solely in the initial conditions. (A) $X_1(0) = 2$, $X_2(0) = 1$; (B) $X_1(0) = 1$, $X_2(0) = 1$; (C) $X_1(0) = 1$, $X_2(0) = 2$; (D) $X_1(0) = 0.5$, $X_2(0) = 0.5$.

$$\mathbf{S4}: \quad \begin{aligned} \dot{X}_1 &= 2X_2 - 2X_1 + 1 \\ \dot{X}_2 &= X_1 - 2X_2 \end{aligned}$$

The only difference between S4 and S1 is an additional constant input with value 1. This change drastically changes the dynamic and the steady-state solutions. A trivial solution like $X_1 = X_2 = 0$ does not satisfy the equations. Instead, setting the second equation qual to zero again yields $X_1 = 2X_2$, and plugging it into the first equation at steady state yields the unique pair of values $X_1 = 1, X_2 = 0.5$. In contrast to the decreasing processes toward 0 in S1, an input is feeding the system, and the system quickly finds its unique balance at (1, 0.5).

$$\mathbf{S5}: \quad \begin{aligned} \dot{X}_1 &= 2X_2 - X_1 + 1 \\ \dot{X}_2 &= X_1 - 2X_2 \end{aligned}$$

This system also has an input of 1, and the only difference with S4 is the coefficient of X_1 in the first equation. Setting the derivatives in both equations equal to zero and adding them yields the requirement 0 = 1, which we know cannot be satisfied. The system has no steady state; not even a trivial one.

The computation of steady states in nonlinear systems is in general much harder. In fact, even if we know that the system has a steady-state solution, we may not be able to compute it with simple analytical means and need to approximate the solution with some "root finding" algorithm. We already saw a deceivingly harmless-looking example in Chapter 2, namely the equation $e^x - 4x = 0$. We could easily envision that this equation described the steady state of the differential equation $\frac{dx}{dt} = e^x - 4x$. If we cannot even solve this single nonlinear equation, imagine how little is possible in a system with dozens of nonlinear equations! You might ask why we don't use linearization and compute the steady state of the linearized system. The reason that this seemingly clever idea won't work is that we need an operating point for the linearization. But if we don't know what the steady state is, we cannot use it as the operating point, and choosing a different point is not helpful.

We may obtain the steady state(s) of a nonlinear system with two numerical strategies. First, we can do a simulation and check where the solution stabilizes. This strategy is very simple and often useful. It has two drawbacks, however. First, if the steady state is unstable, then the simulation will avoid it and diverge away

from it (see S2). Second, in contrast to linear systems, nonlinear system may possess (many) different, isolated steady states. Isolated means that points in between are not steady states. As an example, consider the single nonlinear differential equation

$$\dot{x} = -0.1 \cdot (x-1) \cdot (x-2) \cdot (x-4) \tag{5.25}$$

We can convince ourselves easily that 1, 2, and 4 are steady-state solutions, because the right-hand side becomes zero for each of these values for x. However, values in between, such as $x = 3$, are not steady-state solutions. The simulation strategy for computing the steady states would yield $x = 4$ for all initial values above 2 (see the red lines in **Figure 5.6**). For all initial values below 2, the simulation would yield $x = 1$ as the steady state, and, unless we started exactly at 2 (or used insight and imagination), the (unstable) steady state $x = 2$ would slip through the cracks. In the case presented here, the failure would be easy to spot, but in realistic models of larger size that is not always so, and numerical solution strategies can easily miss a steady state.

The second strategy for finding steady-state solutions uses **search algorithms**, of which there are many different types and software implementations. We will discuss some of them in the context of optimal models (see Chapter 6). The generic idea of many of these algorithms is that we start with a wild guess (the literature calls it an "educated guess" or a "guestimate"). In the example of (5.25), shown in **Figure 5.6**, we might begin with the (wrong) guess that 3.4 could be a steady state. The algorithm computes how good this guess is by setting (5.25) equal to zero and evaluating the **residual error**, i.e., the difference between $-0.1(3.4 - 1)(3.4 - 2)(3.4 - 4)$ and 0, which is 0.2016. Next, the algorithm checks solutions close to the initial guess (for instance, 3.3 and 3.5) and determines the direction in which the solution has the higher chance of improving. In our case, $x = 3.3$ yields a higher (= worse) residual error of 0.2093, while $x = 3.5$ corresponds to a lower (= better) residual error of 0.1875. The algorithm now assumes that, because the error at $x = 3.5$ is lower, the best chances of improving the solution lie with even higher values of x. Thus, the algorithm updates the current guess with a higher number, such as 3.6. This systematic trial-and-error strategy is repeated 100s, 1,000s or, in complicated cases, millions of times until the algorithm cannot find a better solution. The technical details of this type of gradient search (such as the exact direction and distance for the next move) can be very complicated, especially for systems with many equations. More details are presented in Chapter 6. One should just add that these types of searches often succeed but that there is no guarantee. Furthermore, they may not reveal all solutions but converge to the same solutions again and again.

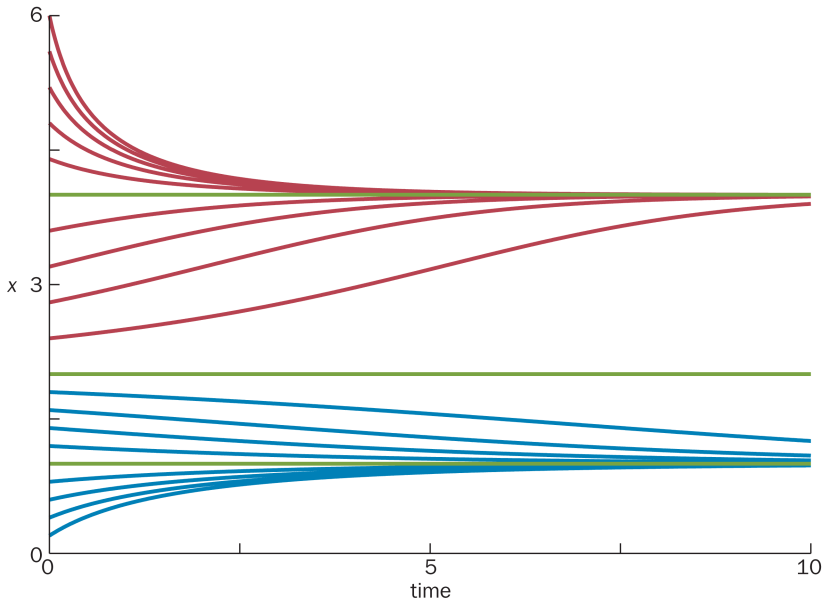


Figure 5.6 Some dynamical (red, blue) and steady-state (green) solutions of (5.25). The steady state at 2 is never reached unless the simulation starts exactly at 2, because it is unstable.

While almost no nonlinear models permit an easy computation of their steady states, some **canonical** models are shining exceptions. The term "canonical" refers to the fact that these forms always have the same mathematical structure, no matter what the application. We will discuss these examples later.

5.2.2 Local Stability Analysis

Stability analysis assesses the degree to which a system can tolerate external perturbations. Obviously, this feature is of great importance in biology, as all bacteria, cells, and organisms are regularly exposed to slight changes in their environment. In the most common case of local stability analysis, one asks whether the system will return to a steady state after a small perturbation. By contrast, global stability analysis asks whether the transients of a system fundamentally change if some parameter value is perturbed; we will return to this question later.

An often-used analogy to a stable steady state is a ball in a half-spherical bowl (**Figure 5.7A**). Left to its own devices, a ball thrown into the bowl will roll up and down the sides a few times, but finally roll to the bottom of the bowl and stay there. If one moves it away from the bottom (if one perturbs it), it will roll back to the stable steady state at the bottom of the bowl. Now suppose the bowl is turned upside down (**Figure 5.7C**). With a very steady hand, one might be able to place the ball on the top of the inverted bowl, but even the slightest perturbation will cause it to roll down. The top of the bowl is a steady state, but it is unstable. As an intermediate case, imagine a ball on a perfectly flat, horizontal surface (Figure 5.7B): any perturbation will cause the ball to roll away and, owing to friction, come to rest at a new location. All points of the flat surface are steady-state points, but none of them is stable. Note that these considerations are true for relatively small perturbations. If one really whacks the ball in the example in Figure 5.7A, it may fly out of the bowl and land on the floor. The same is true for local stability analysis of nonlinear systems: it only guarantees stability for small perturbations. Mathematicians sometimes refer to a **basin of attraction**, which describes the size of a domain from which trajectories converge to the steady state. A system with a large basin of attraction around a stable point can capture a vast range of system space. We will explore these features more in Chapter 10, where we discuss systems with multiple steady states.

Before we discuss technical details of stability, one should note that the terminology "the system is stable" is unfortunate, if not wrong, because a nonlinear system often has several steady states. Thus, the correct terminology is "the system is stable at the investigated steady state" or "the investigated steady state of the system is stable."

Whether a steady state of a dynamical system is stable or not is a priori not an easy question. However, it can be answered computationally in two ways. First, we may start a simulation with the system at the steady state and perturb it slightly. If it returns to the same steady state, this state is locally stable. Second, one can study the system matrix. In the case of a linear system, the matrix is given directly, while nonlinear systems must first be linearized. For a linearized system, the matrix of interest is the Jacobian (5.14), which contains partial derivatives of the system, evaluated at the steady-state operating point. In either case, the decision on stability rests with the real or complex eigenvalues of the matrix, and the rule for stability is simple:

> *If any of the eigenvalues has a positive real part, even if it is just one out of three dozen or so eigenvalues, the steady state is locally unstable. If all real parts are negative, the steady state is stable.*

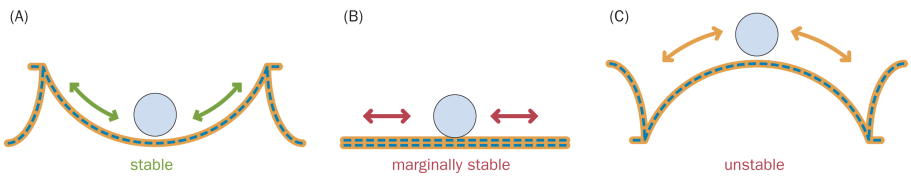


Figure 5.7 Illustration of different degrees of stability. (A) A ball placed in a bowl rolls to the bottom and returns to this position if it is slightly moved. (B) A ball on a perfectly flat surface rolls to a new position when it is tapped. (C) A ball placed perfectly on top of an inverted bowl might stay there for a while, but the slightest perturbation will cause it to roll to one of the sides.

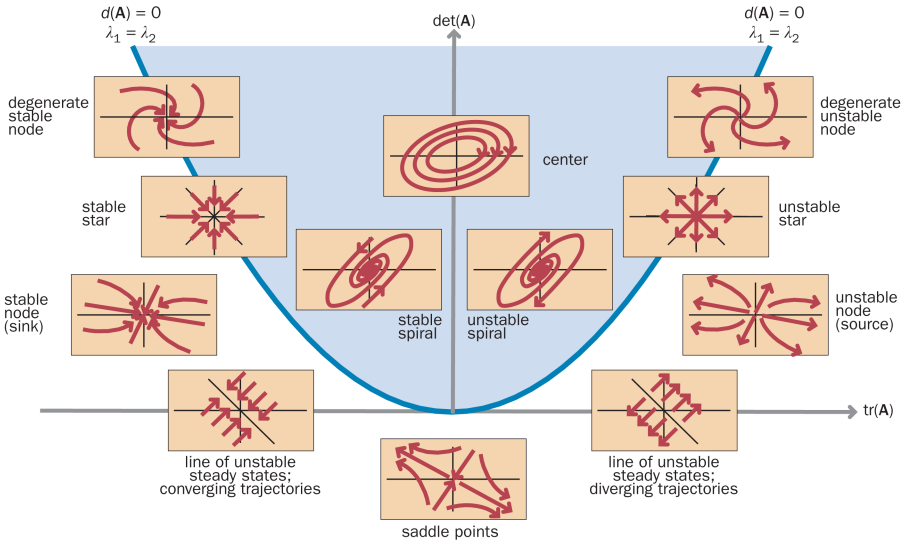


Figure 5.8 Stability patterns in two-dimensional linear differential equation systems without input (see Eqn. 5.4). Only the second quadrant, with *tr*(**A**) < 0 and *det*(**A**) > 0, contains systems with truly stable steady states. The parabola *d*(**A**) = 0 corresponds to "degenerate" cases, where the two eigenvalues are the same. The shaded area above the parabola contains system with complex conjugate eigenvalues, which permit oscillations. Below the parabola, both eigenvalues are real.

The reason behind this rule is that the solution of an n-variable linear system without input is given as a sum of exponentials containing the system's eigenvalues (recall Eqn. 5.6). If t progresses toward infinity, any positive eigenvalue λ_i would make the entire solution diverge toward $+\infty$ or $-\infty$. Thus, for stability, all real parts have to be negative. Cases with eigenvalues having real parts of zero are complicated and require additional analysis or simulation studies. For small systems, it is not too difficult to compute eigenvalues by hand, and software packages like *Mathematica*®, MATLAB®, and PLAS [1], as well as dedicated websites, can also be used to compute eigenvalues.

An important note regarding local stability is that the cut-off values for the real parts of eigenvalues are different for ODE systems and discrete systems: For ODE systems, all real parts must have values below 0, whereas for discrete systems the magnitudes of all real parts must be below 1 (see Chapter 4). The reason is directly related to the different structure of the two modeling frameworks.

In the case of a two-variable linear ODE system without input (Eqn. 5.4) the stability analysis is particularly instructive, because one can visually distinguish all different behaviors of the system close to its steady state point (0, 0). Specifically, the stability of (0, 0) is determined by three features of the system matrix **A**, namely its trace $tr(\mathbf{A}) = A_{11} + A_{22}$, determinant $det(\mathbf{A}) = A_{11}A_{22} - A_{12}A_{21}$, and discriminant $d(\mathbf{A}) = tr(\mathbf{A})^2 - 4det(\mathbf{A})$. These three features are directly related to the two eigenvalues of **A**, λ_1 and λ_2, which can be expressed as $\lambda_{1,2} = \dfrac{tr(A) \pm \sqrt{tr(A)^2 - 4\det(A)}}{2}$.

Depending on the combination of these features of **A**, the system behaviors close to (0, 0) can be quite varied. They may look like sinks, sources, attracting or repelling stars, inward or outward spirals, saddles, or collections of lines (**Figure 5.8**). To see these images in a simulation, one solves the same system (Eqn. 5.4) with the same matrix **A** from many different initial values and superimposes all phase-plane plots (**Figure 5.9**). Corresponding behaviors occur for higher-dimensional systems, where they are however more difficult to visualize.

The first quadrant of the plot in **Figure 5.8** (top half, right of the y-axis) contains unstable systems, where both eigenvalues have positive real parts, while the second quadrant (top half, left of the y-axis) contains the corresponding stable systems, where both eigenvalues have negative real parts. Exactly on the y-axis one finds a special situation, which one may consider a transition case: instead of stable spirals on the left and unstable spirals on the right, the y-axis is the location of *center oscillations*, where all solutions collectively form an infinite set of concentric circles or ellipses that cover the entire plane. The best-known example is the sine-cosine system (5.3).

A different type of transition occurs when crossing the parabola given by $d(\mathbf{A}) = 0$. Exactly on this line, the discriminant vanishes and the two eigenvalues have the same real value: $\lambda_1 = \lambda_2$; this case is sometimes called *degenerate*. On the

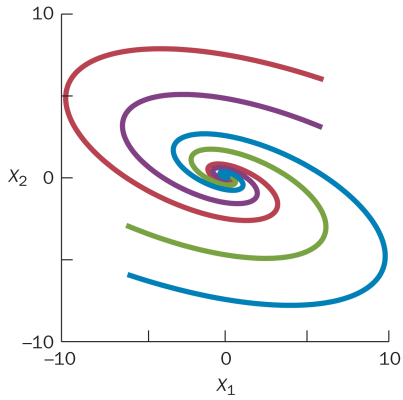


Figure 5.9 Visualization of a stable spiral. The coefficients of **A** are $(A_{11}, A_{12}, A_{21}, A_{22})$ = (−3, −6, 2, 1) and initial values for the four superimposed trajectories are red, (6, 6); purple, (6, 3); green, (−6, −3); and blue, (−6, −6). Other initial values would lead to similar trajectories in between, collectively covering the entire plane.

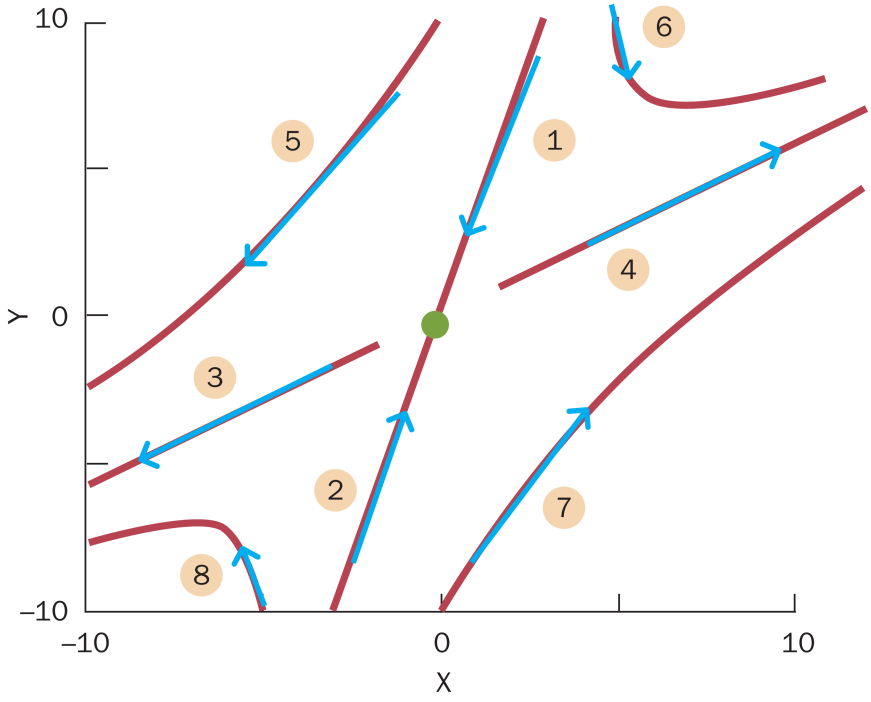


Figure 5.10 Saddle point behavior of a linear or linearized system with matrix $\mathbf{A} = \begin{pmatrix} 2 & -1 \\ 2 & -2 \end{pmatrix}$ in phase-plane plot. Initial conditions ❶ and ❷ are exact multiples of the eigenvector associated with the negative eigenvalue, while ❸ and ❹ are exact multiples of the eigenvector associated with the positive eigenvalue. The remaining initial values lead to curved trajectories that bypass the steady-state point (0, 0). The various initial conditions are:

❶ X = 10(− √2 + 2)/2
Y = 10
❷ X = 10(− √2 + 2)/2
Y = −10
❸ X = −(+ √2 + 2)/2
Y = −1
❹ X = (+ √2 + 2)/2
Y = 1
❺ X = 0
Y = 10
❻ X = 5
Y = 10
❼ X = 0
Y = −10
❽ X = −5
Y = −10

left of the y-axis, the eigenvalue is negative and on the right it is positive. Above the parabola (shaded area), the discriminant is negative, while it is positive below (confirm this!). If the discriminant is positive, both eigenvalues are real, whereas they form a complex conjugate pair for $d(\mathbf{A}) < 0$ (shaded area). Only the latter case permits oscillations in the system, which is due to the fact that the exponential of a complex number, $e^{i\theta}$, equals $\cos(\theta) + i\sin(\theta)$.

All systems in the bottom half of the plot have real eigenvalues, one of which is positive, while the other is negative. This situation leads to *saddle points* with its steady state at (0, 0). Is it stable? Starting exactly on a multiple of the eigenvector associated with the negative eigenvalue, the system moves in a straight line (in phase plane) toward zero, which suggests stability (**Figure 5.10**). But starting exactly on a multiple of the eigenvector associated with a positive eigenvalue, the system diverges in a straight line (in phase plane) toward infinity. Other trajectories may at first seem to approach zero but then curve away from the origin and ultimately diverge toward infinity as well. Thus, the steady state is certainly not stable. The situation is reminiscent of a horse saddle, which at its center has a minimum in the direction of the horse but a maximum in the orthogonal direction.

Crossing the x-axis creates yet a different transition. Systems exactly on the x-axis and in the left half of the plot possess a straight line of marginally stable steady states; all other trajectories converge in straight lines to this line. Systems on the right have a line of unstable steady states; all trajectories diverge from this line in a straight manner. Finally, a unique situation occurs for $tr(\mathbf{A}) = det(\mathbf{A}) = d(\mathbf{A}) = 0$. Here, the two eigenvalues are the same (so there is really only one eigenvalue) with a value of zero. In this case, it could be that $\mathbf{A}$ contains four zero coefficients so that both differential equations equal zero, and wherever the system starts, it stays. It could also be that $\mathbf{A}$ contains other values. Details can be found in [11] and in the exercises.

While 5.17 provides a reference for the dynamic behavior of a two-dimensional linear system, eigenvalues represent more generically classes of behaviors according to the following guidelines (**Table 5.1**):

The eigenvalue analysis just described is very important, but it is also limited. First, in the case of nonlinear systems, the deduction of stability is strictly guaranteed to be correct only for infinitesimally small perturbations. In reality, stable

TABLE 5.1 BEHAVIOR OF LINEAR OR LINEARIZED SYSTEMS, AS DETERMINED BY EIGENVALUES

System Behavior	*Eigenvalues*	
	Real Component	*Imaginary Component*
Stable node	both negative and distinct	0
Stable spiral	both negative	complex conjugate
Center	none	complex conjugate
Saddle	distinct with opposite signs	0
Unstable spiral	positive	complex conjugate
Unstable node	positive	0

systems tolerate moderately large perturbations, but there is no mathematical guarantee. Second, definite conclusions are difficult if one or more eigenvalues have real parts of 0.

There are other types of stability that cannot be addressed with eigenvalue analysis alone. For instance, if one changes the parameter values in a system, it may happen that a stable steady state becomes unstable and that the system begins to oscillate around the unstable steady state. An obvious question is under what conditions something like that might happen. The answer to this question is complicated and constitutes an ongoing research topic in the field of dynamical systems analysis. We will show an example in the section on systems dynamics.

5.2.3 Sensitivity Analysis

Sensitivity analysis is concerned with the generic question of how much a system is affected by small alterations in parameter values. In a biomedical context, such an alteration could represent a mutation or some permanent change in an enzyme activity that could be due to aging or disease. Questions of sensitivity are fundamentally different from local stability analysis. In stability analysis, one or more of the dependent variables are perturbed, the system responds to this perturbation, and one studies the response, which is usually a return to the steady state. In sensitivity and **gain** analysis, by contrast, parameters or independent variables are changed, respectively. These changes are permanent in the sense that the system cannot counteract them, and they almost always cause the system to assume a new steady state. Thus, the question is how different the new steady state is from the old. It may happen that even large variations in parameters do not change the location of the steady state much, but it can also happen that a minute change in some crucial parameter throws the system off course. In a biomedical application, two individuals may be exposed to the same environmental toxicant (perturbation) but respond very differently. One person may not be visibly affected, while the other may become really sick. The difference could be due to different expression levels (parameter values) of some sensitive key genes.

Sensitivity analysis is a crucial component of any systems analysis, because it can quickly show if a model is likely to be wrong. Good, **robust** models usually have low sensitivities, which means that they are quite tolerant to small, persistent alterations, in which a parameter remains altered. Thus, if some of the sensitivities in the model are very high, we typically have cause for concern. However, there are exceptions, for instance in signal transduction systems, where even small changes in signal intensity are greatly amplified (see Chapter 10). Typical sensitivity analyses are again executed by computing the derivative of a system feature with respect to a parameter. While steady-state sensitivities are the most prevalent, it is also possible to compute sensitivities of other features, such as trajectories or amplitudes of stable oscillations.

The easiest way to see how sensitivity analysis works is to study an explicit function. As a specific illustration, let us look at the logistic function $F(x) = a/(b + e^{-cx})$. Changing the parameter c affects the steepness of the sigmoidal function (see **Figure 5.11**). Let us now look at the sensitivity of the function with respect to parameter b. As with c, we can just change b to some arbitrary different value, and

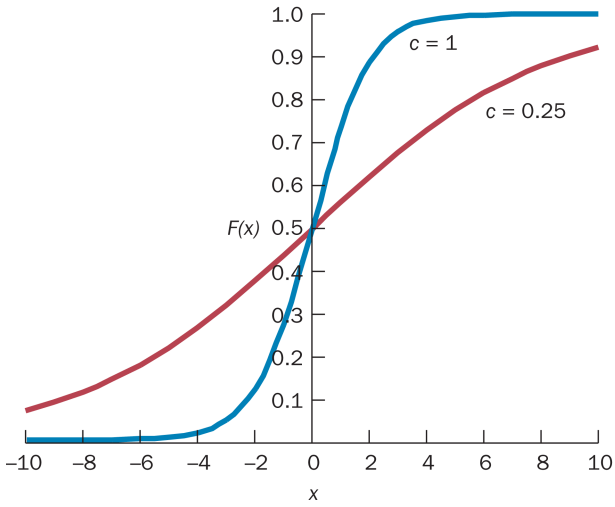


Figure 5.11 Graphs of the logistic function (5.31). Depending on its parameter values, the logistic function models different slopes, final values, and horizontal shifts.

maybe repeat this exploration a few times with different numbers. This strategy gives us some idea in this simple case, but we are interested in a more systematic way to compute the effect of b on F. So, let us study the effect of a generic, small change in b. To make our intent of testing the effects of b explicit, we write the function as $F(x; b)$. The result of a small change Δb in $F(x; b)$ is given as $F(x; b + \Delta b)$. Using the **Taylor approximation** theorem (see **Box 5.1**), we can linearize this expression by using the operating point $F(x; b)$, the slope, and the distance of the deviation. The result in this case of a single-variable function is

$$F(x;b+\Delta b)\approx F(x;b)+\frac{\partial F(x;b)}{\partial b}\cdot\Delta b \tag{5.26}$$

where the expression with the ∂'s indicates the partial derivative of F with respect to b. The change in F caused by a small change in b is

$$\Delta F\approx F(x;b+\Delta b)-F(x;b) \tag{5.27}$$

Rearranging this equation and using (5.26) gives us the desired answer, namely a quantification of how much F changes if b is altered:

$$\Delta F\approx\frac{\partial F(x;b)}{\partial b}\cdot\Delta b \tag{5.28}$$

To put this result in words: a small change in b is translated into a change in F with magnitude $\frac{\partial F(x;b)}{\partial b}$. If F and b have the same unit, the result means the following: if this expression happens to equal 1, the change in F is equal to the change in b. If it is larger than 1, the perturbation is amplified, and if it is smaller than 1, the perturbation is attenuated.

If units are different, or if the sensitivities of several variables in a system are to be compared, it is useful to study **relative sensitivities** by assessing by what percent a variable changes given a 1% percent change in a parameter.

According to Taylor's theorem, these answers are absolutely accurate for infinitesimally small changes in b, but they are sufficiently good for modest—or sometimes even large—alterations in b. The take-home message is that the effect of b on F is driven by the partial derivative with respect to b, and this message is unchanged even for differential equations and systems of differential equations.

As a numerical example with the logistic function, let us assume that $a = 2$ and $c = 0.25$, and that the normal value of b is 1. The partial derivative of F with respect

to b is computed while keeping all other parameters and the variable x constant. Thus,

$$\frac{\partial F(x;b)}{\partial b} = \frac{-a}{(b+e^{-cx})^2} \tag{5.29}$$

Obviously, the right-hand side depends on x, which means that the sensitivity of F with respect to b is different for different values of x. A typical example is the sensitivity of the nontrivial steady state. In the case of F, the function approaches this state for x going to $+\infty$. Maybe we would prefer a finite number for our computation, but we can still compute the sensitivity, because for x going to $+\infty$, the exponential function e^{-cx} goes to zero. Therefore, the sensitivity is

$$\frac{\partial F(x;b)}{\partial b} = -\frac{a}{b^2} \quad x \to +\infty \tag{5.30}$$

Plugging this result into (5.28), along with the values $a = 2$ and $b = 1$, we obtain an estimate for how much the nontrivial steady state of F will change if b is altered. For instance, if $\Delta b = 0.1$, which would mean a 10% perturbation from its normal value of 1, the steady state of F is predicted to change by $-2 \cdot 0.1 = -0.2$. In fact, if we numerically compute $F(x; b = 1.1)$ for x going to $+\infty$, the result is $-2/1.1 = -1.81818$ which, multiplied with 0.1, is close to the predicted 0.2 decrease.

It is not difficult to imagine that our function $F(x; b)$ could be the steady-state solution of a differential equation. Reinterpreting our procedure and results then tells us how to do steady-state sensitivity analysis for a single differential equation, and systems of equations follow the same principles. Software like PLAS [1] automates this computation for S-systems and GMA systems.

The profile of sensitivities is an important metric for assessing biological systems. If the sensitivities with respect to parameter p_1 are relatively high in magnitude, while they are very close to zero for parameter p_2, then it is important to obtain the numerical value of p_1 as accurately as possible, whereas a small error in p_2 has little effect on the system.

If the value of an independent variable is changed, the procedure is the same, but we talk about a gain or logarithmic (relative gain), rather than sensitivity [12].

As an example, consider the two-variable S-system

$$\begin{aligned} \dot{X}_1 &= \alpha_1 X_0 X_2^{-0.6} - \beta_1 X_1^{0.5} \\ \dot{X}_2 &= \alpha_2 X_1^{0.5} - \beta_2 X_2^{0.7} \end{aligned} \tag{5.31}$$

with rate constants $\alpha_1 = \beta_1 = \alpha_2 = 2$, $\beta_2 = 1$and the independent input variable $X_0 = 2$. Draw a diagram corresponding to these equations. We compute the steady state by setting the equations equal to zero and solving for X_{1ss} and X_{2ss}, which yields

$$\begin{aligned} X_{2ss} &= \left(\frac{\beta_2}{\alpha_1 X_0}\right)^{-1/1.3} = 2.904846 \\ X_{1ss} &= \left(\left(\frac{\beta_2}{\alpha_1 X_0}\right) \cdot X_{2ss}^{0.7}\right)^2 = 1.112531 \end{aligned} \tag{5.32}$$

It is useful to leave the results in symbolic form, because we now compute, as an example, the relative sensitivity of X_{2ss} with respect to β_2:

$$\frac{\partial X_{2ss}}{\partial \beta_2} \cdot \frac{\beta_2}{X_{2ss}} = -0.76923 \tag{5.33}$$

PLAS and other software automatically compute this relative sensitivity. We can explore the accuracy of this result by changing β_2, for instance, by 10%, which is actually much larger than an infinitesimally small change that is computed in the

sensitivity analysis and guaranteed to give the exact answer. Changing the second equation to

$$\dot{X}_2 = \alpha_2 X_1^{0.5} - 1.1\beta_2 X_2^{0.7} \tag{5.34}$$

yields the new steady state ($X_{1ss\text{-}new}$, $X_{2ss\text{-}new}$) = (1.21485, 2.69950). The relative change in the steady-state value of X_2, which is given by $[(X_{2ss\text{-}new} - X_{2ss})/X_{2ss}]/[(\beta_{2new} - \beta_2/\beta_2)] = -0.007069$, corresponds to a drop in X_{2ss} of about 0.71%, which is quite similar to the relative sensitivity value computed earlier. The fact that $X_{2ss\text{-}new}$ is lower than X_{2ss} is reflected in the minus sign of the relative sensitivity. One of the exercises analyses smaller changes in β_2.

5.3 DYNAMICS OF NONLINEAR MODELS

Whether systems are linear or nonlinear, continuous representations account for entire time intervals and not just for discrete time points, as we discussed in Chapter 4. Furthermore, continuous systems become the more convenient choice if the numbers of players in the system becomes large. Who wants to count (or account for) every cell in a culture or organism? As in the linear case, the switch to continuous variables leads directly to the formulation of differential equations, but the nonlinear case affords us with incomparably more choices: how do we choose the right-hand sides of the ODEs? Should we use polynomials? Or exponentials? Or combinations of all kinds of functions? Clearly, there are unlimited possibilities, and nature has not provided us with an instruction manual. As a consequence, nobody is entitled to tell us what the best or most appropriate functions are, except for a minute class of very special situations. This fact poses a significant challenge and is the subject of extensive ongoing research (Chapter 6). In most cases, one of two strategies is used. Either one makes assumptions regarding functions that appear to be reasonable, for instance, because they somehow model the mechanisms that drive the phenomenon under investigation (these functions are sometimes called ad hoc, indicating that they were chosen especially for this particular situation and are not claimed to have wider application), or one uses a generic nonlinear approximation that does not require specific assumptions and offers sufficient flexibility in the repertoire of responses it can represent. An approximation of this type is called **canonical**.

5.3.1 Ad Hoc Models

Ad hoc models use functions that for some reason seem to suit the given task the best. Sometimes they are based on the principles of physics and sometimes they simply seem to have just the right shapes and other properties. As an example of the latter, the logistic function

$$F(x) = \frac{a}{b + e^{-cx}} \tag{5.35}$$

is frequently used, sometimes even with $a = b = c = 1$, because it describes a sigmoidal trend as it is often encountered in biology (**Figure 5.11**). Altering a and/or b changes the output range and horizontal shift of the function, while c affects its steepness. Try out a few combinations of a, b, and c.

In the field of population dynamics, the same logistic function is more widely used in the equivalent format of a differential equation of the type

$$\dot{N} = \alpha \cdot N - \beta \cdot N^2, \tag{5.36}$$

where N is the size of a population and α and β are parameters. The first parameter, α, is the growth rate and α/β is the final size or carrying capacity of the population (see Chapter 14).

At first glance, (5.35) and (5.36) don't seem to have much in common. However, if one renames F in (5.35) as N and x as t and differentiates N with respect to t, the result can be manipulated with simple algebra to attain the form (5.36), where the parameters α and β and the initial value N_0 are combinations of a, b, and c. One should also note that (5.36), slightly rewritten as $\dot{N} = \beta N(\gamma - N)$, with $\gamma = \alpha\beta$ looks a lot like the logistic map in Chapter 4, thereby explaining the similarity in name. However, the two are quite different, because the left-hand sides have a different meaning.

The logistic model in (5.36) is the launch pad for very many models of population dynamics (Chapter 14), where different species compete for resources or where one species feeds on another. It is also the base case for canonical Lotka–Volterra systems (see later), which can describe an unlimited variety of nonlinearities.

There is hardly a limit to the possible choices for ad hoc functions, and the selection of a particular function depends heavily on the context and knowledge of the biological phenomenon. Possibilities include trigonometric functions, simple exponentials, logarithms or polynomial functions, or more unusual functions such as spikes or square waves. Indeed, entire catalogs of functions with various shapes have been published [13].

A widely used example of an ad hoc function based on molecular processes is the Michaelis–Menten function of enzyme kinetics, which has the form

$$v_p = \frac{V_{max}\, S}{K_M + S}\,. \tag{5.37}$$

It describes the speed v_P with which a metabolic substrate S is converted into a product P with the help of an enzyme E (**Figure 5.12**). The function has two parameters. $V_{\max}$ is the maximum speed v_P can reach, while K_M is the substrate concentration for which v_P runs with half the maximum speed (see Chapters 2 and 9). A generalization of this function is the Hill function, which has one additional parameter, the Hill coefficient, which allows s-shaped graphs. It has the form

$$v_{Hill} = \frac{V_{max}\, S^n}{K_M^n + S^n} \tag{5.38}$$

and will be discussed in Chapter 9 in detail.

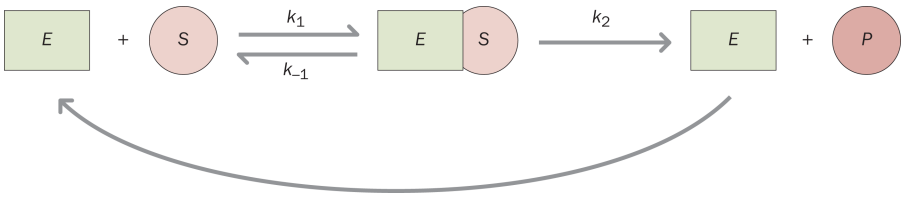


Figure 5.12 Schematic diagram of an enzyme-catalyzed reaction. Substrate S and enzyme E form a complex, which may revert to S and E or break down to yield product P while releasing the enzyme. See Chapter 9 for details.

5.3.2 Canonical Models

A rather different strategy of model selection is the use of canonical models, of which there are only a few types. The designation *canonical* implies that the processes of setting up and analyzing a model follow strict rules. Forced adherence to rules could be seen as a real limitation, but it has significant advantages, as we will see later in this chapter and in some of the application chapters. One attraction of canonical models is that they allow us to get started very quickly with the design of a model from scratch, even if we are faced with a lot of uncertainty and scarce information regarding the details of a system. This advantage becomes most evident if we ask what would happen if we did not have rules and guidelines. As an example, imagine being tasked to come up with a mathematical model describing how the introduction of a new species affects the dynamics of an ecosystem. It is immediately clear that all kinds of players could be involved in the case, but it is very much unclear how exactly they interact. All interactions must follow the laws

of physics and chemistry, but accounting for every physical and chemical detail in the system would quite obviously be impossible—and is also often unnecessary. Canonical modeling provides a compromise between general applicability and simplicity that circumvents some of these problems by using mathematically rigorous and convenient approximations. As a rule of thumb, the less one knows about the details of the system, the more one benefits from canonical models.

The ultimate canonical model is again the linear model, and we would love to use it. Alas, linearity is often too limiting for biological phenomena that deviate a bit from their normal steady state and exhibit responses that are very clearly nonlinear. For instance, a linear model cannot represent stable limit cycle oscillations, as we encounter them in circadian rhythms and our own heartbeat (see later in this chapter). These limitations mean that we must look for nonlinear canonical models. Intriguingly, some canonical models overcome a major challenge of nonlinear analysis; namely, they permit the computation of steady states with methods of linear algebra.

5.3.3 Lotka–Volterra Models

The oldest nonlinear canonical model is the **Lotka–Volterra (LV) model** [14–18]. Interestingly, this model was proposed around the same time as the SIR model for infectious diseases that we introduced in Chapter 2, namely in the 1920s. Its independent authors were Alfred Lotka, an American pioneer in mathematical biology, and Vito Volterra, an applied mathematician from Italy. The rules for setting up an LV model are simple. Enumerate all dependent variables of interest; independent variables are merged with parameters. Assume that each variable may potentially interact with every other variable in the system and formulate each interaction as the product of the two variables and a rate constant. In other words, this mechanism is of mass-action type, as we encountered it in Chapter 2. LV models also permit a linear term in each equation, which could represent birth, death, degradation, or transport out of the system. Thus, each differential equation of an LV model has the same format:

$$\dot{X}_i = \sum_{j=1}^{n} a_{ij} X_i X_j + b_i X_i \ , \tag{5.39}$$

where some (or many) of the parameters may have values of 0. If the system has only one variable, the LV model with negative a_{11} and positive b_1 becomes the logistic function (5.32). Note that the model also contains the "interaction" of a variable with itself, which is represented by a_{ii}. This term, usually with a negative sign, is often interpreted as a crowding term that naturally limits the growth of a population.

The LV model bears good news and bad news. The main good news is that it is easy and intuitive to set up the equations and to compute the steady state (see later). Second, the model has found very many applications in ecological systems analyses, where much of the dynamics, like competition for food or predation, is driven by one-to-one interactions (see [18] and Chapter 14). A further piece of very surprising and intriguing news is that LV equations are extremely flexible: they are capable of modeling any type of differentiable nonlinearities and **complexities**, including all kinds of **damped** or stable oscillations and chaos, if one includes enough auxiliary variables and does some mathematical trickery with them [15, 18, 19]. The bad news is that, without such trickery, many classes of biological systems are not really compatible with the LV format. For instance, suppose we want to model a simple pathway as shown in Figure 5.13. We define four variables $X_1, \ldots, X_4$ and want to formulate appropriate equations. However,

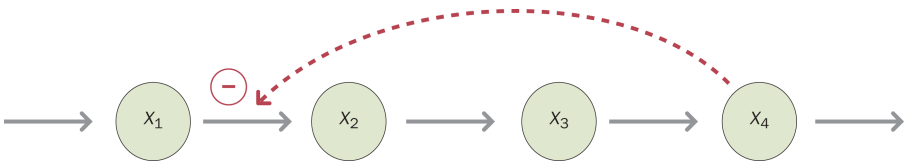


Figure 5.13 Linear pathway with feedback. It is not directly possible to represent this pathway as a Lotka–Volterra system.

this is a problem for the LV format. For instance, the production of X_3 should depend only on X_2, but we are not allowed to include a term like kX_2 in the equation for X_3. Furthermore, the only choice of accounting for the feedback by X_4 in the equation for X_2 would be a "bilinear" term like $\kappa X_1 X_4$, but such a term is not permitted in the equation. As soon as we try to ameliorate this issue, we increase the applicability of the model but compromise the mathematical advantages of the LV format. For instance, if we allow all possible linear and bilinear terms, we arrive at a mass-action system, as we used it for the infectious disease model in Chapter 2. We could also permit multilinear terms of the type $\alpha X_1 X_2 X_4$. In either case, the extension would impede the analytical advantages of the LV model. In particular, the true LV format allows us to compute steady states with methods of linear algebra (see below), whereas the extensions would require the use of nonlinear methods and search algorithms.

The computation of steady states of LV systems is straightforward. Setting the derivatives equal to zero yields

$$\sum_{j=1}^{n} a_{ij} X_i X_j + b_i X_i = 0. \tag{5.40}$$

It is not hard to see that $X_i = 0$ is a solution for the i^{th} equation. Because all n equations have to be satisfied for a non-trivial steady state, $X_i = 0$ must be true for all variables. All other solutions, with some $X_i = 0$, are again encountering a trivial solution. In fact, there are 2^n solutions altogether!

To obtain a non-trivial solution to any one equation, one may employ an old math trick. We acknowledge the trivial solution and then exclude it by fiat, stating that X_i shall not be 0. If so, we are allowed to divide the equation by X_i and the result is a linear equation. Collecting all these linear equations, we obtain the steady states. In particular, if we require that none of the X_i is 0 and divide each equation i by X_i, we obtain a linear system for the steady state where all variable "coexist." Thus, we evaluate the set of n linear equations

$$\sum_{j=1}^{n} a_{ij} X_j = -b_i. \tag{5.41}$$

This is a system of linear algebraic equations and we can apply all the tools of linear algebra.

While the solution without values of 0 is interesting, solutions with some $X_i = 0$ are actually sometimes even more interesting (see Chapter 14 and [20]). Whichever solutions we prefer, it is a rare exception within the vast domain of nonlinear ODE systems that we can analyze all steady states so readily.

5.3.4 Generalized Mass Action (GMA) Models

Much greater flexibility than in LV and mass-action models is afforded by canonical **generalized mass action** (**GMA**) systems within the modeling framework of **biochemical systems theory** (**BST**) [21–26]. In this type of canonical model, each process *vi* is formulated as a product of **power-law functions** of the type

$$v_i = \gamma_{ik} \cdot X_1^{f_{ik1}} \cdot X_2^{f_{ik2}} \cdot \ldots \cdot X_n^{f_{ikn}}. \tag{5.42}$$

As in mass-action and LV models, this format includes a non-negative rate constant γ_{ik}, which describes the turnover rate of the process and a product of variables. In this case, the product may include all, some, or none of the systems variables. Each variable has its own exponent f_{ikj}, which is called a **kinetic order** and may take any real value. If a kinetic order is positive, it signifies a positive or augmenting effect, and if it is negative, it signifies an inhibitory or diminishing effect. If a kinetic order is zero, the variable, raised to zero, becomes 1 and is therefore neutral in the product. Typical kinetic orders are between –1 and +2. The vari-

ables in the term include dependent and independent variables (see Chapter 2) in exactly the same format.

With this type of formulation, each equation in a full GMA model with n dependent and m independent variables has the form

$$\dot{X}_i = \sum_{k=1}^{T_i} \pm \gamma_{ik} \prod_{j=1}^{n+m} X_j^{f_{ikj}}, \tag{5.43}$$

where T_i is the number of terms in the i^{th} equation. The independent variables X_{n+1}, . . . , X_{n+m} are treated just like dependent variables but do not have their own equations. The large "pi" denotes a product; for instance,

$$\prod_{k=1}^{4} X_k = X_1 \cdot X_2 \cdot X_3 \cdot X_4. \tag{5.44}$$

The format of (5.43) looks complicated but is actually quite intuitive once it becomes more familiar. One simply enumerates all processes (terms) affecting a variable, decides which variable has a direct effect on a given process, and includes this variable in the term. The variable receives an exponent, and the whole term is multiplied with a rate constant.

GMA systems can be set up directly from a diagram, as we will see later. They can also be constructed as "power-law approximations" of other models (see **Boxes 5.3** and **5.4**).

As a small example with three dependent variables, consider a branched pathway with one feedback signal, input from the independent variable X_0, and activation by an independent modulator X_4, as shown in **Figure 5.14**. Each equation contains power-law representations of all fluxes v_i that are directly involved. Thus, the GMA equations are

$$\begin{aligned}
\dot{X}_1 &= \gamma_{11} X_0^{f_{110}} X_2^{f_{112}} - \gamma_{12} X_1^{f_{121}} - \gamma_{13} X_1^{f_{131}} \\
\dot{X}_2 &= \gamma_{21} X_1^{f_{211}} - \gamma_{22} X_2^{f_{222}} X_4^{f_{224}} \\
\dot{X}_3 &= \gamma_{31} X_1^{f_{311}} - \gamma_{32} X_3^{f_{323}}
\end{aligned} \tag{5.42}$$

The indices of the rate constants reflect, first, the equation and, second, the term. The kinetic orders have a further index, which refers to the variable with which it is associated. This convention is not always followed though, especially for small system where fewer indices are sufficient.

Some of the terms may be the same, even though they seem different at first glance. For instance, the flux v_2 out of X_1 is the same as the flux into X_2. This equality translates into properties of power-law functions, which in the given case can only be equivalent if $\gamma_{12} = \gamma_{21}$ and $f_{121} = f_{211}$. Similar equalities hold for the two descriptions of the process between X_1 and X_3. The design of a GMA model from a diagram is further illustrated in **Box 5.5**.

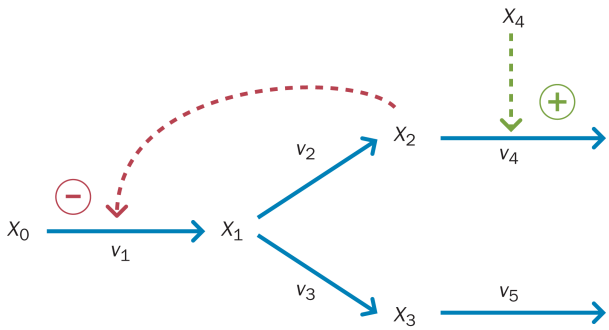


Figure 5.14 Generic pathway with one feedback signal and one external activator. The dependent variable X_1 is produced from a precursor X_0, which is modeled as an independent variable, and is used as substrate for the production of X_2 and X_3. X_2 exerts feedback inhibition on the production of X_1, and X_4 activates the degradation of X_2. Each process is modeled by a function v*i*, which in a GMA model (5.41) has the form of a product of power-law functions (5.38).

5.3.5 S-System Models

Very similar to the GMA format is the so-called **S-system** format [25, 26]. The difference is that, in this format, all processes entering a variable are first merged into one term, which is then represented as a single power-law term, and the same strategy holds for all processes leaving the variable. Biologically, this strategy corresponds to a focus on pools (dependent variables) in S-systems rather than on fluxes in GMA systems: in S-systems, we look at variables and ask "how much is coming in and how much is leaving?" In GMA system, by contrast, we ask "how much material is transported through each flux in the system?"

Owing to the merging of influxes and effluxes in one net flux each, S-systems have at most one positive and one negative term in each equation, which is very significant for further analyses (see later). Therefore, their general S-system form, with its own names for parameters, is always

$$\dot{X}_i = \alpha_i \prod_{j=1}^{n+m} X_j^{g_{ij}} - \beta_i \prod_{j=1}^{n+m} X_j^{h_{ij}} \quad i = 1,2,\ldots,n \tag{5.43}$$

As in GMA systems, the variables $X_{n+1}, \ldots, X_{n+m}$ are independent and do not have their own equations.

For the example in Figure 5.14, the production of X_1 is exactly the same in GMA and S-systems, because it occurs in a single step. Also, the production and degradation processes of X_2 and X_3 remain unchanged. The only difference between the GMA and S-system models occurs for the efflux from X_1: in the GMA system, this efflux consists of two branches, while it consists of one net efflux in the S-system. Thus, the example in Figure 5.14 is described by the S-system model

$$\begin{aligned}
\dot{X}_1 &= \alpha_1 X_0^{g_{10}} X_2^{g_{12}} - \beta_1 X_1^{h_{11}} \\
\dot{X}_2 &= \alpha_2 X_1^{g_{21}} - \beta_2 X_2^{h_{22}} X_4^{h_{24}} \\
\dot{X}_3 &= \alpha_3 X_1^{g_{31}} - \beta_3 X_3^{h_{33}}
\end{aligned} \tag{5.44}$$

Note the similarities and differences between this formulation and the GMA form in (5.42). If the S-system and the GMA system are identical at some operating point, such as a steady state, then the equation $\beta_1 X_1^{h_{11}} = \alpha_2 X_1^{g_{21}} + \alpha_3 X_1^{g_{31}}$ holds at this point. For other values of X_1 they are slightly different, except for some special cases (see Exercises). Generally, the two types of models differ only at converging or diverging branch points and are otherwise identical. The design of a model from a diagram in S-system format, in comparison to the GMA format, is illustrated in **Box 5.5**.

The computation of steady states of S-systems is possible with methods of linear algebra [22]. Suppose for simplicity that the system does not contain independent variables and set the derivatives equal to zero, which yields

$$\alpha_i \prod_{j=1}^{n} X_j^{g_{ij}} - \beta_i \prod_{j=1}^{n} X_j^{h_{ij}} = 0 \quad i = 1,2,\ldots,n \tag{5.45}$$

This steady-state condition does not look linear at all. However, once we move the β-term to the right-hand side and insist that all variables are positive, we can take logarithms on both sides:

$$\ln(\alpha_i) + \ln\left(\prod_{j=1}^{n} X_j^{g_{ij}}\right) = \ln(\beta_i) + \ln\left(\prod_{j=1}^{n} X_j^{h_{ij}}\right) \tag{5.46}$$

Now remember that $\ln\left(\prod_{j=1}^{n} Z_j\right) = \sum_{j=1}^{n} \ln(Z_j)$ is true for any positive quantities Z_j and that $\ln X^p = \ln e^{p \ln x} = p \ln X$ is true for positive X and any real parameter p. Thus, we obtain

$$\ln\left(\prod_{j=1}^{n} X_j^{g_{ij}}\right) = \sum_{j=1}^{n} g_{ij} \ln(X_j) \tag{5.47}$$

and the corresponding equations for terms with h_{ij}. If we rename the variables $y_i = \ln X_i$ and rearrange the equation, we obtain the result

$$\sum_{j=1}^{n} g_{ij} y_j - \sum_{j=1}^{n} h_{ij} y_j = \ln(\beta_i) - \ln(\alpha_i) \tag{5.48}$$

Finally, we rename $a_{ij} = g_{ij} - h_{ij}$ for all i and j and define $b_i = \ln(\beta_i/\alpha_i)$. This algebraic maneuvering reveals the intriguing truth that the steady-state equations of an S-system, which is genuinely nonlinear, are indeed linear:

$$\begin{aligned}
a_{11}y_1 + a_{12}y_2 + a_{13}y_3 + \ \ldots\ + a_{1n}y_n &= b_1 \\
a_{21}y_1 + a_{22}y_2 + a_{23}y_3 + \ \ldots\ + a_{2n}y_n &= b_2 \\
a_{31}y_1 + a_{32}y_2 + a_{33}y_3 + \ \vdots\ + a_{3n}y_n &= b_3 \\
a_{n1}y_1 + a_{n2}y_2 + a_{n3}y_3 + \ \vdots\ + a_{nn}y_n &= b_n
\end{aligned} \tag{5.49}$$

For later biological interpretations, we just must remember to translate back from y's to X's with an exponential transformation, once the analysis is completed. The same procedure works for systems with independent variables. The reader should confirm this or see [25].

A final exceptional case of a nonlinear system with linear steady-state equations is the so-called lin–log model [27]. With the exception of these systems, very few systems of nonlinear models have steady-state solutions that can be computed explicitly with methods of calculus and linear algebra. Even mass-action and GMA systems do not allow us to do this.

As an interesting example of a canonical model from the literature [28], consider a toggle switch model for the SOS pathway in *Escherichia coli*, with which the bacterium responds to stress situations where DNA is damaged by sending out SOS signals; a diagram of the system is shown in **Figure 5.15A**. The system has the same equations in both the GMA and S-system formats.

The main components are two genes, *lacI* and *λcI*, which are transcribed into the regulator proteins LacR and λCI, respectively. The most interesting feature of the system is cross-inhibition: the expression of *lacI* is high when *λcI* has a low expression level and vice versa. Specifically, the promoter P_L of the *lacI* gene is repressed by λCI and the promoter P_{trc} of the *λcI* gene is repressed by LacR, which subsequently leads to increased expression of the gene *lacI*. DNA damage leads to activation of the protein RecA, which increasingly cleaves the λCI repressor protein. Reduced amounts of λCI relieve the inhibition of P_L and thus cause higher expression of *lacI*.

As discussed in Chapter 2, we identify the main players and categorize them as dependent and independent variables (see Figure 5.15B). Clear dependent variables, whose dynamics is driven by the system, are the two regulator proteins X_1 = LacR and X_2 = λCI, as well as X_3 = RecA. The genes, while obviously important biologically, are not modeled, because they serve in the model only as precursors of the proteins. A more difficult debate is whether the promoters should be dependent or independent variables. This is one of those decisions that need to be made during the model design phase that we discussed in Chapter 2. On the one hand, the promoters are involved in the dynamics of the system. On the other hand, we have no information on how they become available or how they are removed from the system. Weighing our options in favor of simplicity, we make the decision to consider the promoters as independent variables $X_4 = P_L$ and $X_5 = P_{trc}$ with constant values. They are included in the production terms of X_1 and X_2. Because the promoters do not respond to changes in the system, we let X_2 directly inhibit the production of X_1 and we let X_1 directly inhibit the production of X_2. In addition to the promoters, X_6 = DNA is modeled as an independent variable.

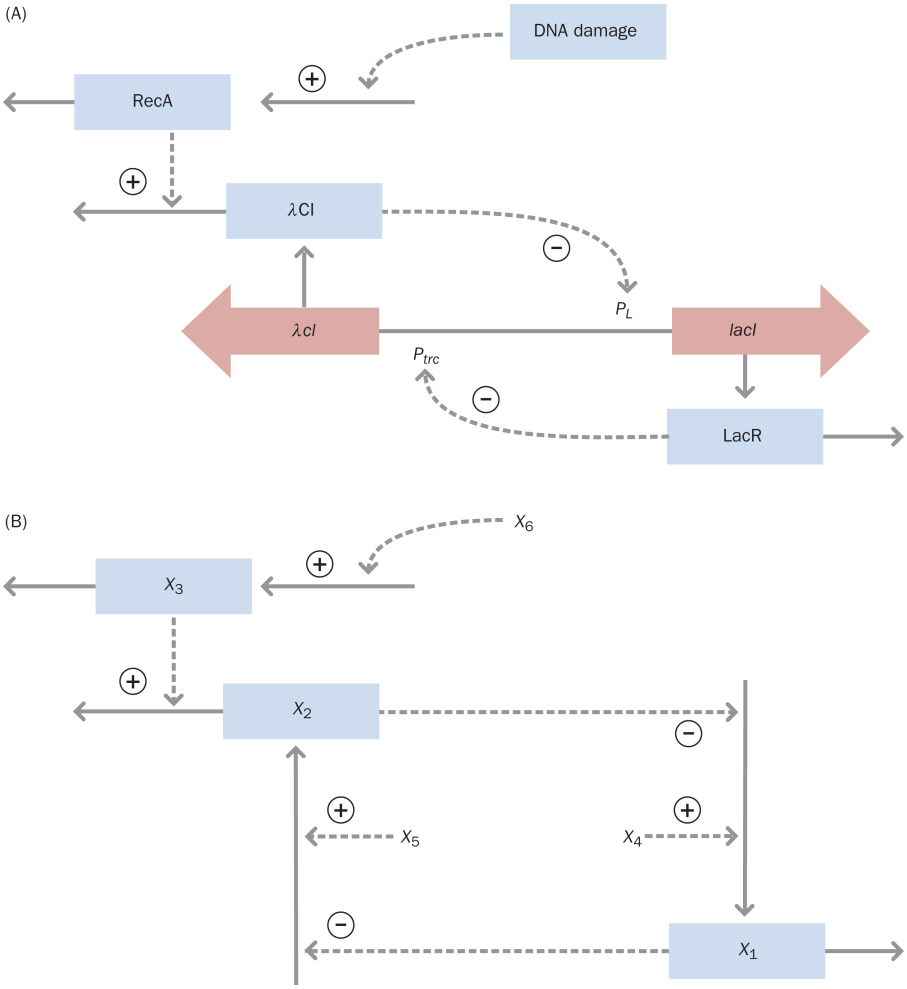


Figure 5.15 Cross inhibition in the SOS pathway of *E. coli*. (A) Diagrammatic representation of the SOS system. (B) Model representation, where the dependent variables are X_1 = LacR, X_2 = λCI, X_3 = RecA, and the independent variables are the promoters $X_4 = P_L$ and $X_5 = P_{trc}$, as well as X_6 = DNA damage. In the simplified model, X_1 and X_2 inhibit each other's production directly, rather than through the promoters (see the text for further discussion). (A, from Tian T & Burrage K. *Proc Natl Acad Sci USA.* 103 [2006] 8372–8377. With permission from National Academy of Sciences, USA.)

To set up the equation for X_1, we identify all production and degradation processes, as well as the variables that directly affect each. Indirect effects are not included; they are realized through the dynamics of the system. The production is affected by its gene's promoter P_L and the inhibition by λCI. Its natural degradation, for instance, through proteases, only depends on its concentration. Thus, the first equation is

$$\dot{X}_1 = \alpha_1 X_2^{g_{12}} X_4^{g_{14}} - \beta_1 X_1^{h_{11}}. \tag{5.50}$$

We do not know the parameter values, but we know already that all are positive, except for g_{12}, which represents the inhibition by λCI. From a point of mathematical frugality, we could merge the independent variable X_4, including its kinetic order g_{14}, with the rate constant α_1, but keeping them separate allows us later to do experiments with the promoter P_L more easily.

The second equation is constructed in a similar fashion. A slight difference here is that its degradation is hastened by X_3:

$$\dot{X}_2 = \alpha_2 X_1^{g_{21}} X_5^{g_{25}} - \beta_2 X_2^{h_{22}} X_3^{h_{23}} \tag{5.51}$$

It would again be more frugal to merge $X_5^{g_{25}}$ with the rate constant, but it might be beneficial to keep them separate, as explained for the first equation.

The third equation describes the dynamics of RecA. Let us begin with its degradation, which depends on its concentration and which we formulate as the power-law term $\beta_3 X_3^{h_{33}}$. For the production of RecA, we assume as a default that it is

constant. However, DNA damage activates this production process, so that X_6 should be included with a kinetic order. Thus, the equation reads

$$\dot{X}_3 = \alpha_3 X_6^{g_{36}} - \beta_3 X_3^{h_{33}}. \tag{5.52}$$

The formulation of the symbolic model (without numerical values) is now completed. Imagine how difficult this step would have been without canonical modeling, that is, if we had been forced to select nonlinear functions for all processes.

For further analysis and for simulations, we need to assign numerical values to all parameters, based on experimental data. This parameterization step is generally rather difficult (see Chapter 6), and we shall skip it for this illustration. Instead, we choose values for all parameters that are typical in BST (see Chapter 5 of [25]) and explore the functionality of the model with these values. The chosen parameter values are $\alpha_1 = 10$, $\alpha_2 = 20$, $\alpha_3 = 3$, $\beta_1 = \beta_2 = \beta_3 = 1$, $g_{12} = -0.4$, $g_{14} = 0.2$, $h_{11} = 0.5$, $g_{21} = -0.4$, $g_{25} = 0.2$, $h_{22} = 0.5$, $h_{23} = 0.2$, $g_{36} = 0.4$, $h_{33} = 0.5$, $X_1(0) = 7$, $X_2(0) = 88$, $X_3(0) = 9$, $X_4 = 10$, $X_5 = 10$, and either $X_6 = 1$ (no DNA damage) or $X_6 = 25$ (DNA damage).

We begin the simulation under normal conditions, where the system operates at its steady state. One hour into the simulation experiment, we induce DNA damage by resetting X_6 from 1 to 25. The system responds to the damage immediately with a fast increase in X_1 and a rapid drop in X_2 (**Figure 5.16**). We furthermore take into account that DNA damage is repaired after 10 hours. In response, λCI and LacR expression return rapidly to their normal states. Thus, as these representative results demonstrate, the model generates high expression of λCI under normal conditions and high expression of LacR under conditions of DNA damage. Moreover, the model exhibits sharp switches from one steady state to another, as has been observed in experiments. While the term "switch" has no firm definition, it is used to distinguish a step-like change from a more gradual transition.

The format for processes in GMA and S-system models was not developed haphazardly. It is again the result of Taylor's general approximation strategy, which is applied to all variables and processes upon a logarithmic transformation. It has turned out to be very convenient for biological systems analyses and at the same time is based on solid mathematical principles (see **Boxes 5.1** and **5.2**). Except for the logarithmic transformation, the approximation is a linearization as we discussed it earlier in the chapter (**Boxes 5.3** and **5.4**). However, the result is nonlinear and usually offers a wider range within which the approximation is good. Moreover, the power-law format of these models is so flexible that it admits models of essentially arbitrary complexity [19].

Of course, the power-law format is not the only one proposed in the literature. Other alternatives are discussed throughout this book. **Box 5.5** summarizes the steps for setting up a dynamic model from a diagram.

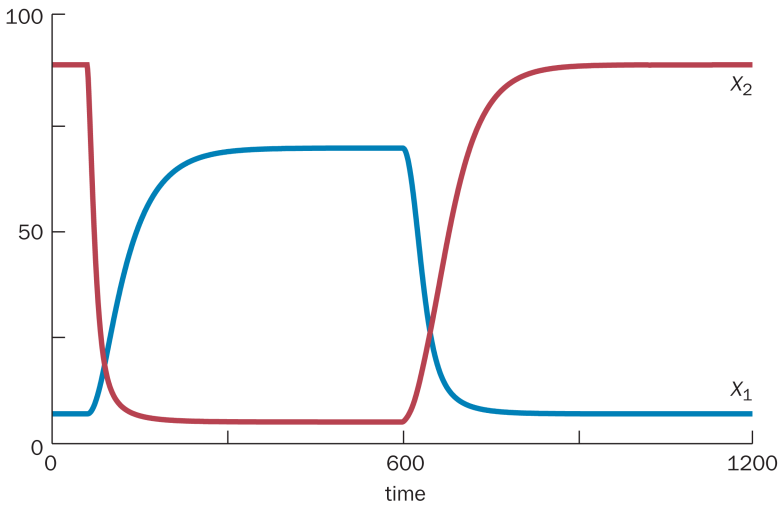


Figure 5.16 Simulation of switches between λCI and LacR expression states. The simulation begins with normal conditions, where the expression levels of λCI and LacR are 88 and 7, respectively. After 60 minutes, DNA is damaged. As a consequence, X_3 increases rapidly (not shown) and cleaves λCI, whose concentration drops in response. The inhibition of P_L is greatly reduced, and LacR rises. After 10 hours, DNA damage in the simulation is repaired, and λCI and LacR expression return to their normal states. Parameter values are given in the text.

BOX 5.3: POWER-LAW APPROXIMATION FOR FUNCTIONS OF ONE VARIABLE

The power-law approximation of a nonlinear function $F(X)$ results in a representation that consists of a power-law function of the type

$$v = \alpha X^g \approx F(X). \tag{1}$$

This approximation is obtained in the following fashion:

1. Take the logarithm of X and also of the approximated function F; both are assumed to be positive.
2. Compute the linearization of the result (see Box 5.1).
3. Take the exponential of the linearization result.

The result is a function v as shown in (1). This procedure is a bit convoluted, but there is a mathematically valid and correct shortcut, which says:

1. The kinetic order (exponent), g, is equal to the derivative of the approximated function F with respect to X, multiplied by X, divided by F, and evaluated at a chosen operating point.
2. The rate constant α is subsequently computed by equating the power-law term and the approximated function F at the operating point and solving for α.

Let us illustrate the short-cut of the power-law approximation procedure with the earlier example of a logistic function

$$F(X) = \frac{a}{b + e^{-cX}} \tag{2}$$

and assume that its parameter values are $a = 0.2$, $b = 0.1$, and $c = 0.3$ (**Figure 1**). We also suppose that X is some physical quantity whose relevant range is restricted to positive values. The task is to find parameters α and g so that αX^g is a power-law approximation of $F(X)$.

Using the shortcut, we must compute the derivative of $F(X)$, and, because there is only one dependent variable, this is an ordinary derivative:

$$\frac{dF}{dX} = \frac{a \cdot c \cdot e^{-cX}}{(b + e^{-cX})^2} \tag{3}$$

Now we multiply this derivative by X and divide by F and evaluate this expression at some operating point of our choosing:

$$g = \frac{dF}{dX} \cdot \frac{X}{F}\bigg|_{OP} = \frac{a \cdot c \cdot e^{-cX}}{(b + e^{-cX})^2} \cdot X \cdot \frac{(b + e^{-cX})}{a}\bigg|_{OP} \tag{4}$$

Algebraic simplification yields

$$g = \frac{c \cdot x \cdot e^{-cX}}{(b + e^{-cX})}\bigg|_{OP} = \frac{c \cdot X}{1 + b \cdot e^{cX}}\bigg|_{OP} \tag{5}$$

Let's pick an operating point—such as $X = 5$—and substitute numerical values for the parameters. The result is the desired kinetic order g:

$$g = \frac{0.3 \cdot 5}{1 + 0.1 \cdot e^{(0.3 \cdot 5)}} = 1.036. \tag{6}$$

The rate constant α is computed in a second step. The function and the approximation must have the exact same value at the operating point ($X = 5$), and with the parameter values chosen earlier, this value is $F(5) = 0.619$. Thus, we equate F and αX^g for $X = 5$ and solve for α:

$$F(X) = \alpha X^g\big|_{OP}$$

$$\alpha = F(X) \cdot X^{-g}\big|_{OP} = 0.619 \cdot 5^{-1.036} = 0.1168\,. \tag{7}$$

If we choose a different operating point, the approximation is different. For instance, for $X = 15$, we obtain

$$g = \frac{0.3 \cdot 15}{1 + 0.1 \cdot e^{(0.3 \cdot 15)}} = 0.4499 \tag{8}$$

and

$$\alpha = 1.8 \cdot 15^{-0.4499} = 0.5323. \tag{9}$$

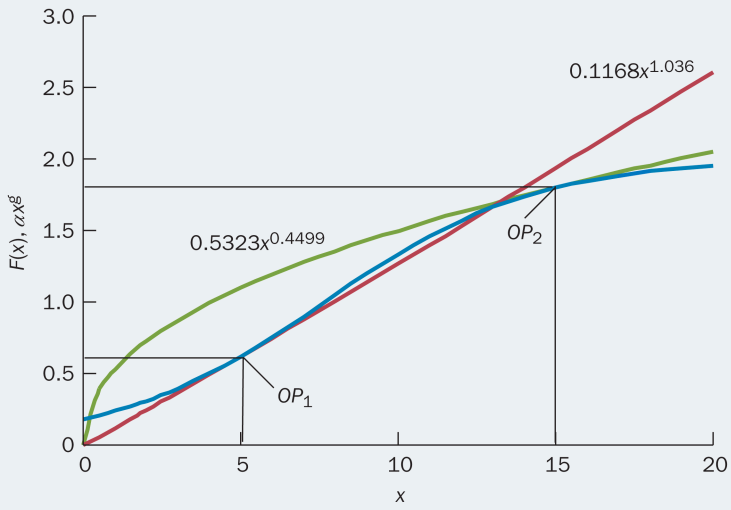


Figure 1 $F(x)$ and power-law approximations at the two chosen operating points. The ranges of the two approximations, where the approximation is acceptable, are quite different.

BOX 5.4: POWER-LAW APPROXIMATIONS OF MULTIVARIATE FUNCTIONS

Power-law approximations of functions with more than one dependent variable are constructed in an analogous fashion, except that we now use partial derivatives. We demonstrate the procedure for functions with two arguments; it is then easy to see how functions with more arguments are to be approximated.

As an illustration, suppose we are interested in a Michaelis–Menten process (5.37) that explicitly includes an enzyme whose activity changes as a consequence of altered gene expression and protein degradation. The first task is to make the enzyme activity E explicit in the Michaelis–Menten function. It is actually embedded in the parameter V_{max}, which equals the product of a turnover rate of the process and the total enzyme concentration, $k{\cdot}E$ (see Chapter 9) so that the function with explicit E reads

$$v_p(S,E)=\frac{k\cdot E\cdot S}{K_M+S} \tag{1}$$

The corresponding power-law representation will have the form

$$V=\alpha\cdot S^{g_1}\cdot E^{g_2} \tag{2}$$

and it is our task to compute its parameter values α, g_1, and g_2 such that V and v_P have the same value at some operating point of our choice and that the slopes in the directions of S and E are also the same. Only then is V a proper approximation of v_P. Fortunately, we may use the shortcut for kinetic orders, which is the same as in Box 5.3. So, we:

1. Compute partial derivatives.
2. Multiply them by the corresponding variable.
3. Divide by v_P at the operating point.

These algebraic operations yield the following:

$$g_1=\left.\frac{\partial v_P}{\partial S}\cdot\frac{S}{v_P}\right|_{OP}=\left.\frac{k\cdot E\cdot(K_M+S)-k\cdot E\cdot S}{(K_M+S)^2}\cdot\frac{S}{v_P}\right|_{OP} \tag{3}$$

$$=\left.\frac{k\cdot E\cdot K_M}{(K_M+S)^2}\cdot\frac{S\cdot(K_M+S)}{k\cdot E\cdot S}\right|_{OP}$$

$$=\left.\frac{K_M}{(K_M+S)}\right|_{OP}$$

$$g_2=\left.\frac{\partial v_P}{\partial E}\cdot\frac{E}{v_P}\right|_{OP}=\left.\frac{k\cdot S\cdot(K_M+S)}{(K_M+S)^2}\cdot\frac{E}{v_P}\right|_{OP} \tag{4}$$

$$=\left.\frac{k\cdot S\cdot(K_M+S)}{(K_M+S)^2}\cdot\frac{E\cdot(K_M+S)}{k\cdot E\cdot S}\right|_{OP}=1$$

The two results demonstrate different features of the power-law approximation. Most importantly, g_1 depends on the operating point, and its value changes depending on where we want to anchor the approximation. For substrate concentrations close to 0, g_1 is close to 1; for $S=K_M$, $g_1=0.5$; and for very large substrate concentrations, g_1 approaches 0. Thus, the kinetic order g_1 in this application is always between 0 and 1. By contrast, g_2 does not depend on the operating point at all, and its value is always 1. The reason for this special situation is that E contributes to the Michaelis–Menten process (1) as a linear function, which corresponds to a power-law function with kinetic order 1. Thus, with respect to E, the two functions V and v_P are the same for all operating values.

For V to be a proper approximation of v_P, the rate constant is again computed following the rationale that v_P and V must have the same value at the chosen operating point. Equating the two at a chosen operating point, for instance, $S=K_M$, leads to

$$v_p(S,E)=V(S,E)=\alpha\cdot S^{g_1}\cdot E^{g_2}\ \text{(for } S=K_M \text{ and some value of } E) \tag{5}$$

$$\alpha=v_P\cdot S^{-g_1}\cdot E^{-g_2}$$

$$=\frac{k\cdot E\cdot K_M}{2\cdot K_M}\cdot K_M^{-0.5}\cdot E^{-1}$$

$$=\frac{k}{2}\cdot K_M^{-0.5} \tag{6}$$

The original rate law and its power-law approximation are shown in **Figure 1** for parameter values $k=1.5$, $K_M=10$, and $E=2$. With respect to E, the power-law representation is exact. In other words, if the substrate concentration S is set at some operating point, the original and the power law are identical for all values of E.

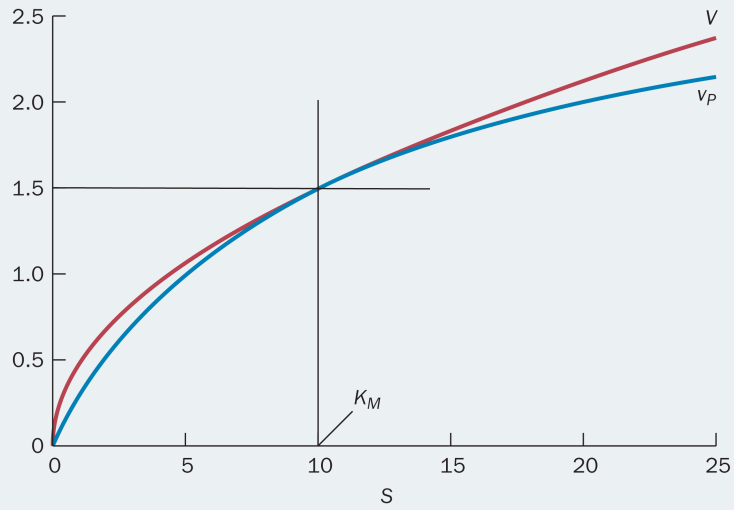


Figure 1 Power-law approximation of a Michaelis–Menten rate law (1) where E is also a dependent variable. At the operating point $S=K_M$, $E=2$, the original function and the corresponding power-law approximation have the same value and slope.

BOX 5.5: CONVERSION OF A SYSTEM DIAGRAM INTO AN ODE MODEL

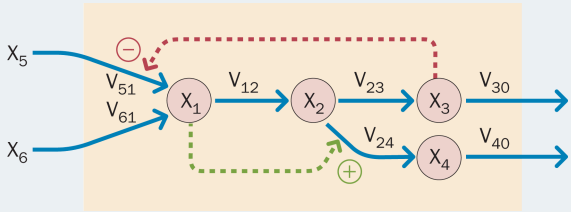


Figure 1 Pathway for illustrating model design in different mathematical formats. It is assumed that the two external inputs are independent variables and that material from X_3 and X_4 leaves the system. The designation of processes (indices) accounts for sources and targets, with 0 referring to the outside.

The design of an ODE model from a diagram is accomplished in three steps. It is demonstrated here with the example in **Figure 1**, which shows a system with two inputs, one feedback inhibition signal (red) and one feedforward activation signal (green). The shaded area reflects the inside of the system while the rest is outside.

1. Determine Dependent and Independent Variables and Set Up Symbolic Equations

Suppose the inputs X_5 and X_6 are independent variables, which are either constant or under the control of the modeler. The remaining four variables are dependent and are assigned differential equations. In a symbolic model, which reflects the structure of the modeled system, functions, and parameters are not yet specified. Thus, the ODEs contain unspecified processes as indicated in the diagram. The nomenclature is not ironclad, but it is convenient to indicate sources and targets in the naming of the processes. Thus, the process converting X_1 into X_2 receives the name v_{12}. Processes leaving the system receive indices of 0.

$$\dot{X}_1 = v_{51} + v_{61} - v_{12}$$
$$\dot{X}_2 = v_{12} - v_{23} - v_{24}$$
$$\dot{X}_3 = v_{23} - v_{30}$$
$$\dot{X}_4 = v_{24} - v_{40}$$

Note that the signals do not show up in these symbolic equations.

2. Choose Appropriate Functions for the Processes

Strictly speaking, this is the hardest step because there are no general rules regarding the optimal types of functions. One may choose functions that others have used in a particular application field, such as mass action or Michaelis–Menten functions in biochemistry, use generic power-law functions, which are known to be valid and rather unbiased approximations, or choose any other type of functions that can be justified. We set up three cases.

2.1. Traditional functions from biochemistry. Assuming that the system describes a biochemical pathway, it is common to use mass action functions (for v_{61}, v_{30}, v_{40}) as in Eqns. (2.1) and (9.2) or Michaelis–Menten rate laws (for $v_{51}, v_{12}, v_{23}, v_{24}$) as in Eqns. (5.33 and 9.10). Note that v_{51} is inhibited. A Michaelis–Menten function with inhibition is discussed in Chapter 9 (Eqn. 9.17) in great detail. If the system represents some other phenomenon, adequate functions must be chosen from the literature or ad hoc. With our choices, we obtain

$$v_{51} = \frac{V_{max51} X_5}{K_{M51}\left(1 + \frac{X_3}{K_3}\right) + X_5}$$

$$v_{61} = r_{61} X_6$$
$$v_{12} = \frac{V_{max12} X_1}{K_{M12} + X_1}$$
$$v_{23} = \frac{V_{max23} X_2}{K_{M23} + X_2}$$
$$v_{24} = X_1 \frac{V_{max24} X_2}{K_{M24} + X_2}$$
$$v_{30} = r_{30} X_3$$
$$v_{40} = r_{40} X_4$$

The model representation contains parameters, which are either simple rates in the mass action equations (r_{61}, . . .) or parameters associated with the Michaelis–Menten functions (V_{max} and K_M), including the inhibition constant K_3. Note that the indices of these parameters are chosen to align with the indices of the processes. Again, different names could be chosen, but any intuitive correspondence between processes and parameters facilitates later diagnostics and the interpretation of results.

2.2. Generalized Mass Action (GMA) representation. If this format is chosen, no pondering of different options for the process representations is needed: the format is prescribed, except for some modest variability in the names of indices. Here, we use the following notation: Each term v_{pq} receives a rate γ_{pq} with corresponding indices. The kinetic orders in such a term have the same indices, plus the index of the involved variable. Thus, the processes in GMA format are written as

$$v_{51} = \gamma_{51} X_5^{f_{513}} X_3^{f_{513}}$$
$$v_{61} = \gamma_{61} X_6$$
$$v_{12} = \gamma_{12} X_1^{f_{121}}$$
$$v_{23} = \gamma_{23} X_2^{f_{232}}$$
$$v_{24} = \gamma_{24} X_1^{f_{241}} X_2^{f_{242}}$$
$$v_{30} = \gamma_{30} X_3^{f_{303}}$$
$$v_{40} = \gamma_{40} X_4^{f_{404}}$$

The only parameter with a negative value is the inhibition parameter f_{513}. Note that we did not assign kinetic orders to the independent variables, as they are typically constant during each computational experiment, so that an additional kinetic-order parameter is superfluous.

2.3. S-system representation. In this case, the notation is entirely prescribed. Every ODE has at most two terms, one for the totality of all incoming processes and one for all outgoing processes. The parameters for the former are indexed α and g, and for the latter indexed β and h. The indices are directly given by the equation in which the term is located and the variables

affecting this term. The S-system representation differs from the GMA representation at the two branchpoints, X_1 and X_2, where converging and diverging fluxes are aggregated into single terms. Because of this aggregation, we directly formulate ODEs:

$$\dot{X}_1 = \alpha_1 X_3^{g_{13}} X_5^{g_{15}} X_6^{g_{16}} - \beta_1 X_1^{h_{11}}$$

$$\dot{X}_2 = \alpha_2 X_1^{g_{21}} - \beta_2 X_1^{h_{21}} X_2^{h_{22}}$$

$$\dot{X}_3 = \alpha_3 X_2^{g_{32}} - \beta_3 X_3^{h_{33}}$$

$$\dot{X}_4 = \alpha_4 X_1^{g_{41}} X_2^{g_{42}} - \beta_4 X_4^{h_{44}}$$

Note that it is not directly possible in all cases to match terms with processes v, due to the aggregation process. Instead, in this format one focuses on variables and represents mathematically all material that is coming in and all that is going out, without identifying specific processes. All parameters are positive except for g_{13}, which represents the inhibition of v_{51} by X_3. From the conservation of mass in the conversion of X_1 into X_2, we know that $\alpha_2 = \beta_1$ and $g_{21} = h_{11}$.

3. Choose Appropriate Parameter Values for All Functions

No matter which format is chosen, the determination of parameter values is always a challenge (Chapter 6). Ideally, the estimates are based on actual measurements, but they are not always available. As a consequence, this step often requires literature study, assumptions, and some guesswork.

Because the differential equations describing a dynamical system are typically nonlinear, we seldom have the luxury of being able to compute explicit solutions, in which each dependent variable is formulated as a function of time. Instead, we need to grind them out on a computer with so-called numerical integrators, which generate sets of numbers $(t_k, X_1(t_k), X_2(t_k), \ldots, X_n(t_k))$ for all time points t_k, which are approximate but usually very close to the truth. Importantly, we cannot write the solutions down as explicit functions, like $X(t) = X_0 e^{at}$, as we did in the case of linear differential equations.

Notwithstanding the computational issues, there is an entire branch of mathematics that studies dynamic phenomena, such as damped and **stable limit cycle oscillations**, bifurcations, and dynamical **chaos** with analytical and numerical methods. Even introductions to these types of investigations tend to require quite a bit of effort and mastery of a variety of mathematical techniques (see, for example, [29]), but we will discuss some of these issues at a somewhat superficial toward the end of this chapter.

While there are some nonlinear ODEs that do have explicit solutions, they are typically restricted to such special constellations of mathematical structures and parameters that they must be considered extremely rare exceptions in the real world. A consequence is that most dynamic analyses in systems biology are done by means of computer simulations. As we have already discussed, rigorous mathematical results are always to be preferred over a collection of numerical results, but in biological applications we can seldom get around computational simulation analyses.

Typical analyses fall into four classes:

1. Bolus experiments.
2. Persistent changes in structure or input (i.e., responses to step input).
3. Comprehensive screening experiments.
4. Analyses of critical points, where the system behavior changes qualitatively.

Bolus experiments usually begin with the system in a stable steady state. At some period into the experiment, one of the variables is altered. If this variable is a dependent variable, the simulation shows whether and how the system returns to the steady state. The "how" here refers to the time it takes to return and the shape of the transient, which is the collective name for the time course between the alteration (stimulus) and the move back to the old steady state, to a new one, or to an oscillatory behavior. Note that the time it takes to return to the steady state is, strictly speaking, infinite in most dynamical models. For instance, if the solution is a decreasing exponential function, the final value of 0 is mathematically only reached when x reaches infinity. A solution to this predicament is to study how long it takes the system to return to within a small percentage of the steady state, such as 1%.

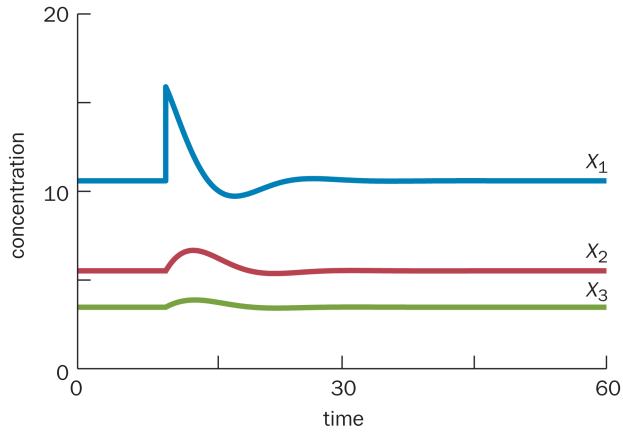


Figure 5.17 Simulation of a bolus experiment with a dependent variable. A 50% bolus of X_1 is supplied at time $t = 10$. All variables change in response, but the system quickly returns to its steady state.

As an example, consider again the simple generic pathway in Figure 5.14, where time units are minutes. Using the GMA representation with parameter values $\gamma_{11} = 20, f_{110} = 1, f_{112} = -0.75, \gamma_{12} = 0.4, f_{121} = 0.75, \gamma_{13} = 2, f_{131} = 0.2, \gamma_{22} = 0.5, f_{222} = 0.5, f_{224} = 1, \gamma_{32} = 2.5, f_{323} = 0.2, X_0 = 1, X_4 = 2$, the steady state is $X_1 = 10.59, X_2 = 5.52, X_3 = 3.47$. As a bolus experiment with a dependent variable, we raise X_1 at time 10 minutes by 50%. Immediately following, the other variables respond, showing slight **overshoots**, and the system returns to within 1% of its steady state in under 20 minutes (**Figure 5.17**).

If the altered variable in the bolus experiment is an independent variable, such as an external input, any alteration in its value will likely affect the steady state. Therefore, the bolus is usually applied only for a few time units. Afterwards, the altered variable is reset to its value at steady state. The simulation shows how the system responds to this temporary change. As an example, increase X_4 in the branched pathway by 50% at time $t = 10$ minutes and reset X_4 after at $t = 45$ minutes. The responses are quite different (**Figure 5.18**). In particular, the system essentially reaches a new steady state after about 35 minutes.

Simulations targeting persistent changes in structure or input are implemented with permanent alterations in parameter values or independent variables. A change in a parameter value could model a mutation or disease, which for instance could affect the rate of a transport process or an enzyme activity in a metabolic model. An alteration in an independent variable could be interpreted as a step change in environmental conditions, which could affect the availability of a substrate or other essential factor. In these types of simulations, one may study transients between the change and the completion of the response, but the target is usually the effect of the alteration on the steady state. An illustration is the previous example of a change in X_4. However, here one would not reset X_4 at $t = 45$ minutes, and the system would stay close to the elevated state we observe at $t = 45$ minutes. A different example would be the change in the parameter γ_{32}, which rep-

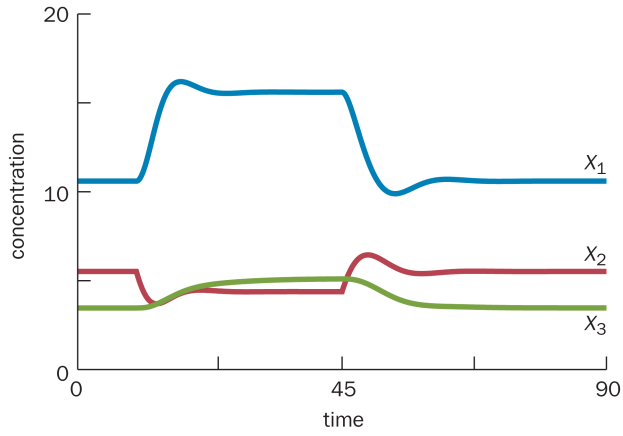


Figure 5.18 Simulation of a bolus experiment with an independent variable. A 50% bolus of X_4 is supplied at time $t = 10$ minutes and stopped at $t = 45$ minutes. During the altered setting, the system assumes a different steady state.

resents the rate of degradation of X_3. We discuss the system feature of adaptation in Chapter 9 more thoroughly.

Comprehensive screening experiments are usually launched if one is not entirely sure what to look for. Many options are possible. In a grid search, every parameter is allowed to take on a certain number of possible values; let us say 6. A complete simulation now computes the system (transients and/or steady state) for every combination of parameter settings, thereby producing a coarse—but in a way comprehensive—picture of possible responses. The problem with this approach is that it easily gets out of hand, because a system with n parameters requires 6^n simulations, and, for instance, $6^{20} = 3.66 \times 10^{15}$. Even for a computer executing 10 simulations per second, it would take over 100 billion hours (more than 11.5 million years) to complete the grid search! The rapid increase in numbers of simulations in these types of studies even has a name: combinatorial explosion. Various statistical techniques for overcoming this problem have been developed. They prescribe how to sample the vast space in a fashion that is manageable and at the same time statistically valid. Probably the best-known methods are the Latin square and Latin hypercube techniques [30]. An alternative strategy is Monte Carlo simulation, which we discussed in Chapter 2.

Analyses of **critical points** address situations where the system behavior changes qualitatively and often abruptly. Most changes in parameter values have an effect on the steady state and on transients between stimuli and responses, but the effects are usually gradual: the values go up or down a little bit, or the transient overshoots more and takes longer to come close to the steady state. However, even some extremely small alterations can throw a system very definitively off kilter. A qualitative change of this type may mean that the system becomes unstable, that one of the variables goes to zero or infinity, or that the system enters a stable oscillation that continues with the same amplitude infinitely. While these are typical examples, many weird things can happen in nonlinear systems [31], and some people have actually referred to the collection as a zoo of nonlinear phenomena.

Points within the space of all parameters (meaning specific combinations of parameter values) that have the characteristic of a threshold where the qualitative behavior changes are called **bifurcation points**, because there is a fork in the parameter space: above the bifurcation value the system does one thing and below the value the system does something else. Much of the field of nonlinear dynamics is concerned with mathematical ways of characterizing such bifurcation points due to their importance in defining abrupt transitions between classes of behavior. As a less elegant but often easier alternative, one can execute a large number of simulations, maybe with Monte Carlo methods, trying to establish domains where all parameter combinations lead to similar solutions, along with boundaries between domains where the system changes qualitatively.

As an example, consider a small system that could describe the mutual activation pattern of two genes (Figure 5.19A) [32]. Corresponding power-law equations with typical parameter values are shown in Figure 5.19B. Notice that one parameter (α_1) is left unspecified. It will be our critical parameter and could represent the strength of activation by a transcription factor. Even though the system has only two dependent variables, its responses can be quite complicated. For values of α_1 below 1, the system has a stable steady state, to which it returns after perturbations (see, for example, the exogenous increase in X_1 at time 10 in Figure 5.19C). As soon as α_1 is increased above 1, the steady state becomes unstable and the system enters an oscillation that will continue until some external force stops it (Figure 5.19D). The mathematical analysis of dynamical systems attempts to characterize these bifurcation points at which the system dynamics changes drastically. This analysis is typically quite involved but happens to be comparatively simple for the specific form of two-variable S-systems [33, 34].

5.4 DYNAMIC MOTIFS

Several chapters in this book have mentioned or will discuss "motifs," which are structural patterns that we encounter time and again in biological systems. The methods described in this chapter allow us to analyze the dynamics of such motifs.

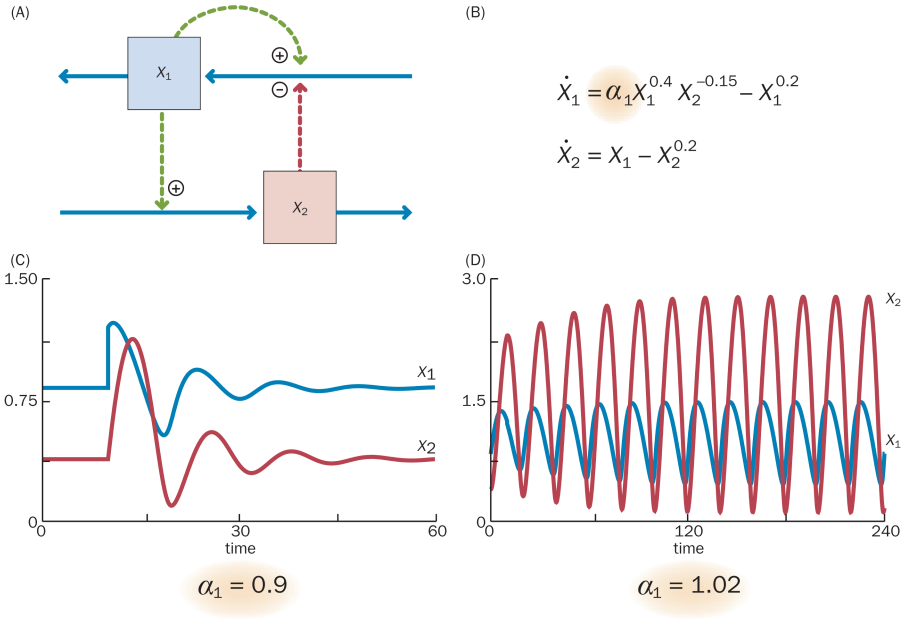


Figure 5.19 Model of a simple gene circuit in which two genes affect each other's expression level. (A) Diagram of the gene circuit. (B) Model equations. Changing the strength α_1 of the effect of a transcription factor on gene expression can lead to qualitatively different model responses (C, D).

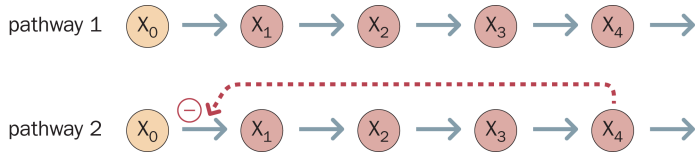


Figure 5.20 Assessment of a feedback motif by comparison. Here, a simple linear pathway without feedback (1) is being contrasted to a pathway with feedback inhibition (2).

A very prominent motif is **negative feedback**, which is ubiquitous in biology and also in many other areas, such as engineering. In this motif, the product of some pathway signals to the beginning of the pathway that not enough or too much of this product is currently available, and this signal alters the activity of the pathway toward correcting the situation. To see the effect of feedback inhibition, it is instructive to compare a simple linear pathway with feedback with one that does not have this feature (Figure 5.20). It does not matter from which application field this feedback system hails.

To understand the role of feedback inhibition, we formulate the two systems in parallel. A formal approach for this procedure is described in Chapter 6. Here, we use the power-law format and assume more or less arbitrary parameter values. For instance,

$$\begin{aligned}
\dot{X}_1 &= 3X_0 X_4^{g_{14}} - 1.2X_1^{0.5} \\
\dot{X}_2 &= 1.2X_1^{0.5} - 0.8X_2^{0.8} \\
\dot{X}_3 &= 0.8X_2^{0.8} - 1.3X_3^{0.6} \\
\dot{X}_4 &= 1.3X_3^{0.6} - 3X_4^{0.6} \\
X_0 &= constant
\end{aligned} \tag{5.53}$$

Note that the two pathways have exactly the same structure and equations, except that the inhibition parameter (shaded) is inactive for Pathway 1 ($g_{14} = 0$) but has a negative value for Pathway 2 (e.g., $g_{14} = -3$).

Not knowing much about the system, we initiate it at $X_i(0) = 1$ for $i = 0, \ldots, 4$ and solve it. The resulting time trends exhibit slight undershoots in X_3 and X_4 and then approach a steady state at about (6.25, 5.22, 4.03, 1.0) (Figure 5.21A). This steady state appears to be stable and, indeed, all eigenvalues have negative real parts and no imaginary parts.

As is common practice, we start the system from now on at its nontrivial steady state. As expected, all values remain constant. For our comparative experiment,

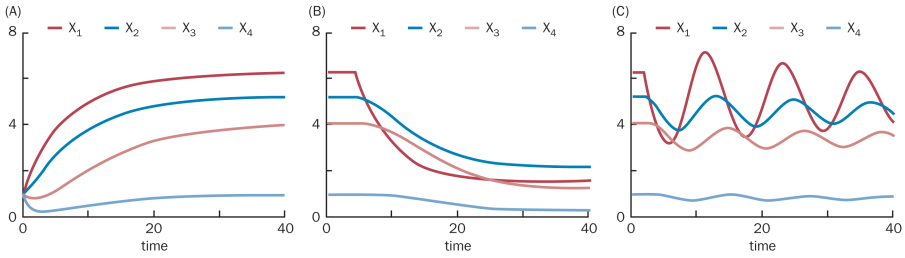


Figure 5.21 Responses of Pathways 1 and 2. Panels A and B: Time courses of the pathway without feedback, starting at (1, 1, 1, 1) (A) or at the steady state, with 50% reduction in input at time $t = 2$ (B). Panel C: Time courses of the pathway with feedback inhibition, corresponding to Panel B. The system now oscillates and the final value of X_4 is much closer to "normal" when the system possesses negative feedback.

we introduce a perturbation. Specifically, we may reduce the input X_0 by 50% at $t = 2$. In response, all variables decrease, with the value of the product X_4 settling in at about 0.315 (Figure 5.21B). This value is much lower than the "normal" steady-state value of 1, but the system has no way of compensating.

To assess the effect of feedback inhibition, we do the same experiment with Pathway 2, setting $g_{14} = -3$. Again, we set $X_0 = 1$ and start the dependent variables at the steady state, which is the same as before, because X_4 in our example has a steady-state value of 1. As before, the variables are constant until we reduce the input X_0 by 50% at $t = 2$. In stark contrast to Pathway 1, Pathway 2 shows damped oscillations, and the value of X_4 eventually settles at 0.825, which is much closer to the "normal" steady-state value of 1 than for Pathway 1, which was 0.315 (Figure 5.21C). Thus, the feedback mechanism has substantially ameliorated the drop in input, although it cannot fully compensate. Expressed differently, the system reduces "noise" by dampening its effect and softens responses to spurious signals. Chapter 10 discusses these features in signaling cascades. One might think that stronger feedback could compensate even more, but there is a limit to what the system can do. Exercise 5.48 asks you to find out what happens if you set $g_{14} = -5$.

Many biological systems involve more than one feedback inhibition signal or different types of **feedforward modulation**. For instance, Savageau [35] showed that a combination of feedback and feedforward inhibition can significantly improve the stability of a pathway, for instance, in the biosynthesis of amino acids. An interesting mechanism of feedforward activation has been found in glycolysis, where glucose 6-phosphate and fructose 1,6-bisphosphate activate the downstream enzyme pyruvate kinase [36]. In the case of the dairy bacterium *Lactococcus lactis*, this mechanism is critical for dealing with the erratic availability of substrate [37]. Mangan and Alon [38] systematically analyzed feedforward loop network motifs with three nodes, where the first node affects both the second and the third node while the second node affects just the third node. They demonstrated that different combinations of positive and negative feedforward effects led to significantly different behaviors. Some of these theoretical possible combinations are frequently found in the gene transcription networks of several organisms including *E. coli* and *S. cerevisiae*, while others are never observed and, indeed, exhibit inferior performance.

An entirely different but also frequently encountered motif consists of a **threshold** at which the system dynamics switches from one mode to another. An all too well-known practical example is exposure to sunlight. For moderate amount of exposure, our skin develops a tan, but if the exposure occurs somewhat longer, we get a sunburn, which is a totally different physiological response.

The biomedical mechanisms of suntan and sunburn are complicated, but the effect can even be demonstrated with a single equation. We will study these types of system in Chapter 10 in detail and here simply show the responses of such a mechanism. As an example, consider the equation

$$\dot{X} = 5 + 10\frac{X^4}{64 + X^4} - 5X^{0.5}, \tag{5.54}$$

which has a constant term, a Hill function (as in 5.38) and a power-law term. Solving the system with $X(0) = 1$, the system quickly reaches its steady state of about 1.02. Modest perturbations of X always return the system to this state. However, if the perturbation is stronger, the system all of the sudden moves to a different

steady state of about 8.43. This steady state is also stable, which we can easily demonstrate by modest perturbations around this higher value. At the same time, reducing X more substantially, the system returns to the lower steady state (**Figure 5.22**). Thus, the system is **bistable**, meaning that it has two stable steady states that are separated by an invisible threshold. We could imagine that the lower state corresponds to the suntan, whereas the higher state reflects the sunburn.

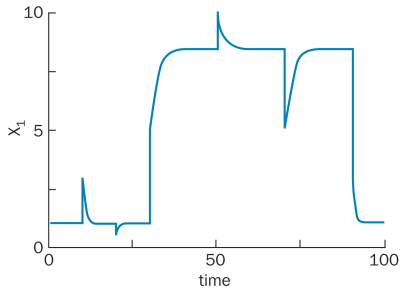


Figure 5.22 A single equation can exhibit switching between stable states. The simulation starts with X close to its low steady state of 1.02. Perturbations at $t = 10$ ($X = 3$) and $t = 20$ ($X = 0.5$) are tolerated, and the system returns to 1.02. However, a larger perturbation at $t = 30$ ($X = 5$) makes the variable jump to a new, much higher steady state of ~8.43. This state is again stable, as perturbation at $t = 50$ ($X = 10$) and $t = 70$ ($X = 5$) demonstrate. Setting $X = 3$ at $t = 90$ causes the system to jump back to the low steady state of 1.02.

5.5 ATTRACTORS OF DYNAMICAL SYSTEMS

As we discussed in Section 5.8, a stable steady state is a point to which trajectories of a system converge. In the language of nonlinear dynamics, a stable steady state is called an **attractor**, whereas an unstable steady state is called a **repeller**, because close-by trajectories are seemingly pushed away from it. Of interest now is that these more generic terms of attractors and repellers also include more complicated sets of points that draw trajectories in or push them away. These more complicated attractors or repellers consist of lines or volumes in two, three or more dimensions. They may even be fractal, which means that their patterns are the same at different magnifications, like the cover illustration of this book. Indeed, some attractors are as difficult to comprehend as they are stunningly "attractive," as a quick Google search for images of "strange attractors" attests.

Two classes of attractors, beyond steady-state points, are particularly important for systems biology. The first consists of limit cycles, which are oscillations with a constant amplitude that attract nearby trajectories, while the second type contains various chaotic systems, which however are not "all over the place" but move unpredictably although they never leave some constrained, often oddly shaped domain. The literature on both types is enormous, and the purely mathematical analysis of chaotic oscillators is quite complicated [39]. However, the basic concepts are easy to explore with simulations. A good introduction is [40].

5.5.1 Limit Cycles

We have several times used the term trajectory, referring to the solution of a system between two points in time and, specifically, between an initial value and a steady-state point. Trajectories may be linear or nonlinear, simply curved, oscillating, or very complicated. One particular type of trajectory is special in that it leaves and returns to the same point again and again. A simple example is the sine-cosine oscillation. As we discussed earlier (5.3), it can be represented as a pair of linear differential equations:

$$\begin{aligned} \dot{x} &= y \quad & x(0) &= 0 \\ \dot{y} &= -x \quad & y(0) &= 1 \end{aligned} \tag{5.55}$$

Instead of displaying this oscillation as a function of time (**Figure 5.23A**), it is often more instructive to show it as a phase-plane plot, where y (cosine) is plotted against x (sine). In the case of a sine oscillator, this plot is a circle (Figure 5.23B). Note that this representation does not directly show time so that one cannot see how fast the system completes a cycle. Neither can we directly see in which direction the solution progresses or where it starts. However, if we look at the time-course plot (Figure 5.23A), we see that the solution starts at (0, 1) and that x increases, while y decreases. Identifying these trends in the phase plane plot shows that the solution starts at "12 o'clock" and moves clockwise. The phase-plane representation explains the terminology of a periodic, closed trajectory, because after one period, the trajectory "closes" and continues on the same path time and again. Such a closed trajectory is also called an **orbit**.

A key feature of this center oscillation is that it is not stable (see Figure 5.8). We can easily test this fact by perturbing the solution. Specifically, we solve the differential equations, stop at some point in time, alter the numerical value of either x or y or both by a bit, and record what happens. A typical result is shown in Figure 5.23C, where variable x is artificially changed from its current value of about –0.959 to a

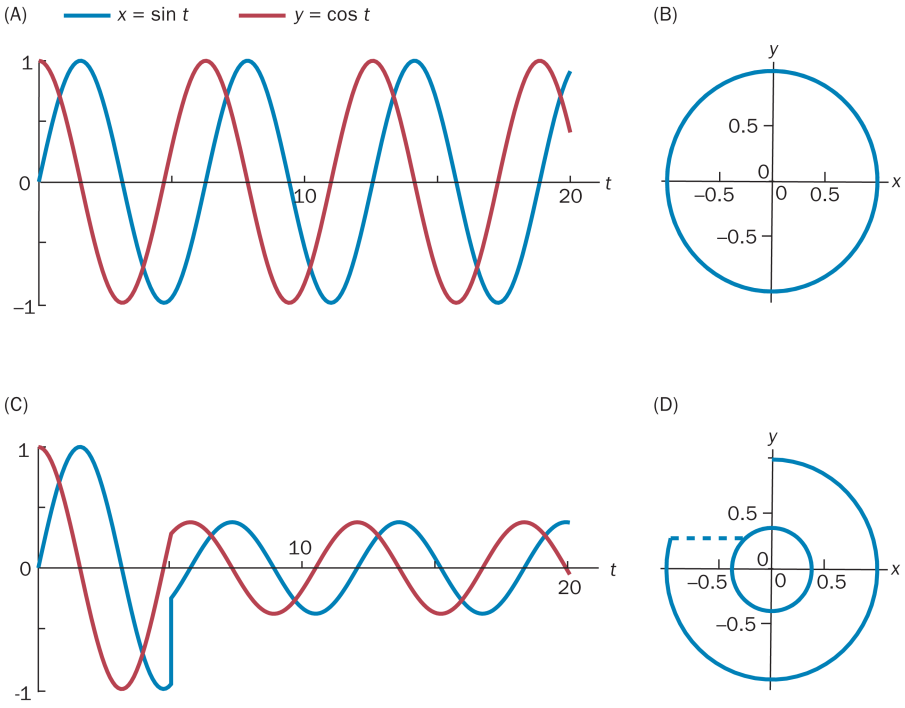


Figure 5.23 Sine–cosine oscillation. (A) The pair of linear ODEs in (5.54) corresponds to a sine–cosine oscillation that repeats every 2π time units. (B) The same oscillation is represented in the phase plane as a circle. (C) When the oscillation is perturbed, the oscillations retain the same frequency—possibly with a shift—but assume a new amplitude. (D) The corresponding phase-plane trajectory jumps (dashed line) to a new circle, which here has a smaller diameter.

value of –0.25 at time $t = 5$. Due to the coupling of x and y, y follows essentially instantaneously. Continuing the simulation, x and y still show sine and cosine oscillations, but the perturbation has done permanent damage: the amplitude has been altered and does not recover. The corresponding phase-plane plot is shown in Figure 5.23D. The solution starts again clockwise at "12 o'clock." At time $t = 5$, it has not yet completed one cycle, which it would do at $t = 2\pi$, but it is artificially perturbed when x is set to –0.25. This perturbation corresponds to the dashed horizontal line segment. From then on, the oscillation continues clockwise along the inner circle and never returns to the original circle, unless it is forced to do so.

More interesting periodic, closed trajectories are **limit cycles**, because they attract (or repel) close-by trajectories. If they attract, they are called stable oscillations or stable limit cycles, whereas repellers in this case refer to unstable limit cycles. Limit cycles are most easily discussed with examples. One famous case was proposed in the 1920s by the Dutch physicist Balthasar van der Pol (1889–1959) as a very simple model of a heartbeat [41–43]. Since his early proposal, many variations of this van der Pol oscillator (vdP) have been developed and applied to a wide range of oscillatory phenomena [40].

The prototype vdP has the format of a second-order differential equation,

$$\ddot{v} - k(1 - v^2)\dot{v} + v = 0, \tag{5.56}$$

where k is a positive parameter. The variable v with two dots represents the second derivative with respect to time, while v with one dot is the typical first derivative that we have used many times before. It is easy to convert this second-order equation into a pair of first-order differential equations, without losing any features of the original. This is done by defining a new variable

$$w = \dot{v}. \tag{5.57}$$

Because the derivative of the derivative of a variable is the second derivative, we immediately obtain

$$\dot{w} = \ddot{v}. \tag{5.58}$$

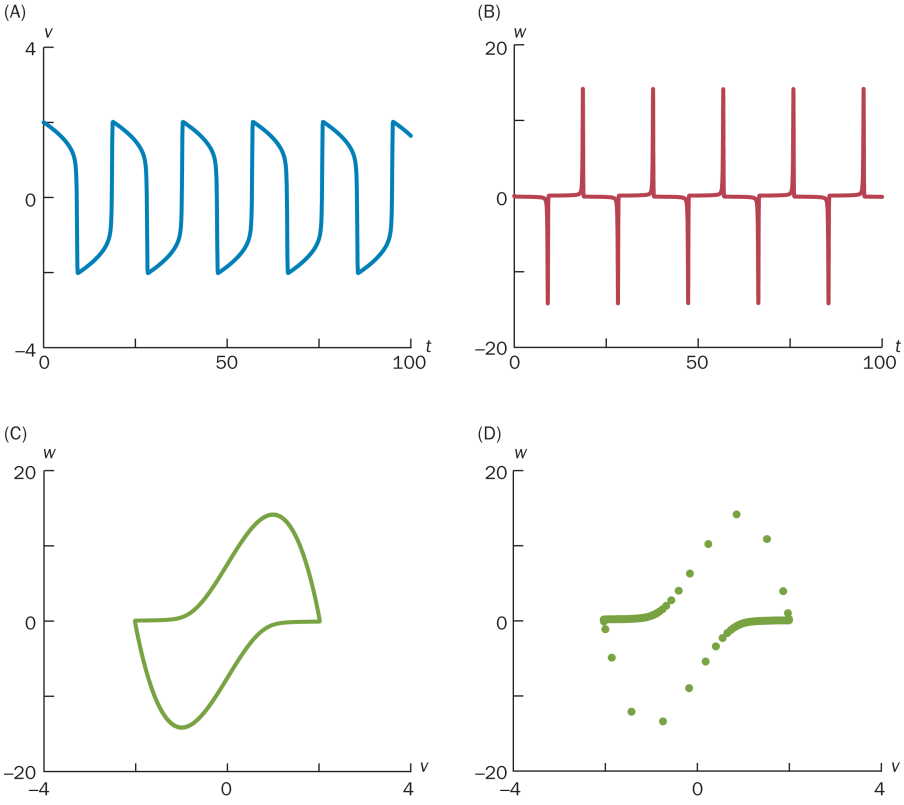


Figure 5.24 Visualization of the van der Pol oscillator (5.55). (A) The original variable v represents a limit-cycle oscillation that was designed to mimic a heartbeat. (B) The plot of the auxiliary variable w has the same frequency, but quite a different shape. (C) Phase-plane plot of w versus v. (D) Plotting the solution shown in (C) at spaced-out time points gives an idea of the speed of the oscillation in different phases of the oscillation. Dense points correspond to slow speed, while more widely separated points indicate a higher speed.

Rearranging Eqn. (5.56) and substituting the results in Eqns. (5.57) and (5.59) for $\dot{v}$ and $\ddot{v}$ results in a first-order system that is exactly equivalent with Eqn. (5.56), if the correct initial values are chosen, namely:

$$\begin{aligned} \dot{w} &= k\left(1-v^2\right)w - v \quad & w(0) &= \dot{v}_0 \\ \dot{v} &= w & v(0) &= v_0 \end{aligned} \tag{5.59}$$

To start the system, initial values must be specified at time 0 for v as well as $\dot{v}$.

Figure 5.24A confirms that the oscillation in v repeats with the same pattern. The secondary variable, w, oscillates with the same frequency, but quite a different shape (Figure 5.24B). Instead of displaying the oscillations as functions of time, we can again use the phase-plane, where w is plotted against v. The result is shown in Figure 5.24C. Starting at the point on the far right, the solution initially decreases clockwise. Displaying the solution only for time points spaced by 0.05 shows that the points are dense in some sections but not in others (Figure 5.24D). In the dense sections, the solution is comparatively slow, while it is fast for points that are farther apart.

Intriguingly, if the van der Pol oscillation is perturbed, it recovers and returns to the same oscillatory pattern. This feature is the key characteristic of a stable limit cycle. For instance, suppose the system is perturbed at t = 18.6. At this point, the system without perturbation has the values $v \sim 0.26$ and $w \sim 10$. Artificially resetting v to 0 yields the plots in **Figures 5.25** and **5.26**. It is evident that the original oscillation is regained, although with some shift in the direction of time. In a similar vein as van der Pol, Fitzhugh [44] and Nagumo [45] proposed a limit-cycle oscillator for action potentials in neurons. It is discussed in Chapter 12.

Limit cycles may have very different shapes. A flexible way to creating such shapes is to start with a limit cycle model in two-variable S-system format, for which Lewis [33] computed mathematical conditions. An example, adapted from [34] is

$$\begin{aligned} \dot{X}_1 &= 1.01\cdot\left(X_2^8 - X_1^{-4}\right) \quad & X_1 &= 1.7 \\ \dot{X}_2 &= \left(X_1^{-3} - X_2^4\right) & X_2 &= 0.9 \end{aligned} \tag{5.60}$$

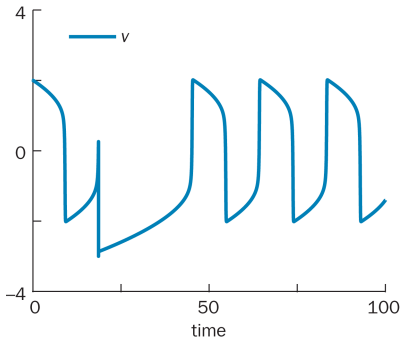


Figure 5.25 Perturbation of the van der Pol oscillator. As is typical for a limit cycle, the oscillation recovers from a perturbation and resumes the original pattern, but the perturbation may lead to a shift in phase.

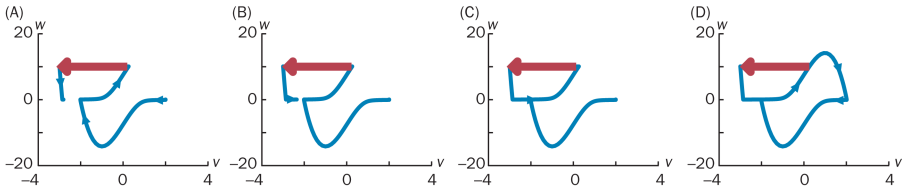


Figure 5.26 Phase-plane visualization of the perturbation in Figure 5.21. The van der Pol oscillator is perturbed at $t = 18.6$, where v is artificially changed from 0.26 to −3 (red arrow). (A), (B), (C), and (D) show, in the phase plane, the same trajectory, which is stopped at $t = 20$, 30, 40, and 50, respectively. The arrowheads have been added for illustration.

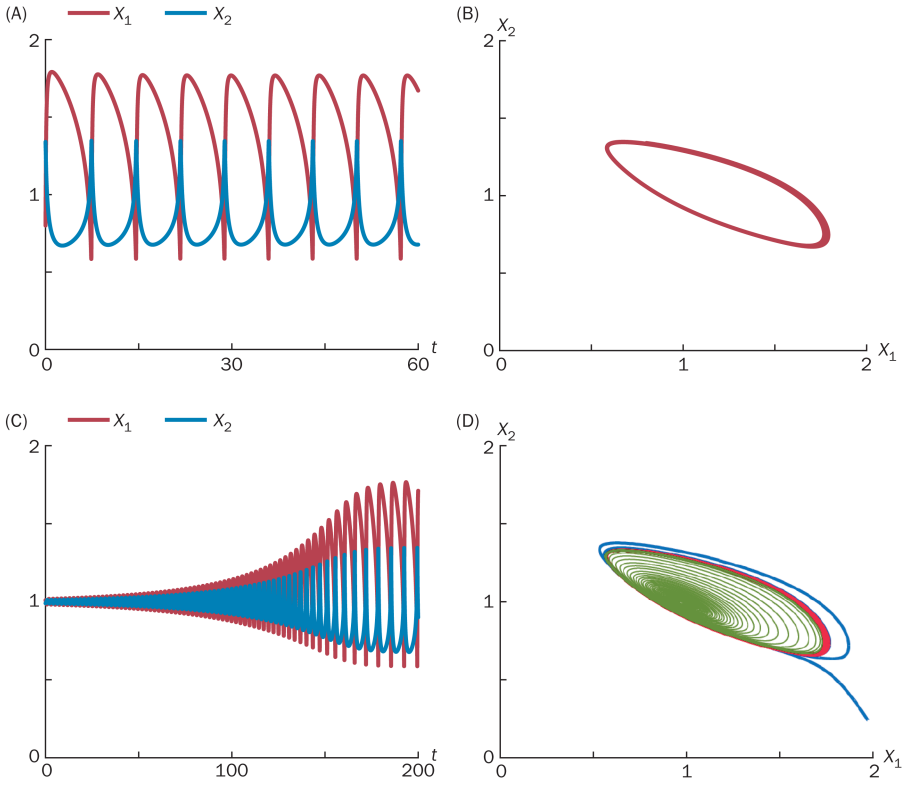


Figure 5.27 Limit cycles can have very different shapes. (A) Time-course representations of the limit cycle in (5.59). (B) The corresponding phase-plane plot. (C) Starting with initial values $(X_1, X_2) = (1.01, 0.99)$, which are close to the unstable steady state (1, 1), the oscillation spirals outward, eventually approaching the limit cycle. (D) Approach to the limit cycle from inside (green) and outside (blue) the limit cycle.

Figure 5.27 shows the oscillations of this system in time-course and phase-plane format. If one selects initial values inside the limit cycle but not exactly at the unstable steady state (1, 1), all trajectories are attracted to the limit cycle. Similarly, oscillations beginning close enough outside the limit cycle are attracted as well. If one starts further away, one of the variables may become zero, and the equations are no longer defined.

By multiplying both right-hand sides of the limit cycle system (5.60) with the same positive functions of X_1 and/or X_2, the appearance of the limit cycle oscillation can be changed quite dramatically (Figure 5.28), although the phase-plane plot remains unchanged (Exercise 5.50). In all cases the limit cycle remains stable and attracts close-by trajectories.

With the same methods as for systems without limit cycles, we can compute the steady states of the system in Eqn. 5.58. Since a negative power is not really defined for $X = 0$, we will not discuss a possibly trivial steady state. The nontrivial steady state is (1, 1). Checking the stability of this state we find a pair of complex conjugate eigenvalues with a real part close to 0, which means that the steady state is unstable. This observation reflects the fact that inside a stable limit cycle lies either at least one unstable steady state or an unstable limit cycle. One also notes that the imaginary parts are not zero, which indicates the potential for oscillations. If the rate constant 1.01 is changed to 1.00, the real parts are zero. At this bifurcation point, the stable limit cycle is "born" when the rate constant is slowly increased from below to above 1. For values smaller than 1, the steady state is stable.

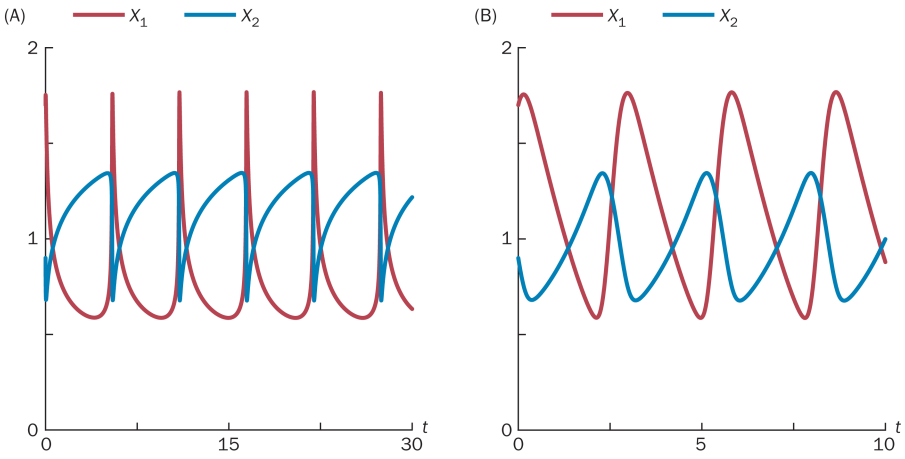


Figure 5.28 Variants of limit-cycle oscillations in (5.60). (A) Both right-hand sides were multiplied by $X_1^6 X_2^{-2}$. (B) Both equations were multiplied by $X_1 X_2^{-4}$. Note the different shapes and scales for the time axes and thus the different speeds of the oscillations.

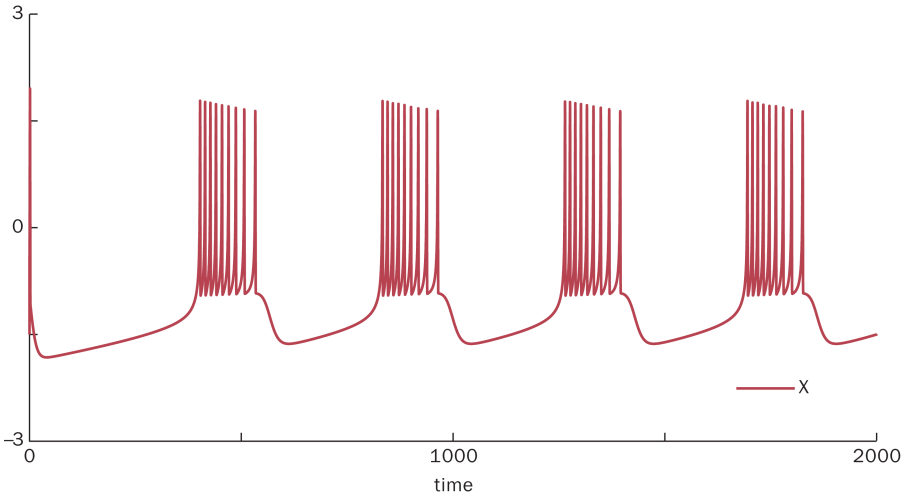


Figure 5.29 Bursting oscillations generated by (5.61). The equations yield a repeated pattern of rapid oscillations, separated by quiet phases. Adapted from [46].

In contrast to the relatively simple, regular oscillations of limit cycles, many neuronal systems display "bursting" dynamics [46], and similar oscillations can be observed as calcium bursts in the pancreas [47, 48] and even in transcription [49]. Often, these types of oscillations are considered biological "pattern generators." Surprisingly simple mathematical models can mimic these bursting oscillations (**Figure 5.29**), which are generated here by a model proposed by Keener and Sneyd [46]:

$$\begin{aligned} \dot{X} &= Y - X^3 + 3X^2 - Z + 2 \qquad X(0) = -1.5 \\ \dot{Y} &= 1 - 5X^2 - Y \qquad Y(0) = 4 \\ \dot{Z} &= 0.004(X + 1.6) - 0.001\,Z \qquad Z(0) = 2.5 \end{aligned} \tag{5.61}$$

5.5.2 Chaotic Attractors

In 1963, the American mathematician and meteorologist Edward Lorenz [50] studied weather patterns and noticed that even minute changes in initial values of describing models could lead to totally different outcomes if one waited for a while. It was Lorenz who introduced the concept of the butterfly effect. Using a simplified model of air flow, Lorenz was able to reproduce this effect with just three differential equations and thereby ushered in an intensive exploration of systems that had features similar to what Lorenz called a **strange attractor**. Trajec-

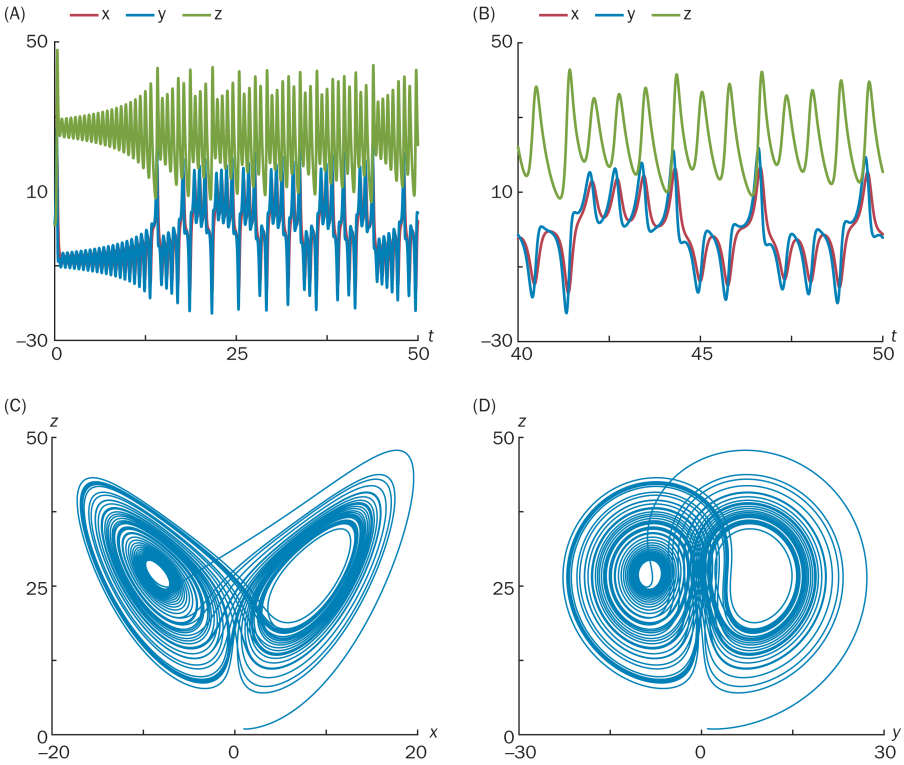


Figure 5.30 Simulation results for the strange attractor proposed by Lorenz [49]. (A) Time-course solution for $t \in [0, 50]$. (B) The same time-course in the interval $t \in [40, 50]$ exhibits irregularities more clearly. (C) and (D) Phase-plane plots corresponding to projections of the three-variable oscillator onto the x–z and y–z planes, respectively.

tories in this and other, similar systems are confined to some volume in the three-dimensional space, but their specific values cannot be predicted unless one solves the equations. The amazing aspect is that it is very easy to simulate these equations, which in Lorenz's notation read

$$\begin{aligned} \dot{x} &= P(y - x) \\ \dot{y} &= R\,x - y - x\,z \\ \dot{z} &= x\,y - B\,z \end{aligned} \tag{5.62}$$

Lorenz used the parameter values $P = 10$, $R = 28$, B = 8/3, along with the initial values (1, 1, 1). Figure 5.30 shows some output that demonstrates the erratic, chaotic trajectories of the system, which never exactly repeat. A common characteristic of chaotic oscillators is the fact that they are extremely sensitive to changes in initial values, and even a change to (1.001, 1, 1) results in a different temporal pattern. With such high sensitivity, even the solver settings can lead to different results. Interestingly, long-term simulations lead to an attractor in three dimensions that is overall always very similar, but different in details. Expressed differently, any trajectory lies somewhere within the attractor, but one cannot predict where exactly it will be without executing a numerical simulation (see Exercises).

The Lorenz attractor actually has three steady states. They are easy to compute and, as it turns out, they are all unstable (see Exercises). Starting a simulation close to one of the steady states fills one of the holes in the attractor with outward spirals that eventually merge into the attractor (Figure 5.31)

The Lorenz system is probably the best-known strange attractor, but many others have been documented. One strategy for creating chaotic oscillators is starting with a limit cycle and adding input from a trigonometric function to it. An example was shown with the blue-sky catastrophe in Chapter 2. A second example is the addition of a function like 0.00165 sin(1.5 t) to a limit cycle (see below). This situation is being considered in Chapter 12. Such an addition sometimes leads to chaos, but not always. It is possible that the oscillation become regular after some while or that one of the variables goes to $\pm\infty$ or 0 (see Exercises).

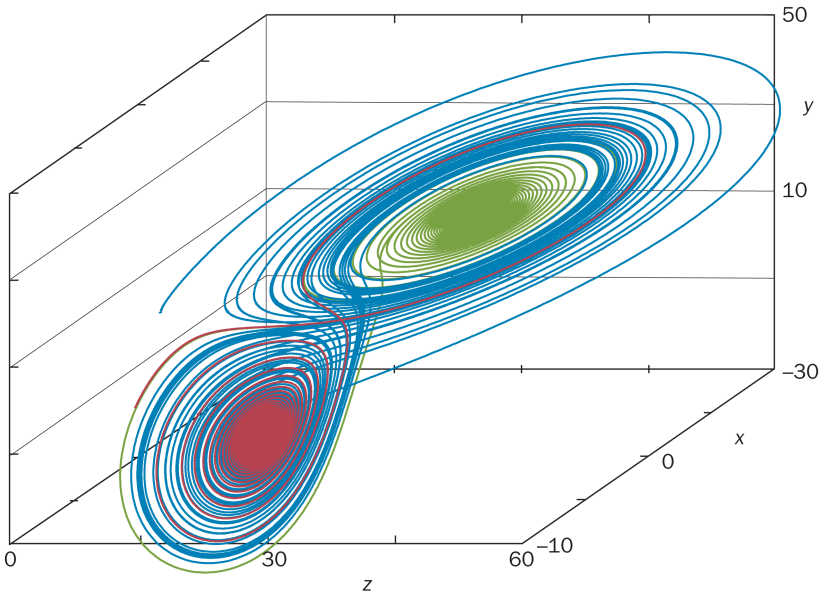


Figure 5.31 Filling the holes of the Lorenz attractor. Starting simulations close to the nontrivial unstable steady-state points yields outward spirals (red and green, respectively) that eventually merge into the attractor. The pseudo-three-dimensional plot demonstrates that the solutions are confined to two "disks." Where exactly the future solution will be within this attractor cannot be predicted; it can only be determined through simulations.

Even systems of differential equations that have a simple structure, such as BST and Lotka–Volterra models, can exhibit chaos (see Chapter 14 and [39, 51, 52]).

A particularly stunning example, due to its apparent simplicity, is the Thomas oscillator

$$\begin{aligned} x' &= -a\,x + \sin(y) \\ y' &= -a\,y + \sin(z) \\ z' &= -a\,z + \sin(x) \end{aligned} \tag{5.63}$$

whose appearance changes dramatically with the choice of the parameter a [53]. The complicated phase plane plot for $a = 0.18$ and initial values $(x, y, z) = (0, 4, 0.1)$ is shown in **Figure 5.32** for $t \in [0, 2000]$. Several other examples, including the systems of the continuous Rössler band [54] and of variations of the Thomas oscillators, as well as the discrete Hénon map [40], are discussed in the Exercises (5.52–5.59).

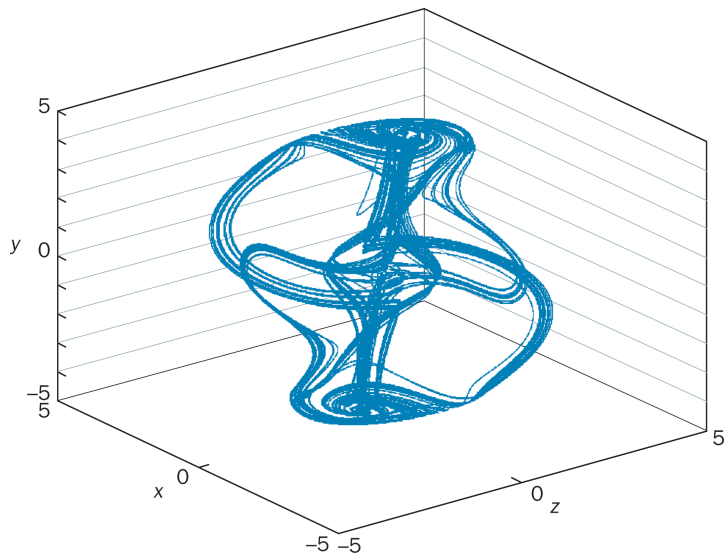


Figure 5.32 The Thomas oscillator. The deceivingly simple set of ODEs in (5.62) leads to a complicated strange attractor.

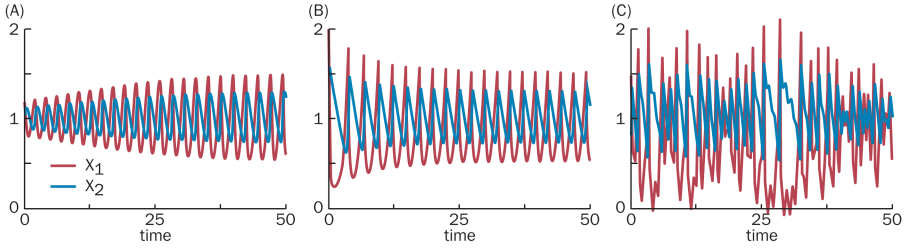


Figure 5.33 Poking a limit cycle. (A) The system oscillates outward toward the stable limit cycle. (B) The system oscillates inward toward the stable limit cycle. (C) Poking the limit cycle with a sine function renders it chaotic.

In addition to known chaotic oscillators, it is often possible to create deterministic chaotic systems by "poking" a limit cycles with a sine function, as mentioned before. Consider the following example. The two equations

$$\dot{X}_1 = 0.9\left(X_1 X_2^{-1} - X_1^2 X_2^5\right)$$
$$\dot{X}_2 = X_1^3 X_2^2 - X_1 X_2 \tag{5.64}$$

describe a stable limit cycle. Initiating the system with (1.2, 1) shows how the oscillations approach the limit cycle from the inside, while the initial conditions (2, 1) demonstrate the approach from the outside (**Figure 5.33A**, B). Now we add a simple sine function $a \cdot \sin(b \cdot t)$ to the first equation. For values of a and b that lead to oscillations that are too fast or too slow or too big or too small, the limit cycle is not affected much or blows up, but if the frequency and amplitude are just right, the oscillations become erratic. For instance, setting $a = 0.45$, $b = 3$, and starting the system at (1.4, 0.8) yields the chaotic oscillations in Figure 5.33C. Sometimes the system recovers from the induced chaos.

The poking mechanism may sound rather artificial but something similar happened in the famous crash of the Tacoma Narrows Bridge in Washington State. Like all structures of this type, this suspension bridge had its own frequency. On November 7, 1940, a steady wind hitting the bridge from the side caused so much resonant frequency that the bridge began galloping sideways with increasing amplitude and ultimately collapsed. Movies of the collapse are literally incredible!

Seeing how complicated the trajectories of seemingly simple systems can become, one may ask how overwhelmingly complicated large systems could be. Indeed, the sky is the limit. At the same time, large systems often contain mechanisms that keep them from "misbehaving," and it easily happens that the behavior of larger systems is quite dull. However, one never knows until one analyzes a system with rigorous mathematical means.

The chaotic systems introduced here are special because they are formulated with ODEs, which are entirely deterministic, which means that they are not affected by any type of randomness. In contrast to these representations, stochastic differential equations explicitly contain probabilistic terms [55]. An example is the Langevin equation, which is often used to describe Brownian motion and various sources of noise [56]. As an alternative, it is possible to formulate differential equations of the time-dependent probability density of stochastic variables. An example of this approach is the Fokker–Planck equation [57], which is typically used for describing the behavior of fluids. As it addresses a spatial phenomenon, the equation is based on partial differential equations [58].

5.6 EXERCISES

5.1 Modify the system representing a sine-cosine oscillator (5.3) in such a manner that the oscillations have a different amplitude and/or oscillate about a value different from 0.

5.2 Linearize a system of the form (5.7), with $f(X,Y) = c_1 - c_2 \cdot Y \cdot X^n / \left(c_3^n + X^n\right)$ and $g(X,Y) = c_4 - c_5 \cdot Y$, where c_i are positive parameters and $n = 1$, 2, or 4. Define initial values. Give a possible interpretation for these equations in terms of a metabolic or cellular system. Compute the functions and their approximations for different sets of parameter values. Consider different operating points. Write a brief report about your findings.

5.3 Compute symbolically, i.e., without particular mathematical formats, the linearization of a

generic two-variable linear system without input, as well as conditions for stability of a steady state.

5.4 Show that $y(x) = c \cdot e^x + e^{2x}$ is the solution of the differential equation $y' - y = e^{2x}$.

5.5 Show that the logistic function (5.35) is indeed the solution of the logistic differential equation (5.36). For this purpose, differentiate $N(t) = a/[b + e^{(-ct)}]$ with respect to t and express this derivative as a function of $N(t)$ itself.

5.6 Explore the effects of changes in α, β, and the initial size N_0 on the shape of the logistic growth function in ODE format (5.36).

5.7 Does it make sense biologically and/or mathematically if N_0 in (5.36) is greater than the carrying capacity? If feasible, simulate this case.

5.8 What happens mathematically in the logistic equation (5.36) if both α and β are negative? Is this situation biologically relevant?

5.9 Linearize the logistic function in (5.36) at its steady state(s) and determine stability.

5.10 Study the role of K_M and V_{max} in (5.37). How do K_M and V_{max} relate to each other?

5.11 Revisit the SOS system (5.50)–(5.52) and consider P_L and P_{trc} as dependent variables. Change the model equations correspondingly. Redo the simulation in Figure 5.16 with the expanded system. Write a brief report on your findings.

5.12 Prove the shortcut in Boxes 5.3 and 5.4 for computing power-law approximations.

5.13 Compute power-law approximations for the Hill function.

$$v_H = \frac{V_{max} S^n}{K_M^n + S^n}$$

for $n = 1$, 2, and 4 at different operating points. Select V_{max} and K_M as you like. Plot the approximations together with the original Hill functions.

5.14 Compute power-law approximations for the function $F(X_1, X_2, X_3) = X_1 \cdot e(X_2 + 2X_3)$ using different operating points. Assess the approximation accuracy for each operating point. Describe your findings in a report.

5.15 Compute the power-law approximation of the following function at the operating point $(X, Y) = (1, -0.5)$:

$$F(X,Y) = 3.2\, X^2\, e^{2Y+1}$$

5.16 Develop a power-law approximation of the function $f(X,Y) = 4X^3Y^{-1}$ at the operating point $(X, Y) = (1, 2)$.

5.17 Develop a power-law approximation of the function $f(X,Y) = 4X^3\sin(Y)$ at the operating point $(X, Y) = (1, 2)$.

5.18 Investigate similarities and differences between the S-system formulation (5.43) and the GMA form in (5.42) at the non-trivial steady state, where the equality $\beta_1 X_1^{h_{11}} = \alpha_2 X_1^{g_{21}} + \alpha_3 X_1^{g_{31}}$ should hold. For values of X_1 other than the steady state, the two sides of the equation are slightly different, but there are some special cases, where both are identical for all values of X_1. Describe and explain these exceptions.

5.19. Formulate symbolic, GMA, and S-system equations for the branched and cascaded pathways shown in the following figure. Use the standard notations as described in the text. Identify parameters with negative values.

5.20 If a Hill function with a Hill coefficient of 2 is approximated with a power-law function, in which

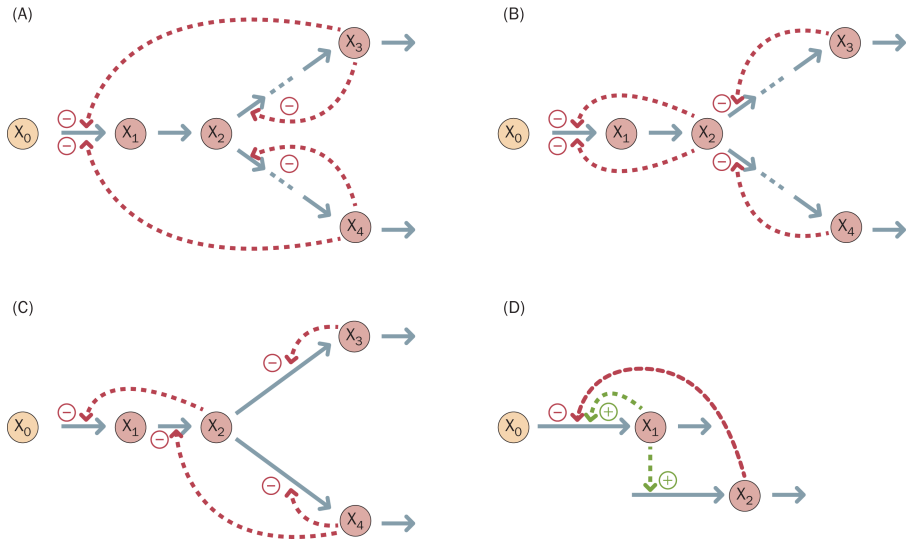


Figure 5.34 Four pathway patterns. These types of systems are encountered in different biosynthetic pathways. A–C contain feedback inhibition in different formations, while D contains feedback inhibition and also activation (green). The dotted lines indicate that there are additional steps in between. Consider X_0 as an independent variable.

numerical ranges do the kinetic order and the rate constant fall?

5.21 Implement the branched pathway system in Figure 5.14 with the following parameter values: $X_0 = 2$; $\gamma_{11} = 2$; $\gamma_{12} = 0.5$; $\gamma_{13} = 1.5$; $\gamma_{22} = 0.4$; $\gamma_{32} = 2.4$; $f_{110} = 1$; $f_{112} = -0.6$; $f_{121} = 0.5$; $f_{131} = 0.9$; $f_{222} = 0.5$; $f_{224} = 1$; $f_{323} = 0.8$. Start with $X_1 = X_2 = X_3 = X_4 = 1$, but explore different values between 0.1 and 4 for X_4.

5.22 Determine the S-system that corresponds to the GMA system in Figure 5.14 with the parameter values in Exercise 5.21. The two systems should be equivalent at the nontrivial steady state and as similar as possible otherwise. Perform comparative simulations.

5.23 Construct a diagram that could correspond to the following GMA system. Is the diagram unique? Discuss your answer.

$$\dot{X}_1 = \alpha_1 X_2^{g_{12}} - \beta_1 X_1^{h_{111}} - \beta_2 X_1^{h_{112}}$$
$$\dot{X}_2 = \beta_1 X_1^{h_{111}} - \beta_3 X_2^{h_{22}}$$

Approximate the GMA system with an S-system. Discuss all conditions under which the S-system is equivalent to the GMA system for all values of X_1 and X_2.

5.24 Compute the steady state of the S-system in (5.43) for arbitrary kinetic orders and for rate constants that are all equal to 1; assume $X_0 = 1$. Show all steps of your computations. Discuss which features (steady state, stability, dynamics . . .) of the system change if all rate constants are set to 10.

5.25 Compute the steady state of the S-system

$$\dot{X}_1 = \alpha \cdot \left(X_2^8 - X_1^{-4}\right)$$
$$\dot{X}_2 = X_1^{-3} - X_2^4$$

for $\alpha = 0.8$, 1, and 1.01. Show all steps of your computations. In each case, determine the stability of the system. Simulate the system in PLAS or some other software.

5.26 Under what conditions are a GMA system and the corresponding S-system equivalent even away from the operating point? List conditions and explain why the two are equivalent.

5.27 Compute all steady states (trivial and nontrivial) of the SIRS model

$$\dot{S} = 0.01 \cdot R - 0.0005 \cdot S \cdot I$$
$$\dot{I} = 0.0005 \cdot S \cdot I - 0.05 \cdot I$$
$$\dot{R} = 0.05 \cdot I - 0.01 \cdot R$$

5.28 Consider the following ethanol producing fermentation system, in which the growth of bacteria B is driven by the availability of substrate (glucose, G):

$$\dot{B} = \mu B - a B^2$$
$$\dot{E} = \frac{V_{max} B}{K_M + B} - c E$$

In this formulation, the growth rate is $\mu = \mu \frac{G^2}{K_I^{\,2} + G_{max}^2}$ and the symbols μ_{max}, K_I, V_{max}, and K_M, are positive parameters. Explain the biological meaning of each term in the two ODEs. Assuming constant glucose, as well as $G, E, B \neq 0$, compute the power-law approximation of the fermentation system at its steady state. Execute comparative simulations with the original system and its power-law approximation.

5.29 Compute the trivial and non-trivial steady state(s) of the Lotka–Volterra system

$$\dot{N}_1 = r_1 \cdot N_1 \cdot K_1 \cdot \left(K_1 - N_1 - a N_2\right)$$
$$\dot{N}_2 = r_2 \cdot N_2 \cdot K_2 \cdot \left(K_2 - N_2 - b N_1\right),$$

where $r_1 = 0.15$, $K_1 = 50$, $a = 0.2$, $r_2 = 0.3$, $K_2 = 60$, $b = 0.6$, $N_1(0) = 1$, $N_2(0) = 1$. Assess the stability of the steady state.

5.30 Enumerate the trivial and nontrivial steady state(s) of a Lotka–Volterra system with three dependent variables.

5.31 Is it possible to construct a model that is at once a Lotka–Volterra, GMA, and S-system? If your answer is no, explain why. If your answer is yes, provide an example.

5.32. Is every Lotka–Volterra system also a GMA system? Is every GMA system also a Lotka–Volterra system? Is every S-system a GMA system?

5.33. Determine matrices **A** for the different regions in Figure 5.8. Implement the systems and solve them, starting with different initial values, as illustrated in Figure 5.9.

5.34. Consider linear or linearized 2-variable ODE models with matrices **A** as specified in the following. For each case, compute eigenvalues and eigenvectors, determinant, trace, and discriminant. Locate the system in the plot of Figure 5.8. Compute sufficiently many trajectories to obtain an impression of the system behavior close to the steady state(s).

5.34.1 $\mathbf{A} = \begin{pmatrix} -1 & 2.5 \\ -1.5 & 1 \end{pmatrix}$

5.34.2 $\mathbf{A} = \begin{pmatrix} -1 & 2.5 \\ -0.4 & 1 \end{pmatrix}$

5.34.3 $\mathbf{A} = \begin{pmatrix} 2 & -1 \\ 2 & -2 \end{pmatrix}$

5.34.4 $\mathbf{A} = \begin{pmatrix} 1 & 1 \\ 4 & 1 \end{pmatrix}$

5.34.5 $\mathbf{A} = \begin{pmatrix} 1 & 1 \\ 1 & 1 \end{pmatrix}$

5.34.6 $\mathbf{A} = \begin{pmatrix} 0 & 0 \\ 0 & 0 \end{pmatrix}$

5.34.7 $\mathbf{A} = \begin{pmatrix} 1 & 1 \\ -1 & -1 \end{pmatrix}$

5.34.8 $\mathbf{A} = \begin{pmatrix} -1 & 1 \\ 1 & -1 \end{pmatrix}$

5.34.9 $\mathbf{A} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$

5.34.10 $\mathbf{A} = \begin{pmatrix} -1 & 0 \\ 0 & -1 \end{pmatrix}$

5.34.11 $\mathbf{A} = \begin{pmatrix} -3 & -1 \\ 4 & 1 \end{pmatrix}$

5.35 Using the concepts of Figure 5.8 and the associated discussion, construct a two-variable S-system that exhibits damped oscillations toward the steady state (1, 1). Sketch a pathway diagram corresponding to the model.

5.36 The following S-system models a stable limit cycle.

$$\dot{X}_1 = 2.2X_1^2X_2^3 - 2.2X_1X_2$$

$$\dot{X}_2 = 2X_1X_2^2 - 2X_1^3X_2^2$$

5.36.1 Confirm this statement.

5.36.2 Use Figure 5.8 and arguments associated with the dynamics of linear systems close to the steady state to show that the system possesses a steady state with unstable spiral behavior close by.

5.37 Calculate the symbolic steady-state equations of the SOS systems (5.50)–(5.52). Enter numerical values and compare with simulation results.

5.38 Does the following system have a trivial steady state? Describe and explain its behavior at its non-trivial steady state.

$$\dot{x} = f(x,y) = 1 - x^2y$$

$$\dot{y} = g(x,y) = xy^2 + y$$

5.39 Use the numerical system from Exercise 5.38 and multiply both right-hand sides first by 10 and then by –1. How would you interpret these operations? Do simulations and interpret the results.

5.40 Use the numerical system from Exercise 5.38 and vary each parameter individually by 5% up and down. Which parameter is most influential with respect to the system's steady state?

5.41 Analyze the system shown in **Figure 5.35**, using the S-system model

$$\dot{X}_1 = X_2^{-2}X_3 - X_1^{0.5}X_2$$

$$\dot{X}_2 = X_1^{0.5}X_2 - X_2^{0.5}$$

The system contains the independent variable X_3, which is constant. Begin the analysis by simulating the system with $X_1(0) = X_2(0) = X_3 = 1$. Compute all steady states and assess their stability. In the next step, vary $X_1(0)$ and $X_2(0)$ up and down and study the system responses. Finally, vary the constant value of X_3 up and down and study the system for different initial values $X_1(0)$ and $X_2(0)$.

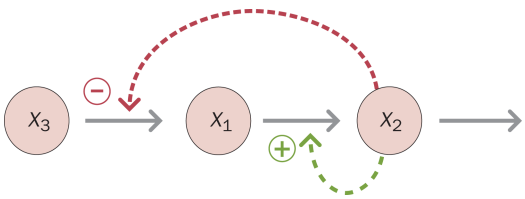


Figure 5.35 Generic pathway with two modulation signals. Even a seemingly simple pathway can exhibit quite complex responses.

5.42 Suppose all terms in a three-variable ODE system are approximated with power-law functions at a nontrivial steady state. Discuss the similarities and differences of the Jacobians of the two systems (the original ODE system and the approximating power-law system) at this steady state.

5.43 Compute the actual change in the steady-state value of X_2 of the system in Eqn. (5.31) if β_2 is increased or decreased by 1% or 0.1%. Compare the values to those computed based on the relative sensitivity.

5.44 Compute the relative sensitivities of the steady-state values of X_1 and X_2 of the system in Eqn. (5.31) with respect to α_1.

5.45 Compute the logarithmic gain of the steady-state values of X_1 and X_2 of the system in Eqn. (5.31) with respect to X_0. For this task, treat X_0 just like a parameter, that is, use the symbolic steady-state equations and compute, for instance, $\frac{\partial X_{2ss}}{\partial X_0} \cdot \frac{X_0}{X_{2ss}}$. Test the accuracy of changing X_1 by 0.1%, 1% and 10%. In PLAS, identify X_0 as independent variable by adding the line

&& X0

to the code and check the gain in the table of sensitivities.

5.46 Why should one typically expect the model of a biological system to be flawed if some of its sensitivities are in a range between 10 and 1,000?

5.47 What does it imply (A) mathematically or (B) for biological experiments associated with the system if a parameter value has very low sensitivities?

5.48 Test whether the feedback system in Eqn. (5.53) compensates better for a change in input if the magnitude of the inhibition parameter g_{14} is further increased, for instance, to $g_{14} = -5$. Compute the eigenvalues of the system and interpret the result.

5.49 Modify the limit cycle system (5.60) by multiplying only one of the two equations with a power function of X_1 and/or X_2. Record the effects on the oscillation and the phase-plane plot.

5.50 Explain why the multiplication of the limit cycle system (5.60) with a function of X_1 and/or X_2 changes the appearance of a limit cycle oscillation in the time domain but leaves the phase-plane plot unchanged.

5.51 Add to the first equation of the limit cycle system (5.60) the function $a \cdot \sin(b \cdot t)$. Start with a = 0.00165, test different values for b, and record the shapes of the oscillations in time-course and phase-plane representations. Subsequently alter a as well and record the effects.

5.52 Simulate the so-called Rössler attractor, which is defined by the differential equations

$$\dot{x} = -y - z$$

$$\dot{y} = x + ay$$

$$\dot{z} = bx - cz + xz$$

with $a = 0.36$, $b = 0.4$, $c = 4.5$. Choose different initial values, beginning with $x = 0$, $y = -3$, $z = 0$. Summarize your findings in a report.

5.53 Compute the steady state(s) of the Rössler system in Exercise 5.48 and assess its/their stability.

5.54 Explore the Lorenz oscillator for slightly changed initial values and parameter values.

5.55 Compute the steady states of the Lorenz attractor. Start simulations at or close to these steady states and study how the trajectories interact with the attractor itself.

5.56 Predict what the phase-plane plot of the blue-sky catastrophe in Chapter 2 looks like. Implement the system in PLAS or some other software and check your prediction.

5.57 As discussed in (5.61), Keener and Sneyd [46] proposed a model of calcium bursting.

5.57.1 Test whether the oscillations form a stable limit cycle. Support your answer with plots.

5.57.2 By altering the values of the parameters q, r, and s, test how the overall pattern of this oscillator can be altered. Support your answers by providing values of altered parameters values and plots. Note that several answers are possible.

5.57.3 Increase the interval between bursts, ideally without drastically changing the numbers of spikes per burst.

5.57.4 Increase the number of spikes within each burst.

5.57.5 Decrease the number of spikes within each burst to 1 or 2.

5.57.6 Create three sets of parameter values, similar to the original (within a factor of 0.5 to 2), for which the bursting pattern of bursts, separated by longer "calm" periods, disappears.

5.58 Explore the Thomas attractor in (5.63) with different, small positive values of. Solve the system with $x(0) = 0$, $y(0) = 4$, $z(0) = 0.5$ and study the phase plane of x and y and the pseudo-three dimensional representation of x, y, and z.

5.59 Explore the Thomas attractor in (5.63) with $a = 0$. Start the system with $x(0) = 0$, $y(0) = 4$, $z(0) = 0.5$ and solve the system first for 500 time units and then for much longer time periods. Study the phase plane of x and y and the pseudo-three-dimensional representation of x, y, and z.

5.60 Draw the pathway diagram corresponding to the system S4 associated with equation 5.24 and dynamics depicted in **Figure 5.5**, using the methods described in Box 5.5.

FURTHER READING

Easy Introductions

Batschelet E. *Introduction to Mathematics for Life Scientists*, 3rd ed. Springer, New York, NY, 1979.

Edelstein-Keshet L. *Mathematical Models in Biology*. SIAM, Philadelphia, PA, 2005.

Klipp E, Herwig R, Kowald A, Wierling C & Lehrach H. *Systems Biology in Practice: Concepts, Implementation and Application*. Wiley-VCH, Weinheim, Germany, 2005.

Voit EO. *The Inner Workings of Life. Vignettes in Systems Biology*, pp. xii + 210. Cambridge University Press, Cambridge, UK, 2016.

Voit EO. *Systems Biology. A Very Short Introduction*. Oxford University Press, Oxford, UK, 2020.

General Texts on Mathematical Biology

Adler FR. *Modeling the Dynamics of Life: Calculus and Probability for Life Scientists*, 3rd ed. Brooks/Cole, Totnes, UK, 2012.

Beltrami E. *Mathematics for Dynamic Modeling*, 2nd ed. Academic Press, Cambridge, MA, 1998.

Britton NF. *Essential Mathematical Biology*. Springer, New York, NY, 2004.

Edelstein-Keshet L. *Mathematical Models in Biology*. SIAM, Philadelphia, PA, 2005.

Ellner SP & Guckenheimer J. *Dynamic Models in Biology*. Princeton University Press, Princeton, NJ, 2006.

Haefner JW. *Modeling Biological Systems: Principles and Applications*, 2nd ed. Chapman & Hall, New York, NY, 2005.

Keener J & Sneyd J. *Mathematical Physiology*. Springer, New York, NY, 1998.

Kremling, A. *Systems Biology. Mathematical Modeling and Model Analysis*. Chapman & Hall, CRC Press, New York, NY, 2013.

Kreyszig E. *Advanced Engineering Mathematics*, 10th ed. Wiley, Hoboken, NJ, 2011.

Murray JD. *Mathematical Biology I: Introduction*, 3rd ed. Springer, New York, NY, 2002.

Murray JD. *Mathematical Biology II: Spatial Models and Biomedical Applications*. Springer, New York, NY, 2003.

Strang G. *Introduction to Applied Mathematics*. Wellesley-Cambridge Press, Wellesley, MA, 1986.

Strogatz SH. *Nonlinear Dynamics and Chaos*. Perseus Books, New York, NY, 1994.

Voit EO. *Computational Analysis of Biochemical Systems: A Practical Guide for Biochemists and Molecular Biologists*. Cambridge University Press, Cambridge, UK, 2000.

Yeargers EK, Shonkwiler RW & Herod JV. *An Introduction to the Mathematics of Biology*. Birkhäuser, Basel, Switzerland, 1996.

REFERENCES

1. Ferreira, A.E.N., *PLAS*, https://arquivo.pt/wayback/20230321162314/http://enzymology.fc.ul.pt/software/plas/, 2000–2024.
2. Grobman, D.M., Homeomorphism of systems of differential equations Doklady Akad. *Nauk SSSR*, 1959. **128**: p. 880–881.
3. Grobman, D.M., Topological classification of neighborhoods of a singularity in n-space. *Mat Sbornik*, 1962. **56**(98): p. 77–94.
4. Hartman, P.A., A lemma in the theory of structural stability of differential equations. *Proc Amer Math Soc*, 1960. **11**: p. 610–620.
5. Hartman, P.A., On the local linearization of differential equations. *Proc Amer Math Soc*, 1963. **14**: p. 568–573.
6. Guckenheimer, J. and P. Holmes, *Nonlinear Oscillations, Dynamical Systems, and Bifurcations and Vector Fields. Applied mathematical Sciences*. Vol. 42. 1983, New York: Springer-Verlag.
7. Atkinson, K., W. Han, and D. Stewart, *Numerical Solution of Ordinary Differential Equations*. 2009, Hoboken, NJ: John Wiley and Sons.
8. Jacquez, J.A., *Compartmental Analysis in Biology and Medicine*. 3rd ed. 1996, Ann Arbor, MI: BioMedware.
9. Chen, C.-T., *Linear System Theory and Design*. 1970, New York: Holt, Rinehart and Winston, Inc.
10. Chong, Y.D., *Complex Methods for the Sciences, Chapter 10.5*. accessed 2023, LibreTexts, https://libretexts.org/.
11. Wiens, E.G., *Egwald Mathematics*, http://www.egwald.ca/nonlineardynamics/twodimensionaldynamics.php, 2011.
12. Savageau, M.A., Concepts relating the behavior of biochemical systems to their underlying molecular properties. *Arch Biochem Biophys*, 1971. **145**: p. 612–621.
13. Von Seggern, D.H., *CRC Handbook of Mathematical Curves and Surfaces*. 1990, Boca Raton, FL: CRC Press.
14. Lotka, A., *Elements of Physical Biology*. 1924, Baltimore: Williams and Wilkins (reprinted as *Elements of Mathematical Biology*. 1956, New York: Dover).
15. Peschel, M. and W. Mende, *The Predator-Prey Model: Do we Live in a Volterra World?* 1986, Berlin: Akademie-Verlag.
16. Volterra, V., Variazioni e fluttuazioni del numero d'individui in specie animali conviventi. *Mem R Accad dei Lincei*, 1926. **2**: p. 31–113.
17. May, R.M., *Stability and Complexity in Model Ecosystems*. 1973, Princeton, NJ: Princeton University Press.
18. Voit, E.O. and M.A. Savageau, Equivalence between S-systems and Volterra-systems. *Math Biosci*, 1986. **78**: p. 47–55.
19. Savageau, M.A. and E.O. Voit, Recasting nonlinear differential equations as S-systems: A canonical nonlinear form. *Math Biosci*, 1987. **87**: p. 83–115.
20. Varga, J., C. Zhao, J.D. Davis, Y. Hao, J. Farrell, J. Gurney, E.O. Voit, and S. Brown, Antibiotics drive expansion of rare pathogens in a chronic infection microbiome model. *mSphere*, 2022. **7**(5): p. 1–18.
21. Savageau, M.A., Biochemical systems analysis. I. Some mathematical properties of the rate law for the component enzymatic reactions. *J Theor Biol*, 1969. **25**(3): p. 365–369.
22. Savageau, M.A., Biochemical systems analysis. II. The steady-state solutions for an n-pool system using a power-law approximation. *J Theor Biol*, 1969. **25**(3): p. 370–379.
23. Savageau, M.A., *Biochemical Systems Analysis: A Study of Function and Design in Molecular Biology*. 1976, Reading, MA: Addison-Wesley Pub. Co. Advanced Book Program, p. xvii, 379.
24. Torres, N.V. and E.O. Voit, *Pathway Analysis and Optimization in Metabolic Engineering*. 2002, Cambridge, UK: Cambridge University Press.
25. Voit, E.O., *Computational Analysis of Biochemical Systems: A Practical Guide for Biochemists and Molecular Biologists*. 2000, Cambridge, UK: Cambridge University Press, p. xii + 530.
26. Voit, E.O., Biochemical systems theory: A review. *Int Scholarly Res Network (ISRN—Biomath)*, 2013. **Article 897658**: p. 1–53.
27. Visser, D. and J.J. Heijnen, The mathematics of metabolic control analysis revisited. *Metab Eng*, 2002. **4**(2): p. 114–123.
28. Tian, T. and K. Burrage, Stochastic models for regulatory networks of the genetic toggle switch. *Proc Natl Acad Sci USA*, 2006. **103**(22): p. 8372–8377.
29. Strogatz, S.H., *Nonlinear Dynamics and Chaos*. 1994, Cambridge, MA: Perseus Books Publishing, LLC.
30. Hinkelmann, K. and O. Kempthorne, *Design and Analysis of Experiments*. 1994, Hoboken, NJ: John Wiley and Sons, Inc.
31. Epstein, I.R. and J.A. Pojman, *An Introduction to Nonlinear Chemical Dynamics. Oscillations, Waves, Patterns, and Chaos*. 1998, Oxford, UK: Oxford University Press.
32. Voit, E.O., Modelling metabolic networks using power-laws and S-systems. *Essays Biochem*, 2008. **45**: p. 29–40.
33. Lewis, D.C., Chapter 16: A qualitative analysis of S-systems: Hopf bifurcations, in *Canonical Nonlinear Modeling. S-System Approach to Understanding Complexity*, E.O. Voit, Editor. 1991, New York: Van Nostrand Reinhold.
34. Yin, W. and E.O. Voit, Construction and customization of stable oscillation models in biology. *J Biol Systems*, 2008. **16**(4): p. 463–478.
35. Savageau, M.A., Feedforward inhibition in biosynthetic pathways: Inhibition of the aminoacyl-tRNA synthetase by the penultimate product. *J Theor Biol*, 1979. **77**(4): p. 385–404.
36. Bali, M. and S.R. Thomas, A modelling study of feedforward activation in human erythrocyte glycolysis. *C R Acad Sci III*, 2001. **324**(3): p. 185–199.
37. Voit, E.O., A.R. Neves, and H. Santos, The intricate side of systems biology. *Proc Natl Acad Sci USA*, 2006. **103**(25): p. 9452–9457.
38. Mangan, S. and U. Alon, Structure and function of the feed-forward loop network motif. 2003. **100**(21): p. 11980–11985.
39. Sprott, J.C. and S.J. Linz, Algebraically simple chaotic flows. *Int J Chaos Theory Appl*, 2000. **52**(2): p. 1–20.
40. Thompson, J.M.T. and H.B. Stewart, *Nonlinear Dynamics and Chaos*. 1986, Chichester: John Wiley and Sons.
41. van der Pol, B. and J. van der Mark, Frequency demultiplication. *Nature*, 1927. **120**: p. 363–364.
42. van der Pol, B. and J. van der Mark, The heart beat considered as a relaxation oscillation. *Phil Mag (7th Series)*, 1928. **6**(Suppl.): p. 763–775.
43. van der Pol, B. and J. van der Mark, The heart beat considered as a relaxation oscillation. *Phil Mag (7th Series)*, 1928. **6**(Suppl.): p. 763–775.
44. Fitzhugh, R., Impulses and physiological states in theoretical models of nerve membrane. *Biophys J*, 1961. **1**(6): p. 445–466.
45. Nagumo, J., S. Arimoto, and S. Yoshizawa, An active pulse transmission line simulating nerve axon. *Proc IRE*, 1962. **50**(10): p. 2061–2070.
46. Keener, J. and J. Sneyd, *Mathematical Physiology*. 1998, New York: Springer-Verlag.
47. Bertram, R., A. Sherman, and L.S. Satin, Electrical bursting, calcium oscillations, and synchronization of pancreatic islets. *Adv Exp Med Biol*, 2010. **654**: p. 261–279.
48. Corrigan, A.M., E. Tunnacliffe, D. Cannon, and J.R. Chubb, A continuum model of transcriptional bursting. *Elife*, 2016. **5**.
49. Tunnacliffe, E. and J.R. Chubb, What is a transcriptional burst? *Trends Genet*, 2020. **36**(4): p. 288–297.

50. Lorenz, E.N., Deterministic nonperiodic flow. *Atmos Sci*, 1963. **20**(2): p. 130–141.
51. Vano, J.A., J.C. Wildenberg, M.B. Anderson, J.K. Noel, and J.C. Sprott, Chaos in low-dimensional Lotka–Volterra models of competition. *Nonlinearity*, 2006. **19**: p. 2391–2404.
52. Voit, E.O., S-system modeling of complex systems with chaotic input. *Environmetrics*, 1993. **4**(2): p. 153–186.
53. Thomas, R., Deterministic Chaos seen in terms of feedback circuits: Analysis, synthesis, "Labyrinth Chaos". *Int J Bifurcation Chaos Appl Sci Eng*, 1999. **9**(10): p. 1889–1905.
54. Rössler, O.E., An equation for continuous chaos. *Phys Lett*, 1976. **57A**(5): p. 397–398.
55. Øksendal, B., *Stochastic Differential Equations: An Introduction with Applications*. 5th ed. 1998, Berlin: Springer-Verlag.
56. Mainardi, F. and P. Pironi, The fractional Langevin equation: Brownian motion revisited. *Extr Math*, 1996. **10**(1): p. 140–154.
57. Medved, A., R. Davis, and P.A. Vasquez, Understanding fluid dynamics from Langevin and Fokker–Planck equations. *Fluids*, 2020. **5**(1): p. 40.
58. Renardy, M. and R.C. Rogers, *An Introduction to Partial Differential Equations,* 2nd ed. 2004, New York: Springer-Verlag.

Taylor & Francis
Taylor & Francis Group
http://taylorandfrancis.com

Optimal Models

6

When you have read this chapter, you should be able to:

- Appreciate different layers of optimality in biological models
- Discuss distinct types of challenges arising in the choice of a model
- Understand and explain the concepts of linear and nonlinear regression
- Describe the concepts of various search algorithms
- Execute ordinary and multiple linear regressions
- Estimate parameters for explicit nonlinear functions
- Estimate parameters for small systems of differential equations
- Realize the challenges of choosing appropriate mathematical representations
- Appreciate efforts to understand optimal designs of biological systems

The introductory chapters discussed that all models are approximations and therefore explicitly or implicitly based on numerous abstractions, simplifications, and omissions. In light of these discussions, the title of this chapter may sound peculiar: is it even possible to design a truly optimal model? Or is this a lofty ideal to be pursued but never achieved? The answer is somewhere in between. There are no absolutely optimal models that represent reality faithfully in every aspect, are computationally tractable, and yield novel insights. Nonetheless, we may be able to design models that are optimal in important aspects, even though they fall short in other aspects. A mathematician might call our pursuit for such models a constrained multi-objective optimization, that is, the search for the best possible solution satisfying several criteria under a set of predefined conditions.

The search for optimal models may coarsely be divided into three domains, which directly mirror the generic model design process described in Chapter 2. The first addresses the question of the best model structure for capturing the essence of a phenomenon. Roughly speaking, this task concerns the initial box-and-arrow diagram in which the model will be grounded (see Chapter 2). The second is the choice of the most appropriate mathematical representations. Should they be deterministic or stochastic? Discrete or continuous? What functional forms should we use? As an example, suppose experimental data display a sigmoidal trend (see later in this chapter): Should we try capturing this trend with a Hill function (Eqn. (9.15) in Chapter 9)? A logistic function (Eqn. (5.31) in Chapter 5)? An arctangent, part of a sine function, or the left half of a normal distribution? All these functions show similar sigmoidal trends, so, which would be best? We will see that this question is very hard to answer in general. Finally, once we have determined the structure of the model and all functional formats, we need to

DOI: 10.1201/9781003402770-6

determine numerical values of all parameters in these functions that are optimal in a sense that they reflect reality, usually in the form of experimental data, as well as possible.

Taken together, these three tasks are complicated and involve numerous conceptual and technical aspects that—while technical—also highlight the "art of modeling." The three tasks have a natural ordering, which reflects the sequence of steps in a typical model design process (Chapter 2). It would seem reasonable to follow this order, but because the first two topics are open frontiers within the field of systems biology, we begin the discussion with the third task of parameter estimation, which uses a set of fairly standardized optimization methods. While discussing these methods, we will assume that we have good reason to believe in our model diagram and the chosen functions.

6.1 PARAMETER ESTIMATION

The previous chapters focused on the analysis of mathematical and computational models without even mentioning where their numerical features and settings came from. In particular, we took the values for all model parameters as given. Clearly, these values are very important, because they connect an abstract structure, the symbolic model, with the reality of biomedical phenomena. Even the best model will not reproduce, explain, or predict biological functionality if its parameter values are wrong. Furthermore, very few analyses can be performed with a model when its parameter values are unknown. The reason for dedicating most of this chapter to parameter estimation is that this key step of modeling is complicated and has therefore been approached from many different angles. Some situations are easy to solve, but many others are not. In fact, if the system under investigation becomes even moderately large, parameter estimation often emerges as a severe and very challenging bottleneck. Despite enormous efforts by several scientific communities, including mathematicians, computer scientists, data scientists, engineers, and essentially all applied science fields, there is still no silver bullet solution for finding the best parameters for a model in a straightforward and efficacious manner. Instead, parameter estimation oftentimes requires different techniques for different aspects that demand dedicated effort easily spanning several months. A poignant example of the many steps involved in a thorough parameter estimation effort is a recent analysis of inflammatory processes [1].

The principles of parameter estimation are well understood, and numerous reviews are available (e.g., [2–5]). We will classify and dissect the estimation problem into manageable tasks, presenting solution strategies, and discussing some of the reasons for why different computer algorithms tend to succeed or fail. Even if you have no intentions to execute actual parameter estimation tasks in your work, you should at least skim through the chapter, in order to gain an impression of the concepts and difficulties of this task, while skipping some of the technical details.

Parameter estimation is an **optimization** task. The objective is to determine the parameter vector (i.e., a set of numerical values, one for each parameter of a model) that minimizes the difference between experimental data and the model. To avoid cancellation between positive and negative differences, the **objective function** to be minimized is usually the sum of the squared differences, rather than the raw differences, between data and model. This function is commonly called **SSE**, which stands for **sum of squared errors**, and methods searching for SSE are called **least-squares methods**.

As for other modeling tasks, the greatest divide in parameter estimation is between linearity and nonlinearity. If a model is linear—or can be transformed so that it is (truly or approximately) linear—then life is good: many methods are available, most of them are very effective, and there are numerous statistical tools characterizing the quality of the solution. By contrast, as soon as a model contains even a few nonlinearities, the vast majority of effective methods become inapplicable. The two dominant solution strategies in these cases are the use of brute computational force (gazillions of iterations of systematic trial and error) and the development of **search methods** that work with reasonable efficiency at least for some classes of problems.

The field has a second divide. First, it is sometimes possible to estimate parameters for each process in the model and then to merge all these local descriptions

in the hope of achieving a well-functioning model that encompasses the interplay of all processes in an adequate fashion. This bottom-up strategy has dominated parameter estimation for biological systems in the past and is still of great importance. However, it often fails, for a variety of suspected or unknown reasons. For instance, kinetic data for metabolic analyses may have come from in vitro experiments that were carried out under different conditions or may even have been obtained from different organisms. The result of this unfortunately typical situation is that the resulting model, which was constructed with the best available information regarding specific details, does not generate reasonable results for an all-encompassing model. The discrepancy requires revisiting all assumptions, as well as the information used to parameterize the model. Sensitivity analysis (Chapter 5) is a useful diagnostic tool for this step because it identifies parameters that affect the solution most and therefore need to be estimated most precisely, but it is seldom sufficient. Experience shows that any successes in applying the bottom-up strategy to moderately large systems with maybe 40 or 50 parameters are almost always the product of an often excruciatingly slow and cumbersome effort that may take months to complete.

A modern, very appealing alternative to bottom-up approaches is becoming increasingly feasible. This alternative depends on experimentally determined **time series** of genomic, proteomic, metabolomic, or physiological data. A general setting for measuring these data is as follows. Typically, the biological system is in its normal state. At the beginning of the experiment, the system is perturbed, and one measures the responses of several components at many time points afterwards. Hidden in these time profiles is very valuable information about the structure, regulation, and dynamics of the system. The trend toward using time series data is to be welcomed, because these data are obtained from the same organism under the same experimental conditions. The catch with time-series-based estimation is that the extraction of information is a difficult task that requires not only powerful computers but also ingenuity and insight into the biological system and into the intricacies of computational algorithms designed to execute the extraction in an effective fashion.

6.2 PARAMETER ESTIMATION FOR LINEAR SYSTEMS

6.2.1 Linear Regression Involving a Single Variable

In the simplest case of a single variable y depending on only one other variable x, the relationship between the two can be plotted in an x–y graph. This **scatter plot** of data gives an immediate impression of whether the relationship might be linear. If so, the method of linear **regression** quickly produces the straight line that optimally describes this relationship (**Figure 6.1**). The reason that this works so well—even on a moderately sophisticated pocket calculator—is that it is straightforward to express the difference Δ_i between each data point (x_i, y_i) and an imagined straight line traversing the data set and to compute the two characteristic features of the line, namely its slope and intercept, such that this line simultaneously minimizes all differences. Specifically, a linear regression algorithm solves the problem

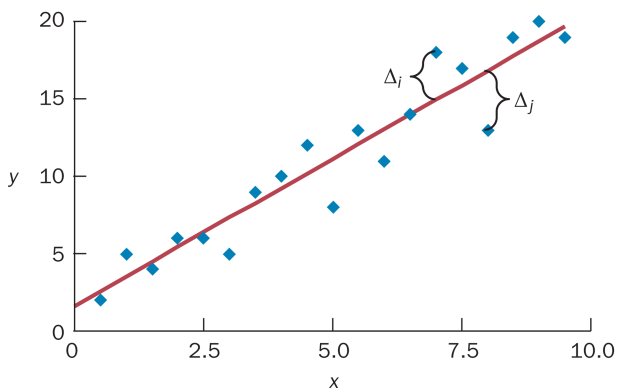


Figure 6.1 Simple linear regression. A more-or less-linear cloud is transected by the regression line, which is optimized in a sense that SSE, the sum of squared errors Δi, is minimized.

sketched in Figure 6.1, which shows data points (symbols) forming a stretched-out cloud, as well as the optimized straight line going through this cloud. This regression line is mathematically defined as the uniquely defined straight line that minimizes the sum of all squared residuals or errors Δ_i. Note that these errors are measured in vertical direction rather than the shortest distance from the regression line. The reason to square these quantities is simply that we do not want a positive Δ_i to compensate a negative Δ_j, and squaring is mathematically more convenient than using absolute values.

SSE is also called an **error function**, and parameter estimation tasks are therefore equivalent with tasks of minimizing error functions. The line **fitting** the data in Figure 6.1 has the representation

$$y = 1.899248x + 1.628571, \tag{6.1}$$

which is easily obtained from data, for instance, with Microsoft Excel®.

Linear regression is a straightforward exercise, and its result allows us to predict the expected value of y for some x that has never been analyzed. For instance, in Figure 6.1, we can rather reliably predict the approximate y-value that is expected for $x = 6.75$. Namely, plugging $x = 6.75$ into (6.1) yields $y = 14.4485$, which fits nicely into the picture.

Two issues require caution. First, extrapolations or predictions of y-values far beyond the observed range of x are not reliable. While we can reliably compute the expected y-value for $x = 6.75$, computing y for $x = 250$ suggests $y = 476.4406$, which is a correct computation but may or may not be a good prediction; we simply don't know. If the example is biological, the solution may be utterly wrong, because biological phenomena usually deviate from linearity for large x values and saturate instead.

The second issue to keep in mind is that the algorithm blindly yields linear regression lines whether or not the relationship between x and y is really linear. As an example, the relationship in **Figure 6.2** appears quite clearly to be nonlinear, maybe following a trend as indicated by the dotted green line. Nonetheless, it is easy to perform a linear regression, which quickly yields the relationship

$$y = 0.707068x + 8.151429. \tag{6.2}$$

It is evident that this model (the red line in Figure 6.2) does not make much sense. Thus, vigilance is in order. Often, simple inspection as in Figure 6.2 is sufficient. But one should also consider assessing a linear regression result with some mathematical diagnostics [6]. For instance, one might analyze the residual errors, which should form a normal distribution if the data really follow a linear trend. One may also study the lengths of "runs" of subsequent data falling below or above the regression line. For instance, in Figure 6.2, all low-valued data fall below the regression line, almost all data in the center lie above the line, and almost all high-valued data are below the line, which would be statistically highly unlikely if the data more or less followed the assumed linear relationship.

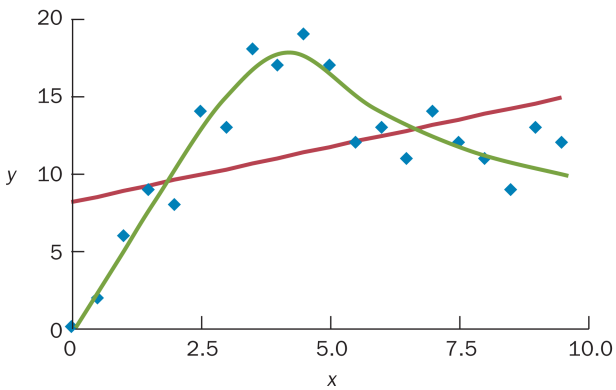


Figure 6.2 Misuse of linear regression. Fed with the presented data (blue), any linear regression algorithm will return the optimal linear regression line (red), even if the true relationship between x and y is presumably nonlinear, as indicated by the green line.

6.2.2 Linear Regression Involving Several Variables

Linear regression can also be executed quite easily in cases of more variables (see, for example, [6]). As the most intuitive example, suppose that some quantity Z depends on the two variables X and Y. For instance, the risk of disease may increase with blood pressure and body mass index. Thus, for every pair (X, Y) there is a value of Z. If the dependence is linear, the triplets (X, Y, Z) are located close to a sloping plane in three-dimensional space (**Figure 6.3**), and one may, for instance, use the **`regress`** function in MATLAB® for a multiple linear regression, which means linear regression of Z as a linear function of several variables. For more than two variables, the result is difficult to visualize, but the **`regress`** function nevertheless computes the higher-dimensional analog of a plane, which is called a hyperplane.

An example for two independent variables is given in Figure 6.3, which shows the data from **Table 6.1** in different perspectives. Red and blue dots in **(A)** and **(C)** correspond to data above and below the regression plane, respectively. The regression plane, computed with **`regress`**, is characterized by the function

$$Z = -0.0423 + 0.4344\,X + 1.1300\,Y. \tag{6.3}$$

It may be surprising that linear regression is applicable to some nonlinear functions. This option applies if a function permits a mathematical transformation that makes it linear. A well-known example is an exponential function, which becomes linear upon logarithmic transformation. An interesting case of linearization is the Michaelis–Menten rate law (MMRL) for enzyme-catalyzed reactions (see Chapters 2, 5, and 9).

As an illustration, suppose that the system of interest consists of a very short metabolic pathway with just two reactions, as shown in **Figure 6.4**. The pathway converts an initial substrate S into metabolite M in a reaction that is catalyzed by an enzyme E, whose activity does not change during our experiment. The metabolite is subsequently used up or degraded, and the degradation products of the reaction are of no particular interest and therefore not explicitly included in the model. A typical mathematical representation for the first process is a Michaelis–Menten function, which is characterized by three quantities: the substrate concentration S, the maximally possible rate or velocity of the reaction, V_{max}, which depends on the (constant) enzyme concentration, and the Michaelis constant K_M,

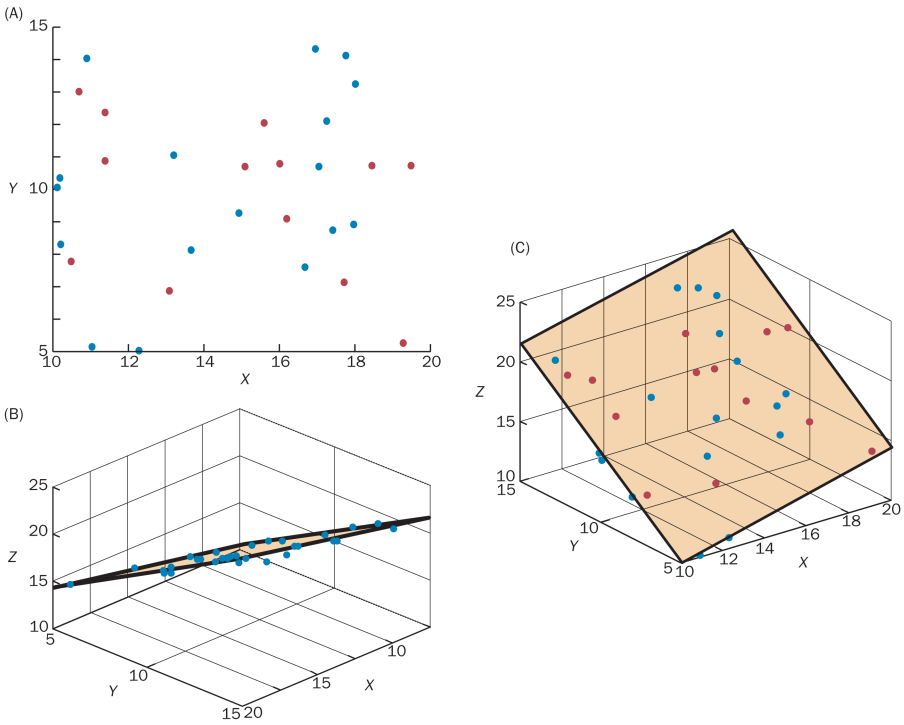


Figure 6.3 Result of a multiple linear regression analysis with the MATLAB® function regress and the data from Table 6.1. (A) The data are distributed throughout the ranges of X and Y. (B) A different perspective shows that all data points are close to a sloping plane, which in this perspective almost looks like a line. (C) In this third view, red dots lie above the regression plane (beige) and blue dots below.

TABLE 6.1: DATA FOR MULTIPLE LINEAR REGRESSION AND CORRESPONDING VALUES OF THE REGRESSION PLANE

X	*Y*	*Z (Data)*	*Z (Regression)*
17.43	8.72	16.73	17.38
10.49	7.75	13.58	13.28
16.69	7.58	15.47	15.77
17.06	10.69	19.32	19.46
13.10	6.85	13.82	13.39
18.46	10.71	21.06	20.07
17.27	12.10	20.55	21.13
16.04	10.77	19.15	19.10
14.95	9.25	16.68	16.90
17.74	7.11	16.30	15.70
17.98	8.89	17.38	17.82
13.67	8.12	14.89	15.07
15.09	10.68	19.42	18.58
10.22	8.28	13.27	13.75
11.04	5.12	10.08	10.54
11.39	12.34	19.60	18.85
11.40	10.87	17.57	17.19
10.71	12.98	19.93	19.28
10.20	10.33	15.45	16.06
12.30	5.02	10.95	10.98
16.97	14.32	22.96	23.51
17.79	14.10	22.68	23.62
13.22	11.05	18.06	18.19
10.91	14.01	20.35	20.53
19.30	5.26	14.31	14.29
15.60	12.04	21.42	20.34
18.02	13.23	22.52	22.73
10.12	10.05	15.15	15.71
19.50	10.73	20.77	20.55
16.22	9.07	17.56	17.25

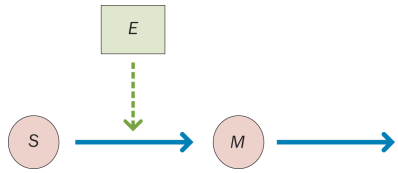


Figure 6.4 A simple model for the dynamics of a metabolite M. *M* is the product of a reaction, which is catalyzed by enzyme *E* and uses substrate *S*. *M* is degraded, but the product of this process is not specified.

which quantifies the **affinity** between the enzyme and its substrate (Figure 6.5); for details see Chapter 9. The default description for the degradation of the metabolite is a so-called **first-order process** with rate constant c, which implies that the speed of degradation is proportional to the current metabolite concentration M (Figure 6.6). Thus, the overall change in metabolite concentration, $\dot{M}$, is given by the difference between production and degradation, and, with the typical functions discussed earlier, it reads

$$\dot{M} = \frac{V_{\max} S}{K_M + S} - cM. \tag{6.4}$$

The model contains three parameters, $V_{\max}$, KM, and c, as well as the concentration of S, which may or may not be constant. Suppose someone had executed experiments by varying levels of S and measuring the production of metabolite (without degradation), thereby generating data consisting of pairs (S, v), where $v = V_{\max} S/(K_M + S)$. For clarity of illustration, let us suppose these data are error-free (Table 6.2 and Figure 6.5A). Our task is to estimate optimal values for the

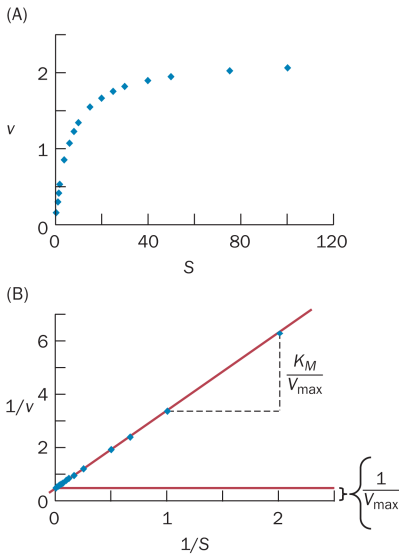


Figure 6.5 Noise-free data of a Michaelis–Menten process. The process is part of the model (6.4) and was measured in a separate experiment. (A) Raw data. (B) Inversely plotted data and determination of slope and intercept.

TABLE 6.2: MEASUREMENTS OF REACTION SPEED *V* VERSUS SUBSTRATE CONCENTRATION *S*

S	*v*
0.5	0.16
1	0.30
1.5	0.42
2	0.52
4	0.85
6	1.06
8	1.22
10	1.34
15	1.54
20	1.67
25	1.75
30	1.81
40	1.90
50	1.95
75	2.03
100	2.07

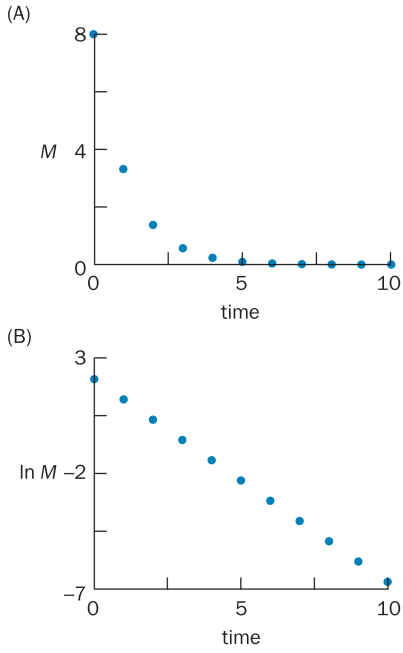


Figure 6.6 Degradation of *M* in (6.4). In Cartesian coordinates (A), the function is exponential. It becomes linear for the logarithm of *M* (B).

TABLE 6.3: TIME MEASUREMENTS OF *M* AND LN(*M*) FOR ESTIMATING PARAMETERS IN (6.4)

t	*M*	*ln(M)*
0	8.000	2.079
1	3.318	1.199
2	1.376	0.319
3	0.571	−0.561
4	0.237	−1.441
5	0.098	−2.321
6	0.041	−3.201
7	0.017	−4.081
8	0.007	−4.961
9	0.003	−5.841
10	0.001	−6.721

parameters $V_{\max}$ and K_M. Clearly, the process is nonlinear, and linear regression seems to be out of the question. However, an old trick for dealing with Michaelis–Menten functions is based on the observation that the function becomes linear if we plot $1/v$ against $1/S$. This inverse plot of the data is shown in Figure 6.5(B); representations of Michaelis–Menten enzyme kinetics in this format are referred to as Lineweaver-Burk plots. Since the data in this representation follow a linear function, we can use linear regression to determine the best-fitting straight line through the data. It turns out from the action of taking the reciprocal of the Michaelis–Menten function that the slope of the regression line corresponds to $K_M/V_{\max}$, whose value here is 2.91 and that the intercept is $1/V_{\max}$ with a value of 0.45. The two values allow us to compute $V_{\max}$ and K_M as 2.2 and 6.4, respectively.

Suppose the degradation process was estimated in a separate step. Artificial data are given in **Table 6.3**. These data exhibit exponential decay, which perfectly matches the formulation of the degradation term in (6.4). Thus, a logarithmic transformation of M into $\ln(M)$ yields a straight-line decay function (see Figure 6.6B) whose parameter values can be estimated by linear regression. The result

of this procedure is the value of c that best fits the degradation data, namely, $c = 0.88$.

Taken together with the estimation of $V_{\max}$ and K_M, we have now estimated all parameters of the system by bottom-up analysis and can substitute them into (6.4) to perform numerical simulations or other analyses. Validation of the estimates and the model requires additional data, for instance in the form of responses of M to changes in S.

Now suppose that no kinetic data are available but that we have instead time-course data for the system in the form of M as a function of time (**Figure 6.7**). In this case, we cannot use the bottom-up approach to estimate systems features from local information (kinetic data, linear degradation) but instead need to use a top-down estimation that works more or less in the opposite direction by using observations on the entire system (namely, time courses $M(t)$) to estimate all unknown kinetic features ($V_{\max}$, K_M, and c) simultaneously. This top-down estimation requires a (nonlinear) regression procedure that uses the time courses, together with information on S, and all at once determines values for $V_{\max}$, K_M, and c that capture the data most accurately. In this simple case, a decent nonlinear regression algorithm quickly returns the optimal values. More methods for this purpose will be discussed later in this chapter.

One of the great features of linear regression is that it also works for higher-dimensional problems, as we discussed before, and multiple linear regression also applies directly to linearized systems. For instance, consider a process described by the two-variable power-law function

$$V = \alpha X^g Y^h. \tag{6.5}$$

To linearize the function, we take logarithms of both sides, with the result

$$\ln V = \ln\alpha + g \ln X + h \ln Y \tag{6.6}$$

which permits the application of multiple linear regression to the new variables ln X, ln Y, and ln V.

When using this strategy, one needs to keep in mind that transformations of whichever type also affect the noise in the data. As a consequence, residual errors that are normally distributed around the true nonlinear function are no longer normal in the linearized form. Furthermore, transformations redistribute sampling across the dimensions, leading to clustering of many points near the origin of the plot of $1/S$ vs $1/V$ and few measurements at higher values that have exaggerated importance in the linear fit. In many practical applications, though, these problems are considered secondary in comparison with the simplification afforded by linearization. If an accurate account of the error structure is important, one may first estimate parameters by linear regression and then use them as start values for a nonlinear regression that does not distort the errors.

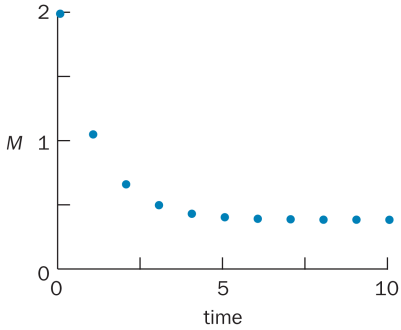


Figure 6.7 Time-course data for the model (6.4). The data cannot be linearized in any obvious fashion and require methods of nonlinear parameter estimation.

6.3 PARAMETER ESTIMATION FOR NONLINEAR SYSTEMS

Parameter estimation for nonlinear systems is incomparably more complicated than linear regression. The main reason is that there is just one linear structure but there are infinitely many different nonlinear functions, and it is not known per se which nonlinear function(s) might model some data sufficiently well, let alone optimally, especially if the data are noisy. Even if a particular function has been selected, there are no simple methods for computing the optimal parameter values as they exist for linear systems. Moreover, the solution of a nonlinear estimation task may not be unique. It is possible that two different parameterizations yield exactly the same residual error or that many solutions are found, but none of them is truly good, let alone optimal.

Because the estimation of optimal parameter values for nonlinear models is challenging, many algorithms for optimization—and thus for parameter estima-

tion—have been developed over the past decades. All of them work well in some situations but fail in other situations, especially for larger systems where many parameters have to be estimated. In contrast to linear regression, it is typically impossible to compute an explicit, optimal solution for a nonlinear estimation task. So, one may ask: if there is no explicit mathematical solution, how is it possible that a computer algorithm can find a solution? The answer is that optimization algorithms iteratively search for ever better solutions, and, if they succeed, the solution is very close to the best possible solution, although it is usually not precisely the truly best solution. The trick of developing good search algorithms is therefore to guide the search in an efficient manner toward the optimal parameter set and to abandon or drastically alter search strategies early when they explore parameter ranges that do not show much promise.

A crucial question regarding the success of nonlinear estimation algorithms is whether the task calls for finding the parameters of a mathematical function or of a system of differential equations. In the former case, many methods work quite well, and we can even use software like the *Solver* option with the *Data Analysis* group of *Microsoft Excel®*, while the latter case is much more complicated, although the basic concepts for both tasks are the same. We will discuss different types of methods, exemplify some of them with the estimation of parameters in explicit functions, and toward the end discuss parameter estimation for dynamical systems.

The currently available search algorithms may be divided into several classes. The first consists of attempts to exhaust all possible parameter combinations and to select the best among them. A second class consists of gradient, steepest-descent, or hill-climbing methods. The term *hill-climbing* implies finding the maximum of a function, while parameter estimation searches for the minimum use steepest-descent methods. Mathematically, these two tasks are equivalent, because finding the maximum of a function F is the same as finding the minimum of $-F$. The basic idea is simple. Imagine finding yourself in a hilly terrain and that it is so foggy that you can only see a few feet in each direction. You are thirsty, and you imagine that water is most likely to be found down in the valley. So, you look around (although you can't see very far) and check which direction is heading down. You walk in this direction until you come to a point where a different direction leads down even more steeply. With time, you have a good chance of getting to the bottom of the valley. Gradient methods imitate this procedure. As a warning, it is easy to realize that even in the hiking example this strategy is by no means failsafe. In fact, gradient search algorithms tend to have problems with rough terrains, in particular if they find themselves in a valley that is not as deep as some of the neighboring valleys. As a result, the algorithms often get trapped in a **local minimum**, which may be better than other points close by but is not as good as the desired solution of the true (global) minimum, which is located in a different valley (**Figure 6.8**).

A third class of parameter estimation methods consists of **evolutionary algorithms**, which operate in a fashion that is entirely different from the previous methods. The concepts here are bio-inspired and gleaned from the natural processes of fitness-based selection, which we believe has time and again promoted the best-adapted individuals and species from among their competitors. The best-known evolutionary method, the **genetic algorithm** (**GA**), begins with a population of maybe 50–100 parameter vectors, which consist of default values or of values supplied by the user. Each parameter vector, entered into the model, represents a numerical implementation of the model. Some of these may be good and others not so. The GA solves the model with each parameter vector and computes the residual error associated with each case. The vectors are then ranked by their **fitness** (how well the model with the parameter values of the vector matches the data), and the fitter a vector, the higher is its probability to "mate." In this mating process, one part of one vector (the mother) is merged with the complementary part of another vector (the father) to produce a new vector (the newborn baby vector). The vectors are furthermore subject to small mutations that permit the evolution of the population. The mating processes within the parent population lead to a new generation of vectors, the model is again solved for each vector, residual errors are computed, the best vectors are selected, and the process continues to the next generation until a suitable solution is found or the algorithm runs out of steam. Gradient methods and genetic algorithms presently dominate the field of

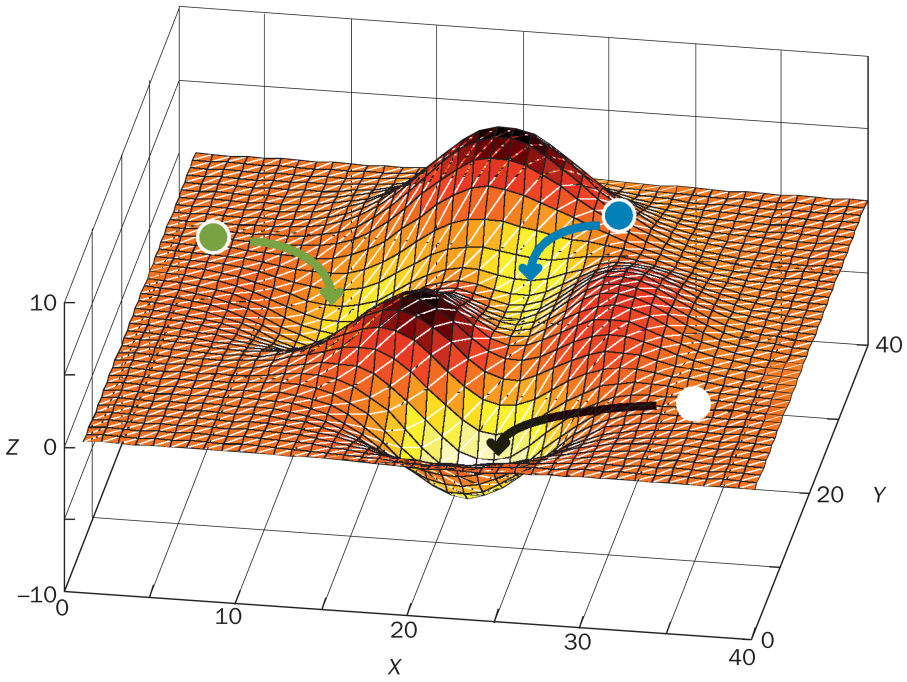


Figure 6.8 Gradient searches may fail in rough terrains. The function shown here (a Laplace function from MATLAB® has three minima, of which the one in the center front ($X = 22$, $Y = 11$, $Z = -6.2$) is the lowest. A gradient search initiated at the white location will likely find this global minimum. However, gradient searches starting at the green or blue locations will easily get trapped in local minima with $Z = 0$ and $Z = -3$, respectively.

search methods for parameter estimation, but there are other alternatives, which we will briefly mention later in this section.

6.3.1 Comprehensive Grid Search

The idea of exhaustive searches is really simple. One determines the possible numerical range for each parameter (for instance, from biological knowledge of the process), evaluates the model with very many combinations of parameter values within these ranges, and selects the parameter set that produces the best result. Because the parameter combinations are usually selected in regular intervals, this type of estimation is called a grid search.

As an illustration, let us return to the example of the Michaelis–Menten rate law (MMRL) $v = \frac{V_{max}}{K_M + S}$ and pretend that our earlier linearization was not possible. Suppose we have measurements of the rate v for several substrate concentrations S. We know from our understanding of the MMRL that the two parameters $V_{\max}$ and K_M must be positive. Let us suppose that we also know that ($V_{\max} \leq 10$ and $K_M \leq 8$. Thus, we have admissible ranges, which we subdivide into even intervals of 1. For the 80 parameter combinations, it is now straightforward to compute the value of v for each substrate concentration S, subtract from this value the corresponding experimental measurement, square the differences, and sum them together. Thus, for $S = 0.5$, the experimental $v = 0.16$ from Table 6.2 and the pair ($V_{\max}$, K_M) = (1, 1), the first quantity of the sum is computed as

$$\left(\frac{1 \cdot 0.5}{1 + 0.5} - 0.16\right)^2. \tag{6.7}$$

Figure 6.9 (A) shows a plot of the sum of squared errors for the given ranges of $V_{\max}$ and K_M. The plot indicates two important results. First, many combinations of $V_{\max}$ and K_M are clearly not contenders for the optimum, because the associated errors are very high. Thus, a considerable amount of computation time has been wasted. Second, it is difficult to discern where exactly the parameter combination with the minimal error is located. It looks to be somewhere close to $V_{\max}$ between 2 and 3 and K_M around 6. We can use this information to refine the ranges for $V_{\max}$ and K_M and repeat the grid search with smaller intervals. **Figure 6.9(B)** indeed

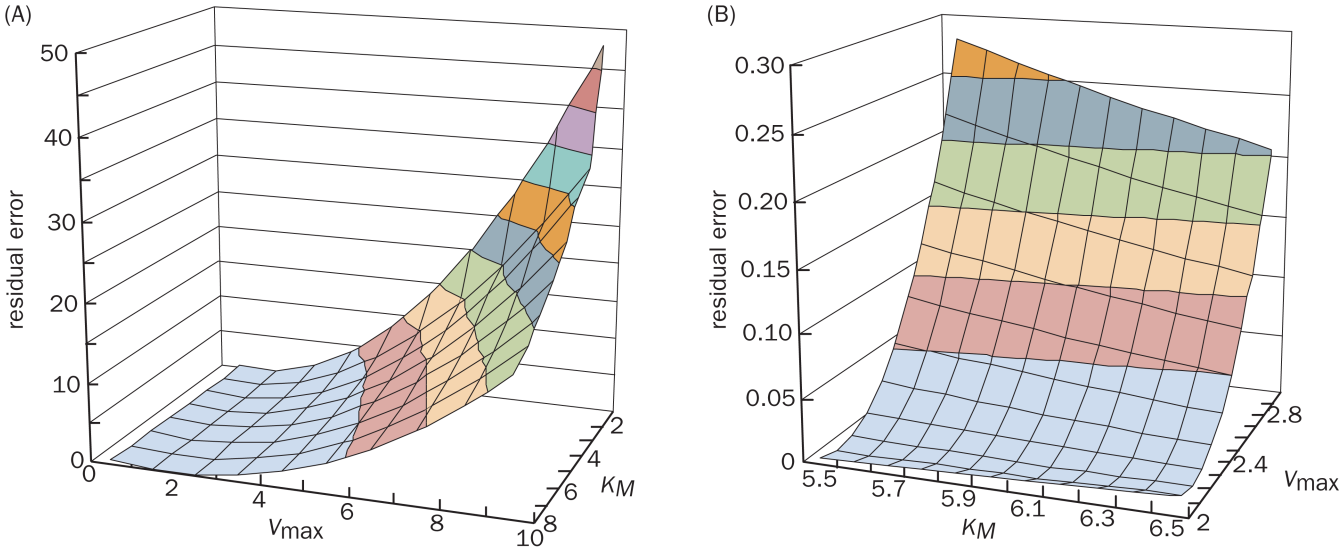


Figure 6.9 Residual errors (SSEs) in a grid search. The search targeted the MMRL parameter values K_M and V_{max} in (6.4), using the data in Table 6.2. (A) Coarse search. (B) Refined search within smaller parameter ranges.

gives a more focused view and identifies the accurate solution $V_{max} = 2.2$ and $K_M = 6.4$. This solution is accurate, because the true values happened to be exactly on the grid we used. If the true values had been $V_{max} = 2.01234567$ and $K_M = 6.54321098$, we would at best have found an approximate, relatively close solution.

This simple example already exhibits the main problems with grid searches. First, one has to know admissible ranges for all parameters, lest one miss the optimal solution. Second, the method is not likely to produce very precise results, unless one iterates the search many times with smaller and smaller intervals. Third, the method wastes a lot of computation time for parameter combinations that are clearly irrelevant (in our case, low V_{max} and high K_M). Fourth and maybe most important, the number of combinations grows very rapidly when many unknown parameters are involved. For instance, suppose the model has eight parameters and we begin by trying ten values for each parameter. Then the number of combinations is already 10^8, which is 100 million. For more parameters and more values to be tried, these numbers become astronomical. The statistical field of experimental design has developed different means of taming this combinatorial explosion [6, 7]. Arguably most popular among them is **Latin hypercube sampling**. The adjective Latin comes from the Latin square, which, for size 5 for instance, contains the numbers 1, 2, 3, 4, and 5 exactly once in each row and each column of a 5×5 grid. The term hypercube refers to the analog of a regular cube in more than three dimensions. The key idea of Latin square sampling is that one analyzes only enough points to cover each row and each column of a grid exactly once. In a higher-dimensional space, one searches a hypercube in an analogous fashion. Latin hypercube sampling can be applied in log scales for each dimension, allowing for more effective grid searching across wide ranges of parameter values.

While grid searches are seldom effective, the attraction of obtaining a **global optimum**, at least in an approximate sense, has triggered research into improving grid searches without getting overwhelmed by combinatorial explosion. As an example, Donahue and collaborators [8] developed methods that search a multidimensional grid more intensively in the most promising domains of the parameter space and only coarsely in domains that are unlikely to contain the optimal solution.

Branch-and-bound methods are significant improvements over grid searches, because they ideally discard large numbers of inferior solutions in each step [9, 10]. Branch-and-bound methods do this with two tools. The first is a splitting or branching procedure that divides the set of candidate solutions into two nonoverlapping "partitions" A and B, so that all solutions are accounted for and

each one is either in A or B but not both. Each solution is characterized by a score for the quantity that is being optimized, which in our case is SSE. The second tool is the estimation of upper and lower bounds for SSE. A lower bound, say for partition A, is a number that is as small as—or smaller than—the SSE of any candidate solution (fitness of a parameter vector) in A, and the analogous definition holds for upper bounds. The key concept of the branch-and-bound method is that if the lower bound of partition A is greater than the upper bound of partition B, then the entire partition A may be discarded, because no solution in A can possibly have a lower SSE (and thus a higher fitness) than any of the solutions in B. The trick is therefore to split the candidate solution sets successively and effectively into partitions and to compute tight lower and upper bounds. In the end, branch-and-bound methods find globally optimal solutions, but their implementation is difficult and they require considerable computational effort.

6.3.2 Gradient Methods

Most nonlinear functions cannot be transformed into linear functions and they are too complicated for exhaustive explorations of their parameter space. Moreover, a direct computation of optimal parameter values, which is the hallmark of linear regression, is no longer possible, and entirely different approaches must be used. The most prominent among these are iterative methods that search in a guided or random fashion for good parameter vectors (see, for example, [11]). Because the parameter space that encompasses all parameter vectors is usually of high dimension and is difficult to fathom, the algorithms initiate the search, assess the quality of fit with the current parameter values, search again, assess the quality of fit again, and thereby operate in an iterative mode that cycles between searching and assessing thousands of times. The algorithm starts with a default or user-provided parameter vector (sometimes collectively called a **guestimate**), which may be quite good or rather inadequate, solves the model with this parameter vector, and computes the residual error. In the next step, the algorithm evaluates the model for a small number of other parameter vectors in a close neighborhood of the start vector; remember the search for water in a hilly, foggy terrain (see Figure 6.8)? The algorithm then uses this information to determine in which direction the error decreases the most. Pointing in this direction, the algorithm selects a new trial parameter vector at some distance from the original vector. This procedure is repeated thousands of times until, ideally, the algorithm converges to a parameter vector that yields a lower SSE than all vectors in its neighborhood as well as all earlier vectors. Understanding these concepts is more important than technical details, because it reveals why search algorithms sometimes do not succeed and because the technical aspects have been implemented in many software packages.

The iterative search strategy is most easily demonstrated in the case of a single parameter. Suppose the error function R has the shape shown in **Figure 6.10**. The plot shows on the horizontal axis the value of the parameter p and on the vertical axis the residual error R, which is a squared quantity and therefore not negative. Our task is to find the position along the horizontal axis (i.e., the optimal parameter p) where R is closest to zero, because this position minimizes the error.

Suppose we are searching for the minimum inside the interval [0, +8] and our initial guestimate is $p = 7$. The algorithm enters $p = 7$ into the model and evaluates

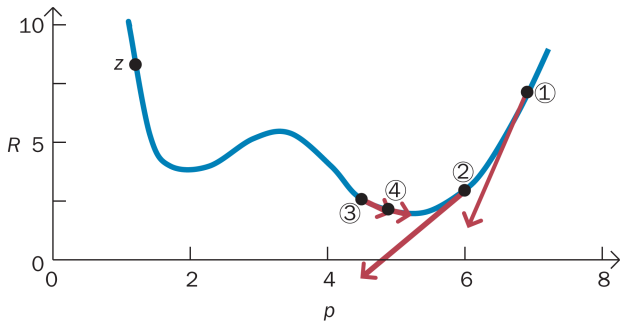


Figure 6.10 Fictitious error landscape for the search for the optimal value(s) of the parameter *p*. The residual error associated with each value of p is denoted by R. Numbers in circles indicate the starting points for different iterations, and arrows show each search direction, with the length of each arrow representing speed. If the search is started at position z, the algorithm may get stuck in a local minimum close to $p = 1.9$.

the residual error R based on experimental data; in the figure, $R = 7.2$ (the circled 1). The algorithm now evaluates the model with two values of p that are a little smaller or a little greater than 7, respectively. From this information, the algorithm concludes that lowering the guestimate has a better chance of reducing R than increasing it. It also estimates the slope of the error function and uses it to determine by how much the original estimate should be lowered, how this determination is accomplished is not so important here. In our example, the algorithm determines the next candidate point as $p = 6$ and computes R at this point as about 2.6 (the circled 2). The slope at this point still suggests lower values for p, and because the same direction is taken again, the speed is increased and the next candidate is about $p = 4.5$ (the circled 3). Now the slope points into the opposite direction, which suggests raising the value of p. The algorithm does this and reaches the point indicated by the circled 4. The process continues in this fashion until the algorithm detects that the slope of the error function is essentially zero, which means that the error can no longer be lowered by moving p a little bit to the left or right. The algorithm terminates and the result is the approximate minimum. It should be kept in mind that the algorithm does not know where the true minimum is. Thus, it has to estimate how far to go at every step and may overshoot or undershoot the target for quite some while.

Figure 6.10 also indicates a potential problem with this type of algorithm. Imagine the initial guestimate is $p = z$. Using the same rationale as earlier, it is easy to see that the algorithm might converge to a value close to $p = 1.9$ with a residual error of about $R = 3.75$. Clearly, this is not the optimal value, but it satisfies the criterion that the slope is zero and that other parameter choices close to $p = 1.9$ are even worse.

Exactly the same principles apply to estimation tasks involving two or more parameters. In higher dimensions, we humans have a hard time visualizing the situation, but iterative search algorithms function in exactly the same fashion as discussed earlier. For two parameters, we can still conceptualize the error function geographically: it is a surface that shows the residual error for each (x, y), where x and y are the two parameters with unknown values (see Figure 6.9 for a simple example). As in Figure 6.8, this error landscape potentially has many hills and valleys, as well as many mountain ridges and craters, and it is easy to imagine how a search algorithm might end up in a local minimum, which, however, is not the truly lowest point. This situation of being trapped in a local minimum is unfortunately quite common. In contrast to our two figures (Figures 6.8 and 6.10), where the mistake is easy to spot, a local minimum is difficult to diagnose in more complicated cases, where for instance six or eight parameters are to be optimized. One obvious, although not failsafe, solution is to start the algorithm many times with different guestimates.

6.3.3 Genetic Algorithms

Using a gradient method for parameter estimation, one is often unsure whether the solution is truly the optimum or whether the algorithm might have converged to a local minimum (see Figures 6.8 and 6.10). Since this issue is directly tied to the core principles of iterative search algorithms, it is unlikely that small changes in implementation would solve the problem. Instead, an entirely different class of global search heuristics has been developed over the years. “Global” refers to their ability to avoid getting stuck in local minima, as nonlinear regression algorithms tend to do. “Heuristics” means that it is often impossible to prove with mathematical rigor that or why these algorithms will indeed succeed.

The best-known global search heuristics are genetic algorithms (see, for example, [12, 13]). They fall into the larger category of evolutionary algorithms, where the attribute of evolution is due to their inspiration by natural selection of the fittest members in a population [14]. Genetic algorithms often find good approximate solutions, but they tend to have difficulties determining very precise solutions. Because this situation is opposite to nonlinear regression, it is sometimes useful to start a parameter search with a genetic algorithm and to refine the solution with a subsequent nonlinear regression (we will discuss an example of this type later).

Just like in natural evolution, genetic algorithms are based on individuals with their specific fitness, and this fitness is taken to define the opportunity to mate and

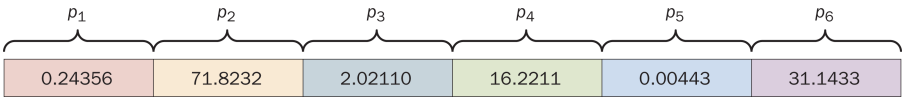


Figure 6.11 A typical individual in the application of a genetic algorithm to parameter estimation. The parameters are lined up either as floating-point variables, as they are shown here, in binary code (see Figure 6.12) or in some other mathematical representation.

produce offspring. Furthermore, the evolution process is subject to mutations. In the specific application of genetic algorithms to parameter estimation, each individual is a parameter vector. In a simplified description, such a vector might look like the diagram in **Figure 6.11**: it lines up the values of all parameters that need to be estimated. In most cases, the numbers are actually converted into binary code, but other encodings, such as floating-point representations, have also been explored.

The starting population contains many such individuals (maybe 20, 50, or 100), which are defined through a random assignment of parameter values that collectively represent the admissible parameter ranges or are "seeded" based on prior knowledge of the problem. Each individual corresponds to a solution of the model, because entering the parameter values into the symbolic model allows us to compute any desired features of this model. In general, the feature of interest could be anything that can be quantitatively characterized with the model, such as the steady state, how fast the model reaches a point of interest, how much the transients of the model deviate from the normal state, or how fast the system oscillates. In the case of parameter estimation, the feature of prime interest is the fit to a set of experimental data and thus SSE. For each individual in a given population, one simulates the model with its parameters and computes the residual error between the data and the corresponding values of the model. This error is used as the measure of fitness: the smaller the error, the fitter the individual. Now comes the time of reckoning. Out of the 20, 50, or 100 individuals in the population, a small number of individuals (maybe 20%) are permitted to mate. They are randomly selected, but the probability of being chosen is proportional to each individual's fitness and thus the SSE. Individuals that were not chosen are discarded. In this scheme, fit individuals have a higher chance of mating, but even unfit individuals may get lucky. Interestingly, the inclusion of a small number of less-fit individuals retains higher genetic diversity and to some degree prevents early termination at a local minimum, which corresponds to a suboptimal parameter vector.

The mating process occurs as follows. Two individuals (parameter vectors) are lined up next to each other, and offspring is produced by using the first few parameters (or parts of the parameter vector) from the first individual and the rest from the second individual, as sketched in Figure 6.12. Any positions within a parent vector may also be mutated; for instance, from 1 to 0 or from 0 to 1 in the case of binary coding. Experience has shown that mutations are very important, but that their rate must be rather low, lest the algorithm wander through the parameter space without ever converging. The result of the mating and mutation process is again an individual in the form of a new parameter vector. The parameter values from this vector are entered in the model and the fitness is computed. The mating process is usually organized such that the population size remains constant. Exceptionally fit individuals may mate several times, and most algorithms automatically move the fittest two or three individuals of the parent generation unchanged into the offspring generation. The process terminates for one of several reasons, the most typical of which are:

- A solution has been obtained that satisfies preset quality criteria. Success!
- Solutions with the highest fitness have reached a plateau where they do not change for many iterations from one generation to the next. This may be good or bad news.

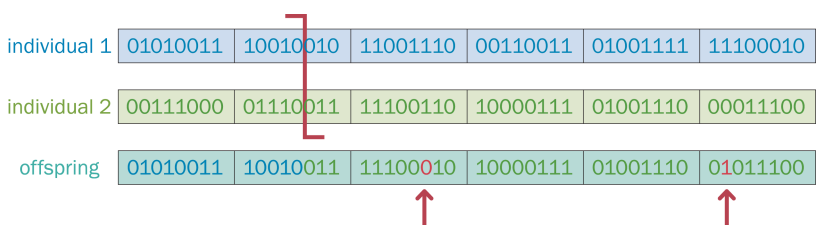


Figure 6.12 Generation of offspring in a generic genetic algorithm. Individuals (parameter vectors) are coded here as binary strings. The new offspring contains a part of individual 1 and the complementary part of individual 2. The newly combined offspring is furthermore subject to mutations before the next iteration (arrows).

- The number of generations has reached a preset limit. It could be started again with the last population of vectors.
- The algorithm encounters some numerical problem, for instance, while integrating differential equations.

In many cases, genetic algorithms work quite well, and, because they are extremely flexible, for instance in terms of fitness criteria, they have become very popular and are implemented in many software packages. Nonetheless, they are certainly no panacea. In some cases, they never converge to a stable population; in other cases, they may not find an acceptable solution; and in almost all cases, these algorithms are not particularly fast. If problems arise, the first attempt at resolution is to reset the algorithmic parameters, such as the population size, the mating probabilities of fit or less-fit individuals, the mutation rate, the number of fittest individuals that directly survive into the next generation, bounds on parameters, and criteria for termination. A specific example will be presented later in this chapter.

6.3.4 Other Stochastic Algorithms

In addition to genetic algorithms, several other classes of methods are available for global optimization and parameter estimation. They include evolutionary programming, for instance implemented as ant colony and particle swarm optimization, as well as simulated annealing and the branch-and-bound methods that we discussed before. In this section, we will discuss the concepts of these methods without giving a lot of detail.

As in genetic algorithms, evolutionary programming is a **machine learning technique** based on evolving populations of solutions. The main mechanisms for improving fitness are mutation and self-adaptation to adjust parameters, and variations may include operations such as combining information from more than two parents. One intriguing method within the class of evolutionary programming is ant colony optimization (ACO; see, for example, [15]). ACO was inspired by the behavior of ants seeking a food source while wandering about their colony. The original exploration is more or less random, but once an ant finds food, it returns to the nest in a more or less direct fashion. Along the way, the ant leaves a pheromone trail that can be perceived by other ants. This type of chemical communication allows other ants preferably to follow the established pheromone trails rather than to follow random paths. If a path indeed leads to food, more and more ants follow it and release pheromone, thereby reinforcing the successful paths. However, if the path does not lead to success or if it is unduly long, it is less traveled, the pheromone evaporates, and its attraction dissipates. Over time, the short, efficient paths incur more traffic and thus enjoy higher concentrations of pheromone. ACO attempts to mimic this dynamic, adaptive mechanism of optimizing the path to a desired destination. Thus, parameters that show up time and again in good solutions are rewarded by increasing their probability of being selected in future generations, while other infrequent parameters become less and less prevalent in the population. Algorithmic issues related to ACO are discussed in [16].

Similar ideas also form the basis for particle swarm optimization (PSO), which was inspired by the flight patterns of birds (see, for example, [17]). Like genetic algorithms and ACO, PSO is a population-based stochastic search procedure, where each particle in the swarm represents a candidate solution to the optimization or estimation problem. The flight of each particle is influenced by the best positions of the particle and the swarm within a high-dimensional space, and the fitness is measured according to each particle's own experience, which is augmented by communication with its neighboring particles. Through this communication within the swarm, all particles share some of the information regarding high-fitness solutions. Specifically, as soon as a particle detects a promising solution, the swarm explores the vicinity of this solution further. As a result, the particles generally tend toward optimal positions, while searching a wide area around them. Expressed differently, PSO is quite efficient because it combines local and global search methods. PSO methods are reviewed in [18].

Simulated annealing (**SA**) is a well-established global optimization technique that grew out of a Monte Carlo randomization method from the 1950s for generating sample states of a thermodynamic system [19, 20]. The inspiration for this method came from the technique of annealing in metallurgy. This technique uses heating and controlled cooling of a material to reduce defects. The heat provides energy that allows atoms to leave their current positions—which are states of locally minimal energy—and to reach states of higher energy. During cooling, the atoms have the chance to find lower-energy states than they had before. Therefore, the system eventually becomes more ordered and approaches a frozen state of minimal energy in a semi-guided random process. The analogy of this heating–cooling process in SA is that a population of solutions traverses the search space randomly, thereby allowing the current solutions to change a bit. The random mutations of each individual solution are tested (i.e., the model is computed with the given parameter values), and mutants that increase fitness always replace the original solutions. Mutations with lower fitness are not immediately discarded but are accepted probabilistically, based on the difference in fitness and a decreasing temperature parameter. If this temperature is high, the current solutions may change almost randomly. This feature prevents the method from getting stuck in local minima. During cooling, the temperature gradually decreases and toward the end the solutions can only vary in a close neighborhood and finally assume local minima. Either the former solutions are regained in the process or new solutions are found that have a higher fitness. Because many solutions settle in their individual minima, it is hoped that the global minimum is among them. The key to a successful implementation is the choice of adequate algorithmic parameters. For instance, if the temperature of the heated system is too low at the beginning or if cooling occurs too rapidly, the system may not have enough time to explore sufficiently many solutions and may become stuck in a state of locally but not globally minimal energy. In other words, SA only finds a local minimum.

It is possible to combine several of these methods. Indeed, it is often useful to start a search with a genetic algorithm in order to obtain one or more coarse solutions, which are subsequently refined with a nonlinear regression. It is also possible to use SA within a standard genetic algorithm by starting with a relatively high mutation rate, which slowly decreases over time. Similarly, Li et al. [21] combined SA with ACO. The advantage of this combination was again that ACO is a global search method, while SA has features characteristic of probabilistic hill climbing.

Box 6.1 contains an example of the estimation of parameter values of a moderately complicated function in Excel.

BOX 6.1: ESTIMATION OF THE PARAMETER VALUES OF A ONE-VARIABLE FUNCTION IN EXCEL

As an example of a parameter estimation in Excel, consider the function

$$F(X) = p_1 + \frac{p_2}{1+[\exp(-(X+p_3)]^{p_4}},$$

which has four parameters: $p_1, \ldots, p_4$. Suppose that data with some noise are available (see **Figure 1**) and the task is to fit the model to these data. The typical Excel workbook contains the data (Columns A and B), the modela with parameter values that are at first unknown (Column C), the squared differences between model and data for each value of *X* (Column D), the parameter vector (Cells F5–I5), and the SSE (Cell D32). To implement these settings, we specify

$$\text{Cell C5 as} = \$F\$5 + \$G\$5 / (1 + \text{EXP}(-(A5 + \$H\$5))^{\wedge} \$I\$5)$$

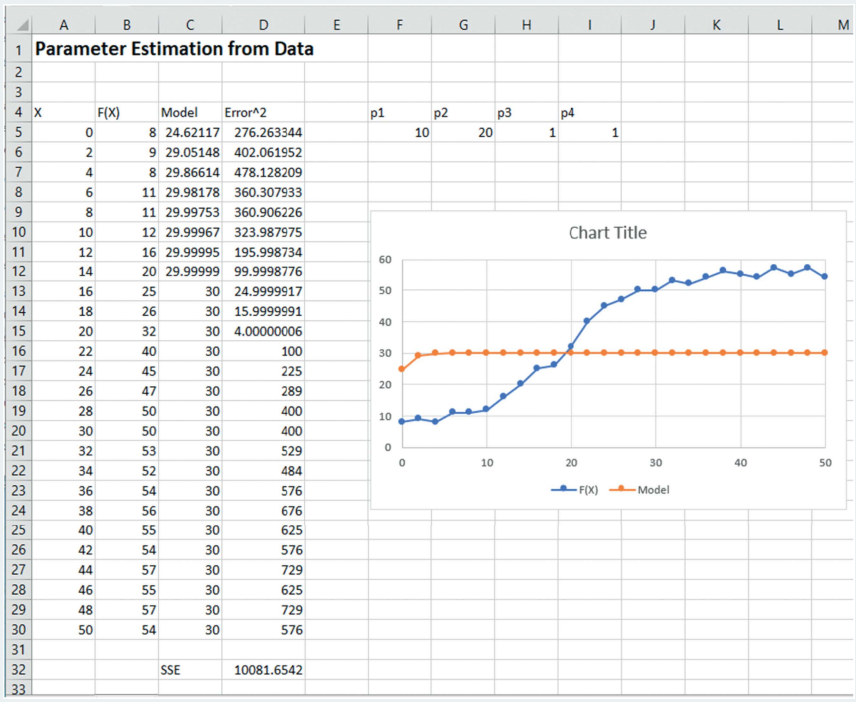
$$\text{Cell D5 as} = (B5 - C5)^{\wedge}2$$

$$\text{Cell D32 as} = \text{SUM}(D5{:}D30)$$

It is furthermore useful to include a chart displaying the data and the model fit. As a side note, Excel can also estimate functions of several variables; the procedure is very similar.

The optimization requires the Add-in *Solver*, which may have to be uploaded to the Data Analysis ToolPak. In the Solver window (**Figure 2**), the specifics for the estimation are to be entered, including SSE, the parameters to be estimated (here called *variables*), and a decision regarding maximization or minimization. It is also possible to add constraints, such as permissible bounds on parameter values. Because we do not know what values the parameters may have under optimal conditions, we make sure that the box "Make Unconstrained Variables Non-Negative" is unchecked.

Starting arbitrarily with initial guesses $(p_1, \ldots, p_4) = (10, 20, 1, 1)$, the non-optimized solution is rather bad (Figure 1). Now clicking *Solve* quickly presents a very good solution with an SSE of 36.8 and a pleasing visual fit; the optimized parameter values are $(p_1, \ldots, p_4) \sim (7.47, 47.97, -19.15, 0.22)$ (**Figure 3**). Starting with many other initial guesses yields the same solution. It is possible, in other cases, that several parameter values yield essentially the same fit.



	A	B	C	D	E	F	G	H	I	J	K	L	M
1	**Parameter Estimation from Data**												
2													
3													
4	X	F(X)	Model	Error^2		p1	p2	p3	p4				
5	0	8	24.62117	276.263344		10	20	1	1				
6	2	9	29.05148	402.061952									
7	4	8	29.86614	478.128209									
8	6	11	29.98178	360.307933									
9	8	11	29.99753	360.906226									
10	10	12	29.99967	323.987975									
11	12	16	29.99995	195.998734									
12	14	20	29.99999	99.9998776									
13	16	25	30	24.9999917									
14	18	26	30	15.9999991									
15	20	32	30	4.00000006									
16	22	40	30	100									
17	24	45	30	225									
18	26	47	30	289									
19	28	50	30	400									
20	30	50	30	400									
21	32	53	30	529									
22	34	52	30	484									
23	36	54	30	576									
24	38	56	30	676									
25	40	55	30	625									
26	42	54	30	576									
27	44	57	30	729									
28	46	55	30	625									
29	48	57	30	729									
30	50	54	30	576									
31													
32			SSE	10081.6542									
33													

Figure 1 Typical Excel Worksheet for parameter estimation. Definitions of key cells are provided in the Text. The displayed solution pertains to the arbitrary initial parameter guestimates in Cells F5–I5.

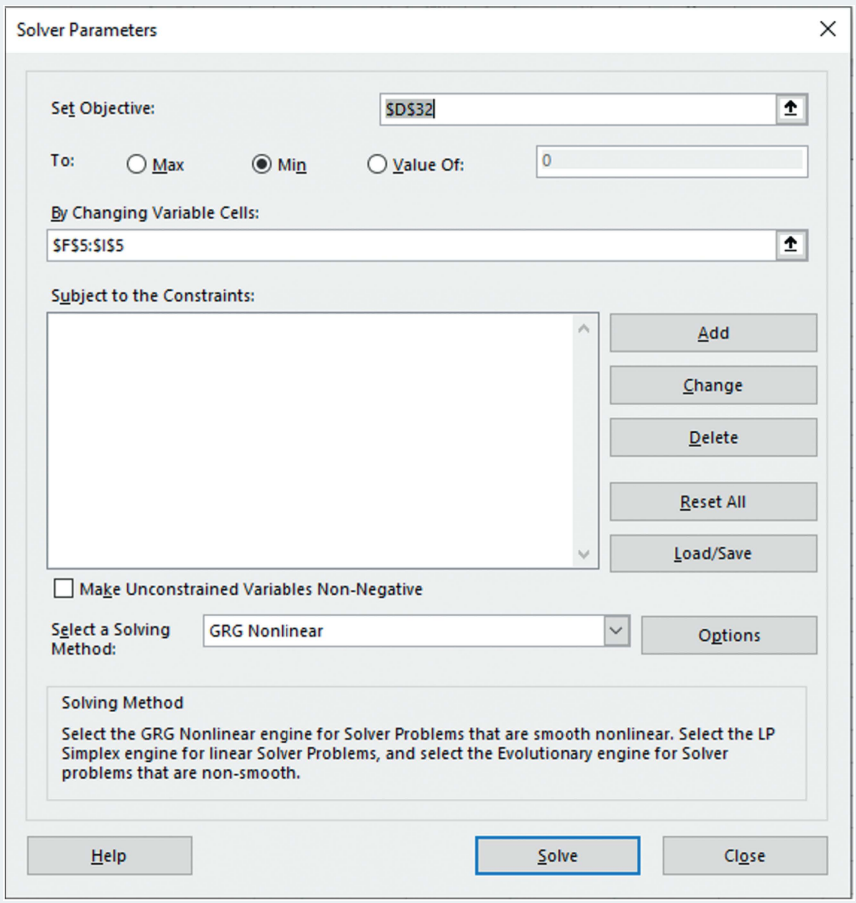


Figure 2 Solver window in Excel with pertinent settings. The Solver requires specification of the SSE, the parameters to be estimated (here called variables), and a decision regarding maximization or minimization.

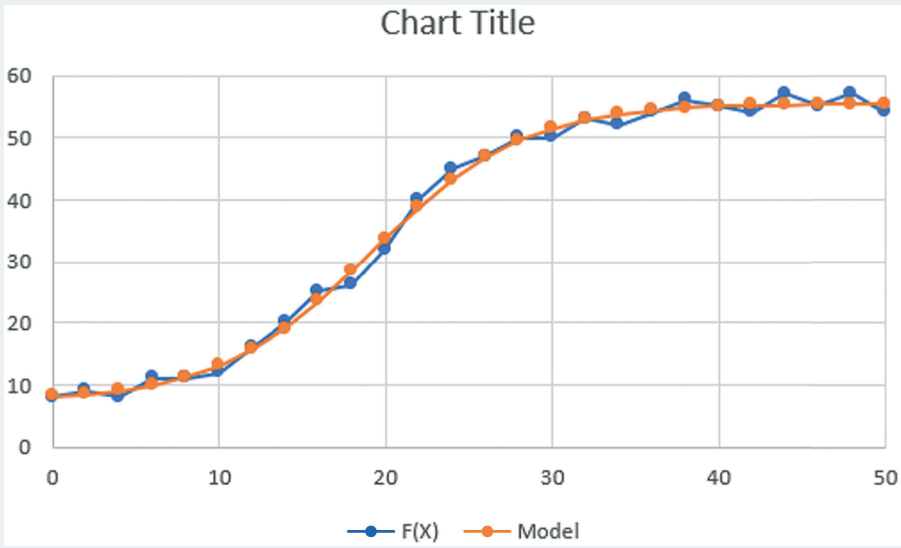


Figure 3 Data fit obtained with Excel Solver. The default chart may be customized with the typical tools in Excel.

6.3.5 Typical Challenges

No matter which search algorithm is used to determine optimal parameter values for a nonlinear model, some challenges come up time and again [22]. One ubiquitous challenge in computational systems biology is **noise** in the data. Although data don't usually come in decibels, this expression is used to describe deviations from an expected (usually rather smooth) trend line, which may be due to variability among beakers, Petri dishes, cells, or organisms and/or to inaccuracies in measurements. It does not require much imagination to infer from plots such as Figure 6.13 that noise may become a challenge for parameter estimation, especially if it is so large that it obscures or hides the true trend line.

A second challenge, which occurs more often than one might think and is actually quite hideous, is the fact that the estimation task may have many solutions with the same SSE. These solutions may all be connected in the parameter space, or they may be distinctly separated. An intuitive example arises when two parameters always appear in the model in the same formation or combination, such as $p_1 p_2$. Clearly, for every p_1 that some algorithm might determine incorrectly, there

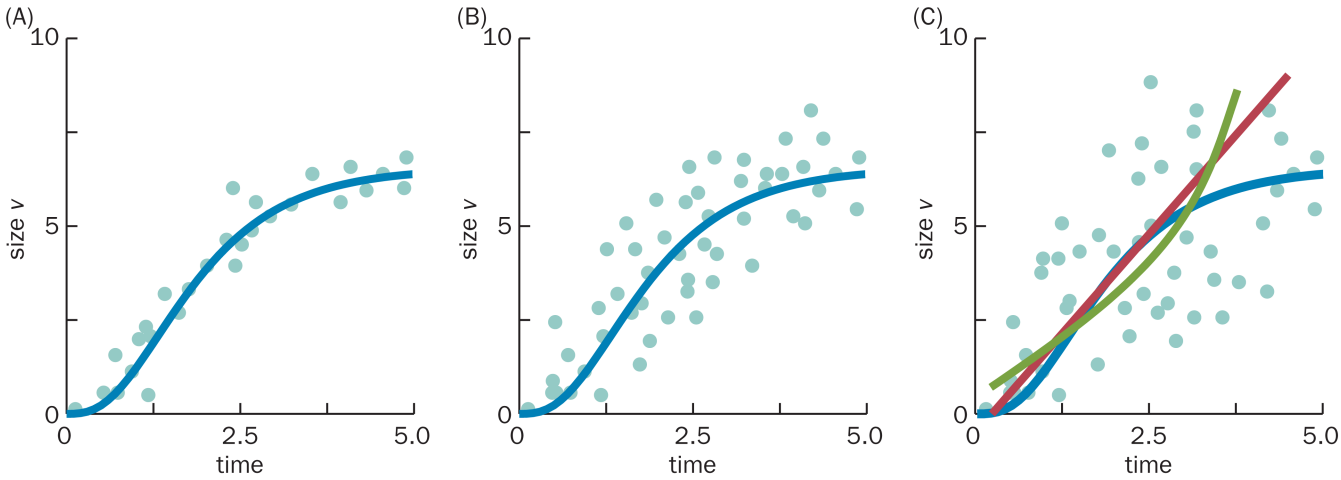


Figure 6.13 Too much noise may hide the true functional relationship and create problems with parameter estimation. (A) The true trend (blue) is clearly perceivable in spite of moderate noise. In the presence of stronger noise (B), the true trend is more ambiguous; in fact, a straight line could possibly fit the data. Even more noise (C) makes it impossible to identify the true trend, and linear as well as drastically different nonlinear trends could be most appropriate.

TABLE 6.4: DATA TO BE MODELED WITH THE POWER-LAW FUNCTION (6.8)

X	Y	f
1	1.75	2.07
2	3.05	4.03
3	4.21	5.95
4	5.31	7.84
5	6.34	9.71
6	7.34	11.57
7	8.30	13.41
8	9.24	15.25
9	10.15	17.07
10	11.04	18.89
11	11.92	20.70
12	12.78	22.50
13	13.62	24.30
14	14.45	26.09
15	15.27	27.88
16	16.08	29.66
17	16.88	31.44
18	17.67	33.21
19	18.45	34.98
20	19.22	36.75

is a p_2 that perfectly makes up for the error (with the exception of $p_1 = 0$). It is useful to express the situation in a slightly different fashion. Suppose that, in the true solution, $p_1 = 3$ and $p_2 = 4$, so that $p_1 \cdot p_2 = 12$. Then, any solution that satisfies $p_2 = 12/p_1$ has exactly the same SSE as the true solution.In the case of $p_1 \cdot p_2$, the culprit is easy to identify, but in other cases it might not be so obvious. For instance, consider the power-law function

$$f = pX^a Y^b \tag{6.8}$$

with three parameters. Suppose data are available for X, Y, and f, as shown in Table 6.4. For clarity of illustration, these data are assumed to be noise-free. Initiating a search algorithm with different guestimates for p, a, and b may lead to distinctly different solutions that are all optimal, which here means that the residual error is 0. The differences in solutions are not the consequence of noise, because the data in this constructed example are noise-free. So, what's the problem? In this case, the redundancy is actually due to the fact that the data for X and Y in Table 6.4 happen to be related by a power-law function, i.e.,

$$Y = \alpha X^{\gamma} \tag{6.9}$$

with parameters $\alpha = 1.75$ and $\gamma = 0.8$ (Figure 6.14). Because of this relationship, Y^b Y^b may be substituted equivalently with $(aX_{\gamma})^b$, no matter what b is. As a consequence, if the parameters of f in (6.8) are to be estimated and the algorithm finds some incorrect value for b, the parameters p and a are able to compensate perfectly. As a numerical example, suppose the true parameters of f in (6.8) are $p = 2.45$, $a = 1.2$, $b = -0.3$. By applying the transformation $Y = 1.75X^{0.8}$, according to (6.9), we obtain

$$\begin{aligned} f &= 2.45X^{1.2}Y^{-0.3} \\ &= 2.45X^{1.2}YY^{-1.3} \\ &= 2.45X^{1.2}(1.75X^{0.8})Y^{-1.3} \\ &= 4.2875X^{2}Y^{-1.3}. \end{aligned} \tag{6.10}$$

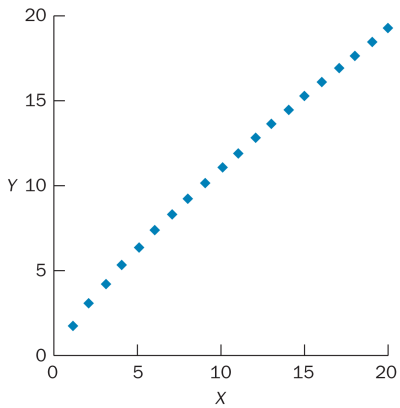


Figure 6.14 Dependences between variables may lead to redundant parameter estimates. The error-free (X, Y) data of Table 6.4 are related to each other by the power-law function (6.9) with $\alpha = 1.75$ and $\gamma = 0.8$.

Thus, we have two distinct solutions to the parameter estimation task, namely the parameter vectors (2.45, 1.2, –1.3) and (4.2875, 2, –1.3), which produce exactly the same values of f. Moreover, these are not the only solutions. Instead of replacing Y with $1.75X^{0.8}$, we could replace any power of Y with the corresponding representation of $1.75X^{0.8}$, and the resulting f would be characterized differently, but generate exactly the same fit to the data.

In the cases discussed so far, all equivalent solutions taken together formed straight or curved lines within the parameter space. A distinctly different situation is possible where two or more solutions are equivalent, and yet the solutions in between are not. The easiest demonstration of this situation is graphical: imagine that the error function R in **Figure 6.15** depends on only one parameter p. Clearly, there are two solutions where the residual error is minimal (with $R = 2$), namely for $p \approx \pm 3.16$, and all other values of the parameter p give solutions with higher errors. It is easy to imagine similar situations for error functions that depend on several parameters.

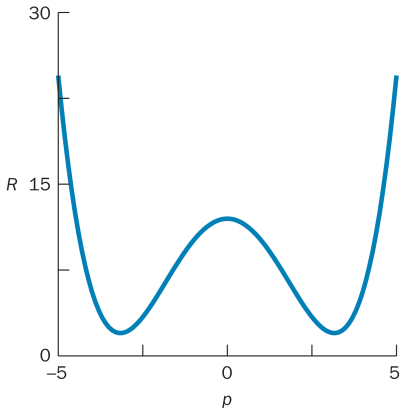


Figure 6.15 Two optimal solutions may be isolated. Here, the error function R assumes the same minimum value of 2 for $p \approx -3.16$ and $p \approx +3.16$. The error is higher for all other values of p.

It is possible that estimating parameters for various subsets of the same data may lead to major differences in the optimized parameter values, an observation called **sloppiness** [23]. Furthermore, and probably surprising, just computing averages of the values from different estimations does not necessarily provide a good overall fit. As an example, consider the task of fitting the function

$$w(t) = (p_1 - p_2 e^{-p_3 t})^{p_4} \tag{6.11}$$

to experimental data. This function, which has been used to describe animal growth [24], has four parameters that describe its sigmoid shape over time. Suppose two datasets were measured, and we fit them separately. The results (**Figure 6.16A, B**) are both good, and the optimized parameter values are given in **Table 6.5**. Next, we average the parameter values obtained from the estimations

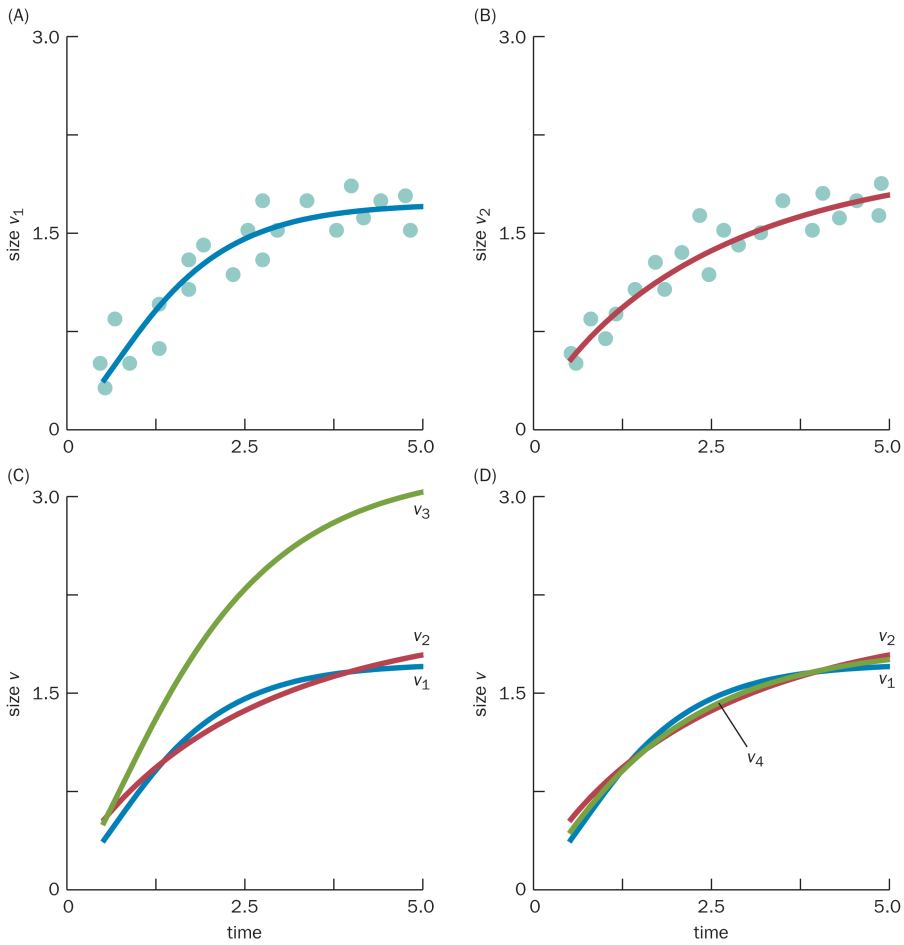


Figure 6.16 Data fits for the model (6.11) with parameter values listed in Table 6.5. (A) Fit v_1 for dataset 1. (B) Fit v_2 for dataset 2. (C) Fit v_3 with averaged parameter values. (D) Simultaneous fit v_4 to datasets 1 and 2.

TABLE 6.5: PARAMETER VALUES FOR DATA FITS WITH THE MODEL (6.11) IN FIGURE 6.16

Dataset/Parameter set	*Figure Panel*	p_1	p_2	p_3	p_4
1	A	1.2	0.8	1	3
2	B	2.5	2.45	0.35	0.8
Average of 1 and 2	C	1.85	1.625	0.675	1.9
simultaneous fit	D	1.68	1.6	0.6	1.2

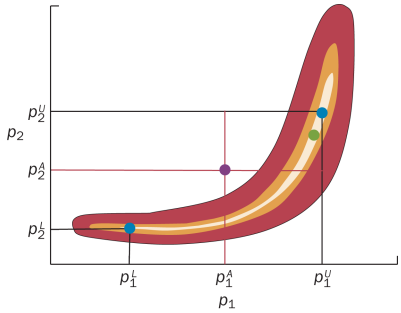


Figure 6.17 Averaging suitable parameter values from different estimations is not necessarily valid. Suppose suitable parameter combinations (p_1, p_2) (with low SSEs) are located within the pale orange region, while the orange and red regions are acceptable and barely satisfactory, respectively. All solutions outside the red region are unacceptable owing to high SSEs. The optimal solution is indicated by the green dot. Even if the solutions at the two blue dots, (p_1^L, p_2^L) and (p_1^U, p_2^U), are both suitable, averaging their parameter values (p_1^A, p_2^A) leads to an unacceptable solution (purple).

for the two datasets. Surprisingly, the fit is rather bad (Figure 5.16C). Indeed, much better parameter values are obtained with the earlier fits for only one dataset or from simultaneously fitting the two datasets (Figure 6.16D). How is that possible? At a coarse level, the answer is again that the parameters in many models are not independent of each other. Thus, if one parameter value is artificially changed, it is to some degree possible to compensate the resulting error by changing a second parameter value, as we encountered with the extreme case of parameters that always form a product $p_1 \cdot p_2$. In more general terms, the acceptable parameter vectors in these cases are actually part of a curved (possibly high-dimensional) cloud whose shape is governed by the dependences among the parameters, and all other points within this cloud are similarly acceptable parameter vectors [25, 26]. Averaging of results does not preserve the relationships within this cloud, and the short take-home message is that it is often not a good idea to average parameter values one by one from two or more fits (Figure 6.17).

Fitting two or more datasets separately does have some advantages. For instance, it indicates how stable an optimized parameter vector is, or how much it is affected by natural variability and experimental **uncertainty**. If the parameter values change drastically from one dataset to the next, none of the parameter vectors is likely to be reliable for predictions of untested scenarios. If many datasets are available, it may be beneficial to use some of the data as a **training set** for estimating parameters, while retaining the remaining, unused data for **validation**. In other words, one obtains parameters with some data and checks with the validation data how reliable the estimates from the training phase are.

If only one dataset is available, it is still to some degree possible to explore the reliability of the estimates. One option is sensitivity analysis (Chapter 5), while the other is the construction of clouds of candidate solutions. The basic concept is a resampling scheme such as the **bootstrap** or **jackknife** methods [27–29]. In bootstrapping, one draws a random sample from the data hundreds of times, estimates the parameters for each sample, and collects all results, which again form clouds of acceptable parameter vectors. In the jackknifing technique, one data point at a time is randomly eliminated and the parameter estimation is executed with the remaining points. Repeating the estimation many times, while leaving out a different point every time, leads to different solutions, which provide an indication of how much the acceptable parameter vectors may vary. The obvious caveat with these techniques is that all information is literally based on a single dataset and therefore does not account for natural variability that comes from different experiments.

6.4 PARAMETER ESTIMATION FOR SYSTEMS OF DIFFERENTIAL EQUATIONS

Essentially all principles and techniques of parameter estimation discussed so far are applicable to explicit functions as well as to dynamical systems consisting of sets of differential equations. Indeed, many of the methods introduced in this chapter are used broadly for optimizing parameters for machine learning algorithms, such as defining the shape of activation functions in convolutional neural networks. However, there are two main reasons why the estimation of dynamical systems is much harder. First, systems of differential equations usually contain many more parameters than individual functions, and we have already

had a glimpse into the challenges that come with larger numbers of parameters and the potential for a combinatorial explosion of possible sets of parameters. We saw this issue most clearly for grid searches but also for evolutionary methods like genetic algorithms that attempt to cover an exhaustive representation of the parameter space. The second issue making the estimation of dynamic systems difficult is that the necessary experimental data now consist of sets of data measured at a series of time points and these are to be compared with the solutions of the differential equations, which therefore require numerical integration of the equations during every parameter updating step. Typical data of this type consist of the expression of genes or the concentrations of metabolites, which are measured every so many seconds, minutes, or hours, following some stimulus. A specific example is a population of bacteria that have been starving for some while. At the beginning of the experiment (t = 0), a substrate such as glucose is added to the medium, and the measurements consist of concentrations of glycolytic metabolites every 30 seconds [30].

6.4.1 Parameter Estimation from Time-Series Data

To estimate parameters from time-series data, one starts again with some parameter vector (or a population of vectors), but, before a comparison between model and data is possible, one must solve the system of differential equations. It does not sound like much effort to perform this solution step, but computation-time studies have shown that this step may use more than 95% of the entire time needed to optimize a parameter vector [31]. To make matters worse, unreasonable parameter vectors, upon which a search algorithm may easily stumble once in a while, sometimes cause the numerical integration of differential equations to become so slow that the algorithm eventually runs out of time before any decent parameter vector has been obtained. These computational issues, added to the combinatorial explosion of large numbers of parameters to be estimated and other issues mentioned previously, render the estimation of dynamical systems a very challenging topic that continues to be the object of much attention within the community of computational biologists [2–5, 32, 33]. Since no general method is presently capable of solving all estimation issues related to dynamical systems, several shortcuts and means of simplification and amelioration have been proposed and are discussed in the following. Nonetheless, modern software does yield a good solution; an example is provided in **Box 6.2**.

BOX 6.2: PARAMETER ESTIMATION OF ORDINARY DIFFERENTIAL EQUATION MODELS IN MATLAB®

As an example for estimating parameter values in an ODE system, consider the pathway in Figure 1.

An ODE model is constructed under the assumption that adequate mathematical formats are known, yielding

$$\begin{aligned}
\dot{X}_1 &= 3 - p_1 X_1^{0.5} X_3^{-2} - \frac{p_2 X_1}{p_3 + X_1} \\
\dot{X}_2 &= p_1 X_1^{0.5} X_3^{-2} - p_4 X_2^{0.5} \\
\dot{X}_3 &= p_4 X_2^{0.5} - p_5 X_3 \\
\dot{X}_4 &= \frac{p_2 X_1}{p_3 + X_1} - p_6 X_4
\end{aligned} \qquad (1)$$

We furthermore assume that the input to the system is known (with a value of 3) and that the kinetic orders in the power-law functions may be set to default values, as discussed in Chapter 5 of [34]. Thus, only six parameters are to be estimated: four rate constants (p_1, p_4, p_5, and p_6) and the V_{max} and K_M of the Michaelis–Menten rate function (here called p_2 and p_3). The initial values could be considered as additional parameters, but we simply use the data at time t = 0.

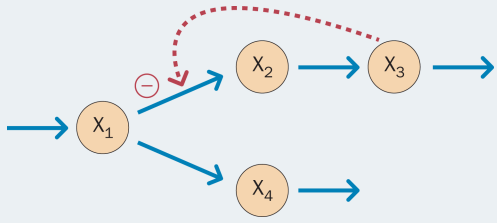


Figure 1 Branched pathway with feedback. The system has four variables and the representation in Eqn. (1), consisting of linear, power-law, and Michaelis–Menten terms (see Chapter 5).

Suppose that noisy data are available for the estimation process. Typical MATLAB® code, including the data, is shown in Figure 2. Because we do not know much about the rate constants, except that they must be positive, we initiate the system with $p_1 = \ldots = p_6 = 20$. The algorithm executes 765 iterations and produces optimized parameter values and the SSE as the "answer." The results are: $(p_1, \ldots, p_6)$ = (1.2579, 245.9365, 62.3755, 1.5464, 1.0329, 3.2094) and *ans* = 3.9780. It is impossible to

```
function z=Matlab_example_noise()
clc

p0=[ 20
     20
     20
     20
     20
     20

];

% z=Cost(p0,1);
options=optimset('Display','iter');
p2=fminsearch(@Cost,p0,options,0);
z=Cost(p2,1);


end
function z=Cost(pp,g)
p=abs(pp);



x0=[2
    0.5
    0.4
    1.5];
Tspan=0:.05:2;

[t,y]=ode23s(@My_ode,Tspan,x0,[],p);

DataE=dataset(3);

DataS=interp1(t,y(:,:),DataE(:,1));

DSE=[DataE(:,2:5)-DataS];

Cost=sum(sum(DSE.^2));

if g==1
    subplot(2,2,1)
    plot(t,y(:,1))
    hold on
    plot(DataE(:,1),DataE(:,2),'o')
    hold off
    xlabel('Time');ylabel('Substrate (X1)'); title('Extracellular')
    subplot(2,2,2)
    plot(t,y(:,4))
    hold on
    plot(DataE(:,1),DataE(:,5),'o')
    hold off
    xlabel('Time');ylabel('Product (X4)'); title('Extracellular')
    subplot(2,2,3)
    plot(t,y(:,2))
    hold on
    plot(DataE(:,1),DataE(:,3),'o')
    hold off
    xlabel('Time');ylabel('Intermediate (X2)');  title('Intracellular')
    subplot(2,2,4)
    plot(t,y(:,3))
    hold on
    plot(DataE(:,1),DataE(:,4),'o')
    hold off
    xlabel('Time');ylabel('Intermediates (X3)');  title('Intracellular')
end

z=Cost;
end
function dxdt=My_ode(~,x,p)

```

Figure 2 Typical MATLAB® code for parameter estimation. The data are listed in lines 82–102. MATLAB® permits numerous options for settings and details of the plots.

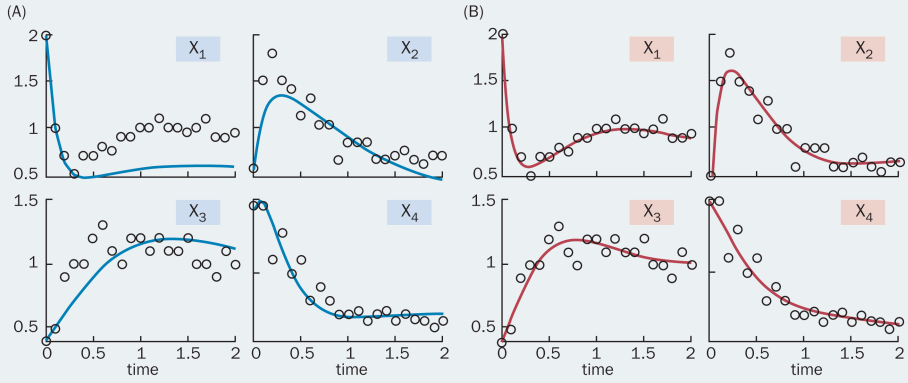


Figure 3 Visual results of MATLAB® parameter estimation for the model in Figure 1 and the data shown in Figure 2. A: Starting with $p_1 = \ldots = p_6 = 20$ does not yield entirely satisfactory fits, especially for X_1. B: However, using these results as start values for a second round of optimization yields good fits.

interpret this SSE as good or bad by itself, but a plot of the four variables, superimposed on the data, is not particularly pleasing (**Figure 3A**). The question is now whether this is the best the model can accomplish. A partial answer comes from additional MATLAB® output stating that the default "maximum number of function evaluations has been exceeded." Running the code again, this time with the solution vector as initial guestimates for the parameter values, quickly leads to a much-improved solution: $(p_1, \ldots, p_6) = (2.4580, 3.4029, 2.6019, 2.7134, 2.1877, 1.5359)$ with $ans = 0.5342$. The plots are now much closer to the data (Figure 3B).

A very effective strategy to counteract the computational challenges of estimating parameters in dynamical systems is the avoidance of integrating the differential equations. The rationale is the following. The derivative on the left-hand side of a differential equation is the slope of the time course of the variable at a given point. Thus, if we can estimate this slope, we can circumvent integrating the differential equation. It may be best to demonstrate the approach with a simple example.

Suppose we are studying the growth of a bacterial population, which begins with two units (maybe two million cells) and grows to 100 units. As a mathematical description, we assume the famous logistic growth function, which was proposed over 150 years ago [35] and has the mathematical formulation

$$\dot{N} = aN - bN^2. \tag{6.12}$$

It contains two parameters, a and b (see also Chapters 5 and 14). For ease of discussion, we pretend to have noise-free data N, which are shown in Figure 6.18A. Because the data are so clean, it is easy to estimate, at each time point, the slope S of the growth curve; the slopes are also shown in the figure. The slope estimation in this case may be done in the following fashion, which is sometimes called the three-point method (Figure 6.18B). For the two subsequent data points N_{i-1} and N_i, which are measured at time points t_{i-1} and t_i, compute the difference $N_i - N_{i-1}$ and divide it by $t_i - t_{i-1}$; denote the result by s_i^-. Do the analogous computation for N_i and N_{i+1}, which are measured at t_i and t_{i+1}, and call the result s_i^+. The average of s_i^- and s_i^+ is an estimate of the slope t_i at (t_i, N_i). This method does not allow the computation of slopes at the first and last data points, but one can often do with two fewer points. Much more sophisticated and accurate methods are available for **smoothing** time courses and estimating slopes even in cases where the data are noisy [36, 37].

The slope estimation leads to an augmented dataset, which consists of three variables, namely t_i, N_i, S_i (**Table 6.6**). Substituting the estimated slopes S_i for $\dot{N}(t_i)$ transforms the estimation task with one differential equation into as many decoupled algebraic equations as there are data points (possibly without the first and last point). The original task is now computationally much easier, because the algorithm no longer needs to solve the differential equation numerically. In the new system, the i^{th} algebraic equation looks like

$$S_i = aN_i - bN_i^2 \tag{6.13}$$

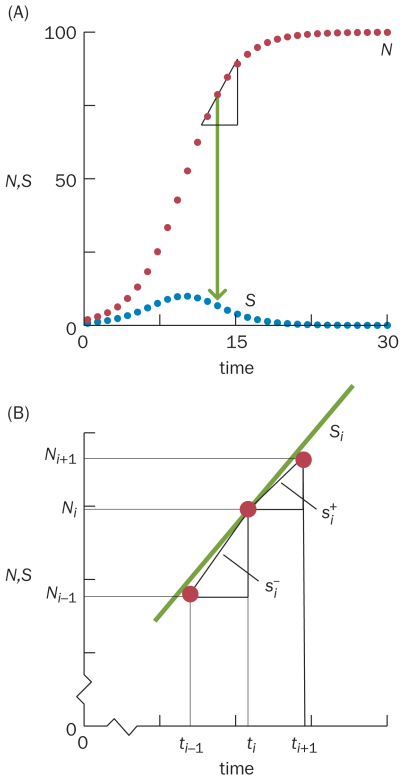


Figure 6.18 Estimation of parameters in (6.12). (A) Growth curve of a bacterial population with size N. At each time point, the slope of the growth curve is estimated and plotted as S. (B) The slope estimation may be performed by averaging the quantities s_i^- and s_i^+, as explained in the text. The result is the estimated slope S_i of the green line at the point (t_i, N_i) of the growth curve.

in symbolic form. For the parameter estimation, the values for N and S are entered into this set of equations. As an illustration, the first two equations of this set, using S from the three-point method, are

$$\begin{aligned} 1.80 &= a \cdot 4.34 - b \cdot 4.34^2 \\ 3.51 &= a \cdot 9.18 - b \cdot 9.18^2 \end{aligned} \tag{6.14}$$

Using a gradient method, such as **`nlinfit`** in MATLAB®, we obtain parameter values very quickly as $a = 0.3899$ and $b = 0.0039$, which are quite close to the true values $a = 0.4$ and $b = 0.004$, which we know in this case from constructing the example.

Importantly, the method of estimating slopes and converting systems of differential equations into algebraic equations also works for systems of more than one variable. Let us illustrate the method for a dynamic system with two variables. For a change, this time the example comes from ecology. Suppose two species N_1 and N_2 are living in the same habitat. N_1 is an herbivore that once in a while is eaten by a predatory carnivore from population N_2. One standard description is a Lotka–Volterra model (see Chapters 5 and 14) of the form

$$\begin{aligned} \dot{N}_1 &= N_1(a_1 - b_1 N_1 - b_2 N_2), \\ \dot{N}_2 &= N_2(a_2 N_1 + a_3 - b_3 N_2). \end{aligned} \tag{6.15}$$

Following standard techniques for setting up the model, N_1 grows exponentially with rate a_1 and has a natural death rate of b_1. It is typical to represent the death term by the square of N_1, which is sometimes interpreted as crowding within the population. The population is also subject to decimation by population N_2 with rate b_2. This process depends on the number of encounters between the two species and is therefore represented by the product of N_1 and N_2. N_2 feeds on N_1 with a rate a_2, and this process augments its growth from other food sources, which is assumed to occur with rate a_3. The predators die with rate b_3. Note that a_2 and b_2 typically have different values, because the growth benefit to the predator is seldom exactly equivalent to the effect on the prey.

TABLE 6.6: DATASET USED FOR ESTIMATING THE PARAMETERS IN (6.12)*

t	*N*	*S (true)*	*S (3 – point)*
0	2	0.78	†
2	4.34	1.66	1.80
4	9.18	3.33	3.51
6	18.36	6.00	6.05
8	33.36	8.89	8.58
10	52.70	9.97	9.47
12	71.26	8.19	7.99
14	84.66	5.19	5.30
16	92.47	2.78	2.95
18	96.47	1.36	1.48
20	98.38	0.64	0.70
22	99.27	0.29	0.32
24	99.67	0.13	0.15
26	99.85	0.06	0.07
28	99.93	0.03	0.03
30	99.97	0.01	†

* The data consist of the numbers of bacteria (in units of millions), the true slopes, which are usually not known and the slopes estimated with the three-point method.
† Slopes at times $t = 0$ and $t = 30$ cannot be obtained with this method.

Suppose we have reason to believe that the model formulation in (6.15) is adequate, but that we do not know appropriate parameter values. We do have data, which stem from a series of measurements around some devastating event for the prey. Specifically, the populations are both at steady state, with about four times as many predators as prey. At time $t = 1$, about 80% of the prey population succumb to some disease. Subsequent measurements of N_1 and N_2 show how both populations respond to the catastrophic event: initially, N_2 lacks food and decreases a little bit, while N_1 begins to recover. Subsequently, both N_1 and N_2 increase and decrease with a few damped oscillations and finally return to their original steady-state values. Since this is just a demonstration, let us begin by assuming that we have comprehensive, error-free measurements, which are given in **Table 6.7**. In addition to the population sizes, the table also lists the slopes of N_1 and N_2, which are denoted by S_1 and S_2, respectively. At time $t = 1$, the abrupt disease event happens, and the slopes at this point are not defined.

The estimation now proceeds by setting up two sets of algebraic equations: one for N_1 and one for N_2. The first two equations of the first set are

$$S_1(1.1) = N_1(1.1)\left[a_1 - b_1 N_1(1.1) - b_2 N_2(1.1)\right],$$
$$S_1(1.2) = N_1(1.2)\left[a_1 - b_1 N_1(1.2) - b_2 N_2(1.2)\right] \tag{6.16}$$

TABLE 6.7: DATA FOR THE ESTIMATION OF PARAMETERS IN (6.15)*

t	N_1	N_2	S_1	S_2
0	139.77	38.88	0	0
0.25	139.77	38.88	0	0
0.5	139.77	38.88	0	0
0.75	139.77	38.88	0	0
1	139.77	38.88	–	–
†				
1	27.95	38.88	–	–
1.1	63.35	26.89	798.41	−44.71
1.2	204.04	30.86	1531.64	143.72
1.3	182.09	45.96	−1350.61	38.67
1.4	107.57	40.61	−158.29	−78.03
1.5	123.83	35.97	376.76	−9.81
1.6	156.03	37.95	125.05	37.21
1.7	145.80	40.31	−214.54	1.83
1.8	132.38	39.13	−25.07	−16.21
1.9	137.55	38.25	89.86	0.12
2	143.28	38.80	7.54	7.37
2.1	140.51	39.17	−41.59	−0.61
2.2	138.19	38.90	−0.11	−3.18
2.3	139.55	38.75	18.30	0.47
2.4	140.50	38.88	−1.14	1.39
2.5	139.82	38.94	−8.07	−0.31
2.6	139.45	38.88	1.06	−0.59
2.7	139.77	38.86	3.51	0.18
2.8	139.92	38.89	−0.71	0.25
2.9	139.76	38.89	−1.51	−0.10

* The data consist of error-free measurements of the sizes of the two populations N_1 and N_2, as well as slopes at each time point.
† At time $t = 1$, population N_1 abruptly falls to about 20% of its steady-state value, for reasons not represented in the model. At this point, slopes are not defined.

Substituting numerical values for S_1, N_1, and N_2 in the first equation yields

$$798.41 = 63.35(a_1 - b_1 63.35 - b_2 26.89). \tag{6.17}$$

Similar equations with numerical values listed in Table 6.7 are composed for time points 1.2, . . . , 2.9. Again using **nlinfit** in MATLAB®, we obtain the exact parameters for the first equation in less than a second. Indeed, although the residual error SSE for the initial guestimate (2, 2, 2) is huge at about 5×10^{10}, the algorithm converges to the correct solution ($a_1 = 40$, $b_1 = 0.008$, $b_2 = 1.0$) within only seven iterations. In exactly the same fashion, the parameter values for the equation for N_2 are computed very quickly as ($a_2 = 0.05$, $a_3 = 0.01$, $b_3 = 0.18$). The optimized solution perfectly matches the noise-free data (Figure 6.19).

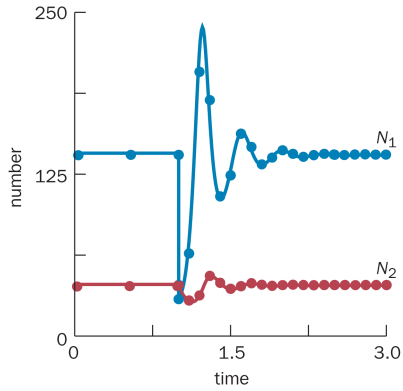


Figure 6.19 Error-free data (symbols) corresponding to the model (6.15) and solution with optimized parameters (lines). At time $t = 1$, 80% of population N_1 succumb to an unexplained disease. By time $t = 3$, both populations have returned to their steady-state values.

It should be mentioned that this specific example would have allowed a simpler method of estimation. Namely, we could have divided all equations in (6.16) by N_1, which would have made the estimation linear [5, 38]. This strategy is left as Exercise 6.17. It is very powerful, because linear regression has an analytical solution and therefore does not require search algorithms, as we discussed before. In fact, this strategy can be used even for very large systems, such as mixed bacterial communities [39] (see Chapter 15). However, the Lotka–Volterra system is a special case, and the slope estimation method does not convert most other nonlinear into linear models.

Of course, actual biological data contain noise, and this noise is usually amplified when slopes are computed. Nonetheless, the slope method is often a good start, especially if the noise is moderate. As an example, suppose noisy data are given, with values starting after the crash of population N_1 (Table 6.8). Using the same techniques of replacing the derivatives with slopes (which are also noisy now), we use **nlinfit** to compute parameters. Again, it does not take long to estimate the parameters, and the results are ($a_1 = 24.0126$, $b_1 = 0.0031$, $b_2 = 0.6159$) and ($a_2 = 0.0376$, $a_3 = -1.6032$, $b_3 = 0.0933$). The first observation is that these numbers are different from the true values, which is to be expected because of the noise in the data and the slopes. Some values are off by quite a bit, and it can be seen in particular that a_3 is actually negative rather than positive, which would change its interpretation from a birth to a death process. However, it is difficult to judge how good these parameter values are just by looking at the numbers. The first test is therefore a plot of the slopes as functions for N_1 and N_2. An example is shown in Figure 6.20, where S_1 is plotted against N_1. At first, the spiral in this plot may be confusing. It is due to the fact that N_1 increases and decreases, thereby moving back and forth along the horizontal axis. At the same time, S_1 moves up and down, there by yielding the spiral pattern. The red spiral shows the true dynamics (with correct parameter values), the blue line with symbols represents the noisy data in Table 6.8, and the green line shows the dynamics of the model with optimized parameter values. Visual inspection suggests that the fit is not so bad. The real test is the use of the optimized parameter values in the model equations (6.15). Three results are shown in Figure 6.21. In panel (A), noisy data and slopes are used only for N_1, while the parameter values for a_3 are taken as the true values. Panel (B) shows the analogous situation for noisy N_2 and correct N_1 data. Panel (C) shows the results of adding noise to both N_1 and N_2. The main observations are that the overall dynamic patterns are retained but that the timing of the curves is not quite right. This deviation in time is a typical outcome of using the slope method. The reason is that when derivatives are substituted with slopes, the regression does not explicitly use time. If there is no noise, the fit is usually very good. However, the consequences of noise often appear in the form of some time warp. This is the cost of the ease and speed of the slope substitution method.

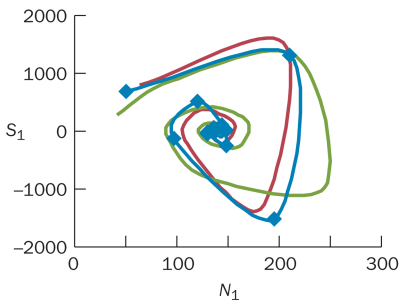


Figure 6.20 Assessing the oscillatory behavior of the estimated system (6.15). Shown here is the plot of the slope S_1 versus the size of population N_1. The red curve shows the true, noise-free relationship, the blue-connected symbols represent noisy data, and the green curve shows the relationship optimized for noisy data.

If biological considerations mandate that a parameter value should be positive, as we would suspect in the case of a_3, the parameter estimation procedure can be set up to enforce solutions to stay within admissible ranges.

As a control, we estimate parameter values from the noise-free and noisy data, using the differential equations (6.15) directly. Again, the result is typical. Assuming the algorithm converges to the correct solution, this solution is usually better than the solution obtained with the slope substitution method. However, it often

TABLE 6.8: DATA FOR ASSESSING THE ROLE OF NOISE IN THE ESTIMATION OF PARAMETERS IN (6.15)*

t	N_1 *with Noise*	N_2 *with Noise*	S_1 *with Noise*	S_2 *with Noise*
1	20	44	–	–
1.1	50	25	690	−60
1.2	210	30	1310	132
1.3	195	48	−1520	30
1.4	97	37	−120	−52
1.5	120	33	510	−18
1.6	144	41	112	50
1.7	148	41	−255	0
1.8	130	44	−10	−22
1.9	136	37	70	0
2	148	35	10	10
2.1	130	36	−50	0
2.2	142	40	0	−2
2.3	140	41	27	4
2.4	136	34	0	2
2.5	137	36	−12	−1
2.6	135	38	4	0
2.7	144	35	4	0
2.8	142	41	0	0
2.9	138	38	0	0

* Noisy measurements of the sizes of the two populations N_1 and N_2, as well as the slopes measured at each time point, starting at time $t = 1$, after population N_1 had abruptly fallen to about 20% of its steady-state value

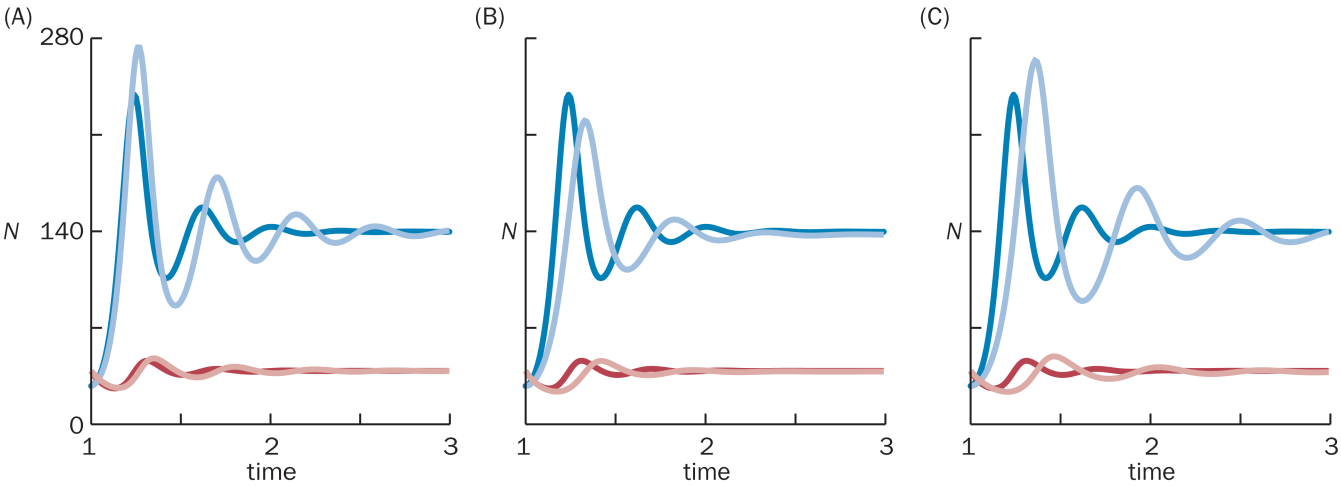


Figure 6.21 Simulations of the population model with parameter values estimated from noisy data. (A) Only N_1 data noisy. (B) Only N_2 data noisy. (C) Both N_1 and N_2 data noisy. The plots start at the same perturbed value at time 1 as in Figure 5.19.

takes much, much longer to obtain this solution, especially for larger systems, and sometimes the algorithms do not converge at all to an acceptable solution.

The website supporting the book (www.garlandscience.com/product/isbn/9780815344674) contains MATLAB® code for a parameter estimation program that first uses a genetic algorithm to obtain a coarse global solution and then hands the results over to a gradient method to refine this solution. The important settings for

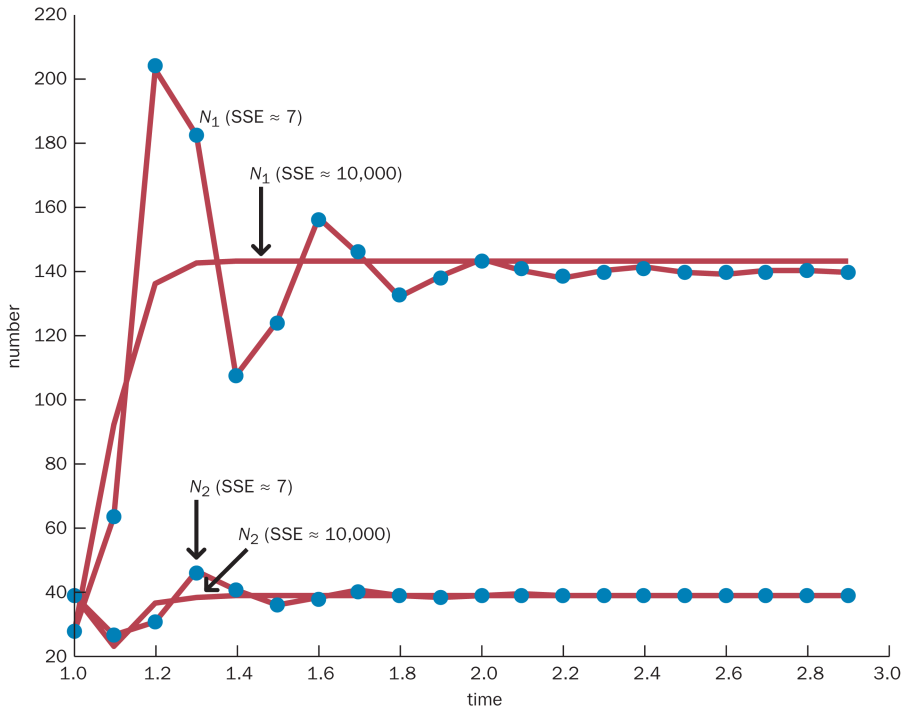


Figure 6.22 MATLAB® output of two optimization runs. Noise-free data are represented as symbols. A "correct" solution with SSE ≈ 7, connected with straight-line segments, essentially hits all points perfectly. However, the same optimization settings may run into a local minimum with SSE ≈ 10,000, which corresponds to an unsatisfactory solution, at least for N_1.

the genetic algorithm are the population size (PopSize), the number of generations (GenSize), the number of best genes retained for the next generation (EliteCount), a limit to the number of consecutive generations if there is no significant improvement in fit (GenLimit), and sets of admissible lower and upper bounds for the parameters. These settings allow a lot of flexibility, and, even in our case, they can make quite a bit of difference. With the settings on the website, the algorithm often actually converges to a good solution. However, because genetic algorithms are stochastic, running the same program with the same settings and data several times typically leads to different solutions. Five examples are given in Table 6.9.

The initial SSE of the best solution automatically created by the GA is of the order of 10^5. It decreases to somewhere between 10^3 and 10^4 at completion of the GA, depending on the random numbers the algorithm uses during the stochastic evolution. Under opportune conditions, the subsequent gradient method reduces this error to between 0 and 10. Interestingly, even if the residual error at the end of the GA phase is very similar in two runs, the gradient method sometimes converges to the correct solution and sometimes to an unsatisfactory local minimum, which corresponds to the simple shoulder curve for N_1 in Figure 6.22. The combined process may take several minutes to complete, which is much longer than

TABLE 6.9: TRUE AND OPTIMIZED SOLUTIONS FOR NOISE-FREE POPULATION DATA (SEE (6.15))

Run	a_1	b_1	b_2	a_2	a_3	b_3	*SSE*
1	89.6965	0.6422	0.0000	0.4296	2.4108	1.6351	13664.2
2	99.8746	0.0000	2.5988	0.4181	0.0000	1.5318	11939.6
3	38.4554	0.0064	0.9601	0.0525	0.0048	0.1879	7.44405
4	38.5346	0.0065	0.9623	0.0524	0.0012	0.1874	6.99142
5	20.7977	0.0000	0.5359	1.2265	4.8767	4.6515	9162.04
True	40	0.008	1	0.05	0.01	0.18	0

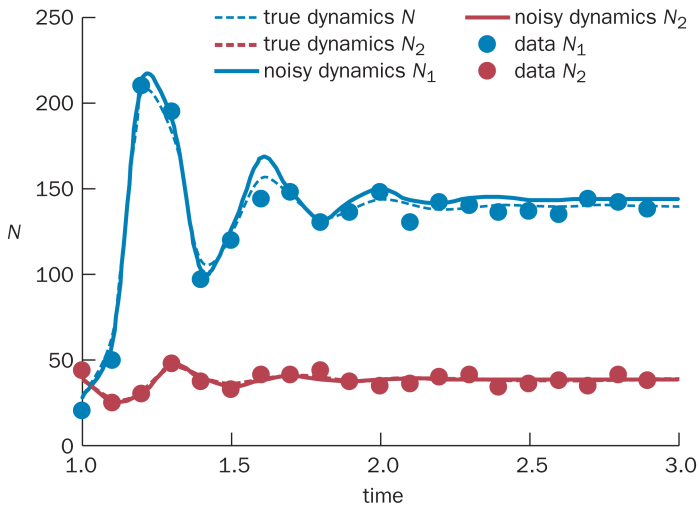


Figure 6.23 Effect of noise in the data on parameter estimates. The figure shows the true dynamics of the population system (dashed lines), noisy data (dots), and the dynamics estimated from these noisy data (solid lines). Even with an SSE of about 1,000, the optimized solutions are not bad. In fact, the fit for N_2 (red) is essentially indistinguishable from the true solution.

the time needed for the decoupled system with estimated slopes. The residual error is almost never 0, even in cases of convergence, but a solution with SSE ≈ 7 is essentially perfect in this case (see Figure 6.22).

Depending on the upper bounds for the unknown parameters, randomly chosen start values may be so far off that the integration of the differential equations stops immediately with an error message. If the GA does get started, it often terminates with a higher SSE than in the cases before. In many cases, the gradient method terminates with a bad solution, but there are also cases where the iterative search is successful. Finally, if the data are noisy, we can of course not expect the SSE to decrease to single digits. If the algorithm converges to the right neighborhood, the SSE in the example is between 900 and 1,000, which looks like a big number but in fact is not a bad solution (**Figure 6.23**).

In the previous examples, we have implicitly assumed that we know the true initial values for integrating the differential equations. This is a common assumption, which may or may not be acceptable. Instead of making the assumption, the initial values may be considered as parameters that are to be optimized along with the parameters we discussed earlier.

6.5 IDENTIFICATION OF THE MOST APPROPRIATE FUNCTIONAL REPRESENTATIONS

Before we can even start estimating parameter values, we need to identify all functions in the model of interest. In some cases, such as purely physical phenomena, appropriate functions are dictated by theory. For instance, we know how to describe forces and processes like heating and cooling. For biological processes, such knowledge is mostly lacking. We could argue that biological systems must satisfy all laws of physics, and while that is certainly true, the assertion is often not helpful because biological phenomena are convolutions of very many physical events [40]. For instance, consider a hormone that causes a change in gene expression in some target cells. The hormone needs to find each target cell and bind to a receptor on its surface. The 3d-shape of the receptor changes, leading to molecular changes that trigger the activation of one or more internal signaling cascades (see Chapter 10). The result of this activation is the translocation of a transcription factor (TF) into the nucleus of the cell. Inside the nucleus, the TF finds one or more DNA sequences with the correct response elements, and upon binding, the following gene is transcribed. Imagine dissecting this process into steps that permit direct physical modeling!

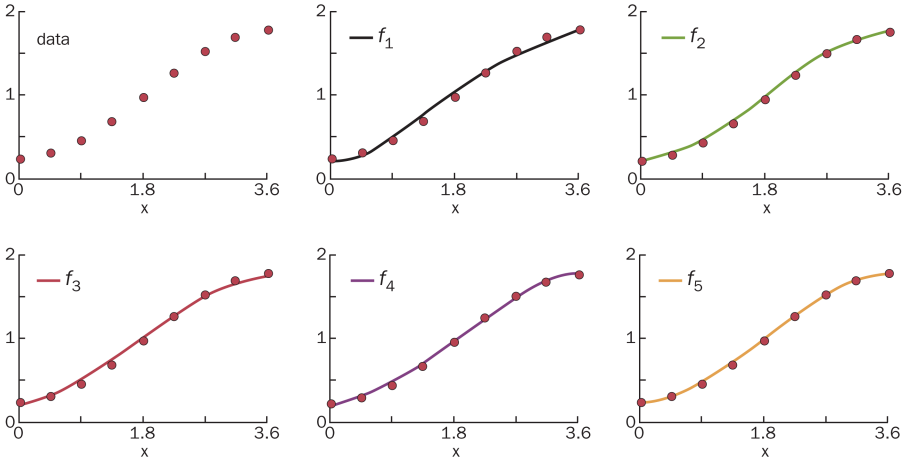


Figure 6.24 Numerous functions provide pleasing fits to the synthetic data in the top-left panel. Synthetic data and fits with a shifted Hill function $f_1 = 0.2 + \frac{2.2\,x^2}{2.3^2 + x^2}$, a shifted arctangent function $f_2 = 1 + 0.77 \arctan(0.9\,x - 1.6)$, a logistic function $f_3 = \frac{1.9}{1 + \exp(2.2 - 1.3\,x)}$, a stretched normal distribution function $f_4 = \frac{8.1}{\sigma\sqrt{2\pi}}\, e^{-\frac{1}{2}\left[\frac{x-u}{\sigma}\right]^2}$, and a shifted sine function $f_5 = 1 + 0.78\sin(0.82\,x + 4.8)$. In addition, other well-fitting functions could be generated. The data fit itself does not suggest which function is most appropriate; see, however, Exercise 6.27 for extrapolation toward larger *x*-values. Adapted from (Torres and Voit (2002)).

Even for much simpler situations in biology, the best functions are seldom prescribed. As an example, consider the artificial data in **Figure 6.24** that exhibit a sigmoidal trend. Functions with very different mathematical formats offer good fits to these data.

Before we attempt to justify criteria for the choice of one function rather than an alternative, we should explore whether it really matters what function we use, as long as it fits the data. As so often, the answer depends on details. For interpolation purposes, that is, for determining the functional values for x-values that were not measured but fall within the measurement range (0 to 3.6 in Figure 6.24), any of the shown functions will do. However, extrapolations beyond this range may cause very dissatisfactory results, as one might expect from sine and normal distribution functions (Exercise 6.27).

For a different example, adapted from [41], consider the small branched pathway in **Figure 6.25** and a model with two alternative functions for one of the processes, namely

$$\begin{aligned}
\dot{X}_1 &= 1.2X_3^{-1} - F(X_1) - 0.2X_1 \\
\dot{X}_2 &= F(X_1) - X_2 \\
\dot{X}_3 &= X_2 - X_3 \\
\dot{X}_4 &= 0.2X_1 - 0.2X_4
\end{aligned} \tag{6.18A}$$

where $F(X_1)$ is either the Hill function

$$H(X_1) = \frac{1.2 \cdot X_1^4}{0.2 + X_1^4} \tag{6.18B}$$

or the linear function

$$L(X_1) = X_1 \tag{6.18C}$$

Thus, the only difference between these models is the function for the conversion of X_1 into X_2 (Eqns. (6.18B, C)).

Clearly, both models represent the synthetic dataset in Figure 6.26A well. However, if we simulate an experiment in which a bolus of 2 units is added to X_1, the two models yield distinctly different outputs (Figure 6.26B). Thus, the model choice is important.

Unfortunately, there are no nature-given guidelines for the choice of a mathematical format, leaving us with three choices.

1. Find some function that matches our data and hope for the best. If additional data become available, we may revisit our choice.

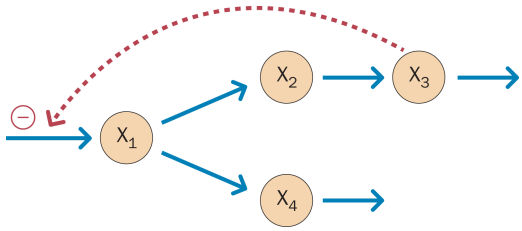


Figure 6.25 Branched pathway. The pathway is modeled with two variations of the same ODEs, as shown in Eqns. (6.18).

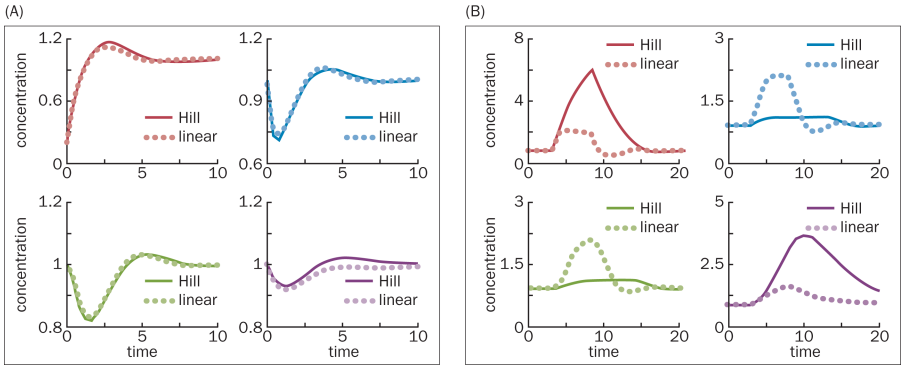


Figure 6.26 Results of modeling the pathway in Figure 6.25 with two model variants of the model in (6.18). (A) Responses of the four system variables to starting the system at $X_1(0) = 0.2$. (B) Responses of the four system variables to a bolus of 2 units into X_1 from $t = 3$ to 8.

2. Search the literature for what others have done in a comparable situation and hope that they had good reason for their choices and that our situation is really similar to theirs. For instance, in the context of enzyme kinetics (Chapter 9), one could use Michaelis–Menten rate laws, Hill functions, or more general constructions [42]. However, it does not take much imagination to see that the numbers of possible candidates and their combinations are limitless [41].
3. Use a generic approximation, such as a power-law function, which is guaranteed by Taylor's theorem to be excellent in the vicinity of an operating point (Chapter 5) and suppose that the system under study does not deviate too far from this operating point.

All three alternatives have been pursued in the literature uncounted times. If the variables of the system indeed do not veer too far from their normal values, the alternative choices often yield rather similar results (e.g., [43–45]), however, there is no guarantee.

Under opportune conditions, the recent method of Dynamic Flux Estimation permits the analysis of metabolic time series data toward inferences of the functional shapes of the various processes within an ODE system [46, 47].

There are no easy solutions to the problem of identifying well-suited functions, and much more research will be required to develop promising solution strategies. Sometimes, biological insight may help with weeding out mathematically unacceptable functions [22]. For instance, a search algorithm may identify a parameter value as negative whereas biological reality mandates a positive value, for instance, in the case of a turnover rate or of a number of items participating in a process. More generically, any single solution computed with a search algorithm sometimes turns out to be unreliable. To address this issue, it has become popular not to search for the uniquely optimal solution but for an entire **ensemble** of parameter settings that all, more or less, have the same SSE [5, 48–50].

Two additional issues arise in the function identification task. First, it might happen that the data are simply insufficient in quantity and/or quality for identifying the true model structure, and many groups have started analyzing this topic of **identifiability** with different approaches (for example, [3, 25, 48, 49, 51, 52]). An effective approach toward this issue is often the computation of the Fisher information matrix, which quantifies the amount of information about a parameter that can be inferred from a variable [53]. Second, related to the identifiability issue, it is possible that the estimated model contains more parameters than can be validly estimated from the data so that predictions toward new data may become unreliable. Methods of cross-validation can help to some degree with this **overfitting** issue [54].

6.6 DISCOVERY OF THE OPTIMAL STRUCTURE

The estimation of parameters in the first part of this chapter required knowledge of the functions whose parameters were to be estimated. The second part addressed the identification of adequate functions for allegedly important processes within a system. In this third part we ask whether our box-and-arrow diagrams faithfully reflect the designs of biological systems in an appropriate manner. An underlying assumption is that biological systems are typically optimized, at least to some extent and compared to alternative designs, because evolution would presumably have selected against a particular design if better alternatives had been available. Some caution is necessary with this statement though because one should expect that evolution is not yet done and that the near or distant future will bring new designs that are even better than what we observe today. Nonetheless, nature has had 3.5 billion years to weed out flawed biological designs, and what we observe today probably offers better guidance regarding design questions than just thinking about them.

The first obvious design question is whether we have included all important variables in our diagram and whether we could omit some variables without compromising the integrity of the system. To some degree, it is possible to use statistical techniques to decide on the inclusion or exclusion of variables in large datasets. This process, called **feature selection**, evaluates significant relationships among variables and eliminates variables that are not very consequential. Ideally, this method reduces the number of variables to an acceptable minimum. Once the list of variables is finalized, one attempts to characterize for each variable which other variables it affects, and, possibly, whether the effects are likely be augmenting or diminishing. This step defines the likely topology of the system, and once this topology has been established, at least in a coarse manner, one might select mechanistic or canonical models for developing a specific symbolic and numeric representation [46]. While all this may sound reasonable, the design of the best box-and-arrow diagram is seldom trivial and, in the end, the ultimate structure of a model becomes a matter of consulting with subject area experts and, yes, some educated guessing.

Because parameter identifiability tends to complicate the discovery of optimal numerical assignments and model fitting, many modelers subscribe to Occam's Razor: the simplest explanation of a system is typically the correct one. Does biology follow this rule? Not necessarily. Genomic events from eons of evolution may result in multiple gene products performing the same function which, on the one hand, provides redundancy that might be considered superfluous. On the other hand, duplication creates robustness, which is a clear advantage.

Many quantitative metrics for selecting the best model out of a domain of candidate structures rely on a penalty for additional parameters (which tend to correspond to larger numbers of variables). Two of the most popular statistical metrics for model selection, the **Akaike Information Criterion (AIC)** [55] and the **Bayesian Information Criterion (BIC)** [56], balance the quality of model fit with the number of parameters.

Finally, an important detail of this design process is the specification of regulation, which is often much less clear in detail than the importance of variables and material fluxes. Guidance for this detail comes from the study of natural **design principles.**

6.6.1 Design Principles

Design principles are "rules that characterize some biological feature exhibited by a class of systems such that discovery of the rule allows one not only to understand known instances but also to predict new instances within the class" [57]. Spearheading the search for such principles, Michael Savageau [58, 59] asked whether objective, quantifiable reasons existed for particular designs that had been observed in nature many times. Pragmatically, the discovery of design principles in dynamic systems is based on comparisons between similar systems. Uri Alon and others provided detailed analyses of the designs of small systems and of recurring **motifs** [60–71].

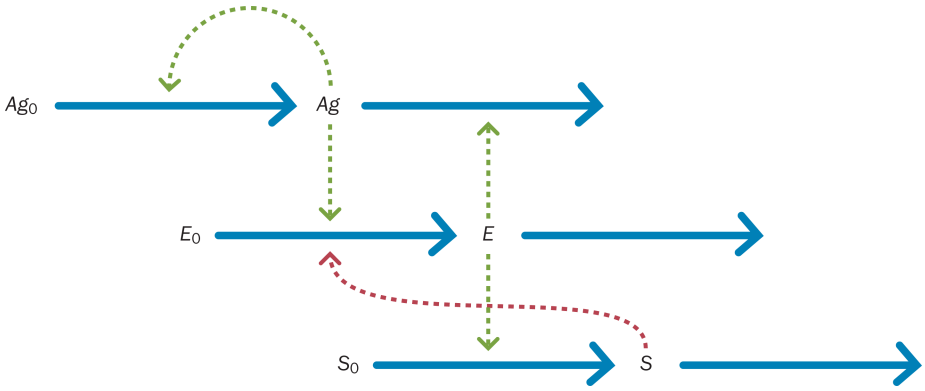


Figure 6.27 Simplified immune cascade. An external source antigen Ag_0 stimulates an increase in the internal antigen level *Ag*. *Ag* triggers the maturation of effector lymphocytes from precursors (E_0) to fully functional cells (*E*). *E* in turn triggers the maturation of precursor suppressor cells (S_0) into functional suppressor lymphocytes *S*. In the observed system (Model A), *S* inhibits the maturation of E_0 (red arrow), while this feedback is missing in the hypothetical alternative system (Model B). See the text for further explanations.

As a representative example for the way one might think about design principles, we revisit an early analysis of a cascaded system involved in immune responses (**Figure 6.27**) [68, 69]. Here a source antigen $Ag_0(= X_0)$ describes a potentially infectious microorganism in the environment and $Ag_0(= X_1)$ refers to microorganisms in the body that disrupt the host's physiological functioning. The variable $E_0(= X_2)$ represents immature effector cells, while $E(= X_3)$ represents mature effector lymphocytes. $S_0(= X_4)$ and $S(= X_5)$ are precursors and mature suppressor lymphocytes, respectively. The variables X_0, X_2, and X_4 in this set-up are constant inputs to the system and therefore independent; that is, they do not require their own differential equations (see Chapter 2).

The question of primary interest was why suppressor lymphocytes inhibit the maturation of effector cells, as suggested by experimental observations. Such a question is very difficult to answer with wet experiments, because one cannot easily eliminate this inhibition signal in a living organism without changing all kinds of other features of the system. Irvine and Savageau therefore developed and used a mathematical approach, called the **method of mathematically controlled comparison** (MMCC), which is particularly powerful for canonical models (see Chapter 5), where each parameter has a clearly defined role and meaning. In this method, two system models are analyzed side by side. One model corresponds to the observed design, while the other, alternative, model is different in the feature of primary interest, which in this case is the inhibition signal. The system is shown in Figure 6.27.

According to the requirement of internal equivalence within MMCC, the parameters that reflect features common to both models receive the same values. The difference between the models is thus manifest in one distinguishing feature, which is typically reflected in one or two terms of the model and its parameters. While many analyses within MMCC can actually be performed symbolically [68], we use for our demonstration a numerical system implementation of an S-system model (see [72] and Chapter 5):

$$\begin{aligned}
\dot{X}_1 &= 2.02 X_0 X_1^{0.9} - 2 X_1 X_3^{0.5}, \\
\dot{X}_3 &= \alpha X_1^{g_{31}} X_2^{g_{32}} X_5^{g_{35}} - X_3, \\
\dot{X}_5 &= X_3^{0.5} X_4 - X_5.
\end{aligned} \tag{6.19}$$

Note that the parameters in the production term for X_3 are not yet numerically specified. We do know that in the system with feedback (which we will refer to as Model A), the kinetic order g_{35} must have a negative value (here $g_{35} = -0.5$), because it represents inhibition, while it is zero in the case without feedback (which we call Model B). This numerical difference changes the value of the production term of X_3 and thereby the steady state of the entire system. To counteract this undesirable side effect, MMCC suggests enforcing external equivalence, which means that the two systems should appear to the outside observer as similar as possible. For instance, adjusting in Model B the rate constant α, which has a value of 1 in Model A, could be used to ensure that both systems have the same steady state. In the present case, we have further degrees of freedom, because the distinguishing term also contains the parameters g_{31} and g_{32}, which in Model A have values of 0.5 and

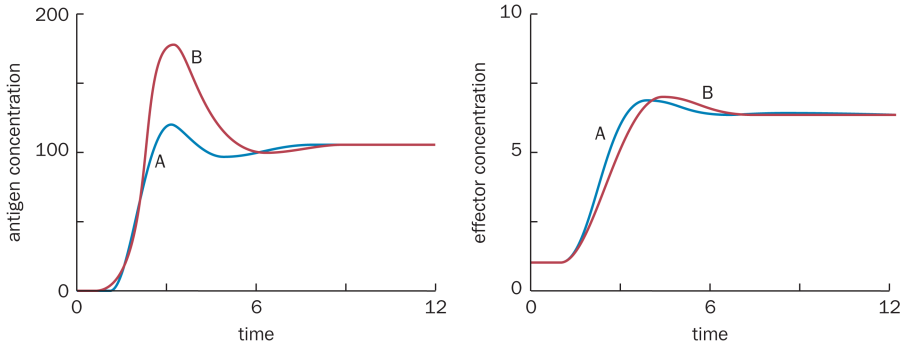


Figure 6.28 Comparison of time courses of two designs of immune cascades. Dynamic responses of the observed system design (Model A) and a hypothetical, alternative, design (Model B), which lacks inhibition of effector lymphocyte maturation, to a fourfold source antigen challenge at time $t = 1$.

1, respectively. In order to achieve maximal external equivalence, it turns out to be sufficient to set g_{31} in Model B to 0.4. Thus, for Model A, the production term for X_3 is $X_1^{0.5}X_2X_5^{-0.5}$, while it is $X_1^{0.4}X_2$ in Model B [68]. With these settings, both systems have not only the same steady state but also the same expenditure in material and energy, which are reflected in the same antigen and effector gains. We now have two systems that to the uninitiated observer appear to be essentially identical but differ internally in their suppression mechanism, with which suppressor lymphocytes inhibit the maturation of effector lymphocytes. What is the relevance of this observed mechanism? MMCC answers this question by formulating objective criteria for desirable system responses. In general, such "criteria for functional effectiveness" depend on the nature of the phenomenon and may include the response time with which the system reacts to a perturbation; characteristic measures of stability, sensitivity, and robustness; the accumulation of intermediate components; or the extent to which the transients between stimulus and the return to normalcy are affected by external perturbations. These criteria provide a set of objective metrics against which the performance of the alternative models is measured.

In the given case of the immune response cascade, the criteria for functional effectiveness were defined as follows [68, 69]: the base levels of systemic antigen and effector lymphocytes should be as low as feasible; the antigen gain and the effector gain, which are the increases in the steady-state levels of systemic antigen and effector lymphocytes following a source antigen challenge, should be minimal; the peak levels of systemic antigen and effector lymphocytes during the response should be as low as possible; and the entire cascade should be relatively insensitive to fluctuations in the environment.

The base levels and gains in Models A and B are identical, because they were equated to maximize external equivalence, so they cannot be used to investigate the superiority of either design. By contrast, the dynamic responses were not constrained and are easy to evaluate with simulations. As an example, one may initiate the two systems at their common steady state ($X_{1S} = 1.033724$, $X_{3S} = 1.013356$, $X_{5S} = 1.0066656$) and raise the source antigen X_0 from 1 to 4 at time $t = 1$. Indeed, the two models respond to the "infection" with different transients in antigens and effector lymphocytes (**Figure 6.28**). The model with suppression (Model A) is deemed superior, because the antigen and effector concentrations show smaller overshoots and the effector concentration responds more quickly. For other source antigen levels, the numerical features are different, but the superiority of the observed design (Model A) persists. Thus, lower peak levels are inherent functional characteristics of suppression. Algebraic analysis furthermore indicates that Model A is less sensitive to potentially harmful fluctuations in the environment [72]. Overall, the observed design has preferable features over the a priori also reasonable design without suppression.

Since its inception, MMCC has been applied to a variety of systems, primarily in the realm of metabolism and signaling, and variations have been proposed to study entire design classes at once (for example, [71, 72]), characterize design spaces [73, 74], or establish superiority of a design in a statistical sense [75]. A particularly intriguing example is demand theory [76–78], which rationalizes and predicts the mode of gene regulation in bacteria based on the microbes'

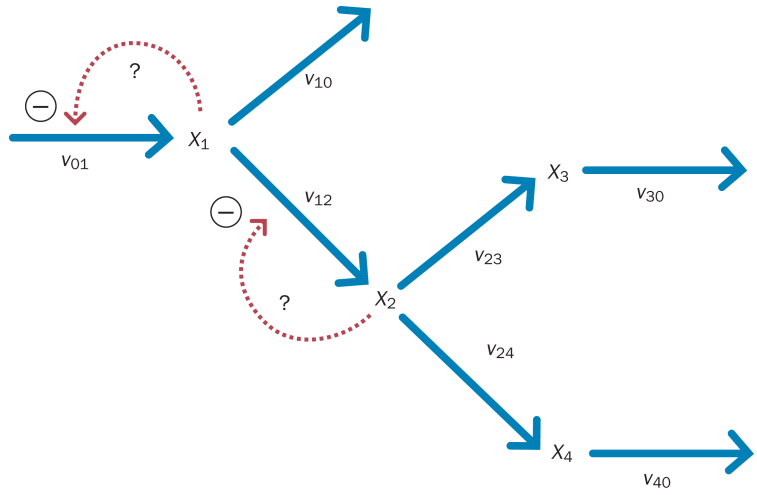


Figure 6.29 Optimal operating strategies depend on both structural and regulatory pathway features. Depending on the existence of no, one, or two feedback inhibition signals (indicated by question marks), the optimal strategy for increasing the flux v_{40} by manipulation of the flux ratios v_{10}/v_{12} and v_{23}/v_{24} is different.

natural environments and thus reveals strong natural design principles. Another example is *E. coli*'s envZ–OmpR two-component sensing system in bacteria, which we will discuss in Chapter 10. A careful analysis of this system [79, 80] led to a deeper understanding of its natural design and provided clues for how to manipulate a system with methods of synthetic biology. For instance, Skerker and collaborators [81] explored the large amount of available sequence data available for this two-component sensing system and identified individual amino acid residues that always varied together as a pair. They speculated that these pairs were critical for sensor specificity. Subsequently, they substituted the specificity-determining residues of one sensor with those of a different sensor, which indeed allowed the sensor's response to be triggered by the "wrong" signal.

6.6.2 Operating Principles

As a variation on the theme of design principles, one may also ask questions regarding **operating principles** for systems with given designs. An example is the controlled comparative analysis of different regulation patterns in a branched pathway (Figure 6.29), where the goal might be to increase the yield v_{40} by altering the ratios of fluxes flowing into different branches, v_{10}/v_{12} and/or v_{23}/v_{24} [82]. It turns out that a variety of manipulations exhibit equal yields if the pathway is not regulated, but that some manipulations are significantly more effective than others if some of the intermediates exert feedback inhibition.

The discovery and analysis of design and operating principles is an ongoing effort. A different aspect than we had discussed so far is the search for rules related to spatial phenomena. For instance, if a protein is to be localized in the cell nucleus, it often contains a high proportion of hydrophobic surface regions and a charge that is complementary to that of the nuclear pore complexes, which control transport between the nucleus and cytoplasm [83]. It may also be that some intracellular gradients and heterogeneous concentration patterns are associated with the spatial design issue. A special issue of *Mathematical Biosciences* [84] contains a collection of papers representing the state of the art.

6.7 EXERCISES

Naturally, many of the exercises in this chapter require computational efforts and access to software, which is available, for instance, in Microsoft Excel®, MATLAB® and Mathematica®, as well as in languages like Python and R. Specific techniques needed in some of the exercises include solutions to ODEs, linear and nonlinear regression, genetic algorithms, and splines. The data tables are available on the website supporting the book (www.garlandscience.com/product/isbn/9780815344674).

6.1 Perform linear regression with the bacterial growth data in Table 6.10. Plot the results. Repeat the analysis after a logarithmic transformation of the data. Compare and interpret the results.

TABLE 6.10: BACTERIAL GROWTH DATA IN SOME UNIT

t	Size	t	Size
0	2.2	1.1	22
0.1	2.9	1.2	21
0.2	3.2	1.3	30
0.3	6	1.4	35
0.4	6	1.5	38
0.5	6.2	1.6	47
0.6	8	1.7	58
0.7	11	1.8	70
0.8	14	1.9	81
0.9	13	2	90
1	18		

6.2 Perform linear regression with the bacterial growth data in Table 6.11. Plot the results. Repeat the analysis after a logarithmic transformation of the data. Compare and interpret the results.

TABLE 6.11: BACTERIAL GROWTH DATA IN SOME UNIT

t	Size	t	Size
0	3	1.1	18
0.1	3	1.2	26
0.2	2.8	1.3	31
0.3	3.7	1.4	36
0.4	3.9	1.5	44
0.5	4.4	1.6	50
0.6	5.6	1.7	64
0.7	7.4	1.8	72
0.8	11	1.9	88
0.9	14	2	96
1	16		

6.3 Confirm that the Michaelis–Menten function becomes linear if one plots $1/v$ against $1/S$. Is the same true for a Michaelis–Menten process with competitive inhibition, which is given by the function

$$v_I = \frac{V_{\max} S}{K_M\left(1+\frac{I}{K_I}\right)+S},$$

where I is the inhibitor concentration and K_I is the inhibition constant? Support your answer with mathematical arguments.

6.4 Use linear regression to confirm the results ($V_{\max} = 2.2$, $K_M = 6.4$) in the text (see (6.4) and Table 6.2).

6.5 Apply a nonlinear regression algorithm directly to the data in Tables 6.2 and 6.3. Compare the results with the linear estimation upon transformation.

6.6 Apply a nonlinear regression algorithm to the differential equation (6.4), using the data given in Table 6.12 and plotted in Figure 6.6. Furthermore, assume $S = 1.2$. Look at the code on the book website for inspiration.

TABLE 6.12: SYNTHETIC DATA DESCRIBING THE TREND IN A METABOLITE (SEE EQN. 6.4)

t	M
0	2.000
1	1.061
2	0.671
3	0.509
4	0.442
5	0.414
6	0.403
7	0.398
8	0.396
9	0.395

6.7 Lineweaver and Burk [85] suggested plotting $1/v$ against $1/S$, thereby reducing the nonlinear estimation problem for Michaelis–Menten rate laws to one of simple linear regressions. Woolf [86] had another idea. If one plots v/S against v, the Michaelis–Menten function also becomes linear (Confirm this!). Even though Woolf came up with this trick, the resulting plot is usually called a Scatchard plot [87]. While the nonlinear method, the Lineweaver–Burk method and the Woolf method, *should* all give the same results, they only do so if the data are error-free. As soon as there is noise in the data, equivalence is not guaranteed; in fact, the results can be quite different. To analyze this situation, use the data in Table 6.13. For all four cases, perform nonlinear regression and the two types of linear regression mentioned earlier. Summarize your findings, along with graphs and regression results, in a brief report.

TABLE 6.13: ERROR-FREE AND NOISY DATA QUANTIFYING THE ENZYMATIC CONVERSION OF A SUBSTRATE (S) INTO A PRODUCT (P)

S	v_0 (No Error)	v_1 (Errors Distributed)	v_2 (Errors at Low End)	v_3 (Errors at High End)
0.2	0.5454552	0.5	1	0.5
0.4	1	1.2	1.2	1.2
0.6	1.384616	1.25	1.5	1.25
0.8	1.714287	1.8	1.35	1.8
1	2	2.2	2.2	2.2
2	3	2.75	2.75	2.75
3	3.6	3.55	3.55	3
4	4	4.2	4.2	3.8
5	4.285715	4.2	4.2	5
10	5	4.85	4.85	5.6
20	5.454546	5.6	5.6	4.9

6.8 Imagine an experiment where a process V is governed by a substrate S and an inhibitor I. In the system, S decreases and I increases over time. Use the data in Table 6.14 and multiple linear regression to compute the parameters for process V, which has the form $V = \gamma S^a I^b$. Display the results in a pseudo-three-dimensional plot. Explain why it seems that the results lie on a slightly nonlinear curve rather than forming a (linear) regression plane.

TABLE 6.14: DATA OF AN ENZYMATIC REACTION WHICH IS MODULATED BY INHIBITOR *I*

t	*S*	*I*	*V*
0	10	1.2	6.3
1	7.5	1.7	4.5
2	5.5	2.3	3.2
3	4	2.9	2.4
4	2.5	3.5	1.8
5	1.5	4.0	1.4
6	1.0	4.5	1.1
7	0.7	4.5	0.8
8	0.4	4.8	0.6
9	0.2	4.8	0.5
10	0.1	4.9	0.3

6.9 (A) The Michaelis–Menten rate law (MMRL) with competitive inhibition has the formula shown in Exercise 6.3. In addition to the parameters of the MMRL, it contains the inhibition parameter K_I and the concentration of inhibitor, which we assume can be controlled in the experiment. Suppose two datasets (Table 6.15) were measured with different inhibitor concentrations I. Using grid searches, determine the parameters from the two datasets separately and from both together. Discuss your findings.

TABLE 6.15: DATA CAPTURING A MICHAELIS–MENTEN RATE LAW WITH COMPETITIVE INHIBITION

S	*v (I = 10)*	*v (I = 20)*	*S*	*v (I = 10)*	*v (I = 20)*
1	1.8	2	11	11.7	10.7
2	2.2	2.2	12	11.5	10.1
3	6.1	3.2	13	11.2	10.2
4	5.5	4.8	14	13	10.2
5	7.1	6.6	15	12.2	12.3
6	9.3	7.9	16	12.4	12.7
7	9	7.5	17	12.6	11.4
8	9.4	8	18	14.8	12.6
9	11.2	8.2	19	14.2	12.3
10	10.6	9.4	20	16.5	15.5

(B) Use a gradient method to perform the parameter estimation. Start with different guestimates. Run the estimation for different numbers of iterations.

6.10 Visualize the error landscape for the nonlinear regression with data v_1 in Exercise 6.7.

6.11 Estimate the 10 parameters of the function

$$F(t) = \sin(at+b)\cdot\left[ct + d(t-e)^2 + f(t-g)^3\right]\cdot e^{h+i(t-j)},$$

which is plotted in Figure 6.30 from the noise-free data in Table 6.16. Compare the requirements, procedures, and results for different methods.

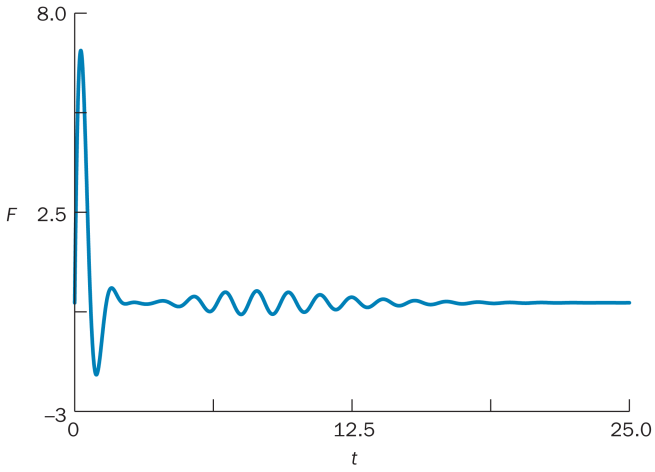


Figure 6.30 The function F(t) with 10 parameters. This strange function is used for estimation purposes.

6.12 Use a genetic algorithm to estimate parameters for the MMRL with inhibition from the data in Table 6.15 (see Exercise 6.3). Look at the website code for inspiration on how to set up the genetic algorithm. Run the algorithm with different bounds on parameter values.

6.13 Estimate the parameters of an MMRL with inhibition from the data in Table 6.17, assuming that $I = 20$. Redo the analysis with the bootstrapping and jackknifing methods. Compare and interpret results. Also, average the parameter values obtained from (A) different bootstrapping runs and (B) different jackknifing runs and assess the fits.

6.14 (A) Use a nonlinear regression algorithm to estimate the parameters of the function f in (6.9) from the data in Table 6.4. Redo the estimation several times with different start guesses. Study the quality of solutions with averaged parameter values.

(B) Transform the function f in (6.9) so that it becomes linear. Try to use linear regression to estimate the parameter values. Report your findings.

(C) Generate more equivalent representations of the function f in (6.9), similar to that in (6.10), using the relationship $Y = 1.75X^{0.8}$. Formulate a functional description that encompasses all possible representations of this type.

(D) Create a different situation of an estimation problem with multiple solutions.

TABLE 6.16: DATA QUANTIFYING FUNCTION $F(t)$

t	F	t	F	t	F	t	F	t	F
0.0	0.000	5.000	0.001	10.000	−0.005	15.000	0.002	20.000	0.000
0.1	4.184	5.100	0.064	10.100	−0.119	15.100	0.031	20.100	−0.004
0.2	6.471	5.200	0.122	10.200	−0.207	15.200	0.052	20.200	−0.007
0.3	6.929	5.300	0.162	10.300	−0.253	15.300	0.062	20.300	−0.009
0.4	5.964	5.400	0.174	10.400	−0.250	15.400	0.060	20.400	−0.008
0.5	4.152	5.500	0.150	10.500	−0.200	15.500	0.047	20.500	−0.006
0.6	2.080	5.600	0.093	10.600	−0.114	15.600	0.026	20.600	−0.004
0.7	0.223	5.700	0.011	10.700	−0.011	15.700	0.002	20.700	0.000
0.8	−1.119	5.800	−0.082	10.800	0.091	15.800	−0.021	20.800	0.003
0.9	−1.836	5.900	−0.166	10.900	0.171	15.900	−0.038	20.900	0.005
1.0	−1.976	6.000	−0.224	11.000	0.216	16.000	−0.047	21.000	0.006
1.1	−1.690	6.100	−0.242	11.100	0.219	16.100	−0.047	21.100	0.006
1.2	−1.170	6.200	−0.211	11.200	0.180	16.200	−0.038	21.200	0.005
1.3	−0.599	6.300	−0.137	11.300	0.110	16.300	−0.022	21.300	0.003
1.4	−0.110	6.400	−0.031	11.400	0.022	16.400	−0.004	21.400	0.000
1.5	0.222	6.500	0.088	11.500	−0.066	16.500	0.014	21.500	−0.002
1.6	0.383	6.600	0.195	11.600	−0.137	16.600	0.028	21.600	−0.004
1.7	0.401	6.700	0.269	11.700	−0.180	16.700	0.036	21.700	−0.005
1.8	0.326	6.800	0.293	11.800	−0.186	16.800	0.036	21.800	−0.005
1.9	0.213	6.900	0.260	11.900	−0.158	16.900	0.030	21.900	−0.004
2.0	0.103	7.000	0.176	12.000	−0.101	17.000	0.019	22.000	−0.002
2.1	0.022	7.100	0.054	12.100	−0.029	17.100	0.005	22.100	−0.001
2.2	−0.020	7.200	−0.082	12.200	0.045	17.200	−0.009	22.200	0.001
2.3	−0.030	7.300	−0.204	12.300	0.107	17.300	−0.020	22.300	0.002
2.4	−0.020	7.400	−0.290	12.400	0.146	17.400	−0.027	22.400	0.003
2.5	−0.004	7.500	−0.321	12.500	0.155	17.500	−0.028	22.500	0.003
2.6	0.009	7.600	−0.291	12.600	0.135	17.600	−0.024	22.600	0.003
2.7	0.012	7.700	−0.204	12.700	0.091	17.700	−0.016	22.700	0.002
2.8	0.006	7.800	−0.077	12.800	0.032	17.800	−0.005	22.800	0.001
2.9	−0.006	7.900	0.066	12.900	−0.029	17.900	0.005	22.900	−0.001
3.0	−0.020	8.000	0.196	13.000	−0.081	18.000	0.014	23.000	−0.002
3.1	−0.031	8.100	0.289	13.100	−0.115	18.100	0.020	23.100	−0.002
3.2	−0.036	8.200	0.327	13.200	−0.126	18.200	0.021	23.200	−0.002
3.3	−0.035	8.300	0.302	13.300	−0.113	18.300	0.019	23.300	−0.002
3.4	−0.026	8.400	0.219	13.400	−0.079	18.400	0.013	23.400	−0.001
3.5	−0.012	8.500	0.095	13.500	−0.033	18.500	0.005	23.500	−0.001
3.6	0.006	8.600	−0.045	13.600	0.017	18.600	−0.003	23.600	0.000
3.7	0.024	8.700	−0.176	13.700	0.060	18.700	−0.010	23.700	0.001
3.8	0.041	8.800	−0.271	13.800	0.090	18.800	−0.014	23.800	0.002
3.9	0.052	8.900	−0.314	13.900	0.101	18.900	−0.016	23.900	0.002
4.0	0.054	9.000	−0.296	14.000	0.093	19.000	−0.014	24.000	0.002
4.1	0.045	9.100	−0.222	14.100	0.068	19.100	−0.010	24.100	0.001
4.2	0.025	9.200	−0.108	14.200	0.032	19.200	−0.005	24.200	0.001
4.3	−0.005	9.300	0.024	14.300	−0.008	19.300	0.001	24.300	0.000
4.4	−0.040	9.400	0.149	14.400	−0.043	19.400	0.007	24.400	−0.001
4.5	−0.073	9.500	0.242	14.500	−0.069	19.500	0.010	24.500	−0.001
4.6	−0.097	9.600	0.287	14.600	−0.080	19.600	0.012	24.600	−0.001
4.7	−0.104	9.700	0.277	14.700	−0.075	19.700	0.011	24.700	−0.001
4.8	−0.089	9.800	0.215	14.800	−0.057	19.800	0.008	24.800	−0.001
4.9	−0.053	9.900	0.114	14.900	−0.029	19.900	0.004	24.900	0.000
								25.000	0.000

TABLE 6.17: DATA QUANTIFYING A MICHAELIS-MENTEN PROCESS WITH INHIBITION

S	*v(I = 20)*
1	2
4	4.8
8	6
10	9.4
12	12
14	10.2
16	13.3
18	12.6
20	12.2

6.15 Construct a function that has at least two minima with the same value (recall the situation in Figure 6.15). Is it possible to have infinitely many, unconnected minima with the same error? If you think yes, provide an example. If not, provide a proof, counterexample, or at least compelling arguments.

6.16 Use a spline function in MATLAB® to connect the data on population sizes N_i in **Table 6.7**, use this function to compute slopes at all time points, and estimate parameter values.

6.17 Estimate the parameter values in (6.16) upon dividing by N_1, which makes the problem linear. Compare the results with those in the text. Discuss the advantages and potential problems of each method.

6.18 (A) Use the data with and without noise (see Tables 6.7 and 6.8) and estimate parameter values with and without slope substitution, starting with different sets of initial guesses for the parameter values.

(B) Estimate parameter values with and without slope substitution, using subsets of the data with and without noise. Use different subsets with 10 and 15 time points. Compare results. Test whether averaging of the parameter values from different runs improves the fits.

6.19 Construct the diagram of a small pathway system that would be described by the following equations:

$$\dot{X}_1 = \alpha_1 X_3^{g_{13}} - \beta_1 X_1^{h_{11}},$$
$$\dot{X}_2 = \alpha_2 X_1^{g_{21}} - \beta_2 X_2^{h_{22}},$$
$$\dot{X}_3 = \alpha_3 X_2^{g_{32}} - \beta_3 X_3^{h_{33}} X_4^{h_{34}},$$
$$\dot{X}_4 = \alpha_4 X_1^{g_{41}} - \beta_4 X_4^{h_{44}}.$$

Is the diagram unique? Estimate parameter values for the system, using the datasets in **Table 6.18**, either one at a time or combined.

TABLE 6.18: DATA QUANTIFYING A SMALL PATHWAY SYSTEM

t	X_1	X_2	X_3	X_4
DATASET 1				
0	1.400	2.700	1.200	0.400
0.2	1.039	3.122	1.591	0.317
0.4	0.698	3.176	1.980	0.250
0.6	0.482	2.992	2.298	0.195
0.8	0.372	2.711	2.519	0.157
1	0.322	2.431	2.643	0.135
1.2	0.302	2.198	2.682	0.125
1.4	0.300	2.025	2.657	0.120
1.6	0.307	1.911	2.591	0.120
1.8	0.322	1.846	2.506	0.123
2	0.340	1.822	2.416	0.126
2.2	0.359	1.826	2.335	0.130
2.4	0.377	1.851	2.269	0.135
2.6	0.393	1.886	2.222	0.139
2.8	0.404	1.924	2.192	0.142
3	0.412	1.959	2.177	0.144
3.2	0.415	1.988	2.174	0.145
3.4	0.415	2.010	2.180	0.146
3.6	0.413	2.024	2.189	0.146
3.8	0.410	2.030	2.201	0.145
4	0.406	2.031	2.212	0.145
DATASET 2				
0	0.200	0.300	2.200	0.010
0.2	0.439	0.833	1.789	0.107
0.4	0.597	1.347	1.569	0.156
0.6	0.685	1.791	1.540	0.185
0.8	0.686	2.125	1.634	0.197
1	0.627	2.327	1.787	0.195
1.2	0.548	2.406	1.954	0.184
1.4	0.479	2.392	2.105	0.170
1.6	0.428	2.324	2.223	0.157
1.8	0.395	2.235	2.304	0.147
2	0.376	2.146	2.349	0.141
2.2	0.367	2.071	2.363	0.137
2.4	0.365	2.013	2.356	0.135
2.6	0.368	1.975	2.336	0.135
2.8	0.373	1.954	2.309	0.136
3	0.380	1.946	2.281	0.137
3.2	0.387	1.949	2.256	0.139
3.4	0.394	1.958	2.237	0.140
3.6	0.399	1.970	2.223	0.142
3.8	0.402	1.982	2.214	0.143
4	0.404	1.993	2.211	0.143

TABLE 6.18: (CONTINUED)

t	X_1	X_2	X_3	X_4
DATASET 3				
0	4.000	1.000	3.000	4.000
0.2	1.864	2.667	2.076	1.909
0.4	0.952	3.151	1.952	0.872
0.6	0.598	3.111	2.094	0.413
0.8	0.447	2.886	2.296	0.232
1	0.371	2.622	2.468	0.165
1.2	0.332	2.376	2.572	0.139
1.4	0.316	2.175	2.607	0.128
1.6	0.313	2.026	2.588	0.124
1.8	0.320	1.926	2.533	0.124
2	0.332	1.870	2.460	0.125
2.2	0.348	1.849	2.385	0.129
2.4	0.365	1.853	2.317	0.132
2.6	0.381	1.874	2.261	0.136
2.8	0.394	1.904	2.221	0.139
3	0.404	1.937	2.196	0.142
3.2	0.410	1.967	2.184	0.144
3.4	0.413	1.992	2.182	0.145
3.6	0.412	2.010	2.187	0.146
3.8	0.411	2.021	2.195	0.145
4	0.408	2.027	2.205	0.145

6.20 Assume again the model structure of Exercise 6.19, but with new parameter values:

$$\dot{X}_1 = \alpha_1 X_3^{g_{13}} - \beta_1 X_1^{h_{11}},$$
$$\dot{X}_2 = \alpha_2 X_1^{g_{21}} - \beta_2 X_2^{h_{22}},$$
$$\dot{X}_3 = \alpha_3 X_2^{g_{32}} - \beta_3 X_3^{h_{33}} X_4^{h_{34}},$$
$$\dot{X}_4 = \alpha_4 X_1^{g_{41}} - \beta_4 X_4^{h_{44}}.$$

Use a gradient-based method, such as `lsqcurvefit` in MATLAB®, to estimate parameter values for the system, using the noisy datasets in Table 6.19, either one at a time or combined. Use as bounds for the parameters $0 \le \alpha_i, \beta_i \le 20$; $-2 \le g_{ij}, h_{ij} \le 2$.

TABLE 6.19: DATA QUANTIFYING A SMALL PATHWAY SYSTEM

t	X_1	X_2	X_3	X_4
DATASET 1				
0	0.109	3.145	1.304	0.817
0.2	0.468	2.336	2.290	0.539
0.4	0.430	1.998	2.778	0.445
0.6	0.354	1.499	3.216	0.329
0.8	0.340	1.131	3.162	0.259
1	0.352	0.871	3.455	0.244
1.2	0.326	0.673	3.223	0.222
1.4	0.353	0.590	3.061	0.233
1.6	0.362	0.513	2.733	0.252
1.8	0.406	0.481	2.620	0.285
2	0.451	0.455	2.473	0.311
2.2	0.460	0.473	2.262	0.336
2.4	0.508	0.508	1.977	0.383
2.6	0.525	0.552	1.951	0.466
2.8	0.559	0.595	1.800	0.492
3	0.576	0.676	1.760	0.547
3.2	0.607	0.707	1.744	0.557
3.4	0.613	0.788	1.674	0.587
3.6	0.616	0.782	1.627	0.646
3.8	0.609	0.794	1.624	0.630
4	0.611	0.848	1.736	0.618
DATASET 2				
0	2.017	3.046	1.143	0.115
0.2	0.711	2.962	2.320	0.966
0.4	0.443	2.652	3.225	0.754
0.6	0.381	2.052	3.249	0.485
0.8	0.385	1.494	3.636	0.312
1	0.338	1.206	3.769	0.275
1.2	0.352	0.835	3.608	0.246
1.4	0.342	0.630	3.448	0.234
1.6	0.353	0.568	3.332	0.245
1.8	0.400	0.541	3.276	0.270
2	0.409	0.472	2.524	0.295
2.2	0.469	0.514	2.580	0.342
2.4	0.517	0.485	2.414	0.329
2.6	0.530	0.544	2.081	0.424
2.8	0.614	0.560	1.848	0.451
3	0.636	0.606	1.927	0.506
3.2	0.644	0.685	1.899	0.585
3.4	0.687	0.724	1.668	0.575
3.6	0.615	0.772	1.748	0.664
3.8	0.650	0.821	1.871	0.626
4	0.602	0.753	1.500	0.619

6.21 For the two-population example in the text (see Table 6.9; and Figure 6.22), what is the absolute deviation of the optimized solution from the true solution at each time point if SSE ≈ 7 or SSE ≈ 10,000?

6.22 Find out from the literature what "sloppiness" means in the context of parameter estimation. Write a brief report.

6.23 Find out details regarding Fisher's information matrix. Write a brief report.

6.24 Write a brief report on overfitting and cross-validation.

6.25 Discuss the advantages and drawbacks of canonical models for structure identification.

6.26 Compare two different functions that could be appropriate for modeling an s-shaped trend in some data. The first is a logistic function and the second a Hill function. To simulate the modeling of actual data, generate artificial data with the logistic function

$$L = \frac{6}{1+\exp(5-0.8S)}$$

for $S = 1, 2, \ldots, 20$ and fit these "data" with a Hill function of the type

$$H = \frac{aS^n}{b+S^n}.$$

How similar are the two functions? What are optimal parameter values for H? What is the residual error?

6.27 Consider the data in Figure 6.24 and suppose that we know that these data saturate at a value of 2.3 for larger values of x. Extrapolate the alternative functions shown for x up to 10. Which functions are clearly wrong and which ones might be suitable, possible with adjusted scaling?

6.28 Consider the following system, which is implemented as the power-law model in **Figure 6.31**. The system has either no feedback or one or both feedback signals shown (see question marks). Thus, there are four scenarios, corresponding to combinations of 0 and –1 for g_{11} and g_{22}.

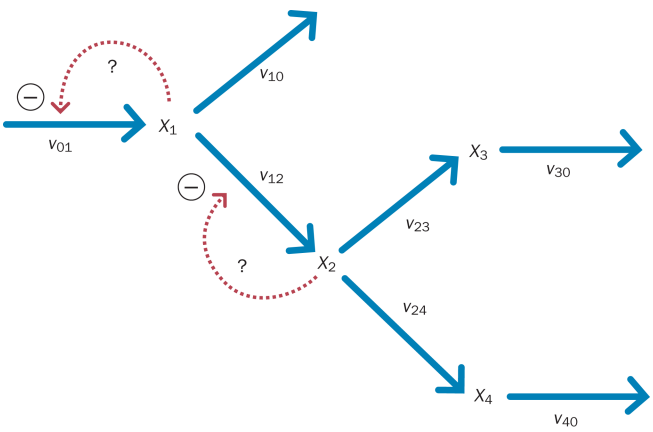


Figure 6.31 Pathway with two branchpoints and uncertain regulation. The goal of this metabolic engineering exercise is to maximize v_{40} under different constraints. Corresponding equations are shown in GMA format (see Chapter 5).

$$\frac{dX_1}{dt} = 8X_1^{g_{11}} - 4X_1^{0.5}X_2^{g_{22}} - 4X_1^{0.5} \qquad X_1(0)=1$$
$$\frac{dX_2}{dt} = 4X_1^{0.5}X_2^{g_{22}} - 2X_2^{0.5} - 2X_2^{0.5} \qquad X_2(0)=1$$
$$\frac{dX_3}{dt} = 2X_2^{0.5} - 2X_3^{0.5} \qquad X_3(0)=1$$
$$\frac{dX_4}{dt} = 2X_2^{0.5} - 2X_4^{0.5} \qquad X_4(0)=1$$

The overall goal of the analysis is to maximize the efflux v_{40}; identify this flux in the equations. To attain this goal, test the effect of different flux split ratios (v_{10}/v_{12} and/or v_{23}/v_{24}) on this efflux v_{40}. Specifically, the permitted manipulation strategies are:

S1: Increase v_{12} four-fold
S2: Decrease v_{10} by a factor of four
S3: Double v_{12} and halve v_{10}
S4: Increase v_{24} four-fold
S5: Decrease v_{23} by a factor of four
S6: Double v_{24} and halve v_{23}
S7: Double v_{12} and v_{24}
S8: Double v_{24} and halve v_{10}
S9: Increase v_{12} and v_{24} and decrease v_{10} and v_{23} by the same factor, with a total effort of four units

Determine which of these strategies generates the highest efflux v_{40} for each scenario:

A: [g11, g22] = [0, 0],
B: [g11, g22] = [–1, 0],
C: [g11, g22] = [0, –1],
D: [g11, g22] = [–1, –1].

Make predictions regarding v_{40} before you simulate.

6.29 For many years, researchers invoked the Central Dogma of biology to justify approximation of abundance levels for a given protein by linear regression with respect to mRNA levels of the corresponding gene. Now that it is possible to quantify individual mRNA strands and protein numbers within a cell, this assumption is being questioned. **Table 6.20** provides mRNA and protein levels measured for 15 genes. Perform linear regression. How well does it do? Determine a transformation to the data that might improve the fit. Discuss the pros and cons of the non-transformed linear regression versus the transformed linear regression.

6.30 The numbers in the previous question are derived from a *Nature* paper [88], in which the authors propose that consideration of synthesis and degradation rates of the mRNA and protein levels for each gene provide a better fit than linear regression. The following figure summarizes the methodology for how the authors model protein and mRNA levels for each gene.

TABLE 6.20: EXPERIMENTALLY DETERMINED QUANTITIES OF PROTEINS AND MRNAS IN A CELL (FROM [88])

Gene Name	*Avg. Protein Number/Cell*	*Avg. mRNA Number/Cell*
Synj1	56.66	3.33
Atr	61.37	2.59
Vps41	75.6	13.73
Esf1	26673.24	13.97
Stambp	26684.63	5.35
Tm9sf3	26704.97	26.82
Creg1	64705.2	15.44
Ddrgk1	64713.32	32.06
Utp6	64769.74	21.83
Rpn2	924200.9	93.01
Cisd1	931091.7	55.32
Pafah1b2	933753.9	24.8
Ldha	13681448	388.36
Rpl18	14414149	408.9
Ywhae	14499283	248.6

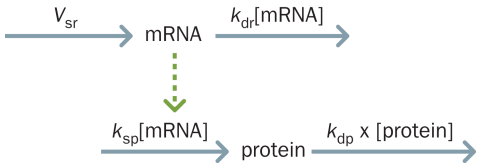


Figure 6.32 Refined model of abundance levels for a given protein with respect to mRNA levels of the corresponding gene. Adapted from [88].

Consider now the possibility that one gene is a transcription factor that regulates the levels of two other gene products. Draw a new diagram derived from Figure 1 that reflects this regulation. At this stage, gene assignment to the network is not necessary and you can keep the network structure generic.

6.31 Continuing the project in Exercises 6.29 and 6.30, some of the synthesis and degradation rates in Table 6.21 are available; others are not. Develop an optimization algorithm using the data in Table 6.22 for determining the missing values and, at the same time, for establishing which gene codes for the transcription factor. *Hint: Start by performing parameter estimation for each network scenario.* Discuss the process that you used to arrive at your parameter estimates for k_{spB}, k_{drC}, and k_{dpA} to establish the proper network structure. Was one process more difficult than the other?

6.32 Table 6.23 contains data from an experiment measuring changes in cell density. Odd columns show time in seconds and even column the corresponding cell density (cells $\cdot 10^6$/mL). A denser table is available on the book's website.

TABLE 6.21: SYNTHESIS AND DEGRADATION RATES ASSOCIATED WITH FIGURE 6.32

	v_{sr}	k_{sp}	k_{dr}	k_{dp}
gene A	1.26	22.4	0.066	
gene B	1.6		0.134	0.211
gene C	5.14	5.66		1.32

TABLE 6.22: DATA FOR DETERMINING MISSING VALUES OF SYNTHESIS AND DEGRADATION RATES

t	$mRNA_A$	$mRNA_B$	$mRNA_C$	*protA*	*protB*	*protC*
0	32.38432	31.88873	33.97225	10352.92	94.53856	100000
10	3682.321	15.14412	14680.14	118560.3	470.148	69510.78
20	7023.812	14.29666	27413.35	198119.3	426.2364	84231.08
30	5500.116	11.48007	21154.91	199997.6	339.6438	95837.88
40	6709.583	12.12782	25246.32	213768.7	353.9472	112996.7
50	6594.395	11.95463	23957.94	201353.6	338.4314	102880.6
60	7314.887	12.57537	24408.44	204937.3	366.591	116923.3
70	6041.073	11.67738	23423.07	197460.4	324.5797	118386.6
80	6926.663	12.56884	24657.44	216272.2	337.3018	100050.4
90	6377.064	11.74027	22968.32	217181.6	331.3738	107838.1
100	6803.352	12.18178	24399.74	218759	345.557	107875.9

- Plot and visually inspect the data.
- Choose three functions that might represent the data well. These can be any functions, such as $\sin(ax) + \sin(bx)$, ae^{bt}, $\sqrt{ax+b}$. The only stipulation is that the function may have no more than three parameters.
- Optimize these parameters with software of your choice. Use the SSE as error function to evaluate the data fits.
- Compare the three functions you chose. Report the errors and plot them together with the data.

6.33 An experiment is conducted in which a constant input of a starting metabolite is added to a cocktail of enzymes and the metabolite concentration of the finished product, X_4, is measured over time. The proposed system is shown in Figure 6.33:

Use a MATLAB® script "Ch6_fit" below (also on the website) or some other software to determine the quality of the fit (SSE) to the corresponding set of mass-action equations. The data from the experiment are included at the end of the file.

A researcher in the lab notices that X_4 is somewhat oscillatory and speculates that one of the metabolites might be regulating the input flux to X_1. Test this suspicion by exploring different possibilities of feedback, using the kinetic order g for this purpose, as it is typical for S-systems.

Provide your best plot and best SSE, as well as the resulting parameters estimated from the optimal network.

TABLE 6.23: EXPERIMENTAL DATA OF CHANGES IN CELL DENSITY

Time	*Density*	*Time*	*Density*	*Time*	*Density*	*Time*	*Density*	*Time*	*Density*
0	5	10	14.34634	20	10.49632	30	6.703047	40	4.943381
0.5	5.7749	10.5	14.3063	20.5	10.25815	30.5	6.572762	40.5	4.892356
1	6.609179	11	14.23538	21	10.0235	31	6.44777	41	4.843894
1.5	7.459296	11.5	14.13673	21.5	9.792851	31.5	6.327941	41.5	4.797883
2	8.296409	12	14.01335	22	9.566635	32	6.21314	42	4.754213
2.5	9.100919	12.5	13.86806	22.5	9.345209	32.5	6.103227	42.5	4.71278
3	9.859539	13	13.70355	23	9.128874	33	5.99806	43	4.67348
3.5	10.5635	13.5	13.52231	23.5	8.917877	33.5	5.897496	43.5	4.636216
4	11.20735	14	13.32669	24	8.712419	34	5.801387	44	4.600892
4.5	11.7881	14.5	13.11885	24.5	8.512654	34.5	5.709588	44.5	4.567417
5	12.30465	15	12.90081	25	8.3187	35	5.621952	45	4.535704
5.5	12.75726	15.5	12.6744	25.5	8.130637	35.5	5.538333	45.5	4.505667
6	13.14728	16	12.44134	26	7.948513	36	5.458588	46	4.477227
6.5	13.47678	16.5	12.20318	26.5	7.772348	36.5	5.382572	46.5	4.450304
7	13.74843	17	11.96132	27	7.602137	37	5.310145	47	4.424825
7.5	13.96527	17.5	11.71706	27.5	7.437851	37.5	5.241168	47.5	4.400718
8	14.13059	18	11.47155	28	7.279442	38	5.175506	48	4.377914
8.5	14.24784	18.5	11.22584	28.5	7.126845	38.5	5.113024	48.5	4.356349
9	14.32054	19	10.98086	29	6.979978	39	5.053594	49	4.33596
9.5	14.35221	19.5	10.73744	29.5	6.838747	39.5	4.997087	49.5	4.316687
								50	4.298472

Figure 6.33 Linear pathway proposed for Exercise 6.33.

Report how the equation of X_1 was altered to achieve your results.

function Ch6_fit

```
% Generate data with true parameters, as-
  sume true parameters unknown
[t,x_obs] = observed_data;
figure(1);
plot(t,x_obs,'o');
% set up optimization parameters, use
  gradient based algorithm
optimization_options = optimoptions(@
  lsqnonlin,'Algorithm','levenberg-marqua
  rdt','MaxFunEvals',1500);
% initial guess of the optimization pro-
  cess
initial_guess = [1;1;1;1;1;1];
% run the optimization algorithm
est_para = lsqnonlin(@(para) myfun(para,
  x_obs, t), initial_guess, [], [], opti-
  mization_options); % x is estimated_pa-
  rameter
display(est_para);
% define objective vector for lsqnonlin
function objectve_vector=myfun(para, x_
  obs, t)
[t_pred,x_pred] = ode15s(@(t,x)
  odefunction(t,x,para), t, [2; 1; 1;
  1]);
if ~isequal(size(x_pred),size(x_obs))
if isempty(t_pred)
x_pred = zeros(size(x_obs));
elseif length(t_pred)==1
x_pred = repmat(x_pred, length(t),1);
else
[x_pred] = interp1(t_pred,x_
  pred,t,'linear','extrap');
end
end
figure(2);
sub=subplot(1,1,1);
plot(t,x_obs(:,1),'b^', t,x_
  pred(:,4),'b-');
set(sub,'YLim',[0 2.1]);
objectve_vector = x_pred(:,4)-x_obs(:);
disp(sum(objectve_vector.^2))
drawnow
function dx = odefunction(t,x,para)
a1 = para(1);
b1 = para(2);
b2 = para(3);
b3 = para(4);
b4 = para(5);
g = para(6);
dx(1,1) = a1-b1*x(1);
dx(2,1) = b1*x(1)-b2*x(2);
dx(3,1) = b2*x(2)-b3*x(3);
dx(4,1) = b3*x(3)-b4*x(4);
```

```
function [t,x_obs] = observed_data
  data = [0,  1.0617
        1, 1.0999
        2, 0.9607
        3, 1.1652
        4, 1.1546
        5, 1.1655
        6, 1.1288
        7, 1.1928
        8, 1.0878
        9, 0.9969
        10,         0.965
        11,         0.953
        12,         0.841
        13,         0.854
        14,         0.9624
        15,         1.0583
        16,         1.0967
        17,         1.0747
        18,         1.0416
        19,         0.9959
        20,         1.0497
        21,         1.1193
        22,         0.9938
        23,         0.9939
        24,         0.9565
        25,         0.8157
        26,         0.9855
        27,         1.0384
        28,         0.9635
        29,         1.0505
        30,         0.9966
        31,         1.0894
        32,         1.1775
        33,         1.1338
        34,         1.017
        35,         0.9761
        36,         1.0015
        37,         0.9614
        38,         0.9518
        39,         1.0108
        40,         0.9623
        41,         1.0489
        42,         0.985
        43,         1.0041
        44,         1.0709
        45,         1.0657
        46,         1.0372
        47,         0.9618
        48,         0.9711
        49,         1.0391
        50,         1.0164
];
  t = data(:,1);
  x_obs = data(:,2);
```

6.34 "Notch" is a 300 kDa transmembrane receptor protein that is increasingly being recognized as a key transducer of signals from adjacent cells that express non-soluble ligands on their surface. Notch is regulated by a number of post-translational modifications (PTMs), and it is thought that the combinations of PTMs create a "code" that relays context-specific instructions.

Consider three unique amino acids A, B, and C, that can undergo O-linked glycosylation (in reality there are dozens). As a consequence, Notch can be unglycosylated, glycosylated in position A, glycosylated in position B, glycosylated in position C, or glycosylated in positions A and B, B and C, A and C, or A, B, and C.

6.34.1 Sketch a diagram of the different states of the Notch protein and the transitions between them.

6.43.2 Design an ODE model describing the transitions and states. How do you ensure that the total amount of protein does not change over time?

FURTHER READING

Alon U. *An Introduction to Systems Biology: Design Principles of Biological Circuits*. Chapman & Hall, CRC Press, New York, NY, 2006.

Aster RC, Borchers B & Thurber CH. *Parameter Estimation and Inverse Problems*, 3rd ed. Elsevier Science, Amsterdam, the Netherlands, 2018.

Chou IC & Voit EO. Recent developments in parameter estimation and structure identification of biochemical and genomic systems. *Math Biosci.* 219 (2009) 57–83.

Eiben AE & Smith JE. *Introduction to Evolutionary Computing*. Springer, New York, NY, 2003.

Glantz SA & Slinker BK. *Primer of Applied Regression and Analysis of Variance*, 2nd ed. McGraw-Hill, New York, NY, 2001.

Kutner MH, Nachtsheim CJ, Neter J & Li W. *Applied Linear Statistical Models*, 5th ed. McGraw-Hill, Irwin, 2004.

Montgomery DC. *Design and Analysis of Experiments*, 7th ed. Wiley, Hoboken, NJ, 2008.

Raol JR, Girija G & Singh J. *Modelling and Parameter Estimation of Dynamic Systems*. Institution of Engineering and Technology (IET), Stevenage, UK, 2004.

Savageau MA. *Biochemical Systems Analysis: A Study of Function and Design in Molecular Biology*, pp. xvii, 379. Addison-Wesley Pub. Co. Advanced Book Program, Reading, MA, 1976.

van der Heijden F, Tax DMJ, De Ridder D & Duin RP. *Classification, Parameter Estimation and State Estimation. An Engineering Approach using* Matlab. John Wiley & Sons, Hoboken, NJ, 2004.

Voit EO. What if the fit is unfit? Criteria for biological systems estimation beyond residual errors. In *Applied Statistics for Biological Networks* (M Dehmer, F Emmert-Streib & A Salvador, eds), pp. 183–200. Wiley, Hoboken, NJ, 2011.

REFERENCES

1. Torres, M., et al., Identifying important parameters in the inflammatory process with a mathematical model of immune cell influx and macrophage polarization. *PLoS Comput Biol*, 2019. **15**(7): p. e1007172.
2. Chou, I.-C. and E.O. Voit, Recent developments in parameter estimation and structure identification of biochemical and genomic systems. *Mathem Biosc*, 2009. **219**(2): p. 57–83.
3. Gennemark, P. and D.D. Wedelin, Benchmarks for identification of ordinary differential equations from time series data. *Bioinformatics*, 2009. **25**(6): p. 780–786.
4. Sun, J., J.M. Garibaldi, and C. Hodgman, Parameter estimation using meta-heuristics in systems biology: A comprehensive review. *IEEE/ACM Trans Comput Biol Bioinform*, 2012. **9**(1): p. 185–202.
5. Davis, J.D., D.V. Olivença, and E.O. Voit, Methods of quantifying interactions among populations using Lotka-Volterra models. *Front Syst Biol*, 2022. **2**: p. 1021897.
6. Kutner, M.H., et al., *Applied Linear Statistical Models*. 5th ed. 2004, Irwin, CA: McGraw-Hill.
7. Montgomery, D.C., *Design and Analysis of Experiments*. 5th ed. 2000, New York, NY: Wiley.
8. Donahue, M.M., G.T. Buzzard, and A.E. Rundell, Parameter identification with adaptive sparse grid-based optimization for models of cellular processes, in *Methods in Bioengineering: Systems Analysis of Biological Networks*, A. Jayaraman and J. Hahn, Editors. 2009, Boston, London: Artech House, p. 211–232.
9. Falk, J.E. and R.M. Soland, An algorithm for separable nonconvex programming problems. *Manage Sci*, 1969. **15**(9): p. 550–569.
10. Land, A.H. and A.G. Doig, An automatic method of solving discrete programming problems. *Econometrica*, 1960. **28**(3): p. 497–520.
11. Glantz, S.A. and B.K. Slinker, *Primer of Applied Regression and Analysis of Variance*. 1990, New York, NY: McGraw-Hill.
12. Goldberg, D.E., *Genetic Algorithms in Search, Optimization and Machine Learning*. 1989, Boston, MA: Kluwer Academic Publishers.
13. Holland, J.H., *Adaptation in Natural and Artificial Systems*. 1975, Ann Arbor, MI: University of Michigan Press.
14. Fogel, D.B., *Evolutionary Computation: Toward a New Philosophy of Machine Intelligence*. 3rd ed. 2006, Hoboken, NJ: IEEE Press, Wiley-Interscience.
15. Dorigo, M. and T. Stützle, *Ant Colony Optimization*. 2004, Boston, MA: MIT Press, p. 319.
16. Catanzaro, D., R. Pesenti, and M.C. Milinkovitch, An ant colony optimization algorithm for phylogenetic estimation under the minimum evolution principle. *BMC Evol Biol*, 2007. **7**.
17. Kennedy, J. and R.C. Eberhart. Particle swarm optimization, in *Proceedings of the Fourth IEEE International Conference on Neural Networks*. 1995, Perth, Australia: IEEE.
18. Poli, R., Analysis of the publications on the applications of particle swarm optimisation. *J Artif Evol Appl*, 2008: p. 1–10, https://doi.org/10.1155/2008/685175.
19. Metropolis, N., et al., Equations of state calculations by fast computing machines. *J Chem Physics*, 1953. **21**(6): p. 1087–1092.
20. Salamon, P., P. Sibani, and R. Frost, *Facts, Conjectures, and Improvements for Simulated Annealing. Monographs on mathematical Modeling and Computation*. 2002, New York: SIAM.
21. Li, S., Y. Liu, and H. Yu, Parameter estimation approach in groundwater hydrology using hybrid ant colony system, in *Computational Intelligence and Bioinformatics (ICIC 2006, LNBI 4115)*, D.-S. Huang, K. Li, and G.W. Irwin, Editors. 2006, Berlin, Heidelberg: Springer-Verlag, p. 182–191.
22. Voit, E.O., What if the fit is unfit? Criteria for biological systems estimation beyond residual errors, in *Applied Statistics for Biological Networks*, M. Dehmer, F. Emmert-Streib, and A. Salvador, Editors. 2011, New York: John Wiley and Sons, p. 183–200.
23. Gutenkunst, R.N., et al., Universally sloppy parameter sensitivities in systems biology models. *PLoS Comput Biol*, 2007. **3**(10): p. 1871–1878.
24. von Bertalanffy, L., Principles and theory of growth, in *Fundamental Aspects of Normal and Malignant Growth*, W.W. Nowinski, Editor. 1960, New York. NY: Elsevier, p. 137–259.
25. Gutenkunst, R.N., et al., Extracting falsifiable predictions from sloppy models. *Ann NY Acad Sci*, 2007. **1115**: p. 203–211.
26. Raue, A., et al., Structural and practical identifiability analysis of partially observed dynamical models by exploiting the profile likelihood. *Bioinformatics*, 2009. **25**(15): p. 1923–1929.
27. Efron, B. and R.J. Tibshirani, *An Introduction to the Bootstrap*. 1993, New York, NY: Chapman & Hall.
28. Good, P., *Resampling Methods*. 3rd ed. 2006, Basel, Switzerland: Birkhäuser Verlag AG.
29. Manly, B.F.J., *Randomization, Bootstrap and Monte Carlo Methods in Biology: Texts in Statistical Science*. 3rd ed. 2007, New York, NY: Chapman & Hall, CRC Press.
30. Neves, A.R., et al., In vivo nuclear magnetic resonance studies of glycolytic kinetics in Lactococcus lactis. *Biotechnol Bioeng*, 1999. **64**(2): p. 200–212.
31. Voit, E.O. and J. Almeida, Decoupling dynamical systems for pathway identification from metabolic profiles. *Bioinformatics*, 2004. **20**(11): p. 1670–1681.
32. Gengjie, J., G.N. Stephanopoulos, and R. Gunawan, Parameter estimation of kinetic models from metabolic profiles: Two-phase dynamic decoupling method. *Bioinformatics*, 2011. **27**(14): p. 1964–1970.
33. Lillacci, G. and M. Khammash, Parameter estimation and model selection in computational biology. *PLoS Comput Biol*, 2010. **6**(3): p. e1000696.
34. Voit, E.O., *Computational Analysis of Biochemical Systems. A Practical Guide for Biochemists and Molecular Biologists*. 2000, Cambridge, UK: Cambridge University Press. p. xii + 530.
35. Verhulst, P.F., Notice sur la loi que la population suit dans son accroissement. *Corr Math Phys*, 1938. **10**: p. 113–121.
36. de Boor, C., *A Practical Guide to Splines. Applied Mathematical Sciences*. Vol. 27. 2001, New York, NY: Springer.
37. Vilela, M., et al., Automated smoother for the numerical decoupling of dynamics models. *BMC Bioinform*, 2007. **8**: p. 305.
38. Voit, E.O. and I.-C. Chou, Parameter estimation in canonical biological systems models. *Int J Syst Synth Biol*, 2010. **1**(1): p. 1–19.
39. Dam, P., et al., Dynamic models of the complex microbial metapopulation of Lake Mendota. *Nature PJ Sys Biol Appl*, 2016. **2**: p. 16007.
40. Voit, E.O., Modelling metabolic networks using power-laws and S-systems. *Essays Biochem*, 2008. **45**: p. 29–40.
41. Voit, E.O., The best models of metabolism. *WIREs Syst Biol Med*, 2017. **9**(6): p. e1391.
42. Schulz, A.R., *Enzyme Kinetics: From Diastase to Multi-enzyme Systems*. 1994, Cambridge, New York: Cambridge University Press.
43. Alvarez-Vasquez, F., et al., Integration of kinetic information on yeast sphingolipid metabolism in dynamical pathway models. *J Theor Biol*, 2004. **226**(3): p. 265–291.

44. Curto, R., A. Sorribas, and M. Cascante, Comparative characterization of the fermentation pathway of *Saccharomyces cerevisiae* using biochemical systems theory and metabolic control analysis: Model definition and nomenclature. *Math Biosci*, 1995. **130**(1): p. 25–50.
45. Curto, R., et al., Mathematical models of purine metabolism in man. *Math Biosci*, 1998. **151**(1): p. 1–49.
46. Goel, G., I.-C. Chou, and E.O. Voit, System estimation from metabolic time-series data. *Bioinformatics*, 2008. **24**(21): p. 2505–2511.
47. Faraji, M. and E.O. Voit, Stepwise inference of likely dynamic flux distributions from metabolic time series data. *Bioinformatics*, 2017. **33**(14): p. 2165–2172.
48. Battogtokh, D., et al., An ensemble method for identifying regulatory circuits with special reference to the QA gene cluster of *Neurospora crassa*. *Proc Natl Acad Sci USA*, 2002. **99**(26): p. 16904–16909.
49. Lee, Y. and E.O. Voit, Mathematical modeling of monolignol biosynthesis in *Populus*. *Math Biosc*, 2010. **228**(1): p. 78–89.
50. Tan, Y. and J.C. Liao, Metabolic ensemble modeling for strain engineers. *Biotechnol J*, 2012. **7**(3): p. 343–353.
51. Gutenkunst, R.N., et al., Universally sloppy parameter sensitivities in systems biology models. *PLoS Comput Biol*, 2007. **3**(10): p. 1871–1878.
52. Vilela, M., et al., Identification of neutral biochemical network models from time series data. *BMC Syst Biol*, 2009. **3**: p. 47.
53. Srinath, S. and R. Gunawan, Parameter identifiability of power-law biochemical system models. *J Biotechnol*, 2010. **149**(3): p. 132–140.
54. Almeida, J.S., Predictive non-linear modeling of complex data by artificial neural networks. *Curr Opin Biotechnol*, 2002. **13**(1): p. 72–76.
55. Bozdogan, H., Model selection and Akaike's information criterion (AIC): The general theory and its analytical extensions. *Psychometrika*, 1987. **52**: p. 345–370.
56. Neath, A.A. and J.E. Cavanaugh, The Bayesian information criterion: Background, derivation, and applications. *WIREs Comput Stat*, 2011. **4**(2): p. 199–203.
57. Savageau, M.A., Design principles for elementary gene circuits: Elements, methods, and examples. *Chaos*, 2001. **11**(1): p. 142–159.
58. Savageau, M.A., *Biochemical Systems Analysis: A Study of Function and Design in Molecular Biology*. 1976, Reading, MA: Addison-Wesley Pub. Co. Advanced Book Program, p. xvii, 379.
59. Savageau, M.A., A theory of alternative designs for biochemical control systems. *Biomed Biochim Acta*, 1985. **44**(6): p. 875–880.
60. Alon, U., *An Introduction to Systems Biology: Design Principles of Biological Circuits*. 2006, New York, NY: Chapman & Hall, CRC Press.
61. Alon, U., Network motifs: Theory and experimental approaches. *Nat Rev Genet*, 2007. **8**(6): p. 450–461.
62. Bromberg, K.D., et al., Design logic of a cannabinoid receptor signaling network that triggers neurite outgrowth. *Science*, 2008. **320**(5878): p. 903–909.
63. Lipshtat, A., et al., Design of versatile biochemical switches that respond to amplitude, duration, and spatial cues. *Proc Natl Acad Sci USA*, 2010. **107**(3): p. 1247–1252.
64. Ma, W., et al., Defining network topologies that can achieve biochemical adaptation. *Cell*, 2009. **138**(4): p. 760–773.
65. Poyatos, J.F., On the search for design principles in biological systems. *Adv Exp Med Biol*, 2012. **751**: p. 183–193.
66. Green, S., Revisiting generality in biology: Systems biology and the quest for design principles. *Biol Philos*, 2015. **30**(5): p. 629–652.
67. Andréen, D. and A. Goidea, Principles of biological design as a model for biodesign and biofabrication in architecture. *Architect Struct Construct*, 2022. **2**(4): p. 481–491.
68. Irvine, D.H. and M.A. Savageau, Network regulation of the immune response: Alternative control points for suppressor modulation of effector lymphocytes. *J Immunol*, 1985. **134**: p. 2100–2116.
69. Schwacke, J.H. and E.O. Voit, The potential for signal integration and processing in interacting MAP kinase cascades. *J Theor Biol*, 2007. **246**(4): p. 604–620.
70. Ma'ayan, A., et al., Ordered cyclic motifs contribute to dynamic stability in biological and engineered networks. *Proc Natl Acad Sci USA*, 2008. **105**(49): p. 19235–19249.
71. Alves, R. and M.A. Savageau, Effect of overall feedback inhibition in unbranched biosynthetic pathways. *Biophys J*, 2000. **79**(5): p. 2290–2304.
72. Irvine, D.H., The method of controlled mathematical comparison, in *Canonical Nonlinear Modeling*, E.O. Voit, Editor. 1991, New York: Van Nostrand Reinhold, p. 90–109.
73. Savageau, M.A., et al., Phenotypes and tolerances in the design space of biochemical systems. *Proc Natl Acad Sci USA*, 2009. **106**(16): p. 6435–6440.
74. Valderrama-Gómez, M.Á. and M.A. Savageau, Phenotype-centric modeling for rational metabolic engineering. *Metab Eng*, 2022. **72**: p. 365–375.
75. Schwacke, J.H. and E.O. Voit, Improved methods for the mathematically controlled comparison of biochemical systems. *Theor Biol Med Model*, 2004. **1**(1): p. 1.
76. Savageau, M.A., Design of molecular control mechanisms and the demand for gene expression. *Proc Natl Acad Sci USA*, 1977. **74**: p. 5647–5651.
77. Savageau, M.A., Demand theory of gene regulation. I. Quantitative development of the theory. *Genetics*, 1998. **149**(4): p. 1665–1676.
78. Savageau, M.A., Demand theory of gene regulation. II. Quantitative application to the lactose and maltose operons of *Escherichia coli*. *Genetics*, 1998. **149**(4): p. 1677–1691.
79. Alves, R. and M.A. Savageau, Comparative analysis of prototype two-component systems with either bifunctional or monofunctional sensors: Differences in molecular structure and physiological function. *Mol Microbiol*, 2003. **48**: p. 25–51.
80. Igoshin, O.A., R. Alves, and M.A. Savageau, Hysteretic and graded responses in bacterial two-component signal transduction. *Mol Microbiol*, 2008. **68**(5).
81. Skerker, J.M., et al., Rewiring the specificity of two-component signal transduction systems. *Cell*, 2008. **133**(6): p. 1043–1054.
82. Voit, E.O., Design principles and operating principles: The yin and yang of optimal functioning. *Math Biosci*, 2003. **182**(1): p. 81–92.
83. Colwell, L.J., M.P. Brenner, and K. Ribbeck, Charge as a selection criterion for translocation through the nuclear pore complex. *PLoS Comput Biol*, 2010. **6**(4): p. e1000747, https://doi.org/10.1371/journal.pcbi.1000747.
84. Alves, R. and A.E. Sorribas, Special issue on biological design principles. *Math Biosci*, 2011. **231**: p. 1–98.
85. Lineweaver, H. and D. Burk, The determination of enzyme dissociation constants. *J Amer Chem Soc*, 1934. **56**: p. 658–666.
86. Woolf, B., *Quoted in J.B.S. Haldane and K.G. Stern: Allgemeine Chemie der Enzyme*. 1932, Steinkopf Verlag: Dresden and Leipzig, p. 19.
87. Scatchard, G., The attractions of proteins for small molecules and ions. *Ann NY Acad Sci*, 1949. **51**: p. 660–672.
88. Schwanhäusser, B., et al., Global quantification of mammalian gene expression control. *Nature*, 2011. **473**(7347): p. 337–342.

Taylor & Francis
Taylor & Francis Group
http://taylorandfrancis.com

Gene Systems

7

When you have read this chapter, you should be able to:

- Discuss the Central Dogma of molecular biology, as well as modern amendments and refinements
- Describe the key features of DNA and RNA
- Identify the roles of different types of RNAs
- Outline the principles of gene regulation
- Set up simple models of gene regulation
- Understand the goals of synthetic biology

7.1 THE CENTRAL DOGMA

Genes are the main carriers of biological information from one generation to the next. They make us unique, are responsible for many of our good features and the bad features of others, contribute to a number of chronic diseases, and are the prime targets of evolution. In the olden days, which in this context means about half a century ago, the role of genes was simple, namely:

> Genes consist of self-replicating deoxyribonucleic acid (DNA), DNA is transcribed into matching ribonucleic acid (RNA), RNA is translated into proteins, and proteins manage the processes of daily life.

This unidirectional **Central Dogma**, proposed by Nobel Laureate Francis Crick [1, 2] became the centerpiece of modern biology.

The Central Dogma is still mostly true today, although we have learned that everything surrounding it is much, much more complicated: some DNA does not code for genes; some organisms inherit their information through RNA; molecular modifications such as DNA methylation can control **transcription**; and alternative splicing in eukaryotic cells may lead to many different RNAs and hence many different proteins from the same DNA. Maybe most significant of all, and in stark contrast to Crick's original ideas, proteins have a huge impact on DNA through their roles as **transcription factors**. In addition, most organisms employ various types of RNAs that affect which gene is transcribed when, where, and how much, and when a gene is to be silenced. Moreover, large and small metabolites can control gene expression. Large signaling metabolites include sphingolipids, which can "sense" environmental stresses and control the up- or down-regulation of appropriate genes in response [3]. An example of a small metabolite is glucose, which indirectly affects the transcription of many genes in organisms from bacteria to humans and, for instance, represses a wide variety of genes in the β-cells of the pancreas, which can be an especially serious problem in individuals with diabetes [4]. In bacteria, any excess of the rare amino acid tryptophan in the environment

DOI: 10.1201/9781003402770-7

shuts down the expression of genes that are responsible for tryptophan production, because this production is metabolically expensive and would be wasteful. These strategies of avoiding unnecessary action are implemented through regulatory mechanisms that interact with the responsible section of the chromosome. In the case of tryptophan, the amino acid directly activates a repressor of the DNA segment that contains the genes that code for the enzymes of the tryptophan pathway [5]. Evidence furthermore suggests that metabolites can affect translation (e.g., [6]). To make matters even more complicated, we have started to discover that about 98% of our DNA does not code for proteins, that genes can overlap each other, that the same stretches of DNA can lead to several alternative transcripts, and that regulatory elements are not necessarily located adjacent to the genes whose transcription they control [7].

Thus, it is still true that genes are responsible for information transfer between generations, but they are clearly not the sole agents (**Figure 7.1**). The genes do determine which proteins can be made by a cell, but they contain no direct information about the quantities and the timing of the process [8]. This information is obviously critical, because otherwise all cells of an organism would continuously produce the same proteins in the same quantities. Instead, genes, together with proteins and metabolites, act as the components of a well-organized system: genes provide the template for the synthesis of RNAs, which lead to proteins and metabolites, which in turn feed back to affect and control gene expression in multiple ways. This shared manner of control is a compromise between the original, unidirectional Central Dogma, where genes were seen as the true drivers of life, and the notion of genes as mere databases or as "prisoners" of the physiological control of the organism, where they have no autonomy or independent control [8].

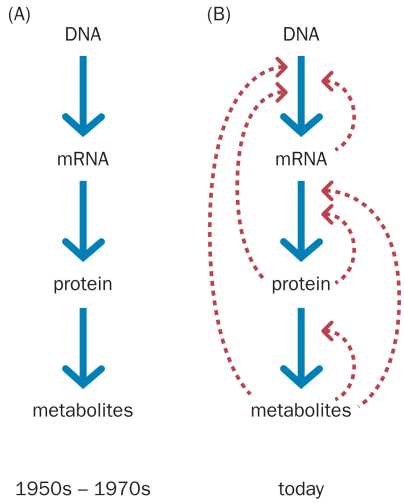


Figure 7.1 The original Central Dogma compared with modern understanding. In the original concept of the Central Dogma, transcription, translation, and enzymatic catalysis were proposed to form a linear chain of processes, although nobody doubted the role of regulation. We know now that a complex feedback structure at every level is crucial for appropriate functioning.

It is worth noting that genetic information is not exclusively transmitted through the organism's chromosomes. For instance, we have learned in recent years that **epigenetic modifications** in the form of methyl or acetyl groups attached to specific locations on the DNA can be inherited (see later in this chapter). Plants contain additional DNA in their chloroplasts, and the mitochondria of eukaryotes contain DNA that is transmitted from the mother to her offspring. Inheritance of male mitochondrial DNA has been reported in some species, although it is rare [9]. The contributions of chloroplast and mitochondrial DNA are generally small but significant nevertheless. In humans, the mitochondrial DNA codes for 13 proteins, 22 transfer RNAs, and two subunits of ribosomal RNA.

Other non-chromosomal genetic elements include **plasmids** and transposable elements. Plasmids are typically circular, double-stranded DNA molecules that are natural in bacteria, although they have also been found in archaea and in eukaryotes [10]. A bacterium may contain hundreds of plasmids at the same time and can share them with other bacteria, even from different species, in a process called horizontal gene transfer, which is crucial for the evolution of new species [11]. Plasmids can replicate independently of the main chromosome. They often contain genes that permit bacteria to survive under hostile conditions, such as exposure to antibiotics, which in hospitals can result in bacterial strains that are resistant to multiple drugs and very difficult to treat [12]. Perhaps the best-known example is methicillin-resistant *Staphylococcus aureus* (MRSA), which is a formidable culprit in many human infections [13]. On the positive side, plasmids have become extremely important research tools in biotechnology and synthetic biology, where they are used as vectors for the transport of genes into foreign species (see *Synthetic Biology* section).

Transposable elements—or **transposons**—are stretches of DNA that can move from one location in the **genome** to another [14]. In some instances, the transposon is cut out of its original location and moved, whereas in other cases, the transposon is first copied and only the copy is relocated. In bacteria, transposons can carry several genes, which might be inserted into a plasmid and make the organism resistant to environmental stresses. Over long time periods, transposons can lead to important mutations and to the expansion of the genome. It has been estimated that 60% of the maize genome consists of transposons. As one might expect, some transposons have been associated with human diseases. The *Alu* sequence can be found over a million times within the human genome and has been associated with diabetes, leukemia, breast cancer, and a number of other diseases [15].

The body of information on genes, their structure, function, regulation, and evolution is overwhelmingly huge and growing at breakneck speed, primarily owing to the rapid evolution of modern laboratory techniques. Some of this information is of great and direct pertinence to systems biology, while other aspects are less so. Arguably most important for systems analyses is an understanding of which genes are expressed when and how this complicated process is regulated by proteins, RNAs, and even metabolites. To hone this understanding in a fashion that is as brief as feasible, this chapter begins with a barebones discussion of the key features of DNA and RNA and afterwards devotes more attention to issues of the control and regulation of gene expression, followed by basic mathematical models that have been developed for gene and genome analysis.

Thus—as with the following chapters on proteins and metabolites—this chapter is bound to disappoint many biologists, because it will only scratch the surface of the fascinating world of genes, genomes, and gene regulatory systems. It does not describe experimental methods and keeps aspects of bioinformatics to a minimum. Issues of sequence analysis, gene annotation, evolution, and phylogeny, as well as strategies of genome analysis and manipulation, will be discussed rather cursorily or not at all. Excellent texts and Internet resources, including [16–20], cover these topics very well.

7.2 KEY PROPERTIES OF DNA AND RNA

7.2.1 Chemical and Physical Features

The basic building blocks of DNA are **nucleotides**, which form two long polymer strands. These strands are oriented in opposite (antiparallel) directions and are stabilized by hydrogen bonds. Because of the physical and chemical features of the nucleotides, the two strands curl slightly, and the result is the famous double helix, which James Watson and Francis Crick discovered in 1953, based in part on Rosalind Franklin's X-ray photographs of DNA and her ideas of two nucleotide strands, and for which Watson and Crick were awarded the 1962 Nobel Prize in Physiology or Medicine; Franklin had already passed away [21, 22].

DNA makes use of four different nucleotides. Each consists of a nitrogen-containing nucleo-base, the five-carbon sugar 2-deoxyribose, and a phosphate group. The nucleo-bases fall into the two biochemical classes of purines and pyrimidines and are called adenine, cytosine, guanine, and thymine, abbreviated as A, C, G, and T. If they are attached to the sugar, they are called **nucleosides**, and if a phosphate group is attached to the sugar as well, the resulting molecules are called **nucleotides**. RNA uses very similar components, but the sugar is ribose rather than 2-deoxyribose and the nucleo-base thymine is replaced with uracil (U). The building blocks are visualized in **Figure 7.2**.

The chemical structure of the nucleo-bases has important consequences. Namely, guanine on one strand of DNA always pairs with cytosine on the antiparallel strand, and adenine always pairs with thymine (**Figures 7.3** and **7.4**). Thus, one speaks of GC and AT **base pairs**, and the number of base pairs has become the most widely used size unit of DNA. Because of the strict pairing, one strand is in some sense the mirror image of the other, and this fact is of utmost importance for two fundamental processes. The first is DNA replication, which precedes cell division: double-stranded DNA opens up and both strands serve as templates for newly created DNA strands that bind to them. As a result of this **hybridization** process, the double-stranded DNA is now available in two copies, which eventually end up in the two daughter cells. The replication process is facilitated by specific proteins (**Figure 7.5**). The second fundamental process is the transcription of DNA into RNA, which follows the same principles, except that uracil replaces thymine.

Each of the two DNA strands is held together by a phosphate–deoxyribose backbone, which defines its direction. Based on the location of the bond on the sugar, they are called the 3-prime end (3′ end) and 5-prime end (5′ end). In Figure 7.3, the backbone molecules are identified by R. The sequence of three base pairs GC-AT-GC is illustrated in Figure 7.4. DNA consists of millions of base pairs that are connected in this fashion of two complementary strands. The chemical

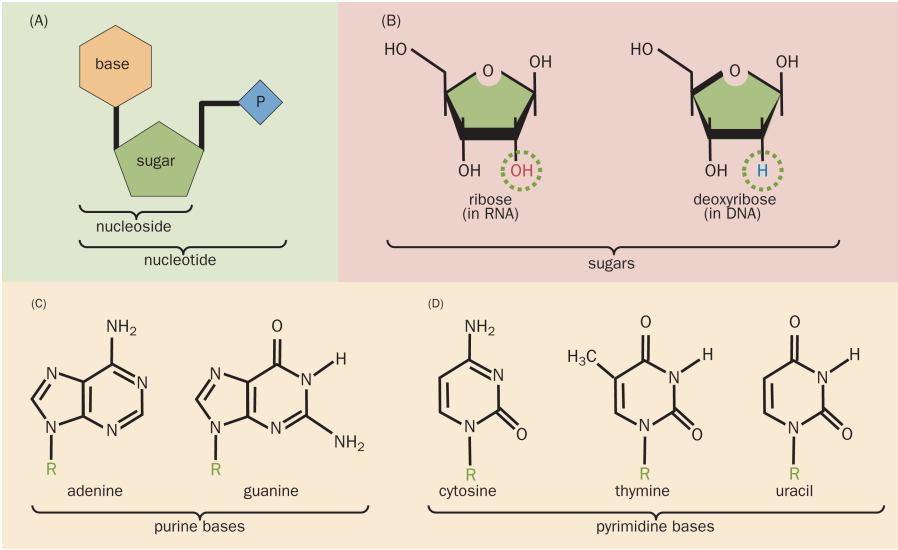


Figure 7.2 Building blocks of DNA and RNA. (A) Generic composition of nucleotides. (B) The sugars in RNA and DNA differ slightly, by one oxygen atom. Both DNA and RNA contain purine (C) and pyrimidine (D) bases. Adenine, guanine, and cytosine are common to DNA and RNA, but the fourth base in DNA is thymine, while in RNA it is uracil.

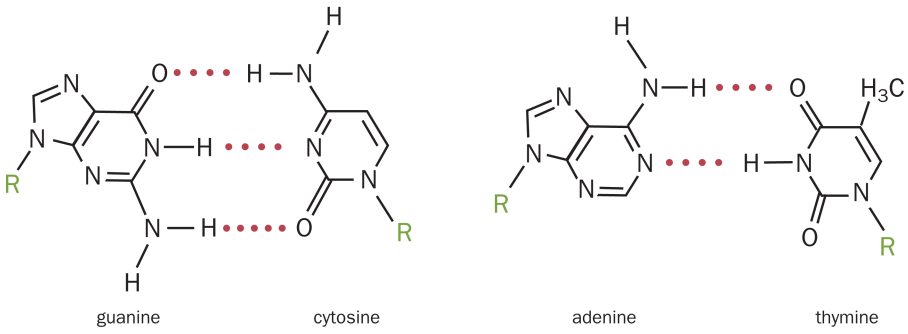


Figure 7.3 Guanine always pairs with cytosine on the anti-parallel strand, while adenine always pairs with thymine. Hydrogen bonds between the partners are shown as dotted red lines. Note that the GC pair contains three bonds, whereas the AT pair contains two. The GC bonding is therefore stronger.

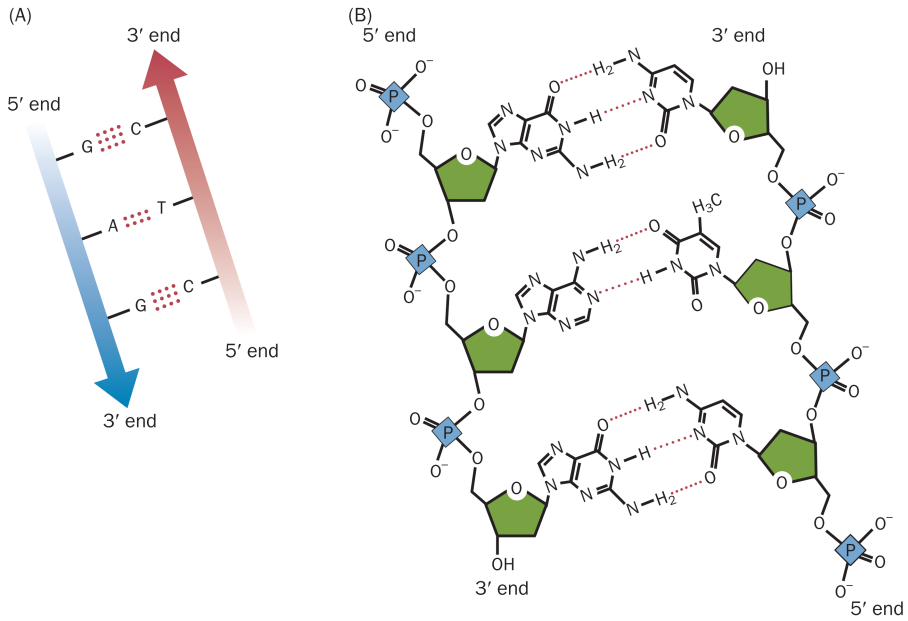


Figure 7.4 Sequence of three base pairs, namely GC, AT, GC, as they are shown in Figure 7.2. The connection between two neighboring bases is made by sugars and phosphates, which form one phosphate–deoxyribose backbone per strand. The backbones run in an anti-parallel fashion. (A) Diagram of backbones (blue and red) and bases with hydrogen bonds. (B) Molecular details.

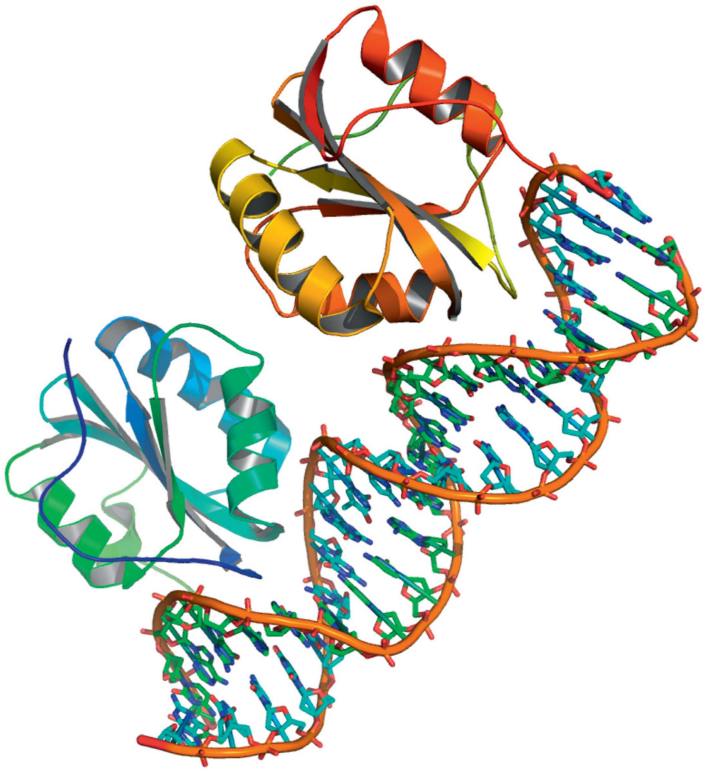


Figure 7.5 Two simian virus SV40 protein domains interact with DNA just before replication. The high-resolution crystal structure shows the origin binding domains (OBDs) of a large T antigen (T-Ag) complexed with the DNA fragment that contains the origin for replication. The core origin contains four binding sites, each consisting of five nucleotides, which are organized as pairs of inverted repeats. In the co-structure, two T-Ag OBDs are oriented in a head-to-head fashion on the same face of the DNA, and each T-Ag OBD engages the major groove. Although the OBDs are very close to each other when bound to their DNA targets, they do not contact each other. (Courtesy of Huiling Chen and Ying Xu, University of Georgia.)

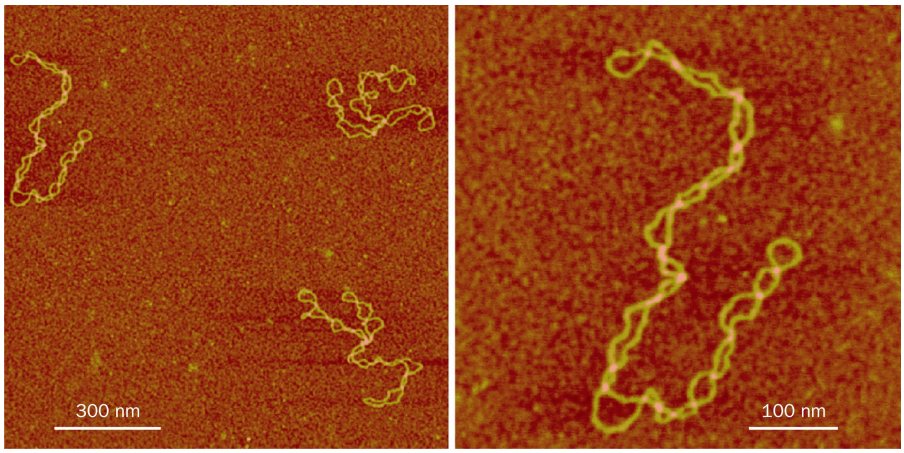


Figure 7.6 Modern methods of atomic force microscopy (AFM) make it possible to visualize supercoiled DNA. The images show 5,600 base pairs of plasmid DNA from *Escherichia coli*. (From Lyubchenko YL. *Micron*. 42 [2011] 196–206. With permission from Elsevier.)

and physical properties of these strands cause double-stranded DNA to coil into a double helix, and this double helix itself curls into supercoils that can be visualized with methods of atomic force microscopy (**Figure 7.6**).

7.2.2 Size and Organization of DNA

Interestingly, the size of a genome has no real correlation with the size or complexity of an organism, and the size of the genome is not even correlated with the number of protein-encoding genes [23]. Human DNA consists of about three billion base pairs, 2% of which form between 20,000 and 25,000 protein-encoding genes. Some plants have many more genes: The black cottonwood, *Populus trichocarpa*, has over 40,000 genes and over 74,000 gene transcripts [24]. The lowly ameba *Polychaos dubium* apparently contains 200 times as much DNA as humans, while the pufferfish *Fugu rubripes*, with roughly 400 million base pairs, has the shortest

genome of any vertebrate species, although it contains about 30,000 genes, which is more than the number of protein-encoding genes in humans [25]. A cogent explanation for these apparent discrepancies cannot be given at present, although there are many partial explanations, mixed with plenty of speculation. For instance, alternative splicing of gene transcripts can ultimately result in a much larger number of different proteins than the number of coding genes might suggest [26]. In this "Lego-block" strategy, short protein sequences (peptides) can be linked together in many variants. The enormous amount of **noncoding** DNA (98% in humans) contains different types of RNAs that are involved in transcription, as well as in many regulatory mechanisms that guide the expression of appropriate genes and the concatenation of appropriate peptides, at the right time and in the right place (see later in this chapter).

Limiting our comparisons of genome sizes to bacteria, one does see some trends. If a bacterial species lives in a stable environment with sufficient nutrition, such as the gut of an animal, the number of genes averages about 2,000. By contrast, aquatic bacteria have about 3,000 genes, and terrestrial bacteria average about 4,500 [27]. An intuitive explanation is that larger genomes permit more—and more complex—metabolic and physiological pathway systems, which are needed if the organism has to survive in drastically varying ecological niches. It has been estimated that bacteria have roughly the same numbers of genes and proteins, but that the number of different metabolites is an order of magnitude lower [28].

Gene sequencing technologies developed over the past few decades have made it possible to determine the order of nucleotides in a DNA strand. This process, which is by now fairly automated, has opened incredible insights into the manner with which nature codes life processes [29]. At first, a DNA sequence looks like a random arrangement of A, C, G, and T, but in truth it contains an enormous amount of information. At the lowest level, triplets of nucleotides in a stretch of coding DNA (called **codons**) correspond to specific **amino acids** in proteins; we will discuss this **genetic code** in greater detail in Chapter 8. Codons can also correspond to start and stop signals for transcription. In other words, the DNA of an organism is a blueprint for the features of the protein for which it codes. Because triplets of four nucleotides permit 64 different combinations and most organisms use only 20 amino acids, there is some degeneracy, and the same amino acid may correspond to several codons. Again, the choice of one of these triplets is not random, and different organisms prefer some triplets over others, a phenomenon called codon-bias or wobble. In fact, the preferences of specific triplets for the same amino acids are so species-specific that it is possible to "barcode" species by their sequences [30]. This barcoding is accomplished in the following fashion. One computationally moves along the DNA sequence, nucleotide by nucleotide, and reports at each nucleotide location the frequencies of all different combinations of four nucleotides and their antiparallel complements, such as GTTC/CAAG, GCTC/CGAG, CTAC/GATG, within a window of the next 1,000 base pairs. Collecting these data, it turns out that these frequencies are far from equal or random and instead form a very specific barcode pattern when they are represented as different shades of gray (**Figure 7.7**). It has been shown that these barcodes correlate with the phylogenetic similarity of organisms and that they can be used to classify DNA from prokaryotes, eukaryotes, plastids, plasmids, and mitochondria. They can also be used to characterize the genomes of mixed microbial communities, which consist of many different cohabitating species (see Chapter 14).

7.2.3 Genes and Noncoding DNA

It was initially assumed that the DNA of an organism consisted of a neat sequence of all genes. However, DNA has turned out to be much more complicated in terms of its encoded information and organization. Indeed, even the concept of a gene has changed over time.

In his revolutionary work on the inheritance of certain features of peas, the Austrian monk Gregor Mendel (1822–1884) noticed that biological traits are inherited from one generation to the next as discrete units. The discrete nature of this process allowed the separation of traits and explained the old observation that sib-

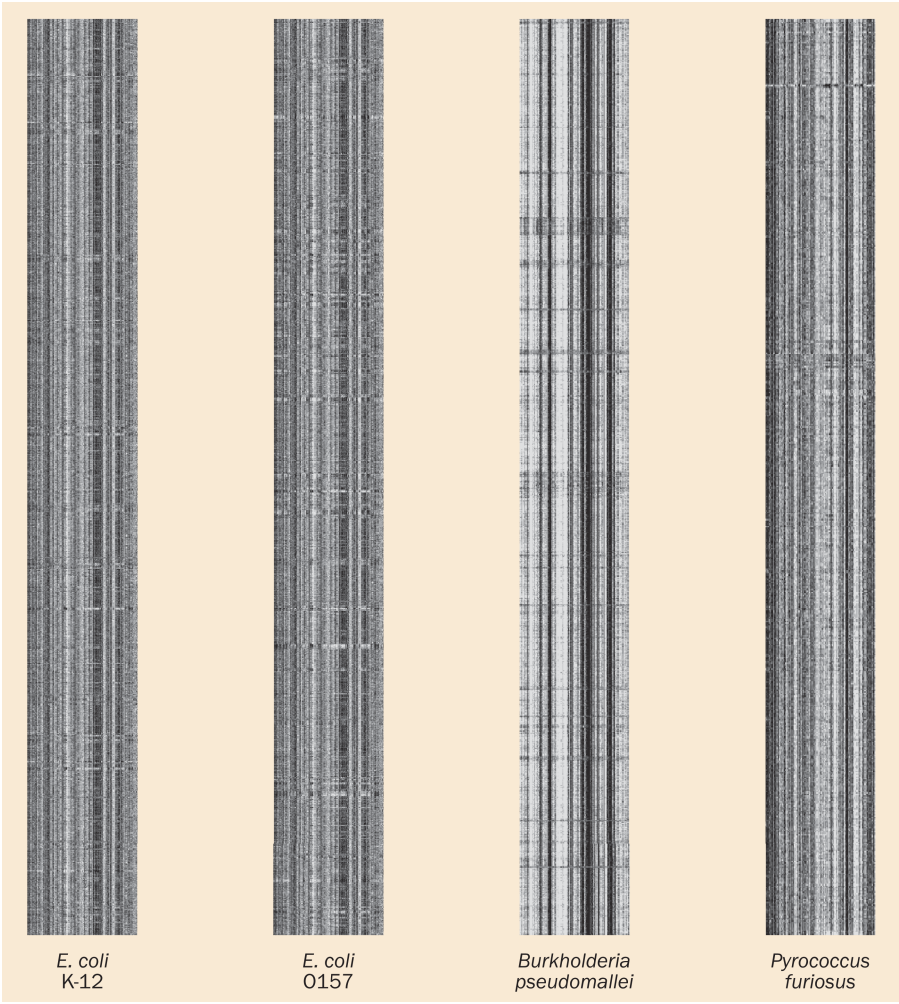


Figure 7.7 Barcodes of four bacterial strains. The *y*-axis in each block represents the location of measurement within the DNA of an organism, while the *x*-axis shows in shades of gray the frequencies of all possible combinations of nucleotides within a window of 1,000 nucleotides, starting at nucleotide *y*. Amazingly, the frequencies are almost independent of the location within the DNA. It seems that the occasional horizontal "stripes" correspond to genes that the organism obtained during evolution from different species through horizontal gene transfer. (Courtesy of Fengfeng Zhou and Ying Xu, University of Georgia.)

lings can have different features. Mendel's findings were more or less ignored for a long time, until DNA was identified as the carrier of inheritance in the 1940s. At that point, genes were seen as the DNA code for traits such as eye color, body height, and possibly for characteristics like personality and intelligence.

Modern biology uses a more specific definition for a gene, namely as a hereditary unit or as a specifically localized region of DNA that codes for a protein. The direct function of a coding gene is therefore the protein that results from its transcription and **translation**. By contrast, the assignment of higher-order function is often questionable, because such functions usually involve many processes that are executed by different proteins. It is therefore too ambiguous and vague to talk about an "eye-color gene," an "intelligence gene," or an "Alzheimer gene." Indeed, the **Gene Ontology** (**GO**) Consortium [31], which assigns functions to genes, states that "oncogenesis is not a valid GO term because causing cancer is not the normal function of any gene." Instead, the GO Consortium uses three categories that characterize different molecular properties of gene products:

- *The cellular component*, which refers to parts of a cell or its extracellular environment where the gene product is located.
- *Molecular function*, which covers fundamental activities of the gene product, such as binding or a specific enzymatic activity.
- *Biological process*, which refers to well-definable operations or molecular events but not to higher functions such as renal blood purification.

It is rare that a single gene is solely responsible for some macroscopic feature or phenotype. Instead, the inheritance of complex, multifactorial traits such as body mass, skin color, or predisposition to diabetes depends on the contribution of

many genes passed from parents to their children. This polygenic inheritance makes genetic studies challenging, because it involves the co-regulation of several genes and the control of possibly complex interactions between genes and the environment through epigenetic effects (see later).

If a gene is a specifically localized region of DNA that codes for a protein, is there DNA that is not a gene? The answer is a resounding "Yes." It is currently assumed that humans possess between 20,000 and 25,000 genes. That sounds like a lot, but it only comprises an estimated 2% of our DNA! Much of the remaining DNA has unknown functions, although one should be very careful calling it "junk DNA," as it was termed in the 1970s [32], because it is likely that much (or all?) of the "junk" might turn out to be good for something. Indeed, the more we learn about this noncoding DNA, the more we find that it has intriguing and very substantial roles in living cells, including regulatory roles at the transcriptional and translational levels. That these long stretches of DNA are not at all useless is also supported by the observation that some of them are remarkably stable throughout evolution and even conserved between diverse species such as humans and pufferfish, which implies that they are under strong selective pressure and therefore most probably rather important [7]. At the same time, other stretches of DNA are not well conserved between species, which simply implies that we do not really understand the hidden signals in some noncoding DNA and their role in gene regulation. Current research therefore focuses on identifying markers for locations of noncoding elements and on characterizing their specific functions. In some cases, we have a vague idea of their roles, as reflected in the names of such DNA stretches—promoters, enhancers, repressors, silencers, insulators, and locus control regions, for example—but we have yet to develop a complete inventory.

Our current understanding of the organization of DNA is the following [18]. The vast majority of DNA is noncoding and only a few percent of DNA codes for the synthesis of proteins in eukaryotes; this percentage increases in less advanced organisms. The average gene size in humans is about 3,000 base pairs but varies widely; the largest known human gene is dystrophin, with 2.4 million base pairs. Coding regions are usually rich in pairs of guanine and cytosine (GC pairs), which are more stable than AT pairs owing to their higher numbers of hydrogen bonds (see Figure 7.3). The distribution pattern of genes within DNA can so far not be explained. Other species have recognizable regulatory patterns of genes and noncoding regions. Long stretches of DNA containing lots of G's and C's apparently flank genes and form a barrier between coding and noncoding regions. The role of noncoding DNA is so far insufficiently understood, but some of this DNA is suspected to serve regulatory roles.

The prediction of genes from sequence data is still a challenge. One piece of evidence for identifying a gene is the open reading frame (ORF), which is a stretch of DNA that does not contain an in-frame stop codon. The presence of a start codon and a following sufficiently long ORF are often used as an initial screening tool for potential coding regions, but they do not prove that these regions will be transcribed and translated into protein. The challenges of gene prediction and annotation are greatly magnified in the new field of metagenomics, in which all genomes of large microbial communities are analyzed simultaneously. Such mixed communities may consist of hundreds if not thousands of microbial species, which coexist in environments such as the human gut, the soil, and the oceans (see Chapter 14).

Even within a eukaryotic gene, the DNA often contains regions that are not translated into protein. These so-called **introns** are typically transcribed but later removed in a process called splicing, whereas the remaining regions of the gene, the **exons**, are merged into one piece of message that is ultimately translated into protein. The splicing process, which is controlled by a variety of signaling molecules, permits the translation from different combinations of RNA pieces and thereby increases the possible number of resulting proteins (alternative splice variants) considerably (**Figure 7.8**). It is currently believed that some introns are transcribed as RNA genes such as **microRNAs**, which can regulate protein-encoding gene expression (see later).

Over the past decades, molecular biology and bioinformatics have developed an enormous repertoire of laboratory techniques and computational tools for

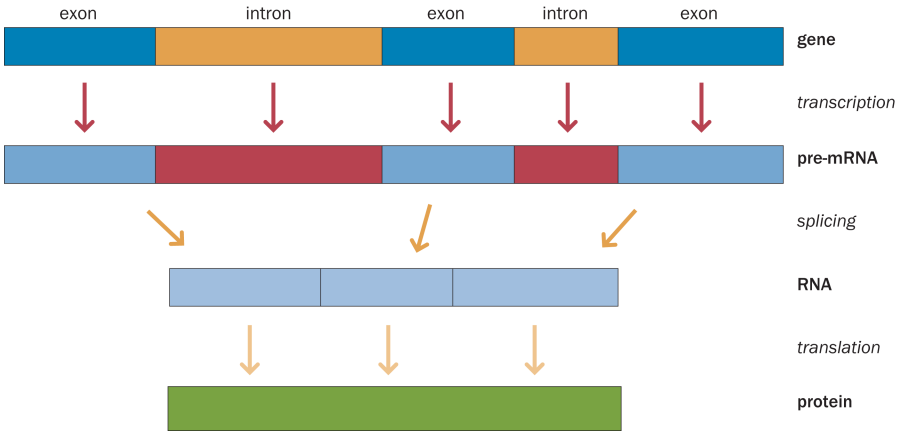


Figure 7.8 Exons and introns. Many eukaryotic genes contain introns, which ultimately do not correspond to amino acid sequences in the protein for which the gene codes. The splicing mechanism at the mRNA level permits alternative combinations and greatly increases the possible number and variability of proteins.

determining DNA sequences for prokaryotic and eukaryotic species, and even for individual humans [33], as well as for comparing DNAs from different species. These sequencing capabilities have allowed us to establish gene inventories for many species, along with their variations; notable, very comprehensive databases with this information include [18, 31, 34–36]. Furthermore, the journal *Nucleic Acids Research* annually publishes a special issue on biological databases. The relative ease of DNA sequencing has also led to much more accurate phylogenetic trees and other evolutionary insights than had been possible before. The methods for sequencing and sequence comparisons are fascinating; they are detailed in many books (for example, [17, 19, 29, 37]) and uncounted technical papers and review articles. An ongoing major challenge is the annotation of genes, that is, the identification of the function of a coding region. The Gene Oncology Consortium [31] collects methods and results of such efforts.

7.2.4 Eukaryotic DNA Packing

In prokaryotes, most of the DNA is arranged in a single chromosome, but some may be located separately, forming plasmids, as we discussed before. These plasmids can be shared among members of the same species or even other species, which leads to a potentially wide distribution of DNA and is a key mechanism of evolution. In higher organisms, DNA is organized in a species-specific number of chromosomes, which consist of DNA together with proteins that help package DNA efficiently and contribute to the control of gene transcription. Of particular importance are histones, which are spherical proteins around which DNA is wound (**Figure 7.9**), and protein scaffolds that are used for the DNA packaging process (**Figure 7.10**) [38]. A basic unit of about 150 DNA base pairs, wound around a histone core, is called a nucleosome (**Figure 7.11**; see also Figure 7.9). The nucleosomes are connected by stretches of about 80 base pairs of free DNA and coiled into higher-order structures, which eventually form chromatin, which makes up the chromosomes (**Figure 7.12**). The component *chrom* in these terms is Greek for color and reflects the fact that chromatin, when stained, becomes visible under a light microscope. The repeating pattern of nucleosomes and free DNA, which can be seen with an electron microscope, permits the astonishing packaging of about 2 meters of eukaryotic DNA into a nucleus with a diameter of only 10 μm. At the same time, the DNA packaging allows selective access to the DNA for transcription, which is controlled by chromatin remodeling motors that consist of proteins and use ATP for energy [39]. Specifically, the dense and well-known packing displayed in Figure 7.12 only has this form during mitosis, when the cell doubles. At other times, the 3D-structure of chromosomes consists of loops of different sizes, spatial compartments and chromosome territories that can interact with nuclear structures [40]. This 3D-arrangement directly affects transcription and changes, for instance, during the progression of cancer or differentiation of cells during development [41, 42].

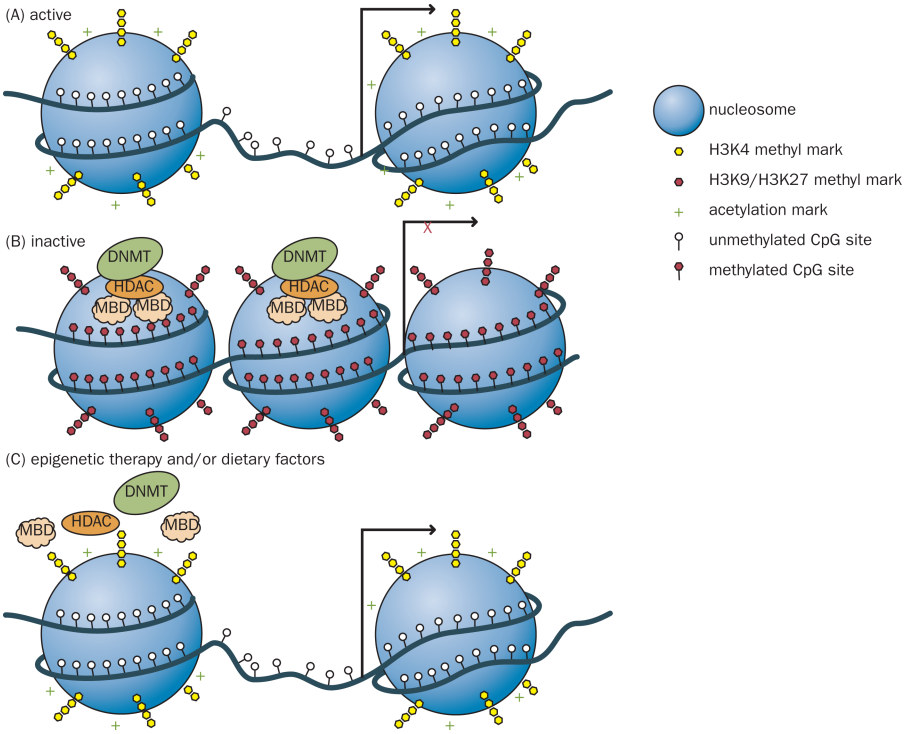


Figure 7.9 Diagram illustrating DNA organization around histones and epigenetic effects influencing the expression of genes close to a nucleosome (blue spheres). (A) Promoters of active genes are often associated with CG pairs that are not methylated (white circles), with acetylated histones (green crosses), and with methylated histone H3 (yellow hexagons); this configuration is conducive to transcription. (B) In some diseases such as cancer, CG sites in the promoter region of tumor suppressor genes are often methylated (red circles). Methyl-CG-binding domains (MBDs), histone deacetylases (HDACs), and DNA methyltransferases (DNMTs) make chromatin inaccessible and lead to repression of transcription. (C) Epigenetic therapy could potentially decrease MBDs, HDACs, and DNMTs and reverse their deleterious effects. (From Huang Y.-W, Kuo C.-T, Stoner K, et al. *FEBS Lett.* 585 [2011] 2129–2136. With permission from Elsevier.)

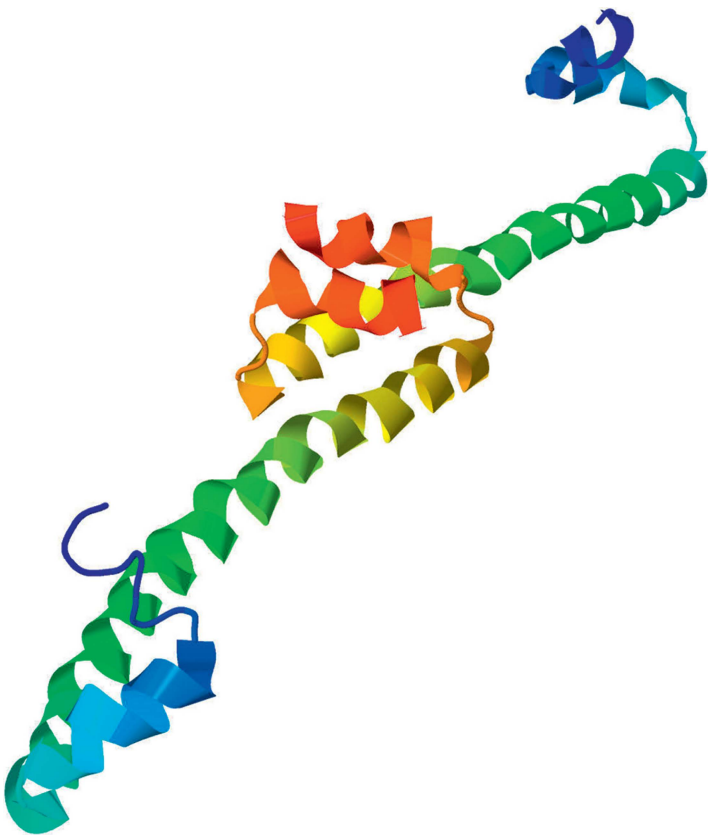


Figure 7.10 The tight packaging of DNA is accomplished with special proteins. Crystal structure of a superhelical protein scaffold that supports the packaging of DNA in the gut bacterium *Salmonella typhimurium*. (Protein Data Bank: PDB 3NR7 [43].)

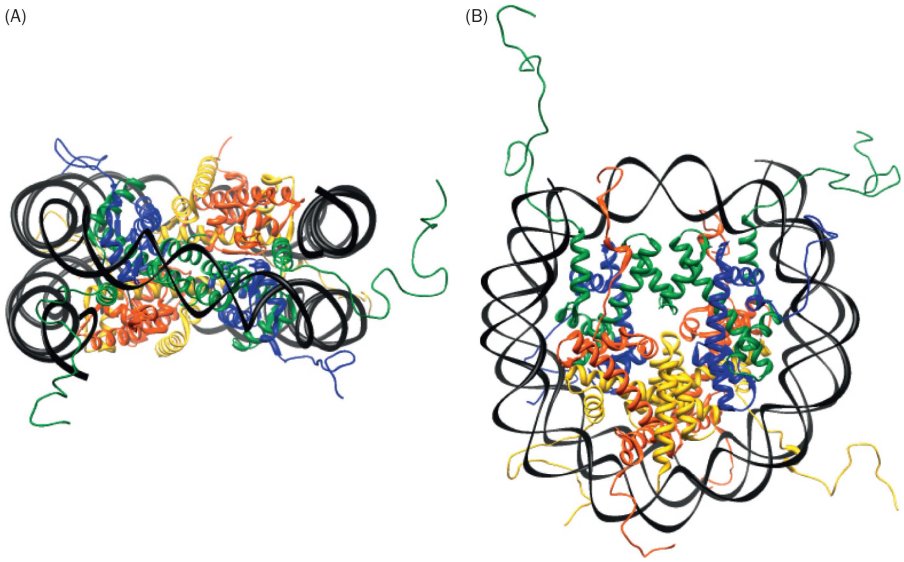


Figure 7.11 Images of a nucleosome. DNA (black) is wound around a protein complex, consisting of histones H2A (red), H2B (yellow), H3 (green), and H4 (blue). (A) Side view. (B) Top view. (From Narlikar GJ. *Curr Opin Chem Biol.* 14 [2010] 660–665. With permission from Elsevier.)

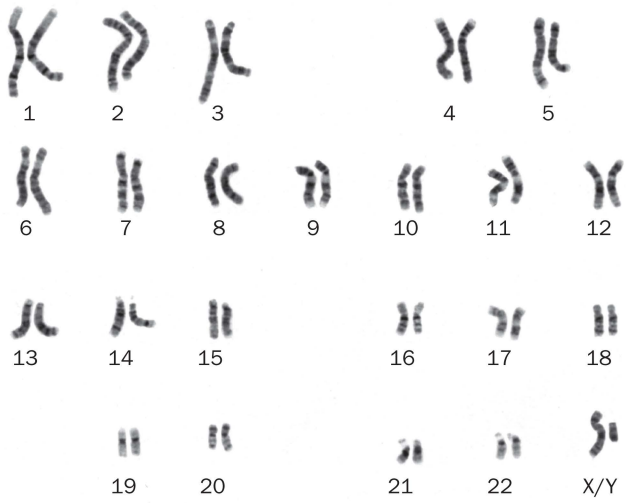


Figure 7.12 Human chromosomes. During mitosis, the chromosomes are densely packed, as shown here. At other times, their structure consists of loops of different sizes and form spatial compartments and chromosome territories that interact with nuclear structures. (Image made by the National Human Genome Research Institute.)

7.2.5 Epigenetics

We are still at the beginning of understanding how the environment, diet, and diseases can affect transcription through **epigenetic** processes that influence where and when genetic information is to be used (see Figure 7.9) [44]. In these processes, small molecules such as methyl or acetyl groups can be attached to DNA as adducts and lead to heritable modifications of the chromatin. This means that it is possible that environmental factors affecting an individual can still be found in children and even grandchildren. The DNA adducts can allow the opening of DNA for transcription or silence gene expression, and such processes have been associated with various diseases and all stages of tumor development, from initiation to invasion and metastasis. Fortunately, it seems possible, at least in principle, to reverse adverse epigenetic effects through clinical treatment and healthy living (see Figure 7.9C) [45]. Epigenetic drugs are being explored for altering epigenetic marks on DNA by inhibiting the enzymes responsible for reading, adding, or removing chemical adducts. While most extensively explored for cancer therapies, this class of drugs is also being investigated for anti-aging and chronic diseases [46].

7.3 RNA

RNA and DNA differ only slightly in their basic biochemical properties: the sugars in their backbone differ by just one oxygen atom, and one of the four bases is thymine in DNA and uracil in RNA (see **Figure 7.2**). Despite their close similarities, DNA and RNA have very distinct roles in biology. DNA is the typical carrier of genetic information, whereas different types of RNA have diverse roles as intermediate information transducers and as regulators. One key difference is that, unlike DNA, most RNA molecules are single-stranded. Because these single RNA strands are not matched up with partners, as in the case of the DNA double helix, they are either small in size, short-lived, or exhibit complex two- and three-dimensional structures, collectively called **secondary structure**, where some of the nucleotides form bonds with other nucleotides of the same RNA strand (**Figures 7.13** and **7.14**).

7.3.1 Messenger RNA (mRNA)

For most organisms, mRNA provides the means of converting the DNA blueprint into proteins. In this transcription process, DNA opens up and serves as a template for RNA, which is generated as a new strand in which one nucleotide at a time is added to match the corresponding nucleotide of the DNA strand. This transcription requires an enzyme, called RNA polymerase, and can be extensively regulated by transcription factors, as we will discuss later.

In eukaryotes, transcription occurs within the nucleus. The new RNA is transported from the nucleus to the cytosol, where ribosomes translate it into a string of amino acids, which form the primary structure of proteins (see Chapter 8). This translation process requires two other types of RNA, namely transfer RNA (tRNA) and ribosomal RNA (rRNA).

7.3.2 Transfer RNA (tRNA)

This type of RNA is relatively short, consisting of only about 80–100 nucleotides. It is synthesized from so-called RNA genes, which are usually located within stretches of noncoding DNA. The role of each tRNA is to transfer a specific amino acid to the growing peptide chain that is being constructed by a ribosome (**Figure 7.15**). Each tRNA contains a triplet of nucleotides that corresponds to a sequence of three DNA nucleotides, that is, a codon, which "codes" for the target amino acid (see Chapter 8 for a more detailed discussion of codons). This amino acid attaches to the 3′ end of the tRNA and, once associated with the ribosome, is covalently bound to the already formed polypeptide, a process that is catalyzed by the enzyme aminoacyl tRNA synthetase.

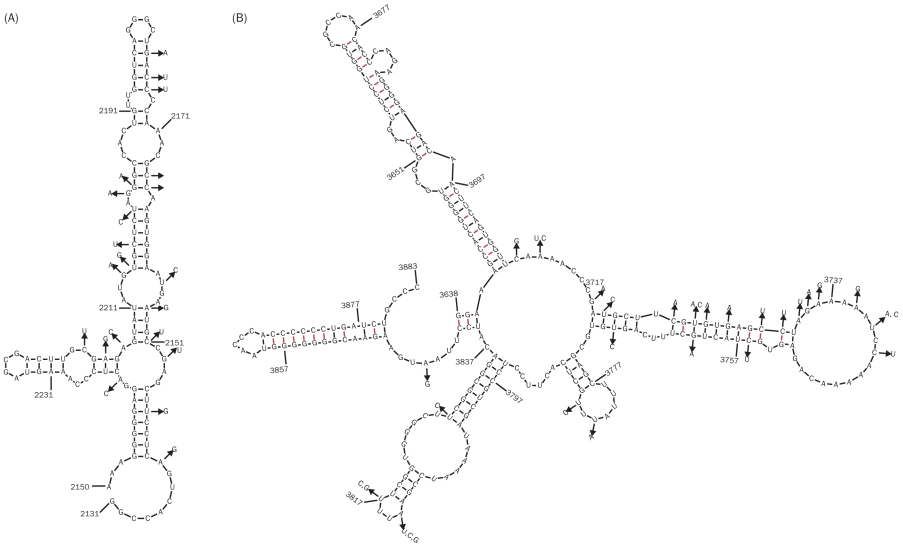


Figure 7.13 Secondary RNA structures. Shown here are structures of a putative promoter (A) and the untranslated region at the 3′ end of the *Pelargonium* line pattern virus (B). (From Castaño A, Ruiza L, Elenaa SF & Hernández C. *Virus Res.* 155 [2011] 274–282. With permission from Elsevier.)

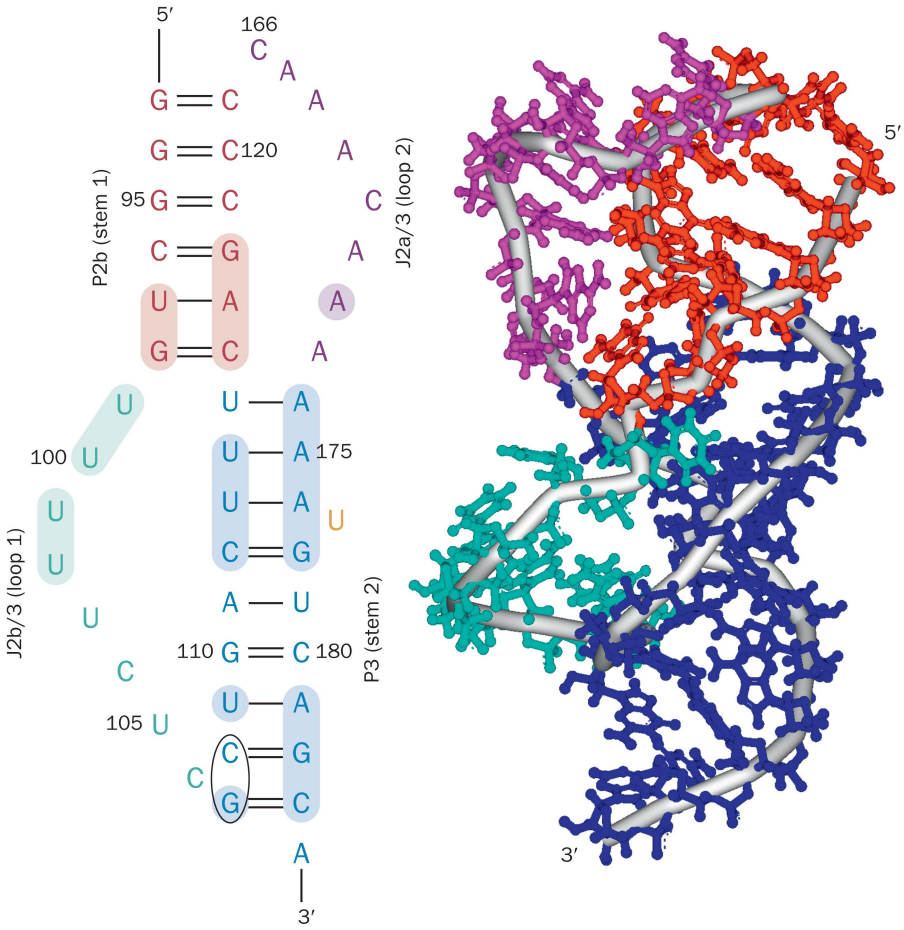


Figure 7.14 Single-stranded RNA often folds onto itself. The folding leads to bonds that give the molecule the stable two-dimensional and three-dimensional structure of a pseudoknot. Different colors of nucleotides indicate their association with stems (red and blue), and two loops (cyan and magenta). (From Yingling YG & Shapiro BA. *J. Mol. Graph. Model.* 25 [2006] 261–274. With permission from Elsevier.)

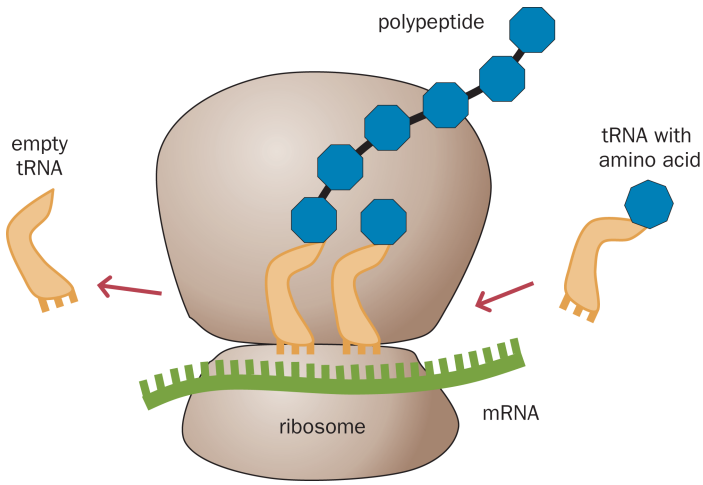


Figure 7.15 Schematic representation of the translation of mRNA into a polypeptide. A ribosome moves along the messenger RNA (mRNA; green). Corresponding to each codon on the mRNA, a tRNA (yellow) with the appropriate amino acid (blue octagons) is recruited. It enters the ribosome and permits the binding of its attached amino acid to the growing polypeptide chain (blue chain).

Each tRNA is single-stranded but forms bonds between its own nucleotides that result in a secondary cloverleaf structure. This structure consists of three stems with loops, a variable loop, and an acceptor arm that permits the amino acid to attach. The center arm contains the anticodon region, whose three nucleotides match the codon on the mRNA and ensure that the correct amino acid is selected. To fit into the ribosome for translation, the tRNA must bend at its variable loop into an L-shape (Figure 7.16).

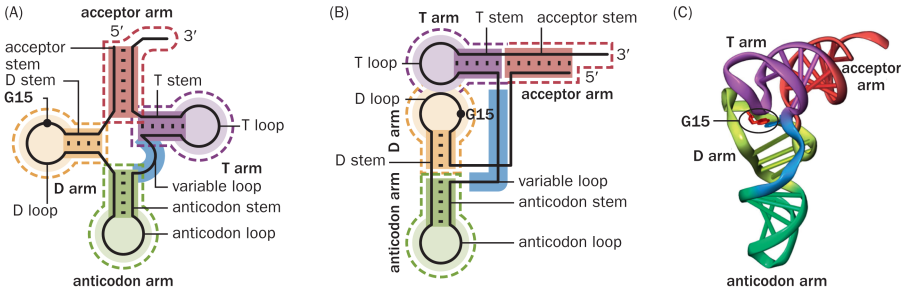


Figure 7.16 Cloverleaf structure of a transfer RNA (tRNA). (A) This secondary structure consists of four arms. The acceptor arm allows the attachment of a specific amino acid, whereas the opposing arm contains the matching anticodon, which mirrors the corresponding codon on the mRNA. (B) To serve its role within the ribosome, the tRNA has to bend at its variable loop into an L-shape. (C) The tRNA's three-dimensional structure. At position G15, this particular tRNA has been modified. (From Ishitani R, Nureki O, Nameki N, et al. *Cell.* 113 [2003] 383–394. With permission from Elsevier.)

7.3.3 Ribosomal RNA (rRNA)

Ribosomes facilitate the process of translating mRNAs into chains of amino acids. They consist of proteins and rRNAs and are composed of two subunits (see Figure 7.15). The smaller one attaches to the mRNA and the larger one associates with the tRNAs and the amino acids they carry. When reading an mRNA is completed, the two subunits separate from each other. Functionally, ribosomes are considered enzymes, belonging to the class of **ribozymes**, because they are catalysts for the linking of amino acids through peptide bonds (**Figures 7.17** and **7.18**). The catalytic activity has led researchers to focus on introducing ribozyme sequences for the design of synthetic biology circuits, where the modularity, expression control, and elimination of delays introduced by translation are advantages over genetically encoded protein components [47].

7.3.4 Small RNAs

Several types of small, double-stranded RNAs have been discovered and characterized in recent years [48]. Because of their small size, they had been overlooked for a long time, but it is now clear that they play important roles in numerous fundamental biological processes and diseases. Small RNAs fall into several classes, the most prominent of which are small-interfering RNA (siRNA) and micro-RNA (miRNA). Interestingly, RNA interference (RNAi) pathways involving these types of RNAs are highly conserved throughout evolution, which underscores their central importance from a different angle. The first discovery of RNAi was actually made in an effort to enhance the flower patterns of petunias [49], demonstrating once again that one can never foretell where some seemingly obscure scientific endeavor may eventually lead.

Small interfering RNA goes by different names, including **silencing** RNA and short interfering RNA, both of which luckily permit the same abbreviation siRNA. These short, double-stranded RNA molecules consist of 20–30nucleotides per strand, of which usually 21–25 form base pairs, while the remaining two or three stick out at either side and are not bound (**Figure 7.19**). The importance of siRNAs derives from the fact that they can interfere with the expression of genes and even

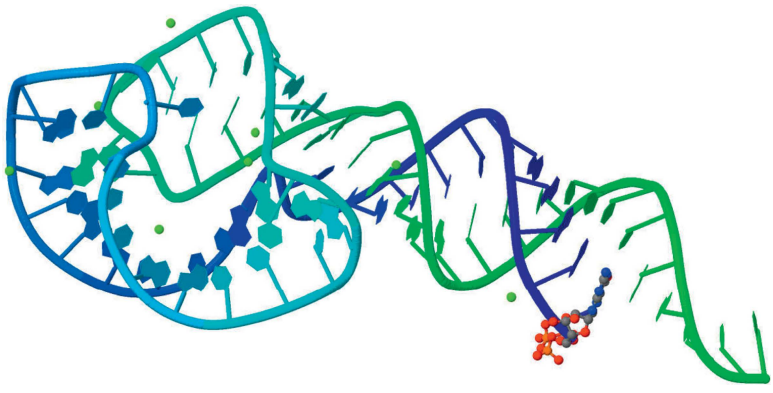


Figure 7.17 Crystal structure of a rather simple "hammerhead" ribozyme from the satellite RNA of the tobacco ringspot virus (sTRSV). This molecule was the first hammerhead ribozyme discovered. It uses a guanine residue as a general base during RNA catalysis. The structure shows how the single-stranded RNA folds and binds to itself. Also seen are guanosine triphosphate (balls and sticks) as a modified residue and magnesium ions (spheres) as ligands. Protein Data Bank: PDB 2QUS [43].

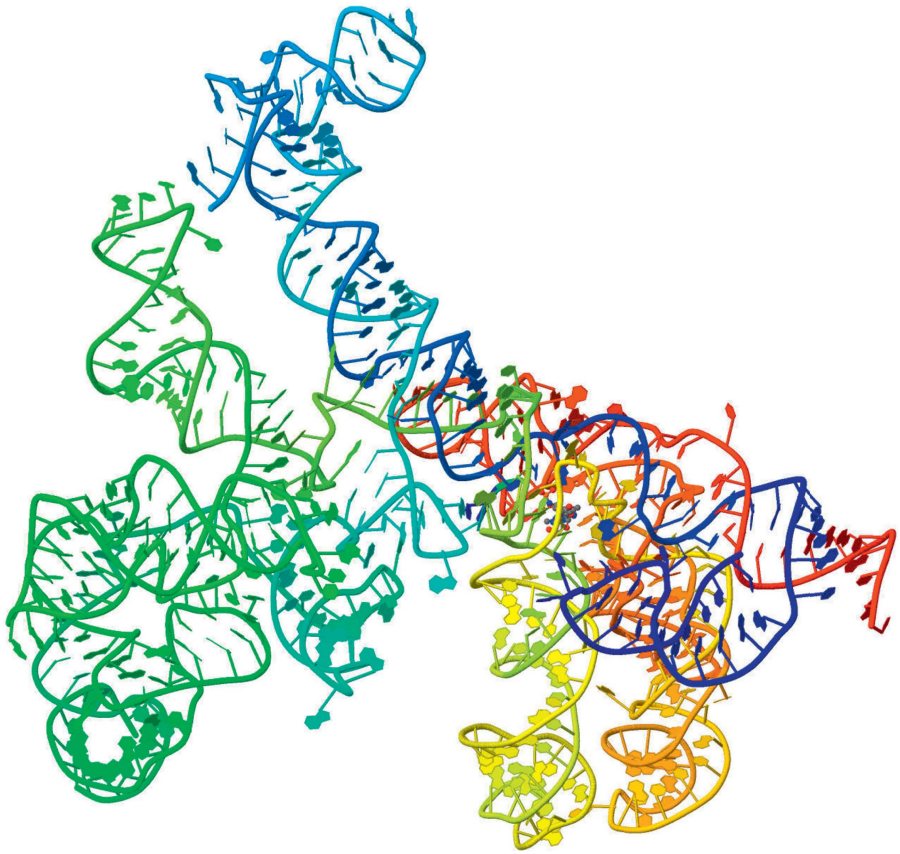


Figure 7.18 Structure of a ribozyme. The atomic structure of the large ribosomal subunit from the Dead Sea archaeon *Haloarcula marismortui* shows that the ribosome is a ribozyme, which catalyzes the formation of peptide bonds during translation. (Protein Data Bank: PDB 1FFZ [43].)

silence it. This interference constitutes a potent regulatory mechanism that allows the cell to slow down or shut down translation at appropriate times [50–52].

Once introduced into cells, siRNAs permit the silencing of precisely targeted gene transcripts. The main mechanism seems to be the degradation of mRNA. This so-called RNA interference process, or RNAi for short, begins with long double-stranded RNA, which is processed by an enzyme called *dicer* into small, sequence-specific siRNA. This siRNA triggers an RNA-induced silencing complex (RISC), which is an enzyme that catalyzes the cleavage of the target mRNA. Because siRNAs can be constructed artificially, the discovery and experimental mastery of RNAi has led to an enormous repertoire of tools for gene expression studies—including so-called loss-of-function techniques—and to a new class of potentially powerful therapeutics.

Other small RNAs have been discovered in recent times. Similar to siRNA are micro-RNAs (miRNAs), which have also been called small temporal RNAs (stRNA). Like siRNAs, miRNAs are produced from double-stranded RNA by a dicer enzyme. Their function is not quite clear, but it seems that they negatively regulate the expression of target transcripts. Apparently, miRNAs modulate protein expression through (a) a slicer mechanism for high-homology mRNA targets, (b) slicer-independent cleavage or degradation of mRNA, or (c) inhibition of translation without appreciable changes in mRNA levels [50]. It has been estimated that the human genome might encode over 1,000 miRNAs, which potentially target half of all genes. This widespread influence is apparently possible because the homology between miRNAs and mRNAs does not have to be perfect. This imperfect matching is a significant difference between siRNAs and miRNAs in animals [52].

The discovery of miRNAs has led to exciting new insights and options for manipulating biological systems and possibly the treatment of cancer [53, 54]. However, one must be careful not to jump to conclusions regarding potential drug treatments too quickly, because the function of miRNAs in complex systems is not predictable from silencing studies executed in simple experimental systems.

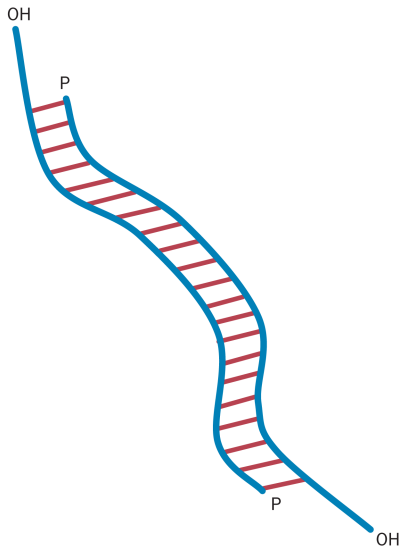


Figure 7.19 Schematic representation of an siRNA molecule. A typical siRNA consists of 21 base pairs and overhangs at both ends. P indicates the phosphate group at the 5′ end, while OH represents the hydroxyl group at the 3′ end.

The reason is that a miRNA may silence some gene *x* whose normal function is the down-regulation of another gene *y*, so that the indirect effect of the miRNA on gene *y* may be activating rather than inhibiting. In fact, to understand the full impact of a miRNA it is necessary to consider the entire system of direct and indirect influences of the miRNA on the genome.

Double-stranded RNAs can also activate gene expression, which has led to the terms saRNA or small activating RNA. Finally, piRNAs (piwi-interacting RNAs) play an important role in germ-line development. It seems that these RNAs may silence transcription by regulating DNA methylation [48]. The characterization of small RNAs and their functional pathways is the subject of intense ongoing research.

Lai and collaborators recently reviewed numerous articles demonstrating how our understanding of the roles of micro-RNAs for gene regulation can be enhanced through mathematical modeling [55, 56].

7.3.5 RNA Viruses

As an exception to the Central Dogma, many viruses use single-stranded RNA instead of DNA as their inheritable material; some even use double-stranded RNA (**Figure 7.20**). The single strand is either arranged in a positive sense, which means that it acts like a messenger RNA that had been transcribed from DNA, or in a negative sense, in which case the strand is first transcribed into a matching positive strand. RNA viruses mutate very easily, because their polymerases do not possess the proof-reading machinery of most DNA polymerases. Thus, transcription errors go undetected but are advantageous in the sense that some of them allow the virus to develop resistance to environmental or pharmacological stressors. The positive strand of RNA is directly translated into a single protein, which the host cell cuts and modifies so that all proteins for virus replication are obtained.

7.4 GENE REGULATION

While essentially all cells of a multicellular organism contain the same genes, it is obvious that not all cells execute the same functions. The key to this variability is the coordinated expression of different sets of cell-type-specific genes at the appropriate times. But even unicellular organisms do not continuously transcribe

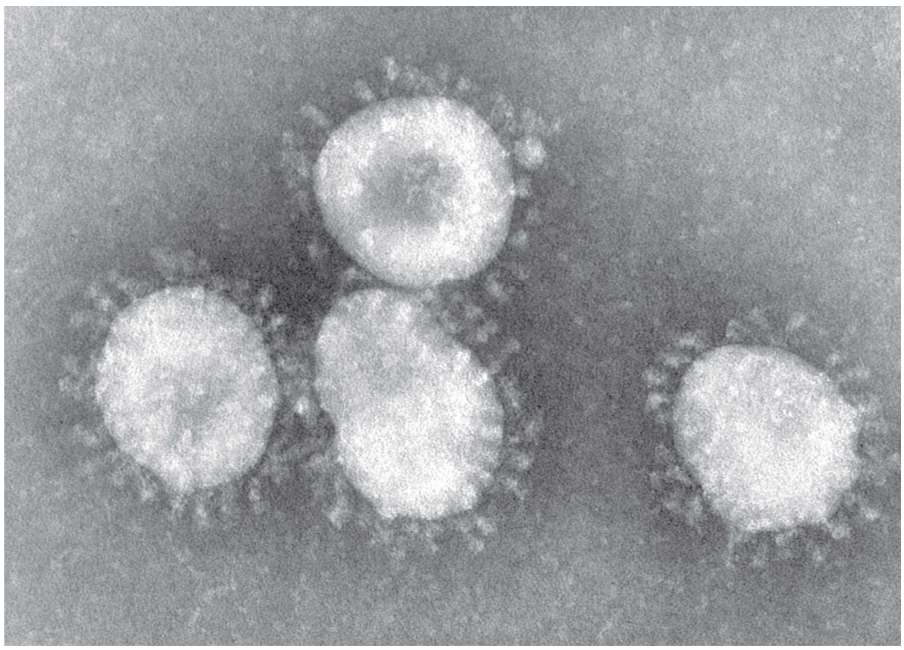


Figure 7.20 Electron micrograph of an RNA-containing coronavirus. The virus causes severe acute respiratory syndrome (SARS), a disease that rapidly spread around the world in 2003. Each virus is about 100 nm in diameter. (From Murray CL, Oha TS & Rice CM. Keeping track of viruses. In Microbial Forensics [B Budowle, SE Schutzer, RG Breeze, et al. eds], pp. 137–153. Academic Press, 2011. With permission from Elsevier.)

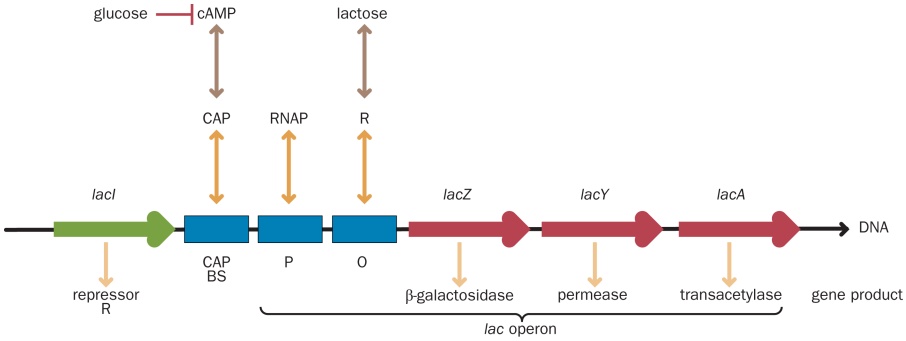


Figure 7.21 Diagram of the structure and function of the lac operon. The coding genes lead to the production of a repressor and three enzymes. Small molecules such as glucose and lactose affect expression. See the text and Table 6.1 for details.

all genes and instead carefully regulate expression to satisfy the current demands for metabolites and for the enzymes that are needed for their production. Because most systems are simpler in microorganisms, very many investigations characterizing the expression of genes have targeted model species such as the bacterium *Escherichia coli* and the baker's yeast *Saccharomyces cerevisiae*, and a vast body of information has been assembled. Indeed, many of the fundamental concepts of gene regulation can be learned best from simpler organisms, as long as one keeps in mind that higher, multicellular organisms use additional means of control and regulation.

A major breakthrough in our understanding of the regulation of gene expression was the discovery of **operons** in bacteria in the 1950s (Figure 7.21). According to the original concept, each operon consisted of a defined stretch of DNA that contained at its beginning at least one regulatory gene that allowed the bacterium to control the expression of all other genes in the operon [57]. This view is still basically true but has been slightly modified in recent years, because it turned out that an operon does not necessarily contain a regulatory gene. Instead, the promoter region of an operon generally has multiple *cis*-regulatory motifs that can bind with either **repressors** or **inducers**.

The regulation of an operon can occur through repressors or inducers. Repressors are DNA-binding proteins that prevent the transcription of genes. These repressors are themselves the products of genes, and this fact creates hierarchies of regulation that we will discuss later. Repressors often have binding sites for specific non-protein cofactors. If these are present, the repressor is effective in preventing transcription; if they are absent, the repressor dissociates from the operator site, and the genes in the operon are transcribed. Inducers are molecules that initiate gene expression. An example is the sugar lactose, which functions as an inducer for the so-called *lac* operon (see next section), which is responsible for lactose uptake and utilization. Lactose inactivates the repressor and thereby induces transcription. Other examples include certain antibiotics, which can induce the expression of genes that make the organism resistant [58].

7.4.1 The *lac* Operon

It is illustrative to study the unit structure of an operon within the genome of an organism in more detail. The first well-characterized example of this structure was the *lac* operon in *E. coli* [59]. Its discovery was a true breakthrough in genetics and molecular biology, and François Jacob, André Michel Lwoff, and Jacques Monod were awarded the 1965 Nobel Prize in Physiology or Medicine for their discoveries associated with this and other operons.

The *lac* operon contains three structural genes that code for enzymes associated with the uptake and utilization of the disaccharide (double-sugar) lactose, which chemically is a β-galactoside and prevalent in mammalian milk. These enzymes are β-galactoside permease, which is a transport protein responsible for the uptake of lactose from the medium (gene *22acy*), galactose transacetylase, which transfers an acetyl group from acetyl coenzyme A (acetyl-CoA) to β-galactosides (gene *lacA*), and β-galactosidase, an enzyme that converts lactose into the simpler sugars glucose and galactose (gene *lacZ*). In addition to these

TABLE 7.1: FUNCTION OF THE *LAC* OPERON UNDER DIFFERENT ENVIRONMENTAL CONDITIONS

Lactose	*Glucose*	*Repressor*	*CAP BS*	*Promoter*	*Operator*	*Transcription*
⇓	⇓	U	B	F	R	off
⇑	⇓	I	B	B	F	on
⇑	⇑	I	F	F	F	off
⇓	⇑	U	F	BF	R	off

Key: Arrows respectively indicate high (⇑) and low (⇓) concentrations in the environment. I, induced; U, uninduced; B, bound (in the case of the CAP binding site, CAP BS, by CAP facilitated by cAMP, whose concentration is inversely related to the glucose concentration in the medium; in the case of the promoter P by RNA polymerase); F, free of binding partner; R, repressed. Transcription occurs only under high-lactose and low-glucose conditions.

structural genes, the *lac* operon contains a promoter region P and an operator region O (Figure 7.21). Also of great importance are a regulator gene *lacI* and a binding region for the catabolite activator protein (CAP). The names of these components are mainly historical.

The bacterium only transcribes the structural genes if their products are needed. The cell manages the decision of whether to transcribe or not to transcribe with a multicomponent regulatory mechanism. If lactose is available in the environment, it serves as an inducer. It binds to the repressor, which is a protein that is coded for by a regulatory gene. In the case of the *lac* operon, the regulatory gene is directly adjacent to the operon, but in other cases it can be located quite distant from it. The repressor is normally bound to the operator region. However, binding of the inducer to the repressor causes the repressor to dissociate from the operator region. Without the repressor bound to the operator, the enzyme RNA polymerase, which synthesizes RNA, can in principle attach to the promoter region and transcribe the structural genes into mRNA. However, the promoter region often also contains response elements in the form of short DNA sequences that permit the binding of transcription factors under the appropriate conditions. Transcription factors recruit RNA polymerase and are of utmost importance for the regulation of gene expression. They will be discussed later in more detail.

In the case of the *lac* operon, the response elements react indirectly to high concentrations of glucose in the medium: glucose can repress the transcription of the *lac* genes. This repression is advantageous because the bacterium saves energy by consuming the more easily digestible glucose rather than lactose. This repression is accomplished in the following fashion. As a consequence of complex metabolic and signaling mechanisms, the cell responds to environmental glucose levels by regulating its internal level of cyclic adenosine monophosphate (cAMP). For low glucose levels, cAMP is high, and for high glucose levels, cAMP is low. The signaling molecule cAMP binds to CAP, which attaches to a specific binding site right next to the promoter region (CAP BS in Figure 7.21). This binding permits the attachment of RNA polymerase to the promoter and transcription of the structural *lac* genes, if lactose is present. By contrast, if glucose in the medium is high, the bacterium does not bother with lactose and waste energy synthesizing enzymes and rather uses the glucose. The high glucose concentration results in low cAMP, RNA polymerase does not attach to the promoter region, and the organism does not transcribe the *lac* genes. Taken together, the system expresses the structural genes only if lactose is available but glucose is not (**Table 7.1**).

Several groups have developed detailed dynamic models of the *lac* operon (e.g., [60–63]). Each model focuses on a particular question, is correspondingly structured, and has a different level of complexity, which again shows that biological questions and data determine the type, structure, and implementation of a model (Chapter 2).

7.4.2 Modes of Regulation

In general, the regulation of an operon may occur through induction or repression, and it may be positive or negative, so that there are four base forms of controlling the initiation of gene transcription.

Positively inducible operons are controlled by activator proteins, which under normal conditions do not bind to the regulatory DNA region. Transcription is initiated only if a specific inducer binds to the activator. The inducer changes the **conformation** (three-dimensional shape; see Chapter 8) of the activator, which allows its interaction with the operator and triggers transcription.

Positively repressible operons are also controlled by activator proteins, but these are normally bound to the regulatory DNA region. When a co-repressor protein binds to the activator, the activator no longer binds to the DNA and transcription is stopped.

Negatively inducible operons are controlled by repressor proteins, which are normally bound to the operator of the operon and thereby prevent transcription. Specific inducers can bind to repressors, change their conformation so that binding to DNA is no longer effective, and thus trigger transcription.

Negatively repressible operons are normally transcribed. They have binding sites for repressor proteins, which, however, only prevent transcription if specific co-repressors are present. Binding of the repressor and co-repressor leads to attachment to the operator and prevents transcription.

The type of regulation found in each particular case appeared to be random at first but eventually a careful analysis led to clear design principles dictating the mode of control (see also Chapter 6). The discovery and interpretation of these design principles resulted in the *demand theory of gene regulation*, according to which there are strict rules for when a particular mode of regulation is beneficial and therefore evolutionarily advantageous [64]. As one example, a repressor mode is advantageous if the regulated gene codes for a function that is in low demand within the organism's natural environment. By contrast, if the function is in high demand, the corresponding gene is under activator control.

Much of what we know about operons and their functions was determined through uncounted laboratory experiments, beginning even before the work of Jacob, Lwoff, and Monod [59]. By contrast, much of what we are now beginning to understand about the higher-order organization of bacterial genomes has been deduced with the help of sophisticated computational approaches. These approaches fall mostly within the realm of bioinformatics; the interested reader is encouraged to consult [37].

7.4.3 Transcription Factors

The most prominent regulators of gene expression are transcription factors (TFs), which include the repressors and activators we discussed before and which must attach to a binding site in the vicinity of the promoter region to execute their function. TFs may up- or down-regulate the operon; some dual-regulator TFs can do both, depending on the conditions. A TF is often activated through phosphorylation by a kinase, which can "sense" specific changes in the intra- or extracellular environment [65]. The kinase and the TF therefore constitute a two-component signaling system that permits the cell to respond adequately to changes in the environment. Chapter 10 discusses such systems in greater detail.

Typical bacterial genomes contain relatively few genes for TFs in comparison with their number of operons, and while some genes can be expressed without TFs, the mismatch between the numbers of TFs and operons has led to the inference that at least some TFs must be responsible for several operons [65]. The collection of all operons under control of the same TF is called the regulon of the TF. Because an operon may be regulated by several TFs, they may belong to different regulons as well. New evidence suggests that bacteria tend to keep operons encoding the same biological pathway—or even systems of related pathways—in nearby genomic locations [66].

The prediction of TF binding sites and regulons from sequence information is a difficult problem, but many algorithms have been developed for this purpose, and numerous databases are available that contain information on known TF binding sites and regulons (see, for example, [65, 67]). In addition to regulons, bioinformaticians have discussed über-operons (where *über* is German for "above"), which are sets of genes or operons that are regulated together even after many rearrangements of the DNA of the organism [68]. Other concepts include stimulons [69], which are sets of genes that respond to the same stimulus and modulons, which are regulons associated with several pathways or functions that are under the control of the same regulator [70].

Responses to environmental stresses often require more than the up- or down-regulation of a single operon. In fact, a complex response to a stress such as starvation may consist of changes in the expression of dozens of genes. The coordination of such responses therefore mandates means for co-regulating genes in different locations of the chromosome. The premier mechanism for this coordination is a hierarchy of TFs. Conceptually, such a hierarchy is not difficult to understand. Let us suppose that six operons $O_1, \ldots, O_6$ need to be called up for a proper response and that they are under the control of one transcription factor each, which we just call $TF_1, \ldots, TF_6$ for simplicity. These TFs are proteins, which are coded for by their own genes $G_1, \ldots, G_6$. If these genes are under the control of a TF at the next level of the hierarchy, say TF_A, then a single manipulation of TF_A will result in expression changes of all six target operons.

A complex example of this hierarchy is the control of processes converting sugars into the amino acids lysine and glutamate in *Corynebacterium glutamicum* [71] (**Figure 7.22**). Another impressive example is the gene regulatory network controlling metabolism in *E. coli*, which consists of about 100 TF genes that control about 40 individual regulatory modules. These modules can be triggered by various external conditions and organize proper responses and control the expression of about 500 metabolic genes that code for enzymes [72, 73].

The hierarchical and context-specific mode of TF control is not restricted to bacteria but can be found throughout evolution. An intriguing example is the analysis of transcriptional networks that regulate transitions between physiological and pathological states. For instance, the activation of genes in malignant brain tumors was found to be governed by about 50 TFs, of which five regulate three-quarters of the brain tumor genes as activators and one as repressor. Two TFs serve as master regulators of the system [74]. Algorithms have been developed to study these types of hierarchical TF networks [75].

The responses of TFs to environmental stimuli are typically achieved through small molecules that ultimately act as inducers for the TF genes. For example, a particular sphingolipid in yeast, which otherwise has no recognized function, was found to regulate several dozen genes associated with the response to heat stress by influencing 11 TFs positively and 2 negatively [3] (also see Chapter 11). Moreover, groups of sphingolipids were associated with the expression of clusters of genes with coherently related functions.

The alteration of TFs in response to environmental stimuli is not a one-time, all-or-nothing event. Instead, TFs can be altered differentially with respect both to magnitude and to time periods following a stimulus. For instance, in the sphingolipid-induced change in TFs, clear time domains for different groups were distinguishable, with some groups being turned on and others turned off about 10–15 minutes after the beginning of heat stress [3].

Numerous other mechanisms can regulate the activity of TFs; for a detailed description see [76]. Cells can sequester the regulator of some physiological function if this function is not needed at the time. For instance, if a sugar like maltose is not available in the medium, the regulator controlling the utilization of maltose is disabled. Mechanisms such as phosphorylation can control the activity of TFs spatially and temporally. Similarly, it is possible to fine-tune the cellular concentration of a TF through its expression and degradation. Finally, a TF or regulator may be translocated within the cell, for instance from the cytosol to the membrane in prokaryotes or between the nucleus and the cytosol in eukaryotes, thereby changing its activity state.

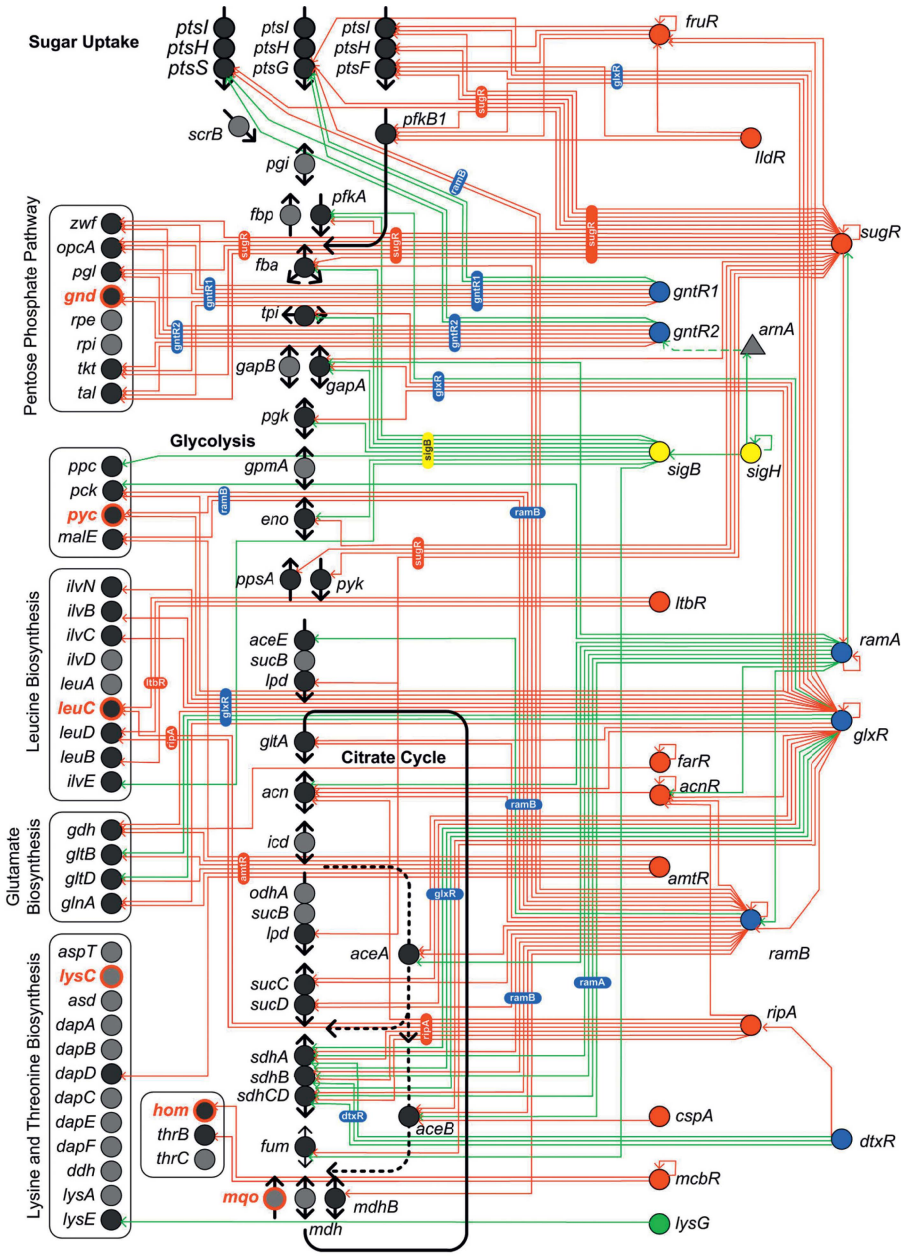


Figure 7.22 Transcription is often hierarchically controlled. Shown here is the hierarchy of transcription regulators involved in the conversion of glucose, fructose, and sucrose into the amino acids lysine and glutamate in *C. glutamicum*. Pertinent metabolic pathways are indicated as black arrows with corresponding gene names. (From Brinkrolf K, Schröder J, Pühler A & Tauch A. *J Biotechnol*. 149 [2010] 173–182. With permission from Elsevier.)

7.5 MATHEMATICAL MODELS OF GENE EXPRESSION AND GENE REGULATION

7.5.1 Generic Models

Two different yet straightforward approaches for modeling gene expression are the use of Boolean networks (see Chapter 3) or differential equations (Chapter 5). The former were originally proposed by Kauffman et al. [77] and used in numerous applications (*e.g.*, [78, 79]). In this approach, the expression level (state) of a gene is either off or on, and this state develops over time as a Boolean function of some or all other genes. Shmulevich *et al.* [80] illustrated the usefulness of this approach with the example of genes controlling the retinoblastoma (Rb) protein, which is a proto-oncogenic tumor suppressor protein that is dysfunctional in several major cancers. The activity of Rb depends on cyclin E and cyclin-dependent kinase 2 (cdk2) both being active, whereas p21/WAF1 must be inactive for Rb to be phosphorylated and become active itself. Furthermore, cdk2 is only active if cdk7 and

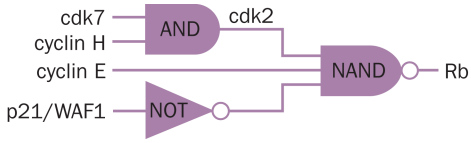


Figure 7.23 Combination of gates in Figure 7.24 controlling the state of Rb. The control mechanism includes an AND gate, a NOT gate, and a NAND (not AND) gate. (Adapted from [77].)

A	B
0	1
1	0

A	B	C
0	0	1
0	1	1
1	0	1
1	1	0

A	B	C	D
0	0	0	1
0	0	1	1
0	1	0	1
0	1	1	1
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	0

Figure 7.24 NOT gate and two-input and three-input NAND gates. The outputs in all three cases are inactive only if all inputs are active.

cyclin H are both active. These dependencies can be formulated as set of Boolean gates: cdk7 and cyclin H form an AND gate activating cdk2, p21/WAF1 is involved as a NOT gate, and a NAND (not AND) gate combines the inputs from cdk2, cyclin E, and p21/WAF1 (**Figure 7.23**).

The diagram contains two types of gates that we did not discuss in Chapter 3. In the NOT gate, the activity status of the output is the opposite to that in input. The typical two-input NAND (not AND) gate yields an inactive output if both inputs are active; otherwise, it is active. For three inputs, the Boolean output is computed in an analogous manner: It is only inactive if all inputs are active (**Figure 7.24**).

While Boolean tables are intuitive, they are inconvenient for simulations of large assemblies of gates, because the required *if-then* statements are computationally expensive. As an alternative, one can define explicit functions producing the same output from given inputs as the gates, assuming that all inputs and outputs are binary with values of 0 or 1. For instance,

NOT gate: $B = 1 - A$

NAND gate: $C = 1 - A \cdot B$

The exercises discuss other gates.

As an extension of the Boolean approach, Shmulevich *et al.* [80] proposed to replace the simple on and off states with probabilities between 0 and 1 for each state, which makes the analysis more realistic, but also much more complicated.

In typical Boolean models, the roles of TFs and their interactions are only implicit, although they are crucial, because genes do not interact directly but by means of their products, including proteins that serve as TFs or their modulators and small signaling molecules, such as calcium, cAMP or sphingolipids. Large-scale gene interaction networks of thousands of genes and their interactions have been reconstructed with experimental and computational means for model species such as yeast and *E. coli* [81, 82].

A powerful approach to understanding the structure of gene regulatory networks is the use of statistical techniques—including Bayesian methods that we discussed in Chapter 3—and their extensions [75, 83, 84]. Detailed reviews are provided in [85–87]. Furthermore, algebraic functions have been devised that condense complex Boolean networks into strings of polynomials, allowing for theoretical analysis of network properties such as *canalization*, the tendency for a system to gravitate towards a stable state from a broad range of initial conditions regardless of perturbations [88, 89].

The most prevalent alternative to a Boolean model is the use of ODEs. An early example is the model of Chen et al. [90], which is based on a simplified diagram of the Central Dogma (Figures 7.1 and 7.25) and was formulated with the equations

$$\frac{d\mathbf{r}}{dt} = f(\mathbf{p}) - V\,\mathbf{r}$$

$$\frac{d\mathbf{p}}{dt} = L\,\mathbf{r} - U\,\mathbf{p}\,. \qquad (7.1)$$

The diagram and equations look rather simplistic, but one needs to be aware that the variables are vectors and potentially encompass all n genes in a genome. Thus, $\mathbf{r}$ and $\mathbf{p}$ are n-dimensional vectors of mRNA and protein concentrations, respectively, $f(\mathbf{p})$ are transcription functions in the form of vector polynomials, L is an $n \times n$ matrix of translational constants, and V and U are $n \times n$ diagonal matrices of degradation rates for mRNAs and proteins. Other feedback, for instance, from mRNAs on gene expression is ignored, since such effects are subsumed in the protein feedback indicated. It was also assumed that the translation mechanism was relatively stable during each mathematical experiment, so the feedback from proteins to mRNAs had no effect. Chen and colleagues investigated this model assuming that $f(\mathbf{p})$ was a linear function, potentially with an included delay. To some degree, linear ODEs of this type can be employed to infer gene regulatory networks from data [91].

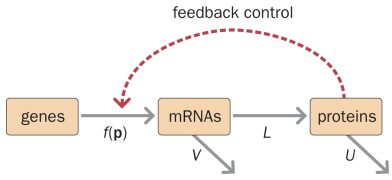


Figure 7.25 Diagram proposed by Chen et al. [90] for modelling gene expression. The model potentially covers all genes in a genome.

Nonlinear ODEs are the method of choice for gene expression modeling if mRNAs, transcription factors, and modulators of transcription are to be included explicitly. Setting up these models from information in the biological literature requires some care. Namely, gene regulatory networks are often represented with genes as nodes and gene-gene interactions as edges, as it is shown in **Figure 7.26**. This representation seems to depict how one gene (or gene transcript) affects another, but it obscures the design of dynamic models, because gene-gene interactions are mediated through transcription factors and other modulators. In particular, these representations do not indicate whether, for instance, a process reduces the transcription into mRNA or hastens the removal of an mRNA. A clearer representation for the two-gene regulatory system in Figure 7.26B uses the instructions from Chapter 2. It is shown in **Figure 7.27** and immediately suggests how to design a corresponding model. For example, equations in power-law format (7.2; Chapter 5) can directly be obtained from this diagram, and the theory behind these functions tells us that all parameter values are positive except for g_2:

$$\begin{aligned}
m\dot{R}NA_1 &= \alpha_1 P_1^{g_1} P_2^{g_2} - \beta_1 mRNA_1^{h_1} \\
m\dot{R}NA_2 &= \alpha_2 P_1^{g_3} - \beta_2 mRNA_2^{h_2} \\
\dot{P}_1 &= \alpha_3 mRNA_1^{g_4} - \beta_3 P_1^{h_3} \\
\dot{P}_2 &= \alpha_4 mRNA_2^{g_5} - \beta_4 P_2^{h_4}
\end{aligned} \qquad (7.2)$$

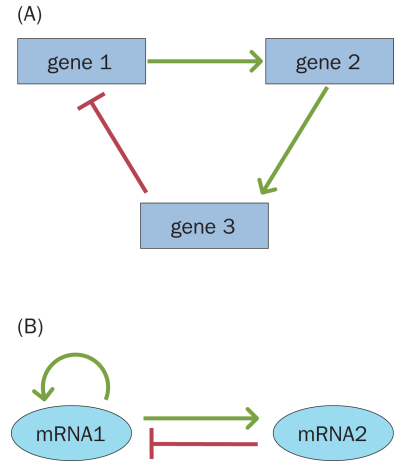


Figure 7.26 Schematic representations of interactions between (A) genes or (B) transcripts. While intuitive, these types of diagrams are not well suited for the design of mathematical models. See instead Figure 7.27.

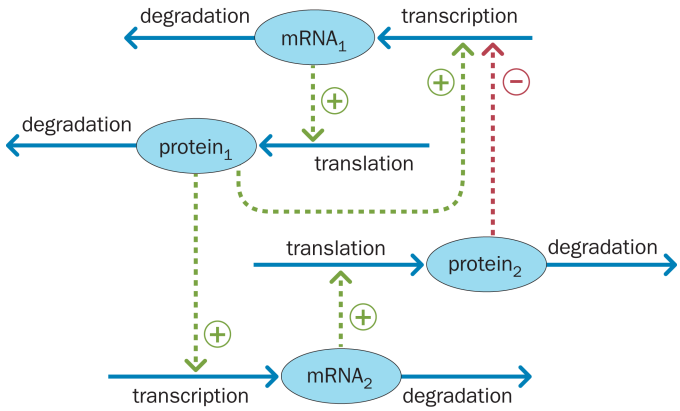
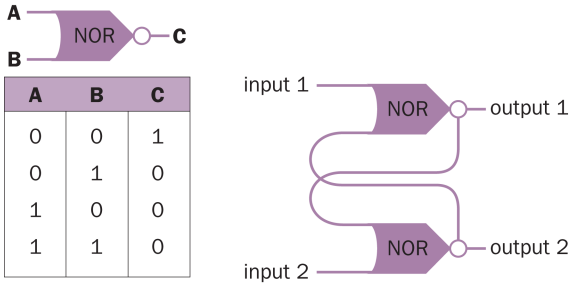


Figure 7.27 A representation of the diagram in Figure 7.26B that is better suited for designing a mathematical model. This representation appears to be more complicated, but it makes the processes involved explicit and immediately prescribes the set-up of model equations such as (7.2).



A	B	C
0	0	1
0	1	0
1	0	0
1	1	0

Figure 7.28 NOR (not OR) gate and two-input-two-output model for a genetic switch [99]. The outputs feed into the opposite gates as inputs.

In the reverse mode, ODE models have been used to infer gene interaction networks from expression data that had been recorded as time series (see, for example, [92, 93]). An intrinsic feature of ODEs is their use of rates, which are valid for large assemblies of molecules. However, if only a few molecules are involved, they may become problematic, and stochastic methods may have to be used [94–96].

7.5.2 Models of Genetic Switches

At first glance, gene expression is either on or off for a while, and several authors have studied this switching between on and off states with different methods (e.g., [94, 97, 98]). Complicating this switching, it has been observed that it frequently has a stochastic component (see [63] and references therein).

The switching itself is typically modeled with a bistable system, which again may be formulated as a Boolean system or an ODE model. Cherry and Adler [99] compared the two approaches side-by-side. The Boolean model (**Figure 7.28**) contains two NOR (not OR) gates, in which the output is active if both inputs are inactive. Suppose Input 1 and Input 2 are 0. For Output 1 to be active, Output 2 must be inactive. By the same token, Output 2 can only be active if Output 1 is inactive. Suppose now that initially Input 1 = Input 2 = Output 2 = 0 and that Output 1 = 1, but that some external signal activates Input 1. Now the inputs to NOR gate 1 turn from (0, 0) to (1, 0) and Output 1 turns off. With Output 1 = 1 and Input 2 = 0, Output 2 turns on. Thus, the system has two stable states: (Output 1, Output 2) = (1, 0) or (0, 1) and the system can switch between the two in response to the activation of one of the Inputs. For completeness, suppose again that Input 1 = Input 2 = Output 2 = 0 and that Output 1 = 1 but that both inputs are activated. In this case, both outputs turn off.

Cherry and Adler [99] then formulated ODEs that serve the same purpose of switching between two stable states. The equations are

$$\dot{X} = \frac{k_1}{1+Y^n} - \mu_1 X,$$

$$\dot{Y} = \frac{k_2}{1+X^n} - \mu_2 Y. \tag{7.3}$$

Here, k_1, k_2, μ_1, and μ_2 are positive rates and $n > 1$. Suppose we set $k_1 = 6$, $k_2 = 8$, $\mu_1 = 2$, $\mu_2 = 4$, and $n = 4$. It is straightforward to perform simulations. For instance, starting with $(X(0), Y(0)) = (0.1, 1)$, the system quickly reaches the steady state $(X_{ss1}, Y_{ss1}) \approx (0.1771, 1.9980)$. Slight changes in initial values lead to the same steady state, and eigenvalue analysis (Chapter 5) shows that this steady state is stable. However, if we start with $(X(0), Y(0)) = (1, 0.1)$, the system converges to $(X_{ss2}, Y_{ss2}) \approx (3.000, 0.0244)$. Thus, in one case, X is large and Y is small, and in the other case the opposite is true: The system has two distinct stable steady states. What determines which one will be approached? The answer is: it is a matter of the initial values (**Figure 7.29**).

We will discuss these types of bistable systems in more detail in Chapter 10 but suffice it to say here: if there are two stable steady states, there must be something in between, and most likely it is an unstable steady state. Unfortunately, trying to

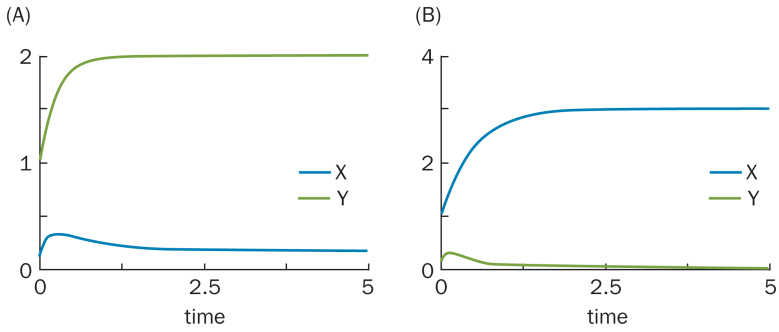


Figure 7.29 Simulation results from the model in (7.3). (A) Starting with ($X(0)$, $Y(0)$) = (0.1, 1), the system reaches the steady state (Xss_1, Yss_1) ≈ (0.1771, 1.9980). (B) Starting with ($X(0)$, $Y(0)$) = (1, 0.1), the system converges to (Xss_2, Yss_2) ≈ (3.000, 0.0244).

compute this steady state algebraically is difficult, due to the format of the equations. Moreover, simulations will not converge to this state because it is unstable, unless one starts right at this point. Trial and error offer some help. Fixing one initial value, e.g., $X(0) = 0.1$ and systematically varying $Y(0)$ between 0.1 and 1 shows that $SS1$ is reached for small $Y(0)$ and $SS2$ for larger ones and that the switch occurs between roughly 0.21 and 0.22. It turns out that the elusive unstable steady state is (X_{ss3}, Y_{ss3}) ≈ (0.8827, 0.1244). But even starting a tiny bit too high or too low sends the system to one of the other two steady states. Thus, the system has three steady states, of which two are stable. Any external system affecting X or Y can push the system from one to the other state, as in the case of the Boolean system. Many authors have studied these types of switching systems; examples include [91, 98, 100, 101].

7.5.3 Models of Genetic Response Circuits

At the center of a slightly more complicated example is the aryl-hydrocarbon receptor (AhR), which senses foreign compounds and danger signals produced by parasites that had invaded the body [102–104]. AhR responds to these signals by ultimately causing the upregulation of various genes, including those coding for so-called cytochrome P450 enzymes, which are instrumental for metabolizing foreign, toxic compounds [105, 106]. The mechanism, slightly simplified, is sketched in **Figure 7.30** [107–110]. The response is triggered when a xenobiotic ligand (L) binds to AhR; the term *xenobiotic* refers to a "foreign biological compound." The binding activates the receptor, and L-AhR subsequently forms a complex with

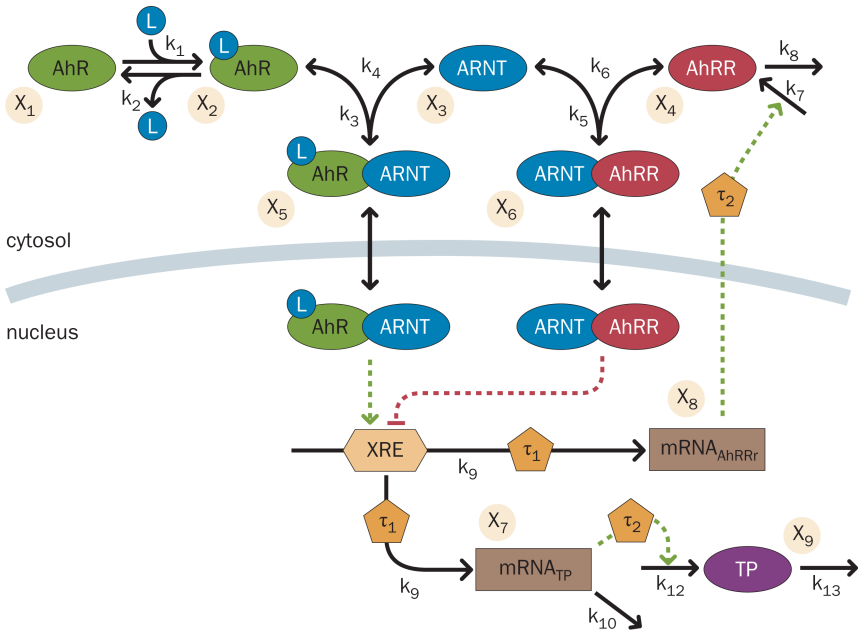


Figure 7.30 Diagram of the mechanism with which AhR responds to a xenobiotic ligand L. L binds to AhR, thereby activating it. The complex binds to the nuclear translocator ARNT, and the resulting complex acts as a transcription factor of genes whose promoter regions contain the xenobiotic response element XRE. These genes code for a variety of target proteins (TP) including, notably, the AhR repressor AhRR. AhRR competes with AhR for ARNT, and the complex inhibits gene expression. Transcription and translation incur delays (τ_1, τ_2). Adapted from [105].

ARNT, the "AhR nuclear translocator," which enables L-AhR to enter the cell nucleus. There, the L-AhR-ARNT complex acts as a transcription factor that binds to a specific DNA sequence—the so-called xenobiotic response element XRE—and initiates the transcription of genes coding for target genes that are needed to respond to the foreign substance and which are collectively called TP in the diagram. Outside these response genes, one of the induced genes codes for the protein AhRR, where the second R refers to "repressor." AhRR competes with L-AhR for ARNT, and the ARNT-AhRR complex represses the expression of the target genes under the control of AhR-ARNT. Taken together, the mechanism triggers the expression of target genes but also inhibits it. How can that make sense? Let's set up a dynamic model to investigate this apparent inconsistency.

Note that the diagram contains pentagons with letters τ_1 and τ_2. These signify delays incurred during transcription and translation. For our simplified ODE analysis, we ignore these time delays as well as stochasticity, which always affects transcription-translation processes [111]. Accounting for delays and/or stochasticity would require different modeling formats (cf. [110, 112]. We set up the ODE model of the sensor system in arguably the simplest format, namely with mass-action equations, which seems appropriate because most of the processes represent the formation or disassociation of complexes. We augment these forms with the repression by the complex ARNT-AhRR)X_6 that is easily modeled as a power-law with negative kinetic order (Chapter 5). To avoid mathematical problems with 0 taken to a negative power, we use $(X_6 + 1)$ instead of X_6. The equations, with rate constants *ki* as shown in Figure 7.30, read

$$
\begin{aligned}
\dot{X}_1 &= -k_1 L X_1 + k_2 X_2 \\
\dot{X}_2 &= k_1 L X_1 - k_2 X_2 - k_3 X_2 X_3 + k_4 X_5 \\
\dot{X}_3 &= -k_3 X_2 X_3 + k_4 X_5 - k_5 X_3 X_4 + k_6 X_6 \\
\dot{X}_4 &= -k_5 X_3 X_4 + k_6 X_6 + k_7 X_8 - k_8 X_4 \\
\dot{X}_5 &= k_3 X_2 X_3 - k_4 X_5 \\
\dot{X}_6 &= k_5 X_3 X_4 - k_6 X_6 \\
\dot{X}_7 &= k_9 X_5 (X_6 + 1)^g - k_{10} X_7 \\
\dot{X}_8 &= k_9 X_5 (X_6 + 1)^g - k_{11} X_8 \\
\dot{X}_9 &= k_{12} X_7 - k_{13} X_9
\end{aligned}
\tag{7.4}
$$

Representative initial conditions and parameter values are presented in **Table 7.2**. The initial conditions reflect the situation where no ligand is present ($L = 0$) and no complexes have therefore been formed yet. This situation constitutes a steady state; confirm this! With these numerical settings, it is easy to do simulations.

As a first test, we reset the ligand concentration to 2 at time $t = 5$. In response, X_1 (AhR) quickly decreases, while the complex X_2 increases (**Figure 7.31**). Almost immediately, X_3 (ARNT) drops in value as the complex X_5 is formed. Shortly after, the mRNAs X_7 and X_8 start to rise, and finally, the concentration of target protein

TABLE 7.2: INITIAL CONDITIONS AND PARAMETER VALUES FOR THE MODEL IN EQN. (7.4)

$X_1(0)$	$X_2(0)$	$X_3(0)$	$X_4(0)$	$X_5(0)$	$X_6(0)$	$X_7(0)$	$X_8(0)$	$X_9(0)$
10	0	8	0	0	0	0	0	0
L	k_1	k_2	k_3	k_4	k_5	k_6	k_7	k_8
0	0.5	0.25	0.1	0.25	2.5	1	1	0.1
k_9	k_{10}	k_{11}	k_{12}	k_{13}	g			
0.25	0.25	0.1	5	0.5	−2			

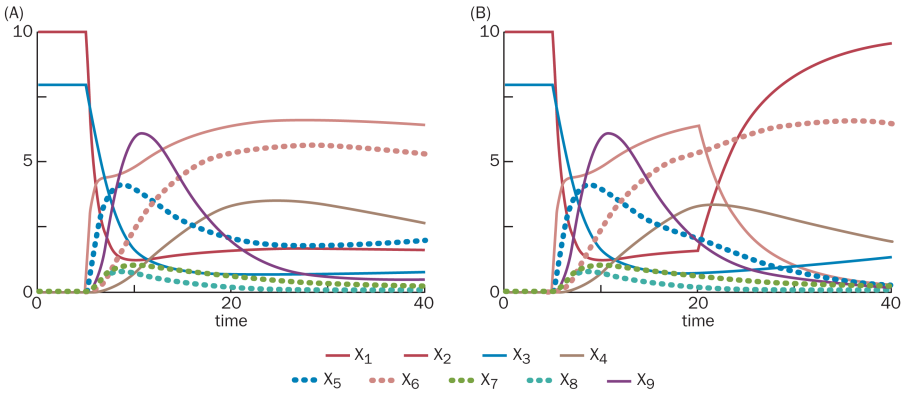


Figure 7.31 Output of simulations with the AhR model (Eqn. 7.4). On the left, the ligand concentration is set from 0 to 2 at time $t = 5$. On the right, the concentration is set back to 0 at time $t = 20$.

(X_9) rises and so does the repressor AhRR (X_4), although a bit more slowly. Interestingly, the target protein soon reaches a peak and then continues to fall. Why is that? The answer consists of three contributing factors. One is the competition by AhRR and the repression of transcription due to the newly formed complex ARNT-AhRR. The second aspect is a drop in the complex L-AhR-ARNT (X_5), which is due to some ARNT being tied up with AhRR. Finally, the system loses mass through the degradation of AhRR (k_8), loss of mRNAs (k_{10} and k_{11}) and the utilization of target protein (k_{13}). An exploration of the different contributions is left as an exercise. Eventually, the system appears to be approaching a new steady state, which is easy to confirm when the simulation is run for a longer time. This new balance is possible, because the system is "fed" with a constant supply of ligand and the synthesis of AhRR (k_7). If we stop the supply of ligand, for instance, by setting $L = 0$ at $t = 20$, mRNAs, the complexes, and the target proteins continue their decline toward zero, while X_1 and X_3 very slowly regain their initial values. Actually, there is a small residual of X_4 and X_6, due to k_7. Figure 7.31B only shows the beginning of this return but it takes until about $t = 1{,}000$ for the system to be close again to the initial settings.

At the beginning of this example, we had wondered how activation and repression of the same process could make sense. Now we have an answer: the delay between activation and repression constitutes a feedback system that triggers the expression of response genes and then turns itself off after a while. If the time delays in Figure 7.30 were more adequately taken into account, the time period of protein synthesis would be much longer [110].

7.6 SYNTHETIC BIOLOGY

Synthetic biology is a relatively recent field of endeavor that attempts to create new organisms from scratch or to modify existing microorganisms, plant cells, or mammalian cells toward some goal of interest, based on a rational design. The first generation of synthetic biology primarily manipulated genes and their expression in new ways and attempted to coerce microorganisms into producing organic compounds that they have never encountered before and that they did not need. In some sense, this idea was close to the goals of metabolic engineering (see Chapter 9). If the early successes in this subdiscipline of biotechnology [113] are any indication, we have every reason to expect that the field will explode over the next decades as we develop faster and more finely tuned means of inserting foreign DNA into other organisms, coax organoids into new shapes and with new functionality [114], and even move DNA between prokaryotes and eukaryotes [115–117]. In the process, researchers are learning to master the control of the timing and extent of translating this DNA into RNA, proteins, and ultimately desired physiological functions.

The main attraction of any gene-based strategy is that it allows rather precise alterations in the activities of key enzymes, which themselves are specific and very effective. Even complex metabolic conversions, which would require a long sequence of steps in conventional synthetic chemistry, can sometimes be catalyzed in a single enzymatic reaction [118]. Furthermore, it is becoming possible

to combine several enzymatic reactions, thereby eliminating the need for intermediate purification steps. Finally, if microorganisms can be coaxed into producing complex organic compounds, the products are usually enantiomerically pure and not, for example, in racemic mixtures of L- and D-forms, which often result from chemical synthesis and are very expensive to separate [119]. Even though many of the current methods might be outdated and replaced in a few years, it is instructive to read good reviews of what has been accomplished so far, such as [120–123].

The more we learn about the natural control of genes and pathways, the more targets for manipulation we will discover for the advancement of genetic and metabolic engineering. Barely known until rather recently, small interfering or **silencing** RNAs, **microRNAs** and other noncoding RNAs, as well as ribozymes that we discussed before [47], aptamers and **riboswitches**, promise a slew of new tools for controlling transcription and translation [20, 124–126] (see also **Box 7.1**), and inversion recombination elements such as *fim* and *hin* are even able to rearrange DNA sequences within the genome [127].

BOX 7.1: RIBOSWITCHES

Riboswitches are gene regulatory elements that are found as sections of 5′ untranslated regions of mRNAs. They are broadly distributed across bacteria and, for instance, account for the regulation of more than 2% of all genes in *Bacillus subtilis*, underscoring their importance in the control of cellular metabolism. The 5′ untranslated region of many mRNAs of genes involved in the metabolism and transport of purines contains a guanine-responsive riboswitch that directly binds guanine, hypoxanthine, or xanthine to terminate transcription. **Figure 1** shows the crystal structure of the purine-binding domain of the guanine riboswitch from the xpt-pbuX operon of *B. subtilis* bound to hypoxanthine, a prevalent metabolite in the bacterial purine salvage pathway. The structure reveals a complex RNA fold containing several phylogenetically conserved nucleotides that create a binding pocket that almost completely envelops the ligand. Hypoxanthine functions to stabilize this structure and to promote the formation of a downstream transcriptional terminator element, thereby providing a mechanism for directly repressing gene expression in response to an increase in intracellular concentrations of metabolite.

Figure 1 Structure of a riboswitch. The natural guanine-responsive riboswitch in *B. subtilis* is shown here in complex with the metabolite hypoxanthine (mRNA_metabolite_1u8d). (Courtesy of Ying Xu and Huiling Chen, University of Georgia.)

As genetic engineering techniques become more mature, it will be useful, indeed mandatory, to develop experimental protocols and standards, as well as to share information and materials through registries of biological components [128] and standardized formats called datasheets, which succinctly characterize the features of building blocks for synthetic biology, such as sensors, actuators, and other logical elements [129]. Important features to be documented include the purpose of the device module, along with its operational contexts, interfaces, and compatibility with other devices; quantitative measures of its functions, such as the characteristics of its input–output transfer function; properties of its dynamic behavior; and indicators of its reliability [130]. Standards are important because they characterize components in sufficient detail to permit predictions for larger systems composed of these components. Standards also allow abstracted representations, as we know them from electric wiring diagrams.

Along with design standards and the creation of standardized modular parts [128], future synthetic biology will use methods of computer-aided design (CAD) that have been tremendously successful in various engineering disciplines for a long time. For instance, platform-based design (PBD), a special case of CAD, has proven very useful in the design of embedded electronic systems [131], and one

might expect that PBD methods will support the de novo creation of robust, reliable, and reusable biological systems within given design constraints and will usher in procedures for automatically optimized designs, given the specifications of the intended system [130, 132].

The sky is the limit when it comes to creating novel gene circuits, and one should expect a steady increase in the number of ingenious inventions. For instance, James Collins's group created an *E. coli* strain that can count [133]. Christopher Voigt's and Andrew Ellington's laboratories engineered bacteria that were enabled to see light, detect edges between light and dark, and essentially act like a photographic film [134]. The genetic constructs consisted of a combination of a chimeric light-sensitive protein with components from the freshwater *cyanobacterium* (blue–green alga) *Synechocystis* and the bacterium *E. coli*, the LuxI biosynthetic enzyme from the marine bacterium *Vibrio fischeri*, which produces a cell–cell communication signal, a transcriptional repressor from bacteriophage λ, and a lacZ system producing black pigment. The implementation of the edge-detection algorithm consisted of three plasmid backbones carrying about 10,000 DNA base pairs. A good how-to review of strategies for engineering bacterial sensor systems and synthetic gene circuits can be found in [135].

In a different application, Voigt's laboratory developed a method for designing synthetic ribosome binding sites, which allowed them to control protein expression in a rational, predictable, and accurate fashion [136]. This type of control could have enormous implications for connecting gene circuits and for the targeted production of organic compounds in metabolic engineering.

In nature, bacteria communicate with each other and often exhibit coordinated, population-based behaviors through a process called **quorum sensing (QS)** (see Chapter 14). In this process, bacteria produce and release species-specific or universal chemical signaling molecules. Other bacteria of the same or another species sense these molecules and correspondingly regulate their gene expression in a manner that depends on the amount of a chemical signal and thus on the cell density in the environment. The result is a population response that occurs only if sufficiently many bacteria are on board. Natural examples of QS can be seen in the formation of **biofilms**, which exist everywhere from human teeth to sewage pipes, in bacterial pathogenicity, and in virulence. It seems possible, at least in principle, to use a detailed understanding of the QS process as a weapon against some of these undesired bacterial phenomena [137]. It also seems that QS could have great potential for the creation of novel microbial systems that react collectively as a population to stimuli of interest. For instance, bacterial population sensors might be used for the detection of environmental chemicals or pathogens or for the creation of new materials [138].

Many species of bacteria employ only one QS circuit, while others use two or more circuits for communication. As we learn more about the details of these circuits, more targets for possible disruption at the genetic or signaling level emerge. The reward for such interruption could be significant, because biofilms are often undesirable and many pathogenic species develop drug resistance by means of QS processes. A promising approach to developing interruption strategies and directing multicellular functioning is the re-creation of computational and molecular QS systems with methods of synthetic biology [138, 139]. One way to accomplish this task is by rewiring of the input–output relationships of a signaling pathway. For instance, exchanging the sensors for different signaling pathways permits targeted engineering of the input to a microbial system, while altering the molecule carrying the signal from the receptor to downstream effectors affects its output [120].

7.6.1 Artificial Gene Circuits

Artificial gene circuits [140–143] have become an important research topic within the field of synthetic biology. These circuits are created by inserting foreign genes into microorganisms and changing their transcription and/or translation in a controlled fashion. One of the first working examples was a genetic toggle switch [97], which was constructed such that the circuit permitted bistability and memory, as

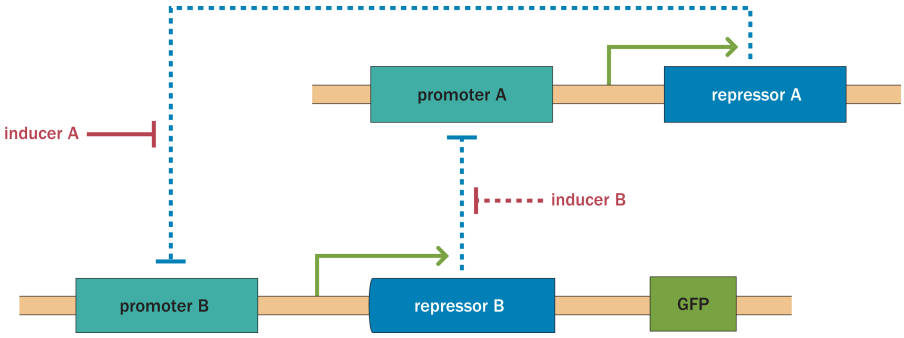


Figure 7.32 Schematic representation of a toggle switch [97]. Synthetic biology has advanced far enough to create gene circuits that exhibit responses as predicted by mathematical models.

we discussed earlier. Specifically, the system consisted of two genes that each coded for a transcriptional repressor of the other (**Figure 7.32**). An external inducer allowed the inhibition of one repressor, thereby moving the system to a stable state where one gene was being transcribed and the other repressed. The stability of this state also led to hysteresis (see Chapter 10), where the system remained in the particular on–off state without further exposure to the external inducer, until another perturbation forced the system into the second steady state. Gardner and collaborators [97] represented the experimental system with the same bistable model we discussed before in Eqn. (7.3).

Elowitz and Leibler [144] created in *E. coli* a low-copy-number plasmid encoding an oscillating gene circuit, called the **repressilator**. This system contained a compatible higher-copy-number reporter plasmid with a tetracycline-repressible promoter fused to a gene coding for green fluorescent protein (GFP). The system was stimulated with a transient pulse of IPTG, which is known to interfere with the repression by *LacI*. The repressilator circuit thus contains three gene modules. The first gene produced the repressor protein *LacI*, which inhibits the transcription of the second repressor gene, *tetR*, which was obtained from a tetracycline-resistance transposon and whose product inhibits the expression of the λ-phage gene *cI*. The product, CI, inhibits *lacI* expression, thereby creating a cyclical system (**Figure 7.33**). The oscillations were visible in single cells under a fluorescence microscope. They were sustained for a few cycles—but not synchronized—and varied quite a bit in frequency and amplitude, even between sibling cells.

Stripping away some details, Elowitz and Leibler [144] modeled the system with six ODEs, three for mRNAs and three for the corresponding proteins. Exercise

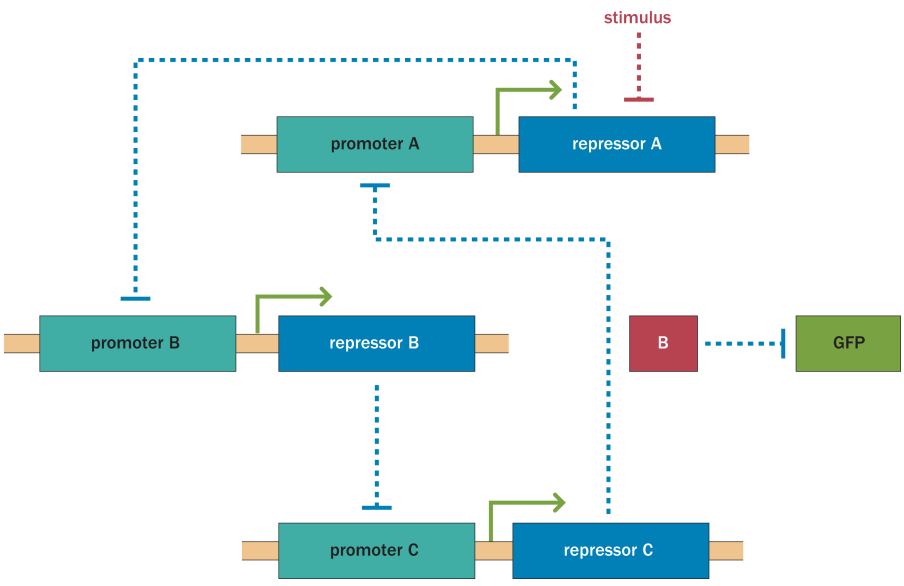


Figure 7.33 Schematic representation of the repressilator. This interesting gene circuit consists of three genes that inhibit each other's expression in a circular manner. The result is the ability to exhibit oscillatory expression patterns [144].

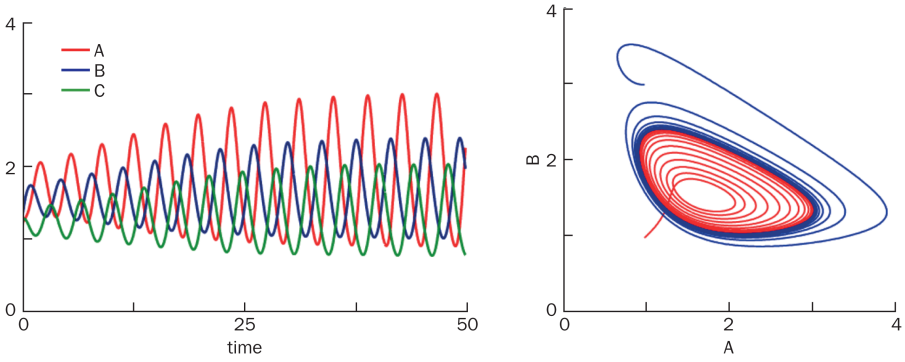


Figure 7.34 Dynamics of the repressilator (Eqn. 7.5) in the time domain and as phase-plane plot. Note that one of the trajectories in the latter starts outside the limit cycle and then spirals out, while the other one shows an initial curl. These features are due to the fact that the system has three variables, and the phase-plane plot is a projection onto the plane.

7.37 explores this system. For illustration purposes, a simplified system with the three equations suffices. It takes the form

$$\dot{A} = \frac{\alpha_A}{1+C^{\beta_C}} - s_A A$$

$$\dot{B} = \frac{\alpha_B}{1+A^{\beta_A}} - s_B B$$

$$\dot{C} = \frac{\alpha_C}{1+B^{\beta_B}} - s_C C \qquad (7.5)$$

Simulation of the system with parameter values $\alpha_A = \alpha_B = \alpha_C = 6$, $\beta_A = 2$, $\beta_B = 3$, $\beta_C = 4$, and $s_A = s_B = s_C = 1$ indeed reveals a stable limit-cycle oscillation (**Figure 7.34**). However, these oscillations can change for different parameter sets (see Exercises).

Combining properties of the toggle switch and the original oscillator, Atkinson and collaborators [141] improved on the simple repressilator with a more complex circuit design that led to predictable oscillation patters that were robust to perturbations and even synchronized in large populations. The design contained an activator and a repressor module (**Figure 7.35**). The activator module consisted of a *LacI*-repressible *glnA* promoter, which controlled the expression of the activator NRI. Because phosphorylated NRI in turn activates the *glnA* promoter, this module is auto-activated. The repressor module consisted of a phosphorylated NRI-activated *glnK* promoter fused to the gene coding for *LacI* expression. Thus, the output of the activator module was connected to the input of the repressor module. The researchers demonstrated that this circuitry reproducibly generated synchronous damped oscillations. More importantly, the behavior of the oscillator was consistent with sophisticated theoretical predictions. For instance, as predicted by the model analysis, the oscillation changed to a toggle switch when the

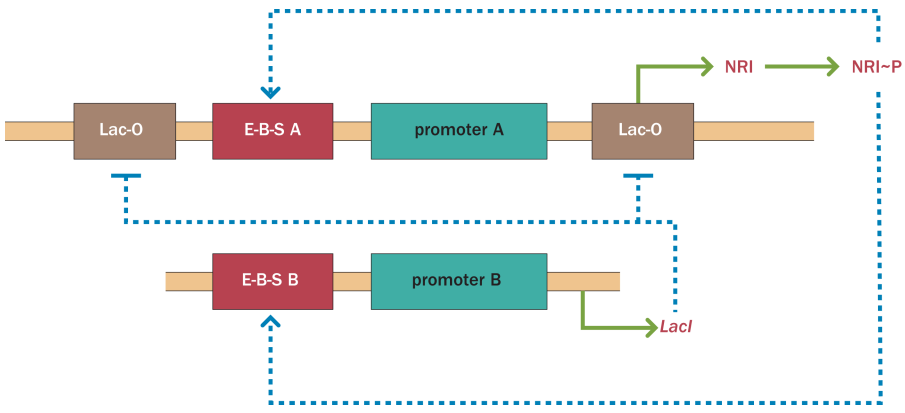


Figure 7.35 Schematic representation of a gene circuit that exhibits sustained, synchronized oscillations. Lac-O is a *LacI* operator site, and E-B-S A and E-B-S B are enhancer-binding sites. (Adapted from Atkinson MR, Savageau MA, Myers JT & Ninfa AJ. *Cell* 113 [2003] 597–607. With permission from Elsevier.)

enhancer-binding site of the repressor system was removed and the production of *LacI* thereby became constitutive.

7.6.2 Case Study

The laboratory of Jay Keasling at Berkeley has been at the forefront of synthetic biology since the inception of the field. One of the hallmark projects of the laboratory has been the production of an antimalarial drug, artemisinin, at a cost that allows developing countries to buy the drug for their malaria patients [145, 146]. Malaria has been a persistent problem for a long time. Hippocrates described the disease about 2,500 years ago, and yet even today nearly 500 million people in the tropics and subtropics become infected every year and nearly three million patients die [147].

Artemisinin is a natural secondary metabolite of the Chinese sweet wormwood *Artemisia annua* and has been used in Chinese treatments of malaria for a long time. It kills *Plasmodium falciparum*, the most prevalent malaria parasite, by releasing large doses of reactive oxygen species inside a patient's red blood cells. Annually, about 100 million treatment doses are being sold, but harvesting the drug from *A. annua* is not feasible at a cost that developing countries can afford [148]. Even in microbes, the total synthesis of artemisinin would be expensive and difficult, and the Keasling laboratory therefore set out to re-engineer the baker's yeast *Saccharomyces cerevisiae* to produce the immediate precursor of artemisinin, artemisinic acid, in sufficient quantities and at an acceptable price.

The biochemical context for this precursor is given by terpenoids, sesquiterpenoids, and isoprenoids. Sequencing of *A. annua* clones indicated that enzymes from the mevalonate and farnesyl pyrophosphate (FPP) pathways are involved in artemisinin biosynthesis [149]. FPP is found in many fungi and other organisms and can be converted into about 300 different structures. One of these, amorpha-4,11-diene, is the precursor of artemisinin.

Thus, the task became to re-engineer yeast toward three goals (**Figure 7.36**). First, it was necessary to increase the amount of FPP. This was accomplished by enhancing the activities of enzymes and thereby increasing the flux into the endogenous mevalonate pathway. The initial substrate of this pathway is acetyl-CoA, which is readily produced from simple sugars. The second goal was to introduce a gene coding for amorphadiene synthase, which converts FPP into amorpha-4,11-diene. Finally, it was necessary to recreate in yeast the three-step conversion of amorpha-4,11-diene to artemisinic acid, which in *A. annua* is accomplished with a cytochrome P450 mono-oxygenase and the corresponding redox partner NADPH:cytochrome P450 oxidoreductase. As mentioned earlier, FPP is the starting compound for many other products, and in order to channel FPP toward amorpha-4,11-diene, it was furthermore necessary to down-regulate the initial step of ergosterol biosynthesis from FPP, which was accomplished through reduced transcription of squalene synthetase, which converts FPP into squalene. The down-regulation was achieved through reduction of the expression of the corresponding gene *ERG9* by placing it under the control of the methionine-repressible PMET3 promoter and including a gain-of-function mutation in the gene *UCP2*, which regulates sterol production. It turned out that yeast transports the synthesized artemisinic acid out of the cell, which facilitated its harvesting and purification.

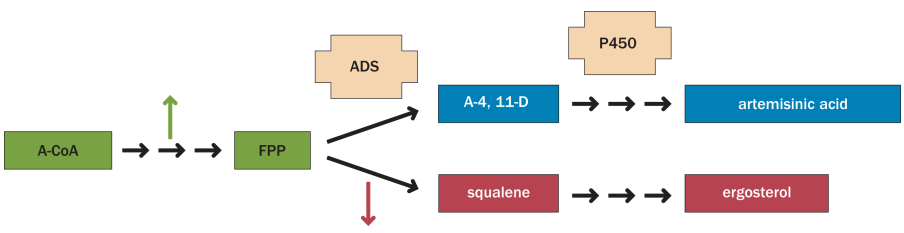


Figure 7.36 Simplified representation of the steps toward generating artemisinic acid in yeast. The task involves enhancing the flux of the mevalonate pathway from acetyl-CoA (A-CoA) to farnesyl pyrophosphate (FPP) (green arrow), down-regulating the flux from FPP toward squalene and ergosterol (red arrow), and introducing two new enzyme systems (pale orange): amorphadiene synthase (ADS), which converts FPP into amorpha-4,11-diene (A-4,11-D) and the complex (P450) of a specific cytochrome P450 mono-oxygenase and its NADPH-oxidoreductase partner.

With concerted efforts, the engineered yeast eventually became capable of producing artemisinic acid at a significantly higher specific productivity than *A. annua*. In fact, the yield of amorpha-4,11-diene was increased roughly 10,000 times, and it seems feasible to improve yield by another order of magnitude or more. It remains necessary to scale up the processes to an industrial level and to optimize yield further, in order to reduce the cost of artemisinin sufficiently. Unfortunately, recent reports indicate that the *Plasmodium* parasite is already beginning to show resistance, for instance, in Cambodia [150]. However, it is hoped that anti-malarial drug cocktails, as recommended by the World Health Organization, rather than monotherapy, will delay or even prevent resistance and permit the effective use of artemisinin in the future.

Similar to the development of drugs against diseases like malaria, geneticists have been using synthetic biology to create transgenic foods with added benefits. Specifically, almost 50% of the world population suffers from vitamin deficiency. As is to be expected, this problem occurs mostly in developing countries, where the typical cereal-based diet does not change from day to day. An early and very famous example of a transgenic crop was Golden Rice, a rice variety that was genetically engineered by the Swiss research group of Ingo Potrykus to synthesize β-carotene, which is a precursor of vitamin A [151]. The name of this transgenic strain reflects its golden color, which is due to the synthesized carotene. Just 5 years later, its successor, Golden Rice 2, generated over 20 times the amount of the original transgenic crop [152]. In a similar vein, the group of Paul Christou set out to create transgenic staple crops with increased vitamin content. They have succeeded in altering a South African corn variety containing 169 times the normal amount of β-carotene, six times the normal amount of vitamin C, and twice the normal amount of the B vitamin folate [153]. The Golden Rice Project [154] is dedicated to research efforts of this type.

7.7 OUTLOOK

Genomes have been and will continue to be a very exciting area of biological research. The developments in the field are often mind-boggling in their scope and speed, and yet one might expect that we have so far only scratched the surface. In the past, most gene studies have focused either on individual genes or on static gene regulatory networks. The field has now begun to move toward the roles of small RNAs—which comprise much more specific and sophisticated regulatory systems than anyone had expected—and toward other means of regulating and controlling gene expression throughout the genome so that the right gene is transcribed to the right degree, at the right time, and in the right location. The regulation and dynamics of these networks—and the localization and integration of their sub-networks—will without doubt be a cornerstone of experimental and computational systems and synthetic biology throughout the foreseeable future.

Exciting applications of gene research are synthetic biology and metabolic engineering (Chapter 9), which one could consider applications of systems biology. In the present early stages of synthetic biology, we are primarily mimicking natural systems. Soon, we might be able to create novel, fully functioning systems from basic components [128, 129]. Already, we have at our disposal a variety of means for controlling microbial function at the genomic level [121, 122], and we are beginning to hone our capabilities of fine-tuning expression at the RNA and protein levels [136, 155]. On a small scale, we have developed the base capabilities to design functional circuits based on predictions from mathematical models and implement them in actual microbial systems [123]. In metabolic engineering, the traditional trial-and-error approaches are being successfully supplanted with rational methods for altering existing genomic and metabolic designs and with the creation of new pathways and their control. What would have seemed to be a fantasy only a few decades ago is becoming reality. Yet, it is only the beginning. There is no doubt that the manipulation of genes and genomes is on a long and exciting journey whose destination we cannot even imagine.

7.8 EXERCISES

7.1 Identify and describe examples of overlapping genes.

7.2 Collect information on biological materials other than DNA that can be passed from one generation to the next.

7.3 Explore differences and similarities between DNA transposons and retrotransposons. Summarize your findings in a report.

7.4 GC base pairs contain three hydrogen bonds and AT pairs only two. Find out how much stronger GC base pairs are than AT base pairs.

7.5 Discuss in a report what phylogenetic trees are and how they are established.

7.6 Gregor Mendel is sometimes called the "Father of Genetics." Summarize his major accomplishments in a one-page report.

7.7 Use Gene Ontology or some other web resource to determine the function of gene Hs.376209.

7.8 Explain why a search for "Parkinson's disease" as a GO term fails, while searching for "Parkin" as a "gene or protein" leads to genes associated with the disease.

7.9 Compare genetic and epigenetic inheritance. Summarize your findings in a report.

7.10 Find out how cells attempt to ensure that genes are correctly translated into mRNA and proteins. Write a brief report.

7.11 Without searching on the Internet, speculate on how it is possible that only miRNAs can specifically target half of all human genes. Afterwards, check in the literature and the web how close your speculations were to reality.

7.12 Establish a list of diseases that are associated with RNA viruses.

7.13 Determine which cells in the human body do not contain the same genes as most other cells. Are there living human cells without genes?

7.14 Construct a Boolean rule for the expression of the structural genes of the *lac* operon. Then assign to lactose, glucose, and transcription each a value of 0 if they are absent or 1 if they are present. Test the outcomes from different combinations of inputs.

7.15 Consider the hypothetical situation of industrial waste remediation. In this example, researchers find a remarkable bacterial microorganism, living in waste containers, which they name *Slimophagum avarum*. In response to the harsh selective pressure of its environment, *Slimophagum* appears to have evolved a unique regulatory mechanism in the presence of a slimy molecule in the sludge, SMβ, that is periodically stored in the container. SMβ is harnessed by the bacterium as a substitute for membrane lipids.

Slimophagum has a *bme* operon that encodes for three inducible genes—bmeA, bmeB, and bmeC—as well as a repressor bmeR. Proteins A, B, and C are lipid synthesizing enzymes that help produce cell membrane lipids. Protein R is constitutively produced and—in the absence of SMβ—R is not bound to DNA and freely diffuses in the cytoplasm so that lipid synthesis operates normally. If sludge is available, however, the presence of SMβ results in repressor binding to the operator region of the *bme* operon of normal lipid synthesis and SMβ is harnessed by *Slimophagum* as a substitute for the lipid products synthesized by A, B, and C.

Draw a proper network diagram for this novel operon system that contains all necessary components and processes, which should be labeled appropriately. It is not necessary to account explicitly for the repressor-operator complex in the diagram. Represent all genes as their mRNAs, which are transcribed and can degrade. Use solid arrows for material transport or conversions and dashed lines to represent influence edges; for the latter, indicate their roles with appropriate signs.

7.16 Draw a network diagram for a positively repressible operon that regulates gene A, depending on the abundance of metabolite M. Enzyme A is encoded by gene A, while repressor protein, R, is encoded by gene R. Enzyme A binds to substrate S to catalyze the production of metabolite M, and M can degrade. When the repressor R binds to the operator O, the transcription of gene A is inhibited, but the repressor-operator complex does not need to be explicitly accounted for in the diagram. Represent genes as mRNA that are transcribed and eventually degrade. The translation of R and A is activated by its corresponding mRNA. Use dashed lines to represent influence edges and indicate the appropriate signs.

7.17 Bacteria often use iron-restriction as an environmental signal for coordinated expression of virulence genes. In many bacteria, the ferric uptake regulator (Fur) plays a role in this regulation by controlling iron homeostasis. In the presence of its corepressor iron, the Fur protein forms a dimer and binds to specific sequences in its target promoters. Under low iron conditions, the iron storage protein ferritin as well as Fur itself are transcriptionally downregulated. When iron is abundant, the transcription of Fur-regulated genes is blocked by the binding of Fur. Draw a schematic diagram of this operon based upon the information provided.

7.18 Determine the implications of mitochondrial DNA for studies of the evolution of the human race. Explain why studies of human evolution pay particular attention to genes located on the Y chromosome.

7.19 Search the literature for a study where the connectivity of a gene network was inferred with statistical methods from observation (or artificial) data. Write a one-page summary of methods, results, and shortcomings.

7.20 Search the literature for a study where the connectivity of a gene network was inferred from

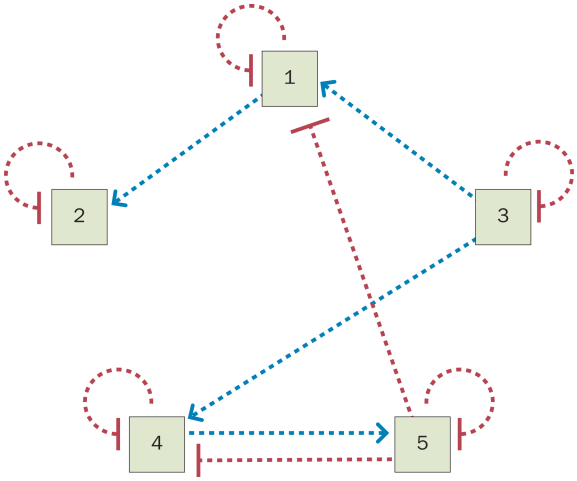


Figure 7.37 Artificial genetic network, used for testing an inference algorithm. (Adapted from Kimura S, Sonoda K, Yamane S, et al. *BMC Bioinform* 9 (2008) 23. With permission from Biomed Central.)

observation (or artificial) data with methods based on differential equations. Write a one-page summary of methods, results, and shortcomings.

7.21 Kimura and collaborators [156] described a method for inferring the structure of genetic networks. Like many other authors they created an artificial genetic network (**Figure 7.37**), where blue and red arrows are supposed to indicate "positive" and "negative regulations." What does "regulation" in this case mean? What would be its biological basis?

7.22 Discuss different options for developing a mathematical model for the gene network in Figure 7.37. Set up a mathematical model in symbolic form. What kinds of data would you need to estimate parameter values for a mathematical model? What type of algorithm would you need for the estimation?

7.23 Read the review of models for gene regulation by Schlitt and Brazma [86] and write a brief report about the use of difference models in the field.

7.24 Suppose two genes indirectly regulate each other's expression in the fashion shown in **Figure 7.38**, where green arrows indicate direct or indirect

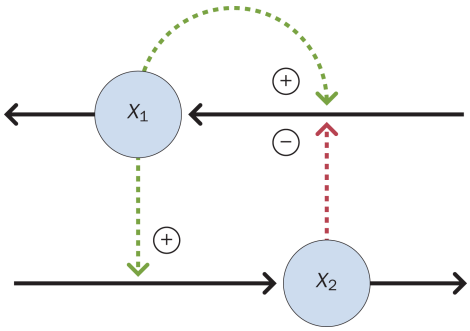


Figure 7.38 Artificial system in which two genes indirectly regulate each other's expression.

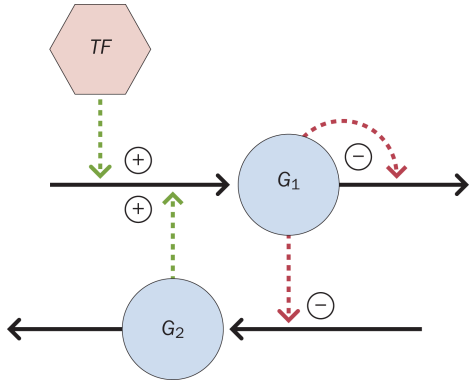


Figure 7.39 Artificial gene network. Although the system consists of just two genes and one transcription factor, it can generate interesting dynamics.

activating effects and the red arrow indicates repression. A possible model for this system has the form

$$\dot{X}_1 = \alpha_1 X_1^{0.4} X_2^{-0.15} - X_1^{0.2},$$
$$\dot{X}_2 = X_1 - X_2^{0.2}.$$

Begin by setting α_1 and the initial values equal to 1. Study the responses of this system for different values of the parameter α. Argue whether α may be positive or negative. Alter the kinetic orders and study the effects. Summarize your findings in a report.

7.25 Implement a model for the artificial gene network in **Figure 7.39**. Begin with the model

$$\dot{G}_1 = \alpha_1 TF\, G_2 - \beta_1 G_1^{-0.5},$$
$$\dot{G}_2 = \alpha_2 G_1^{-1} - \beta_2 G_2^{0.5},$$

and set all rate constants, as well as the initial values and the value of the transcription factor *TF*, equal to 1. Initiate the system with different initial values. Second, study the effects of changes in the value of *TF*. Finally, study how the systems dynamics changes with alterations in parameter values, including the negative kinetic orders associated with G_1.

7.26 Set up an ODE model for the simple metabolic pathway shown in **Figure 7.40A**, using power-law functions (see Chapter 5). Choose kinetic orders between 0.2 and 0.8 and adjust the rate constants to your liking. Study the responses of the pathway to changes in the input. In the second phase of the exercise, use the same system, but augment it to reflect the system in Figure 7.40B, where the end product *Z* affects the production of a transcription factor *TF*, which turns on a gene *G*, which codes for an enzyme *E*, which catalyzes the conversion of *X* into *Y*. Choose appropriate parameter values to your liking. Before you simulate, make predictions of the effects of the addition of *TF*, *G*, and *E*.

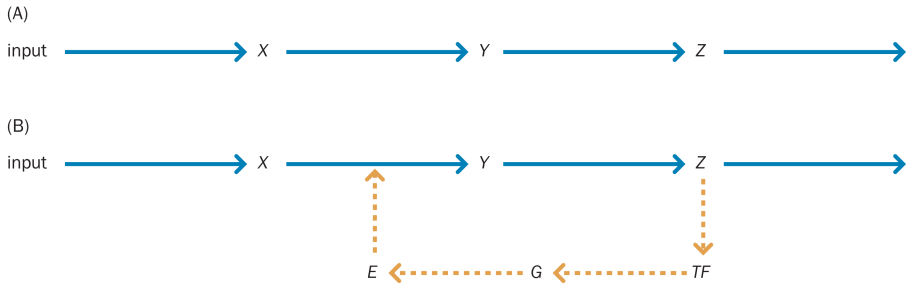


Figure 7.40 Pathways with and without genomic feedback. The responses of the system in (A) to changes in the input are easy to predict, but the same is not true for the system in (B).

7.27 Define functions that reflect the input-output relationships of AND, OR and NAND gates, where the latter has three inputs (see Figure 7.24).

7.28 The production of target proteins in the AhR sensing system (7.4) decreases after a while because some of the ARNT becomes tied up due to appearance of AhRR and also because it loses mass through the degradation of AhRR (k_8), loss of mRNAs (k_{10} and k_{11}) and the utilization of target protein (k_{13}). Explore the magnitude of the different contributions to the dynamics of the target proteins.

7.29 Implement the repressilator model (7.5) in some software. Set all α parameters equal to 6 and all s parameters equal to 1 and study the effect on the systems dynamics. Determine three sets of βs: one where the system does not oscillate, one where the oscillation is damped, and one so that the system oscillates as a limit cycle. In the latter case, demonstrate that the result is not a center.

7.30 Reduce the repressilator model (7.5) to a corresponding system with only two variables. Can you determine (positive or negative) parameter values of the βs so that the system oscillates? Can the system have a limit cycle?

7.31 Define the Boolean table and a function that reflect the input-output relationships in the retinoblastoma example in Figure 7.23.

7.32 Yao and co-workers [157] studied the restriction point within the mammalian cell cycle, where the cell either commits to entering S-phase or remains in G_1. To analyze this bistable behavior, they constructed a simplified model, whose core consisted of the self-activation of the E2F transcription factor, which was inhibited by the retinoblastoma tumor suppressor protein (Rb). Read the original article and implement the differential equation system that is given in the article. Analyze which components are essential for bistability. Summarize your findings in a report. Note that the equations in the article contain several typos. That's life, and it's a good opportunity to learn troubleshooting. Compare the equations with the table of processes and the diagram of the system to detect and reconcile inconsistencies.

7.33 Explore examples of drug development through metabolic engineering and synthetic biology. For instance, investigate the engineered production of the cancer drug paclitaxel or of polyketides, a family of compounds containing antibiotics like erythromycin. Describe your findings in a report.

7.34 Explore the specific techniques of metabolic engineering and synthetic biology that were used in the creation of Golden Rice, which has helped address vitamin A deficiency in developing countries whose people rely on rice as the primary staple. Describe your findings in a report.

7.35 A stable form of tetracycline, called doxycycline, is now considered a staple reagent in molecular biology for controlling gene expression in cells. Scientists have borrowed the natural properties of the tetracycline operon from bacteria to engineer "tet-on" plasmids that can be transduced into mammalian cells and triggered by the addition of doxycycline (**Figure 7.41**). More background description of tet-on systems can be found at the site https://www.tetsystems.com/science-technology/origin-of-the-tet-technology.

7.35.1 Draw a diagram of the dynamics involved in the commercial Dox induction system depicted in Figure 7.41. Include relevant influence arrows and variables that pertain to *both transcription and translation*. Clearly distinguish in your diagram the mass fluxes from influence signals.

7.35.2 Convert your schematic diagram into a mathematical representation in the format of either an GMA- or an S-system (see Chapter 5).

7.35.3 Implement your equations in a simulation using simple numerical assignments for all parameters. Describe the behavior you observe for this operon in the presence and absence of Dox. How does this operon and its regulation resemble the lac operon? How does it differ?

7.36 A synthetic biologist intends to design a logic circuit that consists of two OR, AND, NOR and/or NAND gates, G_1 and G_2 (**Figure 7.42**). The system

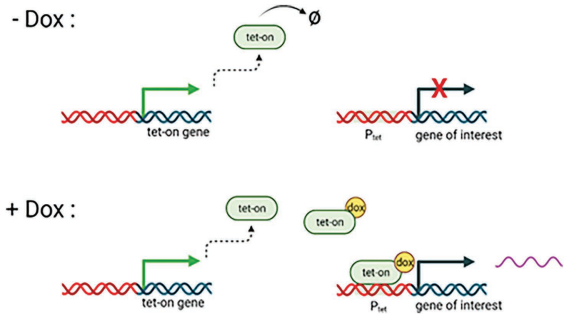


Figure 7.41 Tet-On design. A Tet-On transactivator protein is expressed constitutively but is unable to bind to the PTet promoter region in the absence of doxycycline (Dox; top panel). When bound by Dox (supplied in the culture medium), the transactivator undergoes a conformational change, binds to the DNA at PTet, and activates transcription of the transgene of interest downstream (bottom panel).

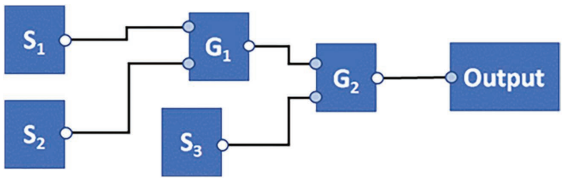


Figure 7.42 Design of a Boolean network with predefined behavior. The system consists of OR, AND, NOR and/or NAND gates in series.

should generate the following input-output behavior: If the two inputs S_1 and S_2 are *on* but the third input S_3 is *off*, there will be no output. Furthermore, if S_3 is *on* and both S_1 or S_2 are *off*, there will be no output. Determine whether G_1 and G_2 should be OR, AND, NOR, and/or NAND gates. Is the solution unique? If yes, argue why. If not, provide at least two combinations of G_1 and G_2 that yield the desired output. In either case, determine what happens if S_1 and S_3 are *off* but S_2 is *on*?

7.37 A complicated challenge for synthetic biologists is that the introduction of molecular circuits into cells requires the delivery of plasmids but that the number of introduced plasmids can be quite variable. The concern is that this variability in component numbers might lead to suboptimal or even undesired system performance.

For this exercise, consider the responses of a repressilator circuit, as proposed by Elowitz and Leibler [144], which we discussed in the text in the simplified form of Eqn. (7.5). The more detailed equations account for mRNAs (*Am*, *Bm*, *Cm*) and corresponding proteins (*Ap*, *Bp*, *Cp*). The model is defined as

$$\dot{Am} = \frac{a}{S + C^n} - Am$$

$$\dot{Bm} = \frac{a}{S + A^n} - Bm$$

$$\dot{Cm} = \frac{a}{S + B^n} - Cm$$

$$\dot{Ap} = b(Am - Ap)$$

$$\dot{Bp} = b(Bm - Bp)$$

$$\dot{Cp} = b(Cm - Cp)$$

Here, the parameter values are $a = 300$, $b = 0.2$, and $n = 2$. S is a signal that normally has a value of 1. Suppose the initial values are $Am = 1$, $Bm = 60$, $Cm = 2$, $Ap = 1$, $Bp = 20$, $Cp = 30$.

7.37.1 Implement the system in PLAS or some other software and run it. Describe the result in words.

7.37.2 To explore the effects of copy number variation, suppose that *Am* and *Cm* are synthetically engineered components with variable copy numbers, while *Bm* is endogenous and therefore not (as) variable. Because the genes for *Am* and *Cm* are on the same plasmid, assume that the variation in copy number affects *Am* and *Cm* in the same manner. Thus, we multiply a factor F to the initial values of *Am* and *Cm*.

Perform a Monte-Carlo simulation, resetting F to different random values. You do not have to consider variable copies to be integer values but can use a standard random number generator with positive real values. How much does the variability in mRNAs affect the output?

7.37.3 Suppose an inhibiting signal is introduced at $t = 50$. Model this change by setting S to 10. How does the behavior of the circuit change when variable copy numbers are implemented as in part 7.37.2? Visualize the result in a pseudo-three-dimensional plot.

7.38 Most of the accomplishments in the brief history of synthetic biology have been achieved in bacteria. Why? What are the challenges in designing and implementing circuits in eukaryotes?

FURTHER READING

Brown TA. *Genomes*, 3rd ed. Garland Science, New York, NY, 2006.

Carro MS, Lim WK, Alvarez MJ, et al. The transcriptional network for mesenchymal transformation of brain tumours. *Nature.* 463 (2010) 318–325.

de Jong H. Modeling and simulation of genetic regulatory systems: A literature review. *J Comput Biol.* 9 (2002) 67–103.

Durban R, Eddy S, Krogh A & Mitchison G. *Biological Sequence Analysis: Probabilistic Models of Proteins and Nucleic Acids*. Cambridge University Press, Cambridge, UK, 1998.

Huang Y-W, Kuo C-T, Stoner K, et al. An overview of epigenetics and chemoprevention. *FEBS Lett*. 585 (2011) 2129–2136.
Noble D. *The Music of Life; Biology Beyond Genes*. Oxford University Press, Oxford, UK, 2006.
Schlitt T & Brazma A. Current approaches to gene regulatory network modelling. *BMC Bioinform*. 8 (2007).
Zvelebil M & Baum JO. *Understanding Bioinformatics*. Garland Science, New York, NY, 2007.

REFERENCES

1. Crick, F.H.C., On protein synthesis (An early draft of an article describing the Central Dogma). *Symp Soc Exp Biol*, 1956. **XII**: p. 139–163, http://profiles.nlm.nih.gov/SC/B/B/F/T/_/scbbft.pdf.
2. Crick, F.H.C., Central dogma of molecular biology. *Nature*, 1970. **227**: p. 561–563.
3. Cowart, L.A., et al., Revealing a signaling role of phytosphingosine-1-phosphate in yeast. *Mol Syst Biol*, 2010. **6**: p. 349.
4. Schuit, F., et al., Glucose-regulated gene expression maintaining the glucose-responsive state of β-cells. *Diabetes*, 2002. **51**(Suppl. 3): p. S326–S332.
5. Merino, E., R.A. Jensen, and C. Yanofsky, Evolution of bacterial TRP operons and their regulation. *Curr Opin Microbiol*, 2008. **11**(2): p. 78–86.
6. Rook, F., et al., Sucrose-specific signalling represses translation of the *Arabidopsis* ATB2 bZIP transcription factor gene. *Plant J*, 1998. **15**(2): p. 253–263.
7. Elgar, G. and T. Vavouri, Tuning in to the signals: Noncoding sequence conservation in vertebrate genomes. *Trends Genet*, 2008. **24**(7): p. 344–352.
8. Noble, D., Claude Bernard, the first systems biologist, and the future of physiology. *Exp Physiol*, 2008. **93**(1): p. 16–26.
9. Breton, S., et al., The unusual system of doubly uniparental inheritance of mtDNA: Isn't one enough? *Trends Genet*, 2007. **23**(9): p. 465–474.
10. Siefert, J.L., Defining the mobilome. *Methods Mol Biol*, 2009. **532**: p. 13–27.
11. Keeling, P.J., Functional and ecological impacts of horizontal gene transfer in eukaryotes. *Curr Opin Genet Dev*, 2009. **19**(6): p. 613–619.
12. Nikaido, H., Multidrug resistance in bacteria. *Annu Rev Biochem*, 2009. **78**: p. 119–146.
13. Tang, Y.W. and C.W. Stratton, *Staphylococcus aureus*: An old pathogen with new weapons. *Clin Lab Med*, 2010. **30**(1): p. 179–208.
14. Sinzelle, L., Z. Izsvak, and Z. Ivics, Molecular domestication of transposable elements: From detrimental parasites to useful host genes. *Cell Mol Life Sci*, 2009. **66**(6): p. 1073–1093.
15. Batzer, M.A. and P.L. Deininger, Alu repeats and human genomic diversity. *Nat Rev Genet*, 2002. **3**(5): p. 370–379.
16. Alberts, B., et al., *Molecular Biology of the Cell*. 2002, New York: Garland Science.
17. Brown, T.A., *Genomes 3*. 3rd ed. 2007, New York, London: Garland Science.
18. HGP, *Human Genome Project*, http://www.ornl.gov/sci/techresources/Human_Genome/project/journals/insights.shtml, 2010.
19. Zvelebil, M. and J.O. Baum, *Understanding Bioinformatics*. 2008, New York, London: Garland Science, p. xxiii + 772.
20. Buehler, L.K. and H.H.E. Rashidi, *Bioinformatics Basics: Applications in Biological Science and Medicine*. 2nd ed. 2005, New York, NY: CRC Press.
21. Maddox, B., The double helix and the wronged heroine. *Nature*, 2003. **421**: p. 407–408.
22. Watson, J.D. and F.H.C. Crick, A structure for deoxyribose nucleic acid. *Nature*, 1953. **171**: p. 737–738.
23. Gregory, T.R. and P.D. Hebert, The modulation of DNA content: Proximate causes and ultimate consequences. *Genome Res*, 1999. **9**(4): p. 317–324.
24. EnsemblPlants, *Populus Trichocarpa Assembly and Gene Annotation*, https://plants.ensembl.org/Populus_trichocarpa/Info/Annotation/, accessed 2023.
25. Fugu-Project, http://www.fugu-sg.org/, 2010.
26. Black, D.L., Mechanisms of alternative pre-messenger RNA splicing. *Annu Rev Biochem*, 2003. **72**: p. 291–336.
27. Wu, H. and E. Moore, Association analysis of the general environmental conditions and prokaryotes' gene distributions in various functional groups. *Genomics*, 2010. **96**(1): p. 27–38.
28. Vaidyanathan, S. and R. Goodacre, Metabolome and proteome profiling for microbial characterization, in *Metabolic Profiling: Its Role in Biomarker Discovery and Gene Function Analysis*, G.G. Harrigan and R. Goodacre, Editors. 2003, Boston: Kluwer.
29. Durban, R., et al., *Biological Sequence Analysis*. 1998, Cambridge, UK: Cambridge University Press.
30. Zhou, F., V. Olman, and Y. Xu, Barcodes for genomes and applications. *BMC Bioinform*, 2008. **9**: p. 546.
31. GO, *Gene Ontology*, http://www.geneontology.org/, 2010.
32. Ohno, S., So much "junk" DNA in our genome. *Brookhaven Symp Biol*, 1972. **23**: p. 366–370.
33. Wheeler, D.A., et al., The complete genome of an individual by massively parallel DNA sequencing. *Nature*, 2008. **452**(7189): p. 872–876.
34. GenBank, *NCBI*, http://www.ncbi.nlm.nih.gov/genbank/, accessed 2023.
35. NCBI (National Center for Biotechnology Information), *Databases*, http://www.ncbi.nlm.nih.gov/guide/all/, accessed 2023.
36. DDBJ, *DNA Data Bank of Japan*, http://www.ddbj.nig.ac.jp/, 2010.
37. Xu, Y. and J.P. Gogarten, eds. *Computational Methods for Understanding Bacterial and Archaeal Genomes. Series on Advances in Bioinformatics and Computational Biology*. Vol. 7. 2008, Singapore: Imperial College Press.
38. Arold, S.T., et al., H-NS forms a superhelical protein scaffold for DNA condensation. *Proc Natl Acad Sci USA*, 2010. **107**(36): p. 15728–15732.
39. Narlikar, G.J., A proposal for kinetic proof reading by ISWI family chromatin remodeling motors. *Curr Opin Chem Biol*, 2010. **14**(5): p. 660–665.
40. McCord, R.P., et al., SnapShot: Chromosome organization. *Mol Cell*, 2022. **82**(12): p. 2350–2350.e1.
41. San Martin, R., et al., Chromosome compartmentalization alterations in prostate cancer cell lines model disease progression. *J Cell Biol*, 2022. **221**(2).
42. Zheng, H. and W. Xie, The role of 3D genome organization in development and cell differentiation. *Nat Rev Mol Cell Biol*, 2019. **20**(9): p. 535–550.
43. RCSB Protein Data Bank, https://www.rcsb.org/, accessed 2024.
44. Huang, Y.-W., et al., An overview of epigenetics and chemoprevention. *FEBS Lett*, 2011. **585**(13): p. 2129–2136.
45. Yoo, C.B. and P.A. Jones, Epigenetic therapy of cancer: Past, present and future. *Nat Rev Drug Discov*, 2006. **5**: p. 37–50.
46. Minocha, T., et al., Chapter 7: Epigenetic drugs based on antiaging approach: An overview, in *Anti-Aging Drug*

Discovery on the Basis of Hallmarks of Aging, S.K. Singh, C.-L. Lin, and S.K. Mishra, Editors. 2022, Academic Press, p. 127–138.
47. Park, S.V., et al., Catalytic RNA, ribozyme, and its applications in synthetic biology. *Biotechnol Adv*, 2019. **37**(8): p. 107452.
48. Siomi, H. and M.C. Siomi, On the road to reading the RNA-interference code. *Nature*, 2009. **457**(7228): p. 396–404.
49. Napoli, C., C. Lemieux, and R. Jorgensen, Introduction of a chimeric chalcone synthase gene into petunia results in reversible co-suppression of homologous genes in trans. *Plant Cell*, 1990. **2**(4): p. 279–289.
50. Liu, Q. and Z. Paroo, Biochemical principles of small RNA pathways. *Annu Rev Biochem*, 2010. **79**: p. 295–319.
51. Deng, W., et al., Organization of the *Caenorhabditis elegans* small non-coding transcriptome: Genomic features, biogenesis, and expression. *Genome Res*, 2006. **16**(1): p. 20–29.
52. Pillai, R.S., S.N. Bhattacharyya, and W. Filipowicz, Repression of protein synthesis by miRNAs: How many mechanisms? *Trends Cell Biol*, 2007. **17**(3): p. 118–126.
53. Chen, J., et al., Overexpression of miR-429 induces mesenchymal-to-epithelial transition (MET) in metastatic ovarian cancer cells. *Gynecol Oncol*, 2011. **121**(1): p. 200–205.
54. Gregory, P.A., et al., MicroRNAs as regulators of epithelial-mesenchymal transition. *Cell Cycle*, 2008. **7**(20): p. 3112–3118.
55. Lai, X., et al., A systems' biology approach to study microRNA-mediated gene regulatory networks. *Biomed Res Int*, 2013. **2013**: p. 703849.
56. Lai, X., O. Wolkenhauer, and J. Vera, Understanding microRNA-mediated gene regulatory networks through mathematical modelling. *Nucleic Acids Res*, 2016. **44**(13): p. 6019–6035.
57. Mayer, G., Bacteriology—Chapter 9: Genetic regulatory mechanism, in *Microbiology and Immunology On-Line*, R.C. Hunt, Editor. 2010, Columbia, SC: University of South Carolina School of Medicine.
58. Adam, M., et al., Epigenetic inheritance based evolution of antibiotic resistance in bacteria. *BMC Evol Biol*, 2008. **8**: p. 52.
59. Jacob, F., et al., Operon: A group of genes with the expression coordinated by an operator. *C R Hebd Seances Acad Sci*, 1960. **250**: p. 1727–1729.
60. Wong, P., S. Gladney, and J.D. Keasling, Mathematical model of the lac operon: Inducer exclusion, catabolite repression, and diauxic growth on glucose and lactose. *Biotechnol Prog*, 1997. **13**(2): p. 132–143.
61. Vilar, J.M., C.C. Guet, and S. Leibler, Modeling network dynamics: The lac operon, a case study. *J Cell Biol*, 2003. **161**(3): p. 471–476.
62. Yildirim, N. and M.C. Mackey, Feedback regulation in the lactose operon: A mathematical modeling study and comparison with experimental data. *Biophys J*, 2003. **84**(5): p. 2841–2851.
63. Tian, T. and K. Burrage, Stochastic models for regulatory networks of the genetic toggle switch. *Proc Natl Acad Sci USA*, 2006. **103**(22): p. 8372–8377.
64. Savageau, M.A., Design of molecular control mechanisms and the demand for gene expression. *Proc Nat Acad Sci USA*, 1977. **74**(12): p. 5647–5651.
65. Su, Z., G. Li, and Y. Xu, Prediction of regulons through comparatice geneome analyses, in *Computational Methods for Understanding Bacterial and Archaeal Genomes*, Y. Xu and J.P. Gogarten, Editors. 2008, Singapore: Imperial College Press, p. 259–279.
66. Yin, Y., et al., Genomic arrangement of bacterial operons is constrained by biological pathways encoded in the genome. *Proc Natl Acad Sci USA*, 2010. **107**(14): p. 6310–6315.
67. RegulonDB, http://regulondb.ccg.unam.mx/, 2010. Cambridge, Massachusetts.
68. Che, D., et al., Detecting uber-operons in prokaryotic genomes. *Nucleic Acids Res*, 2006. **34**(8): p. 2418–2427.
69. Godon, C., et al., The H2O2 stimulon in *Saccharomyces cerevisiae*. *J Biol Chem*, 1998. **273**(35): p. 22480–22489.
70. Igarashi, K. and K. Kashiwagi, Polyamine modulon in *Escherichia coli*: Genes involved in the stimulation of cell growth by polyamines. *J Biochem*, 2006. **139**(1): p. 11–16.
71. Brinkrolf, K., et al., The transcriptional regulatory repertoire of *Corynebacterium glutamicum*: Reconstruction of the network controlling pathways involved in lysine and glutamate production. *J Biotechnol*, 2010. **149**(3): p. 173–182.
72. Covert, M.W., et al., Integrating high-throughput and computational data elucidates bacterial networks. *Nature*, 2004. **429**(6987): p. 92–96.
73. Samal, A. and S. Jain, The regulatory network of *E. coli* metabolism as a Boolean dynamical system exhibits both homeostasis and flexibility of response. *BMC Syst Biol*, 2008. **2**: p. 21.
74. Carro, M.S., et al., The transcriptional network for mesenchymal transformation of brain tumours. *Nature*, 2010. **463**(7279): p. 318–325.
75. Basso, K., et al., Reverse engineering of regulatory networks in human B cells. *Nat Genet*, 2005. **37**: p. 382–390.
76. Martínez-Antonio, A. and J. Collado-Vides, Comparative mechanisms for transcription and regulatory signals in archaea and bacteria, in *Computational Methods for Understanding Bacterial and Archaeal Genomes*, Y. Xu and J.P. Gogarten, Editors. 2008, Singapore: Imperial College Press.
77. Kauffman, S., et al., Random Boolean network models and the yeast transcriptional network. *Proc Natl Acad Sci USA*, 2003. **100**(25): p. 14796–14799.
78. Somogyi, R. and C.A. Sniegoski, Modeling the complexity of genetic networks: Understanding multigenic and pleiotropic regulation. *Complexity*, 1996. **1**(6): p. 45–63.
79. Chen, L., D. Kulasiri, and S. Samarasinghe, A novel data-driven Boolean model for genetic regulatory networks. *Front Physiol*, 2018. **9**.
80. Shmulevich, I., et al., Probabilistic Boolean networks: A rule-based uncertainty model for gene regulatory networks. *Bioinformatics*, 2002. **18**(2): p. 261–274.
81. Babu, M., et al., Systems-level approaches for identifying and analyzing genetic interaction networks in *Escherichia coli* and extensions to other prokaryotes. *Mol Biosyst*, 2009. **5**(12): p. 1439–1455.
82. Tong, A.H., et al., Global mapping of the yeast genetic interaction network. *Science*, 2004. **303**(5659): p. 808–813.
83. Geier, F., J. Timmer, and C. Fleck, Reconstructing gene-regulatory networks from time series, knock-out data, and prior knowledge. *BMC Syst Biol*, 2007. **1**: p. 11.
84. Yeung, M.K., J. Tegner, and J.J. Collins, Reverse engineering gene networks using singular value decomposition and robust regression. *Proc Natl Acad Sci USA*, 2002. **99**(9): p. 6163–6168.
85. de Jong, H., Modeling and simulation of genetic regulatory systems: A literature review. *J Comput Biol*, 2002. **9**(1): p. 67–103.
86. Schlitt, T. and A. Brazma, Current approaches to gene regulatory network modelling. *BMC Bioinform*, 2007. **8**(Suppl. 6).
87. Le Novère, N., Quantitative and logic modelling of molecular and gene networks. *Nat Rev Genet*, 2015. **16**(3): p. 146–158.
88. Li, Y., et al., Boolean nested canalizing functions: A comprehensive analysis. *Theor Comput Sci*, 2013. **481**: p. 24–36.

89. Dimitrova, E., et al., Revealing the canalizing structure of Boolean functions: Algorithms and applications. *Automatica*, 2022. **146**: p. 110630.
90. Chen, T., H.L. Jhe, and G.M. Church, Modeling gene expression with differential equations. *Pacific Symp Biocomput*, 1999: p. 29–40.
91. Gardner, T.S., et al., Inferring genetic networks and identifying compound mode of action via expression profiling. *Science*, 2003. **301**(5629): p. 102–105.
92. Gasch, A.P., et al., Genomic expression programs in the response of yeast cells to environmental changes. *Mol Biol Cell*, 2000. **11**(12): p. 4241–4257.
93. Nakatsui, M., et al., Method for inferring and extracting reliable genetic interactions from time-series profile of gene expression. *Math Biosci*, 2008. **215**(1): p. 105–114.
94. McAdams, H.H. and A. Arkin, It's a noisy business! Genetic regulation at the nanomolar scale. *Trends Genet*, 1999. **15**(2): p. 65–69.
95. Gillespie, D.T., Stochastic simulation of chemical kinetics. *Ann Rev Phys Chem*, 2007. **58**: p. 35–55.
96. Wolkenhauer, O., et al., Modeling and simulation of intracellular dynamics: Choosing an appropriate framework. *IEEE Trans NanoSci*, 2004. **1**: p. 200–207.
97. Gardner, T.S., C.R. Cantor, and J.J. Collins, Construction of a genetic toggle switch in *Escherichia coli*. *Nature*, 2000. **403**: p. 339–342.
98. Plahte, E., T. Mestl, and S.W. Omholt, A methodological basis for description and analysis of systems with complex switch-like interactions. *J Math Biol*, 1998. **36**: p. 321–348.
99. Cherry, J.L. and F.R. Adler, How to make a biological switch. *J Theor Biol*, 2000. **203**(2): p. 117–133.
100. Becskei, A., B. Seraphin, and L. Serrano, Positive feedback in eukaryotic gene networks: Cell differentiation by graded to binary response conversion. *EMBO J*, 2001. **20**(10): p. 2528–2535.
101. Keller, A.D., Model genetic circuits encoding autoregulatory transcription factors. *J Theor Biol*, 1995. **172**(2): p. 169–185.
102. Simon, T.W., R.A. Budinsky, and J.C. Rowlands, A model for aryl hydrocarbon receptor-activated gene expression shows potency and efficacy changes and predicts squelching due to competition for transcription co-activators. *PLoS One*, 2015. **10**(6): p. e0127952.
103. Stevens, E.A., J.D. Mezrich, and C.A. Bradfield, The aryl hydrocarbon receptor: A perspective on potential roles in the immune system. *Immunology*, 2009. **127**(3): p. 299–311.
104. Gupta, A., M.R. Galinski, and E.O. Voit, Dynamic control balancing cell proliferation and inflammation is crucial for an effective immune response to Malaria. *Front Mol Biosci*, 2022. **8**: p. 800721.
105. McDonnell, A.M. and C.H. Dang, Basic review of the cytochrome p450 system. *J Adv Pract Oncol*, 2013. **4**(4): p. 263–268.
106. Stavropoulou, E., G.G. Pircalabioru, and E. Bezirtzoglou, The role of cytochromes P450 in infection. *Front Immunol*, 2018. **9**.
107. Evans, B.R., et al., Repression of aryl hydrocarbon receptor (AHR) signaling by AHR repressor: Role of DNA binding and competition for AHR nuclear translocator. *Mol Pharmacol*, 2008. **73**(2): p. 387–398.
108. Huang, G. and C.J. Elferink, A novel nonconsensus xenobiotic response element capable of mediating aryl hydrocarbon receptor-dependent gene expression. *Mol Pharmacol*, 2012. **81**(3): p. 338–347.
109. Zudaire, E., et al., The aryl hydrocarbon receptor repressor is a putative tumor suppressor gene in multiple human cancers. *J Clin Invest*, 2008. **118**(2): p. 640–650.
110. Voit, E.O. and D.V. Olivença, Discrete biochemical systems analysis (dBST). *Front Molec Biosc*, 2022. **9**: p. 874669.
111. Raj, A. and A. van Oudenaarden, Nature, nurture, or chance: Stochastic gene expression and its consequences. *Cell*, 2008. **135**(2): p. 216–226.
112. Mocek, W.T., R. Rudnicki, and E.O. Voit, Approximation of delays in biochemical systems. *Math Biosci*, 2005. **198**(2): p. 190–216.
113. Voigt, C.A., Synthetic biology 2020–2030: Six commercially-available products that are changing our world. *Nat Commun*, 2020. **11**(1): p. 6379.
114. Trentesaux, C., et al., Harnessing synthetic biology to engineer organoids and tissues. *Cell Stem Cell*, 2023. **30**(1): p. 10–19.
115. Lartigue, C., et al., Creating bacterial strains from genomes that have been cloned and engineered in yeast. *Science*, 2009. **325**(5948): p. 1693–1696.
116. Wang, F. and L.S. Qi, Applications of CRISPR genome engineering in cell biology. *Trends Cell Biol*, 2016. **26**(11): p. 875–888.
117. Mojica, F.J.M. and L. Montoliu, On the origin of CRISPR-Cas technology: From prokaryotes to mammals. *Trends Microbiol*, 2016. **24**(10): p. 811–820.
118. Koeller, K.M. and C.H. Wong, Enzymes for chemical synthesis. *Nat Rev Genet*, 2001. **409**: p. 232–240.
119. Duncan, T.M. and J.A. Reimer, *Chemical Engineering Design and Analysis: An Introduction*. 1998, Cambridge, UK: Cambridge University Press.
120. Mukherji, S. and A. van Oudenaarden, Synthetic biology: Understanding biological design from synthetic circuits. *Nat Rev Genet*, 2009. **10**(12): p. 859–871.
121. Drubin, D.A., J.C. Way, and P.A. Silver, Designing biological systems. *Genes Dev*, 2007. **21**(3): p. 242–254.
122. Keasling, J.D., Synthetic biology for synthetic chemistry. *ACS Chem Biol*, 2008. **3**(1): p. 64–76.
123. Marner, W.D., Practical application of synthetic biology principles. *Biotechnol J*, 2009. **4**(10): p. 1406–1419.
124. Ni, S., et al., Recent progress in aptamer discoveries and modifications for therapeutic applications. *ACS Appl Mater Interfaces*, 2021. **13**(8): p. 9500–9519.
125. Kavita, K. and R.R. Breaker, Discovering riboswitches: The past and the future. *Trends Biochem Sci*, 2023. **48**(2): p. 119–141.
126. Mukherjee, S., et al., A computational approach for the identification of distant homologs of bacterial riboswitches based on inverse RNA folding. *Brief Bioinform*, 2023. **24**(3).
127. Ham, T.S., et al., Design and construction of a double inversion recombination switch for heritable sequential genetic memory. *PLoS One*, 2008. **3**(7): p. e2815.
128. Registry, *Registry of Standard Biological Parts*, http://partsregistry.org/, accessed 2023.
129. Arkin, A., Setting the standard in synthetic biology. *Nat Biotechn*, 2008. **26**(7): p. 771–774.
130. Canton, B., A. Labno, and D. Endy, Refinement and standardization of synthetic biological parts and devices. *Nat Biotechn*, 2008. **26**(7): p. 787–793.
131. Densmore, D., A. Sangiovanni-Vincentelli, and R. Passerone, A platform-based taxonomy for ESL design. *IEEE Des Test Comput*, 2006. **23**(5): p. 359–374.
132. Densmore, D., et al., Algorithms for automated DNA assembly. *Nucleic Acids Res*, 2010, https://doi.org/10.1093/nar/gkq165.
133. Friedland, A.E., et al., Synthetic gene networks that count. *Science*, 2009. **324**: p. 1199–1202.
134. Tabor, J.J., et al., A synthetic genetic edge detection program. *Cell*, 2009. **137**: p. 1272–1281.
135. Salis, H., A. Tamsir, and C. Voigt, Engineering bacterial signals and sensors. *Contrib Microbiol*, 2009. **16**: p. 194–225.
136. Salis, H.M., E.A. Mirsky, and C.A. Voigt, Automated design of synthetic ribosome binding sites to control protein expression. *Nat Biotechnol*, 2009. **27**: p. 946–950.

137. March, J.C. and W.E. Bentley, Quorum sensing and bacterial cross-talk in biotechnology. *Curr Opin Biotechnol*, 2004. **15**: p. 495–502.
138. Hooshangi, S. and W.E. Bentley, From unicellular properties to multicellular behavior: Bacteria quorum sensing circuitry and applications. *Curr Opin Biotechnol*, 2008. **19**: p. 550–555.
139. Li, J., et al., A stochastic model of *Escherichia coli* AI-2 quorum signal circuit reveals alternative synthesis pathways. *Mol Syst Biol*, 2006. **2**: p. 67.
140. Nandagopal, N. and M.B. Elowitz, Synthetic biology: Integrated gene circuits. *Science*, 2011. **333**(6047): p. 1244–1248.
141. Atkinson, M.R., et al., Development of genetic circuitry exhibiting toggle switch or oscillatory behavior in *Escherichia coli*. *Cell*, 2003. **113**(5): p. 597–607.
142. Xie, M. and M. Fussenegger, Designing cell function: Assembly of synthetic gene circuits for cell biology applications. *Nat Rev Mol Cell Biol*, 2018. **19**(8): p. 507–525.
143. Sedlmayer, F., D. Aubel, and M. Fussenegger, Synthetic gene circuits for the detection, elimination and prevention of disease. *Nat Biomed Eng*, 2018. **2**(6): p. 399–415.
144. Elowitz, M.B. and S. Leibler, A synthetic oscillatory network of transcriptional regulators. *Nature*, 2000. **403**: p. 335–338.
145. Martin, V.J.J., et al., Engineering a mevalonate pathway in *Escherichia coli* for production of terpenoids. *Nat Biotechnol*, 2003. **21**(7): p. 796–802.
146. Ro, D.K., et al., Production of the antimalarial drug precursor artemisinic acid in engineered yeast. *Nature*, 2006. **440**(7086): p. 940–943.
147. NIAID, *Understanding Malaria*, http://www3.niaid.nih.gov/topics/Malaria/understandingMalaria/.
148. Graham, I.A., et al., The genetic map of *Artemisia annua L.* identifies loci affecting yield of the antimalarial drug artemisinin. *Science*, 2010. **327**(5963): p. 328–331.
149. Bertea, C.M., et al., Isoprenoid biosynthesis in *Artemisia annua*: Cloning and heterologous expression of a germacrene A synthase from a glandular trichome cDNA library. *Arch Biochem Biophys*, 2006. **448**(1–2): p. 3–12.
150. Dondorp, A.M., et al., Artemisinin resistance in *Plasmodium falciparum* malaria. *N Engl J Med*, 2009. **361**(5): p. 455–467.
151. Ye, X., et al., Engineering the provitamin A (beta-carotene) biosynthetic pathway into (carotenoid-free) rice endosperm. *Science*, 2000. **287**(5451): p. 303–305.
152. Paine, J.A., et al., Improving the nutritional value of Golden Rice through increased pro-vitamin A content. *Nat Biotechnol*, 2005. **23**(4): p. 482–487.
153. Naqvi, S., et al., Transgenic multivitamin corn through biofortification of endosperm with three vitamins representing three distinct metabolic pathways. *Proc Natl Acad Sci USA*, 2009. **106**(19): p. 7762–7767.
154. Golden Rice Project, http://www.goldenrice.org/, 2011.
155. Beisel, C.L. and C.D. Smolke, Design principles for riboswitch function. *PLoS Comput Biol*, 2009. **5**(4): p. e1000363.
156. Kimura, S., M. Tokuhisa, and M. Okada-Hatakeyama. Genetic network inference using hierarchical structure. *Front Physiol*, 2016. **7**: p. 57.
157. Yao, G., et al., A bistable Rb–E2F switch underlies the restriction point. *Nature Cell Biol*, 2008. **10**(4): p. 476–482.

Taylor & Francis
Taylor & Francis Group
http://taylorandfrancis.com

Protein Systems

8

When you have read this chapter, you should be able to:

- Discuss the different types of proteins and their roles
- Describe the basic chemical properties of proteins
- Explain the four hierarchical levels of protein structure
- Retrieve from databases information about the crystal structure of proteins
- Design simple models of protein activation
- Understand the concepts of signal transduction with proteins

Biological systems are complicated machines, and proteins supply most of their gears, gaskets, valves, sensors, and amplifiers. Indeed, proteins are arguably the most versatile and fascinating molecules in living cells. Although all proteins have generically the same molecular composition, their sizes, functions, and roles could not be more diverse, and their importance for life is hard to overestimate. Proteins are constantly generated and destroyed, with some existing just for minutes or less and others, such as the proteins in the lens of the human eye, persisting for an entire human life. Owing to their central importance and versatility, an astounding amount of information has been accumulated over the past two centuries, since the Swedish chemist Jöns Jacob Berzelius coined the term protein (Greek: primary) in 1838, and uncounted databases for different aspects of proteins are available (see, for example, [1–4]). Proteins have been the subject of intense investigation in different fields: biochemistry and protein chemistry; molecular, structural, and computational biology; metabolic engineering; bioinformatics; proteomics; systems biology; and synthetic biology. Yet, what we know today might just be the tip of the iceberg. Thus, this chapter will not come close to portraying all aspects of proteins in a just manner, and it will not even present much detail about relevant topics as far as they are covered comprehensively in texts on computational biology and bioinformatics (e.g., [5]). Instead, it will focus on some of the main features of proteins and discuss those roles that are of particular importance for the dynamic functioning of biological systems. We begin with some chemical properties and the assembly of proteins according to the building instructions provided in the genome, describe the different roles of proteins, and end with some basic modeling approaches. A good starting point for exploring the known properties of a protein is the Human Protein Atlas [6].

DOI: 10.1201/9781003402770-8

8.1 CHEMICAL AND PHYSICAL FEATURES OF PROTEINS

Proteins are polymers composed of amino acids, which are joined into a linear string by **peptide bonds** (Figure 8.1). Each amino acid unit (or residue of the polymer) has a particular molecular side chain, denoted by R in Figure 8.1. The chemical properties of the side chains determine the features of the amino acids and, collectively, of the protein. Importantly, the side chains can interact with each other through different noncovalent mechanisms, such as hydrogen bonding and van der Waals forces. Furthermore, the spatial arrangements of amino acid residues are affected by hydrophobic effects, which tend to maximize the packing density of the protein, as well as by covalent links and ionic bonds, called salt bridges. The interactions between the side chains of distant amino acids of the same or a different protein lead to cross-linking and the formation of the **tertiary** (three-dimensional) **structure** of the protein (see later). They are also responsible for the stability of a protein. Of note is the most prevalent covalent link, namely a disulfide bond between two cysteine residues, which is so important that the resulting dimeric (two-component) amino acid has its own name: cystine. Almost all of nature uses 20 L-amino acids (Table 8.1). Alas, as so often in nature, there are exceptions: a very few organisms make use of two additional amino acids, selenocysteine and pyrrolysine [7, 8].

As we discussed in Chapter 7, proteins are generated from stretches of DNA within the organism's genome through the distinct processes of transcription, translation and, frequently, subsequent "posttranslational" modifications. Specifically, each amino acid within a protein corresponds to a DNA codon, consisting of a triplet of nucleotides within the genetic code, which is transcribed into a matching RNA codon. Because DNA consists of four nucleotides, $4 \times 4 \times 4 \times 64$, different codons are possible. A few of these serve as start or stop signals for transcription, but more codons than necessary are left to code for the 20 amino acids, with the consequence that most amino acids may be encoded by more than one codon (Table 8.2). In fact, in cases such as alanine, the third nucleotide of the codon is virtually, although not totally, immaterial. While different codons can thus represent the same amino acid,

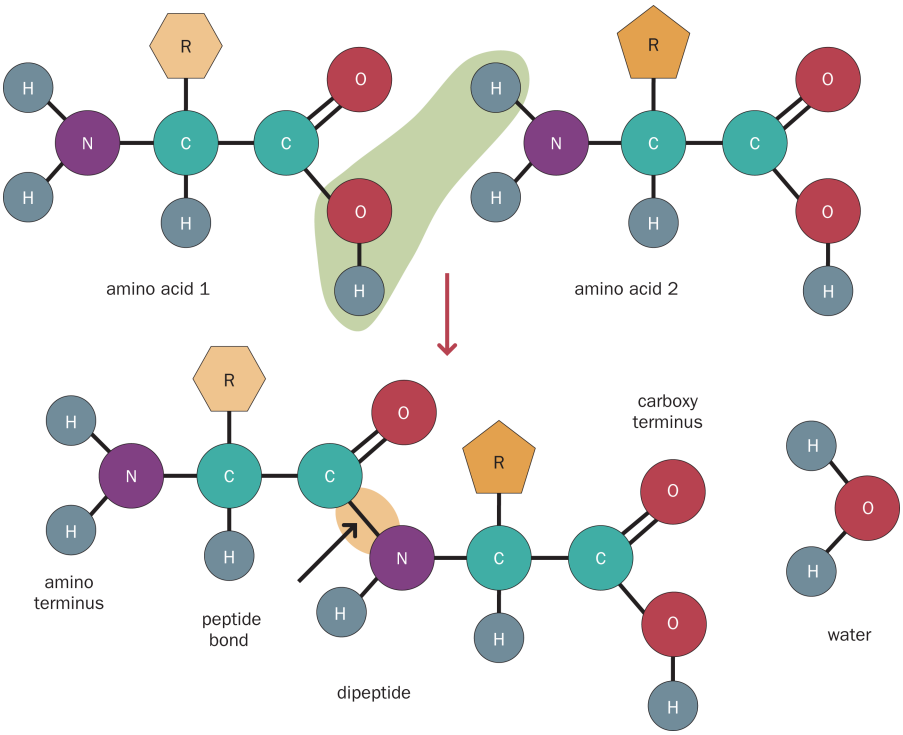


Figure 8.1 Generic representation of a peptide bond between two amino acids. R represents the specific side chain of each amino acid.

TABLE 8.1: AMINO ACIDS IN ALPHABETICAL ORDER WITH THEIR ABBREVIATIONS

The generic backbone is shown in blue. Color coding of the amino acid's name represents types of side chains: orange, hydrophobic; green, positively charged; red, negatively charged; yellow, polar uncharged; turquoise, special cases; purple, nonstandard amino acids.

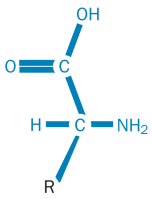
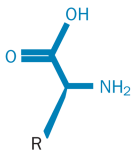
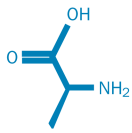
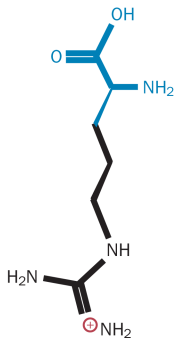
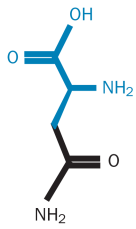
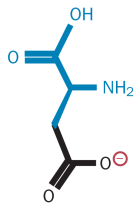
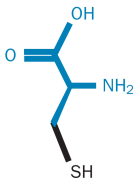
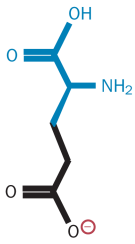
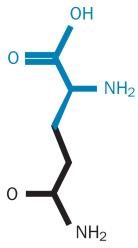
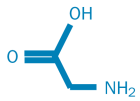
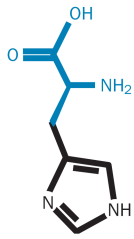
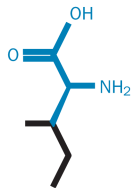
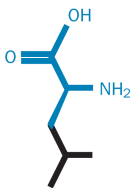
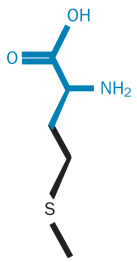
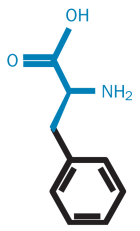
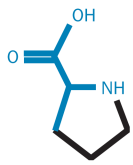
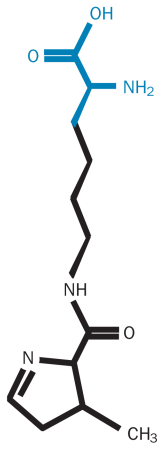
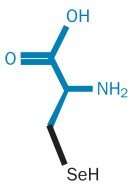
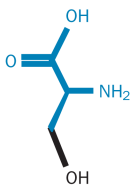
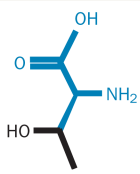
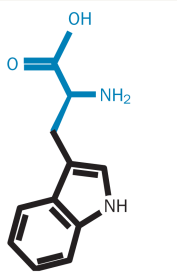
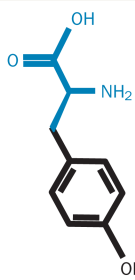
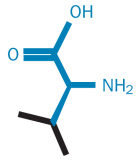
Generic Amino Acid	Generic Amino Acid (simplified notation)	Alanine (A/Ala)	Arginine (R/Arg)	Asparagine (N/Asn)	Aspartic Acid (D/Asp)
					
Cysteine (C/Cys)	**Glutamic Acid (E/Glu)**	**Glutamine (Q/Gln)**	**Glycine (G/Gly)**	**Histidine (H/His)**	**Isoleucine (I/Ile)**
					
Leucine (L/Leu)	**Lysine (K/Lys)**	**Methionine (M/Met)**	**Phenylalanine (F/Phe)**	**Proline (P/Pro)**	**Pyrrolysine (O/ Pyl)**
					
Selenocysteine (U/Sec)	**Serine (S/Ser)**	**Threonine (T/Thr)**	**Tryptophan (W/Trp)**	**Tyrosine (Y/Tyr)**	**Valine (V/Val)**
					

TABLE 8.2: AMINO ACIDS AND THEIR CORRESPONDING RNA CODONS			
Ala	GCA, GCC, GCG, GCU	**Leu**	CUA, CUG, CUC, CUU, UUA, UUG
Arg	AGA, AGG, CGA, CGC, CGG, CGU	**Lys**	AAA, AAG
Asn	AAC, AAU	**Met**	AUG
Asp	GAC, GAU	**Phe**	UUC, UUU
Cys	UGC, UGU	**Pro**	CCA, CCC, CCG, CCU
Gln	CAA, CAG	**Ser**	UCA, UCC, UCG, UCU, AGC, AGU
Glu	GAA, GAG	**Thr**	ACA, ACC, ACG, ACU
Gly	GGA, GGC, GGG, GGU	**Trp**	UGG
His	CAC, CAU	**Tyr**	UAC, UAU
Ile	AUA, AUC, AUU	**Val**	GUA, GUC, GUG, GUU
START	AUG	**STOP**	UAA, UAG, UGA

a phenomenon called wobble, the reverse is not true: every codon codes for one and only one amino acid, or, as a mathematician might say, the transcription–translation system implements a "many-to-one mapping." Nonetheless, the choice of a codon is apparently not random. For instance, different species often prefer particular codons over alternatives, a phenomenon called codon usage bias (see Table 3 in [9]). In prokaryotes, this bias is so pronounced that species can be barcoded based on codons [10] (see Chapter 7). Also intriguing, recent research seems to suggest that alternative codons, while resulting in the same amino acid, can nevertheless affect the function of a protein in some cases [11].

Each side chain makes an amino acid chemically unique. Because there are 20 standard amino acids and because proteins contain many of them, the potential variability among proteins is enormous. It might sound exaggerated to consider the possible effects of single amino acid substitutions, but it is well known that even a single change can indeed alter the structure of a protein drastically [12]. For instance, sickle cell disease is caused by a nucleotide mutation that results in a single amino acid substitution yet has dramatic physiological effects.

Proteins vary tremendously in size. On the low end of the size scale are **peptides**, which consist of up to about 30 amino acids; this boundary between peptides and proteins is not clearly defined. On the high end, the largest protein that has so far been characterized is titin, which is a constituent of muscle tissue and consists of about 30,000 amino acids. To gauge the degree of variability among proteins, compute how many different proteins would be possible if they consisted of exactly (or at most) 1,000 or 30,000 amino acids. The size of a protein is usually not given in terms of the number of amino acids but in atomic mass units or Daltons. Because of their large sizes, proteins are actually measured in kilodaltons (kDa). As an example, titin has a molecular mass of about 3,000 kDa.

The physical and chemical characteristics of two neighboring amino acid side chains slightly bend their peptide bond in a predictable fashion, and all these bends collectively cause the protein to fold into a specific three-dimensional (3D) structure or **conformation**. This structure can contain coils, sheets, hairpins, barrels, and other features, which we will discuss later. Rearranging the same amino acids in another order would almost certainly result in a different 3D structure. One should note in this context that some proteins contain unstructured peptide regions of various lengths that allow different bending and folding, which in turn increases the repertoire of roles a protein can assume [13].

Half a century ago, Nobel Laureate Christian Anfinsen postulated the so-called thermodynamic hypothesis stating that, at least for the class of small globular proteins, the structure of a protein is determined exclusively by the amino acid sequence so that under normal environmental conditions the protein structure is unique, stable, and such that it minimizes the total Gibbs free energy [14]. This assertion was based on in vitro experiments where he and his colleagues showed that the protein ribonuclease, which had been unfolded with urea, spontaneously refolded into the right conformation once the urea was removed. One could think

that disulfide bonds might be a driving force in this process, but it is our current understanding that they are rather the result of correct folding and serve to stabilize a protein, whether it is folded correctly or incorrectly.

The automatic folding posited by Anfinsen's Dogma poses an interesting puzzle, called Levinthal's Paradox. Cyrus Levinthal computed that even if a protein consisted of only 100 amino acids, it would allow so many conformations that correct folding would take 100 billion years [15]. Of course, that is out of the question and, in reality, the process is remarkably quick: it takes only a bit more than one minute to fold a protein of 300 amino acids! While we do not completely understand how this is possible, we do know that the folding of a newly created amino acid chain into the proper 3D structure is facilitated by two means. First, a special class of cytosolic proteins, called chaperonins, supports the folding task as the newly formed chain of amino acids exits the **ribosome**. A prominent example is the GroEL/ES protein complex. GroEL consists of 14 subunits that form a barrel, the inside of which is hydrophobic, while its partner GroES resembles a lid. The unfolded amino acid chain and adenosine triphosphate (ATP) bind to the interior rim of a cavity in the barrel-shaped GroEL and trigger a change in conformation, as well as an association of the growing protein with GroEL. Binding between the two proteins causes the subunits of GroEL to rotate and to eject the folded protein, and GroES is released. Proteins often undergo several rounds of refolding until the appropriate conformation is reached. Modern methods of cryo-electron-microscopy permit a glimpse into the structure of proteins like GroEL/ES (see **Figure 8.2**).

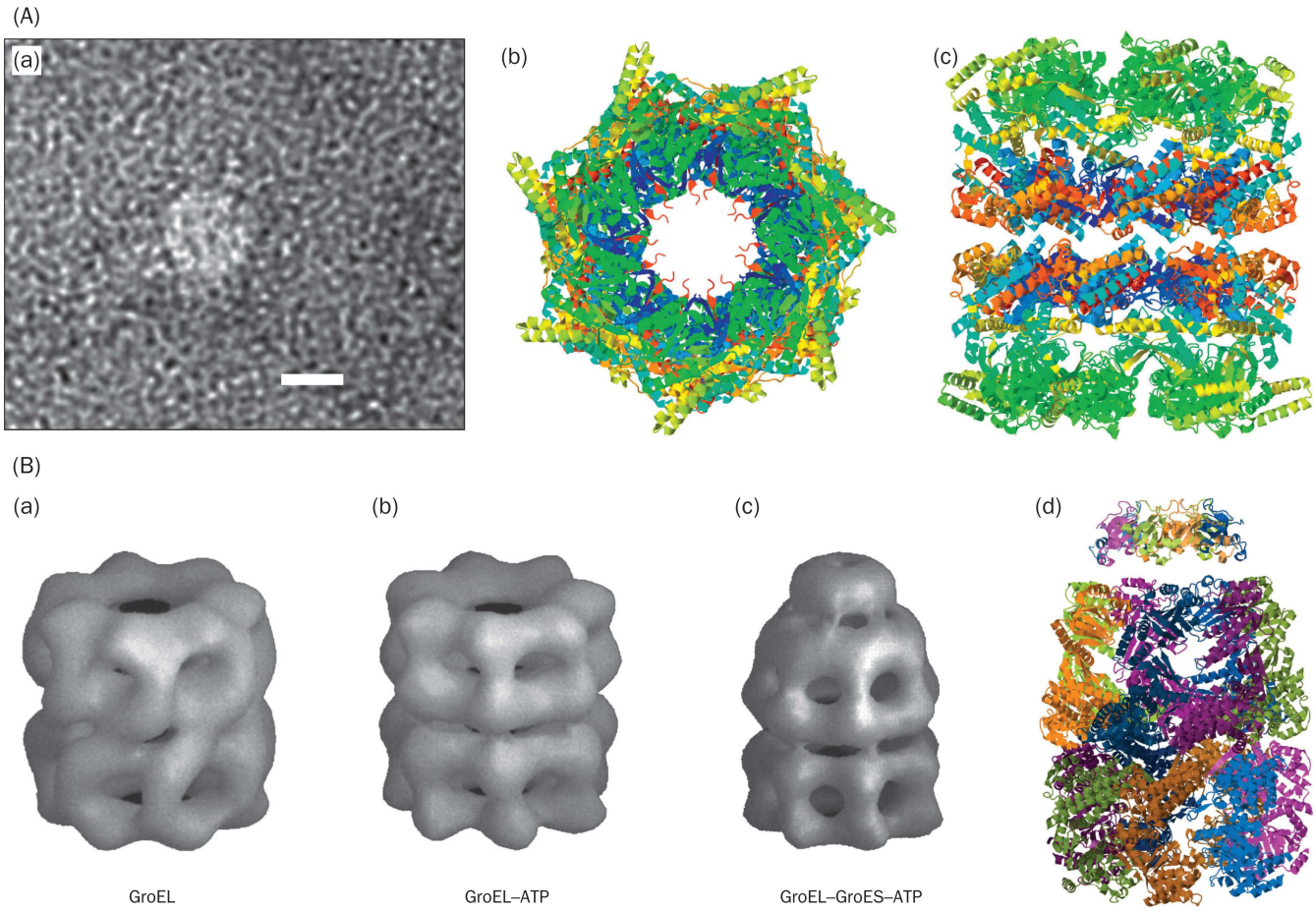


Figure 8.2 Cryo-EM results for the molecular chaperone GroEL. (A) Electron micrograph (a) and the inferred ribbon structure of the protein GroEL in the end-view orientation (b) and in a side view (c) (Protein Data Bank: PDB 2C7E [16]). GroEL measures 140 Å in diameter in the end-on orientation; the scale bar represents 100 nm. Courtesy of Inga Schmidt-Krey, Georgia Institute of Technology. (B) Cryo-EM-based, surface-rendered views of 3D reconstructions of (a) GroEL, (b) GroEL–ATP, and (c) GroEL–GroES–ATP. The GroES ring is seen as a disk above the GroEL. (d) Corresponding ribbon structure. It consists of a β-barrel structure with a β-hairpin forming the roof of the dome-like heptamer (Protein Data Bank: PDB 2CGT [17]). ([a–c] from Roseman AM, Chen S, White H, et al. *Cell* 87 [1996] 241–251. With permission from Elsevier.)

Chaperonins belong to a larger class of molecules, called molecular **chaperones**. These not only facilitate folding, they also protect a protein's 3D structure in situations of stress, such as elevated heat. In reference to this role, chaperones belong to the class of heat-shock proteins (HSPs).

The second mechanism facilitating proper folding is accomplished through special enzymes in the endoplasmic reticulum of eukaryotes. These enzymes, called protein disulfide isomerases (PDIs), catalyze the formation and breakage of disulfide bonds between cysteine residues within the protein [18]. This mechanism helps proteins find the correct arrangement of disulfide bonds. PDIs also serve as chaperones that aid the refolding of wrongly arranged proteins. In the process of forming a disulfide bond, the PDI is chemically reduced. The protein Ero1 subsequently oxidizes PDI and thereby refreshes its active state.

While the chain of amino acids in a protein corresponds to the nucleotide sequence in the corresponding mRNA and DNA, a protein may be chemically altered once the transcription and translation processes are completed. Collectively these alterations are called **post-translational modifications (PTMs)**. They consist of the attachment of biochemical groups, for example, phosphate, acetate, or larger molecules such as carbohydrates or lipids. Furthermore, amino acids may be removed from the amino end of the protein, enzymes may cut a peptide chain into two, and disulfide bridges may form between two cysteine residues. It even happens that the amino acid arginine is changed into a different amino acid, citrulline, which has no corresponding codon. Hundreds of PTMs are listed in a document file within the UniProt Knowledgebase [4], and proteome-wide statistics on PTMs are available at [19]. PTMs are of enormous importance, because they expand the repertoire of protein functions manifold and permit fine-tuned, dynamic control of the specific roles of proteins. A prominent example of the functionality of a PTM is phosphorylation or dephosphorylation in a specific location of a protein, which in the case of an enzyme or signaling protein can lead to activation or deactivation (see models at the end of the chapter).

The turnover of proteins can be very fast or very slow, depending on the particular protein. If a protein is no longer needed, it is marked for degradation by the attachment of arginine [11] or of multiple copies of a specific regulatory protein called **ubiquitin**. This marking is recognized by a cellular protein recycling organelle, called the **proteasome**, which disassembles the protein into peptides and amino acids [20]. Aaron Ciechanover, Avram Hershko, and Irwin Rose received the 2004 Nobel Prize in Chemistry for the discovery of this process. Interestingly, ubiquitin and ubiquitin-like proteins are directly involved in the assembly of ribosomes [21]. This coupling of the nuclear ubiquitin–proteasome system with the assembly of new ribosomal subunits ensures that these two central processes of protein production and destruction are well coordinated.

Thus, the genetic code determines the amino acid sequence, the amino acid residues and the protein's environment determine the base structure of a protein, and the structure and regulatory modifications determine its role. Special proteins help other proteins fold into the right conformation, affect their activity, and mark them for destruction in an ongoing dynamic process. During their short or long lifetimes, the proteins in a functioning cell or organism have roles that are amazingly diverse, as we will see throughout this and other chapters.

8.2 EXPERIMENTAL ASSESSMENTS OF PROTEINS

Uncounted studies on proteins have been published, and we will not even try to do justice to the field. As a very coarse overview, the majority of experimental efforts in protein research may be categorized in four classes: what does a specific protein look like? Which proteins are present in some system like a cell? Where are specific proteins localized? How do proteins interact with each other and with other molecules?

8.2.1 Protein Structure

We know that proteins consist of chains of amino acids. But in contrast to DNA and RNA, we do not know directly what their three-dimensional (3D) structure looks

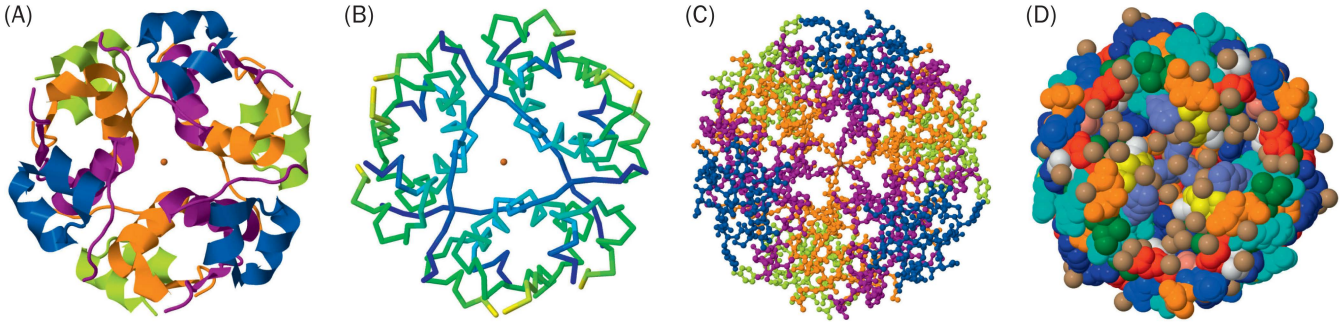


Figure 8.3 Different PDB-Jmol representations of the crystal structure of human insulin, complexed with a copper ion (in the center). (A) Ribbon cartoon representation. (B) Backbone representation. (C) Ball-and-stick representation. (D) Sphere representation. The last of these, also called the space-filling or Corey–Pauling–Koltun (CPK) representation, uses a convention showing atoms in defined colors (for example, C, gray; O, red; N, blue; Cu, light brown). (Protein Data Bank: PDB 3IR0 [25, 26].)

like, as it is driven by the physico-chemical properties of all amino acids in the protein, which bend the amino acid chain due to their shapes and charges. Experimental methods for this task include X-ray crystallography [22], NMR (nuclear magnetic resonance) spectroscopy [23], and cryo-electron-microscopy (cryo-EM) [24]. Because of the large number of amino acids in a protein, there is a huge repertoire of potential protein shapes and functions. Good software tools, such as the open-source Java viewer Jmol in the Protein Data Bank (PDB) [2, 25], have been developed to visualize the 3D shapes of protein crystal structures (**Figure 8.3**). In spite of concerted efforts of the scientific community, the 3D structure cannot always be obtained with experimental means.

A distinct alternative is the attempt to determine the structure with computational means, based on detailed knowledge of the physical and chemical properties of amino acids. While it may seem like a small step to concatenate information of each amino acid within a chain, the computational prediction is extremely difficult and requires very sophisticated computing. The task can be structured into the analysis of four types of structures. The primary structure of a protein refers to its amino acid sequence, which is by and large known from the sequence of the gene coding for the protein. The secondary structure is a localized feature involving regular 3D shapes or motifs. The best understood of these motifs are α-helices, β-sheets, and turns. Each α-helix is the consequence of regularly spaced hydrogen bonds between backbone oxygen and amide hydrogen atoms, while a β-sheet occurs when two strands are joined by hydrogen bonds and each strand contains alternating residues. Turns and bends are due to three or four specific sequences of amino acids that permit tight folds that are stabilized with hydrogen bonds. The local motifs are usually interrupted by less-defined stretches within the protein, and all defined and undefined portions combine to form the tertiary structure, that is, the overall folding of the protein.

The tertiary structure of a protein is maintained by a hydrophobic core, which prevents the easy dissolution of the structure in the aqueous medium of the cell and furthermore through chemical bonds, such as hydrogen bonds, disulfide bonds, and salt bridges. One should note that this tertiary structure is not iron-clad, and post-translational modifications, changes in environmental conditions, and various cellular control mechanisms of protein activation and deactivation can alter it significantly [11, 13]. This flexibility in folding is crucial, because it is directly related to the function of the protein.

Finally, many proteins consist of subunits and are functional only as complexes. The structure formed by two or more protein subunits is called the quaternary structure of a protein. The functional rearrangements, which may affect both the tertiary and quaternary structure, are often called conformations, and transitions between such states are called conformational changes.

The prediction of the 3D structure is very difficult, but it is even more difficult to predict the function of a protein from knowledge of its amino acid sequence is very difficult. While the nucleotide sequence provides the initial code for peptides and proteins, one must remember that many modifications can occur between the

BOX 8.1: ALPHAFOLD FOR PRIMARY SEQUENCE-TO-STRUCTURE PREDICTION

In 2021, protein structure prediction experienced a landmark achievement [27]. For decades, predictions from primary sequences had been based upon physical principles of seeking the conformational state that achieves minimization of free energy. Often, the "threading" of a primary sequence into a template of a related, resolved protein structure was used as a starting point for simulated annealing (as described in Chapter 6) and physics-based modeling of atomic-level interactions and chemical bonding within the macromolecular polymer chain of amino acid residues. This approach was hindered by the small fraction of protein structures available as templates and the uneven coverage of solved structures across the range of protein families.

Recent breakthroughs came from the application of artificial intelligence to this long-standing challenge [28, 29]. Researchers trained neural networks on the Protein Database (PDB), which contains 170,000 experimentally resolved structures. Improvements in deep learning allowed for fitting millions of parameter values to capture the high-dimensional degrees of freedom of motion possible for a given sequence. The current algorithms, AlphaFold and RoseTTAFold, approach experimental accuracy of atomic position on the order of 1.5 angstroms, roughly the width of an atom [28, 29]. The open release of the AlphaFold2 platform to academic researchers turned out to be transformative for identifying sequence-to-function relationships, with over 200 million predicted entries across 47 organisms available as of 2024 [30]. While many of these predictions will never be experimentally validated, the sheer scale of the newly available structures is suggesting comparisons with the impact of the Human Genome Project on biomedical research.

translation of mRNA and the availability of the fully functional protein. These modifications can affect many physical and chemical properties of the protein and thereby alter its folding, stability, activity, and function. Furthermore, peptides may be spliced together in different ways, thereby forming distinct proteins. Presently available methods for predicting the so-far uncharacterized function of a protein include a variety of similarity, homology, and structure comparisons with proteins of known sequence and function [31]. Also helpful in many cases are phylogenetic approaches and information from the Gene Ontology Project [32] (see Chapter 7). All these methods fall within the realm of bioinformatics and structural biology, and readers interested in the topic are encouraged to consult texts such as [5]. An extension of these computational methods is the field of **molecular dynamics**, which attempts to describe how molecules, for instance in proteins, move in order to achieve a specific function [33].

8.2.2 What Is There?

The proteome is the totality of proteins in a system, and **proteomics** attempts to characterize it. An important aspect of this characterization is an account of the inventory and the abundance of all (or at least many) proteins at a given point in time. As is to be expected, both inventory and abundance can significantly differ among the tissues of an organism. Extensive research has addressed the spatial distribution and localization of target proteins, as well as temporal changes due to physiological trends, such as differentiation and aging, or in response to infection or disease. It is hoped that an early identification of these spatial and temporal changes might suggest diagnostic biomarkers that show up before a disease manifests.

Proteomics is a grand challenge, because it is not known how many proteins exist in model organisms. It is not even clear how to count them. For instance, should the same protein with two different post-translational modifications count as one or two proteins? Many signaling proteins and enzymes can be phosphorylated or dephosphorylated, and these processes change the activity state of the protein. Should both forms be counted separately? Methylation, glycosylation, arginylation, ubiquitination, oxidation, and other modifications raise similar questions. Finally, newly formed transcripts can be spliced in different ways, thereby leading to different proteins. These splice variants are presumably the reason that humans can manage with only about 20,000 to 25,000 genes, while some trees are thought to possess twice as many [34].

Proteomics methods are quite diverse. Gel electrophoresis [35–38] separates proteins by molecular size and possibly by their isoelectric point, which is the pH at which the protein has a neutral charge. Other separation techniques use different variations of liquid chromatography [39–42]. Mass spectrometry, in different implementations, identifies the mass of proteins and sometimes permits the identification of post-translational modifications and protein complexes [43–46]. For a recent review, see [47].

8.2.3 Localization

To serve their specific functions, proteins must be in the correct location within a cell or tissue. It is therefore of great interest to study which proteins are located where and when. An intriguing method is the attachment of the small **green fluorescent protein (GFP)** to a target protein [48–50]. This attachment can be achieved through genetic engineering by expressing in the cell of interest a fusion protein consisting of the natural target protein with a GFP reporter linked to it. GFP consists of a β-barrel containing a chromophore, which is a molecule that absorbs light of a certain wavelength and emits another wavelength. GFP is not limited to locating proteins but is widely used in molecular biology for reporting the presence of a variety of cellular components and the expression of genes that are of particular interest. In fact, the discovery of GFP has truly transformed cell biology, especially in conjunction with automated real-time systems for fluorescence microscopy in living cells. For example, Singleton and colleagues used enhanced GFPsignaling sensors to investigate the spatial and temporal processes of 30 signaling intermediates, including receptors, kinases, and adaptor proteins, within individual T cells during their activation [51]. Martin Chalfie, Osamu Shimomura, and Roger Y. Tsien were awarded the 2008 Nobel Prize in Chemistry for the discovery and development of GFP. While GFP was the first "coloring agent" for proteins, many other colors were subsequently developed [52, 53].

A distinctly different methodology was developed by Richard Caprioli's laboratory at Vanderbilt University. This methodology scans tissue slices and applies mass spectrometry individually to very many small areas in a tight grid. The result is a peptide spectrum for every grid point, and visualization reveals a very detailed picture of peptide abundances within the tissue section. This imaging method even permits the creation of 3D pictures of peptide abundances in tissues. For such an analysis, the tissue is thinly sliced and every slice is analyzed with mass spectrometry imaging. Computer software is used to stack up all these images to create a 3D impression [54].

8.2.4 Interactions Involving Proteins

Proteins can interact with about any other biological molecules. In the case of DNA, they serve as transcription factors; the translation of RNA involves proteins, the entire field of enzyme kinetics assesses how proteins catalyze organic compounds, and many types of molecules can "dock" (bind) as "ligands" to proteins, for instance in the form of receptors on cell membranes. Proteins also serve as transporters for all kinds of molecules, including metals [55]. Many of these interactions have already been discussed or will be discussed later in this book because there is great interest in the question how molecules, whether proteins, peptides, or small molecules dock to proteins (**Figure 8.4**). This question is important because the docking may potentially be used for therapeutic purposes.

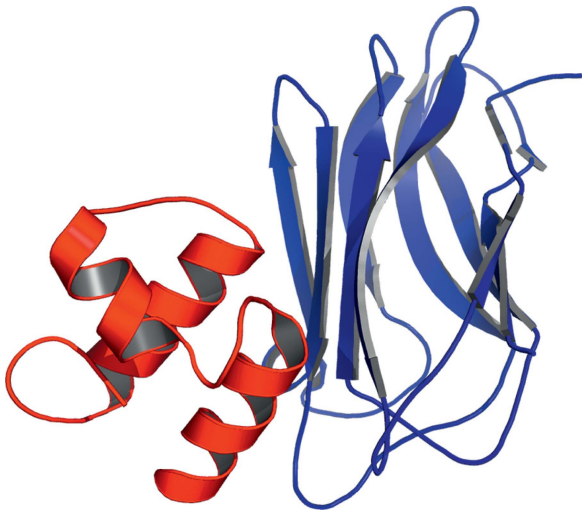


Figure 8.4 Interaction between two proteins forming the cohesin–dockerin complex of the cellulosome in the anaerobic bacterium *Clostridium thermocellum*. The cellulosome is a multi-protein complex for the efficient degradation of plant cell walls. The β-sheet domain of cohesin (blue) interacts predominantly with one of the helices of the dockerin (red). The structure provides an explanation for the lack of cross-species recognition between cohesin–dockerin pairs and thus provides a blueprint for the rational design, construction, and exploitation of these catalytic assemblies. (Courtesy of Huiling Chen and Ying Xu, University of Georgia.)

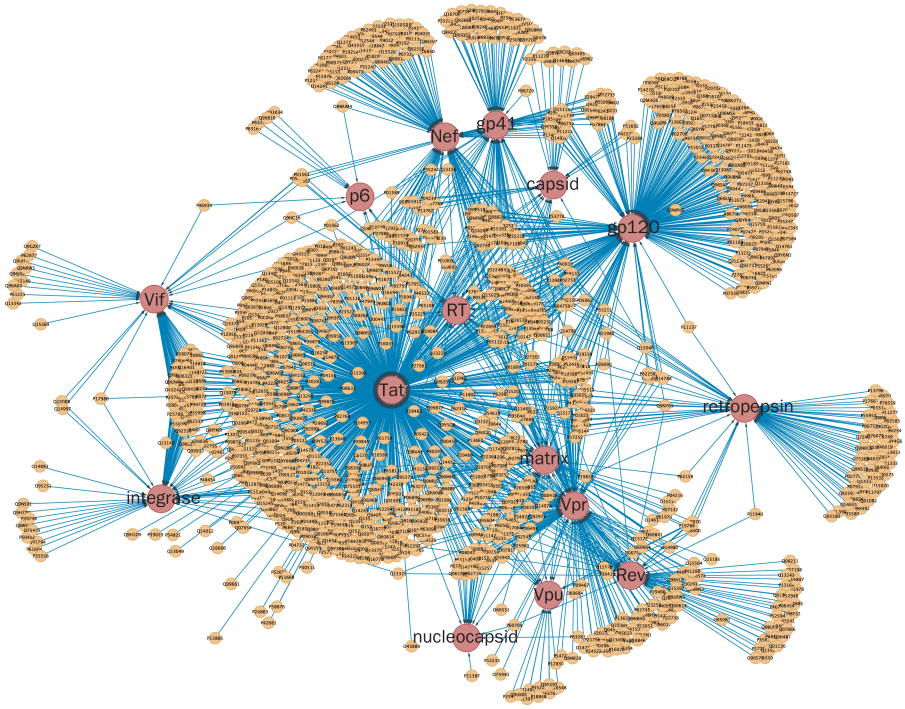


Figure 8.5 Network of 1785 unique protein–protein interactions between HIV and the human host. The graph was assembled from a database of the National Institute of Allergy and Infectious Diseases, Division of AIDS. Red nodes represent HIV proteins and light orange nodes represent host factors. It is unclear at present which interactions are direct and which are functionally relevant. (From Jäger S, Gulbahce N, Cimermancic P, et al. *Methods.* 53 [2011] 13–19. With permission from Elsevier.)

We focus here on protein-protein interactions (PPIs), which exhibit many aspects worth investigating. PPI research may be categorized into two types. First, enormous effort in recent times has been devoted to PPI networks of medium to very large sizes (**Figure 8.5**). The rationale is that deciphering which proteins interact with each other, possibly in some specific location, will provide insights into the functioning of a cell as a complex system. Probably the most prominent method for assessing large-scale PPIs has been the yeast two-hybrid assay, which uses artificial fusion proteins in a yeast cell to infer the binding partners of a protein [56, 57]. The assay is scalable, which permits the screening of interactions among very many proteins, but it often generates high numbers of false positives, which means that the test suggests many more interactions than truly exist in a living cell. For relatively small interaction networks, the antibody-based method of immunoprecipitation can verify or refute these interactions [58]. It is also possible, at least to some degree, to use computational validation methods [59].

Second, the analysis of PPIs becomes much more complicated when their dynamics is of interest or even if one just needs to know the directionality of the edges in the network. These types of analyses are challenging even if they involve only a dozen proteins or so. An example of a dynamic PPI is the transport of a protein by another protein, for example between the nucleus and the cytoplasm. Somewhat similar is the process of ubiquitination, which marks a protein for degradation. Other very important PPIs are involved in signal transduction (**Figure 8.6**). Base modules of such interactions are analyzed in a later section with simple dynamic models. Much more detail can be found in Chapter 10, which is entirely devoted to different signaling systems.

8.3 AN INCOMPLETE SURVEY OF THE ROLES AND FUNCTIONS OF PROTEINS

It is impossible to rank proteins by importance, because removal of any class or type of proteins would be disastrous. Nonetheless, if one were to sort all proteins purely by their abundance, the clear winner would be RuBisCO, a protein that most people never heard of. If you are surprised, consider that RuBisCO (ribulose-1,5-bisphosphate carboxylase oxygenase) is the first enzyme in the Calvin cycle, which is the metabolic pathway responsible for photosynthesis. It has been estimated that the

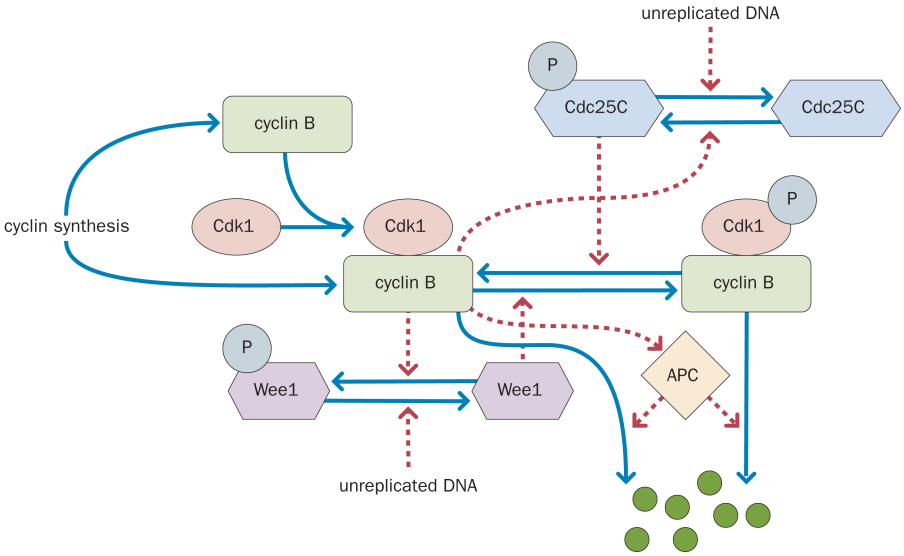


Figure 8.6 Protein–protein interaction network controlling the entry of cells into mitosis. Leland Hartwell, Tim Hunt, and Paul Nurse received the 2001 Nobel Prize in Physiology or Medicine for their discoveries of key regulators of this process. While the details are not critical here, the diagram illustrates that there is clear directionality in some signals. Unreplicated DNA blocks the entry of cells into mitosis. The process is governed by positive feedback between the Cdk1–cyclin B complex and Cdc25 and by double-negative feedback between Cdk1 and Wee1. This arrangement renders it possible for gradual mitotic cyclin synthesis to lead to a switch-like increase in Cdk1 activation. Cdk1 is inactivated owing to a negative feedback in which Cdk1 indirectly targets cyclin B for degradation through APC. (Adapted from Zwolak J, Adjerid N, Bagci EZ, et al. *J Theor Biol.* 260 [2009] 110–120. With permission from Elsevier.)

total photosynthetic biomass production on Earth is over 100 billion tons of carbon per year [60], and RuBisCo is instrumental in all of it. Photosynthetic cyanobacteria (blue algae) alone comprise an estimated biomass between 40 and 50 billion tons. This is an enormous amount, considering that all humans together make up only about 250 million tons. The most abundant protein in the human body is collagen, which accounts for between a quarter and a third of the body's total protein content [61]. Within human cells, actin is the winner [62]. We can see that even the most abundant proteins are distributed among different functional classes.

When discussing the function of a protein, one should be aware that proteins often serve more than one role. For instance, the extracellular matrix consists of a number of proteins and other substances that maintain a cell's shape. The matrix also aids the movement of cells and molecules, it is crucial for signal transduction, and it acts as a storage compartment for growth factors, which can be released by the action of proteases if they are needed for growth and tissue regeneration. The same release processes are probably also involved in the dynamics of tumor formation and cancer metastasis. Specific examples of matrix proteins are fibronectins, which attach to collagen and other proteins called integrins on the surfaces of cells. Fibronectins allow cells to reorganize their cytoskeleton and to move through the extracellular space. They are also critical in blood clotting, where they bind to platelets and thereby assist in wound closure and healing.

The following highlights the main function of some representative proteins, but many of these have secondary, nevertheless important, roles—for instance, by serving as enzymes and as structural proteins that have a strong effect on signal transduction.

8.3.1 Enzymes

Enzymes are proteins that catalyze chemical reactions; that is, they convert organic substrates into other organic products. For instance, the first enzyme of glycolysis, hexokinase, takes the substrate glucose and converts it into the product glucose 6-phosphate (see Chapter 9 for an illustration). This conversion would not occur in any appreciable amount without the enzyme, because glucose 6-phosphate has a higher energy state than glucose. The enzyme makes the conversion possible by coupling the reaction to a second reaction. Namely the required energy cost is paid for by the energy-rich molecule adenosine triphosphate (ATP), which donates one of its phosphates to the product and is thereby dephosphorylated to adenosine diphosphate (ADP), which contains less energy. A key element of this coupling of processes is that the enzyme itself is unchanged in the reaction and directly available again for converting further substrate molecules. The vast majority of all metabolic conversions in a living cell are catalyzed by enzymes in this fashion. We will discuss details of such conversions in Chapter 9, which addresses metabolism.

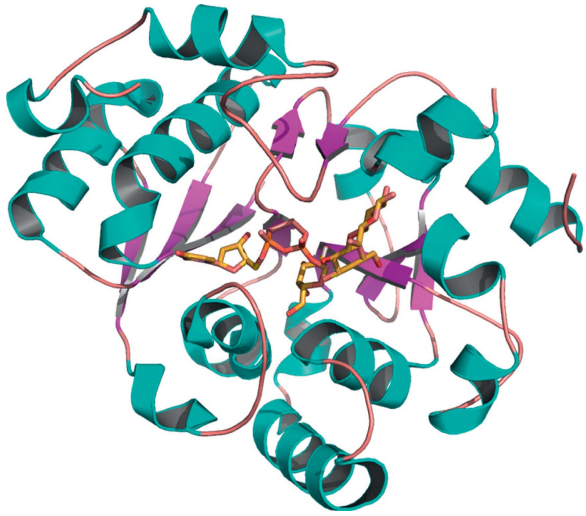


Figure 8.7 Many pathogenic bacteria produce lipo-oligosaccharides, which are composed of lipid and sugar components and resemble molecules on human cell surfaces. This molecular mimicry can deceive membrane receptors in the host, which do not recognize the invaders as foreign. As a consequence, the bacteria attach to the unsuspecting receptors and manage to evade the host's immune response. An example is the meningitis-causing bacterium *Neisseria meningitideis*, and a key tool of its success is the enzyme galactosyltransferase (LgtC), whose crystal structure is shown here. The enzyme catalyzes an important step in the biosynthesis of a specific lipo-oligosaccharide; namely, it transfers *α-d*-galactose from UDP-galactose to a terminal lactose. The enzyme (turquoise ribbons) is shown in complex with sugar analogs (UDP 2-deoxy-2-fluoro-galactose and 4′-deoxylactose; stick molecules), which respectively resemble the true donor and acceptor sugars encountered in nature. Investigations of such complexes, combined with methods of directed evolution and kinetic analysis, offer important insights into the mechanisms with which bacteria outsmart the mammalian immune system and provide a starting point for targeted drug design. (Courtesy of Huiling Chen and Ying Xu, University of Georgia.)

With respect to the enzyme itself, it is of interest to study how a protein is capable of facilitating a metabolic reaction. The details of such a process depend of course on the specific enzyme and its substrate(s), but one can generically say that the enzyme contains an active site to which the substrate molecule binds. This active site is a physical 3D structure, which typically consists of a groove or pocket in the surface of the enzyme molecule and whose shape very specifically matches the substrate (**Figure 8.7**). Because of this specificity, enzymes typically catalyze just one substrate—or one class of substrates—with high efficiency. Initially, the active site was assumed to be rather rigid, but the current understanding is that it is more flexible, thereby allowing the transient formation of a complex between enzyme and substrate. Chemical residues near to the active site act as donors of small molecular groupings or protons, which are attached to the substrate during the reaction. Once attached, the modified molecule falls off the enzyme as the product of the reaction. Other classes of enzymes act on substrates by removing molecular groups, which are then accepted by residues close to the active site. Once the product is formed, the residues are returned to their original state.

The involvement of an enzyme in a metabolic reaction is of enormous importance for several reasons. First, very many reactions have to be executed against a thermodynamic gradient. In other words, such a reaction would not be possible without an enzyme. Second, even if the product has a lower energy state than the substrate, the energy state of the substrate is still lower than a transition state that needs to be overcome before the product can be formed. An enzyme is usually needed to surmount this energy barrier. Specifically, the enzyme lowers the energy requirement of the reaction by stabilizing the transition state. Third, the cell can regulate the enzyme and thereby affect its activity. Through this mechanism, the cell is able to steer substrates into alternative pathways and toward products that are in highest demand. For instance, pyruvate may be used as the starting substrate for the generation of oxaloacetate, which then enters the citric acid cycle, or it may be used to create acetyl-CoA, which can subsequently be used in fatty acid biosynthesis or different amino acid pathways. Pyruvate can also be converted into lactate and used by some organisms for the generation of ethanol (**Figure 8.8**).

The "decision" of how much substrate should be channeled toward alternative pathways and products is the result of complex regulation. The most immediate controllers are the enzymes that catalyze the various possible conversions of the substrate. Indirectly, the decision is much more distributed among factors that regulate the activities of these enzymes. This regulation is implemented in various, distinct ways. First, metabolites within or outside the target pathways can enhance or inhibit the activity of an enzyme. This modulation occurs essentially immediately. In some cases, the product of a pathway directly binds to the active site on the enzyme, thereby competing with the substrate and slowing down the conversion of substrate to product. An alternative is **allosteric** inhibition or activation, where the

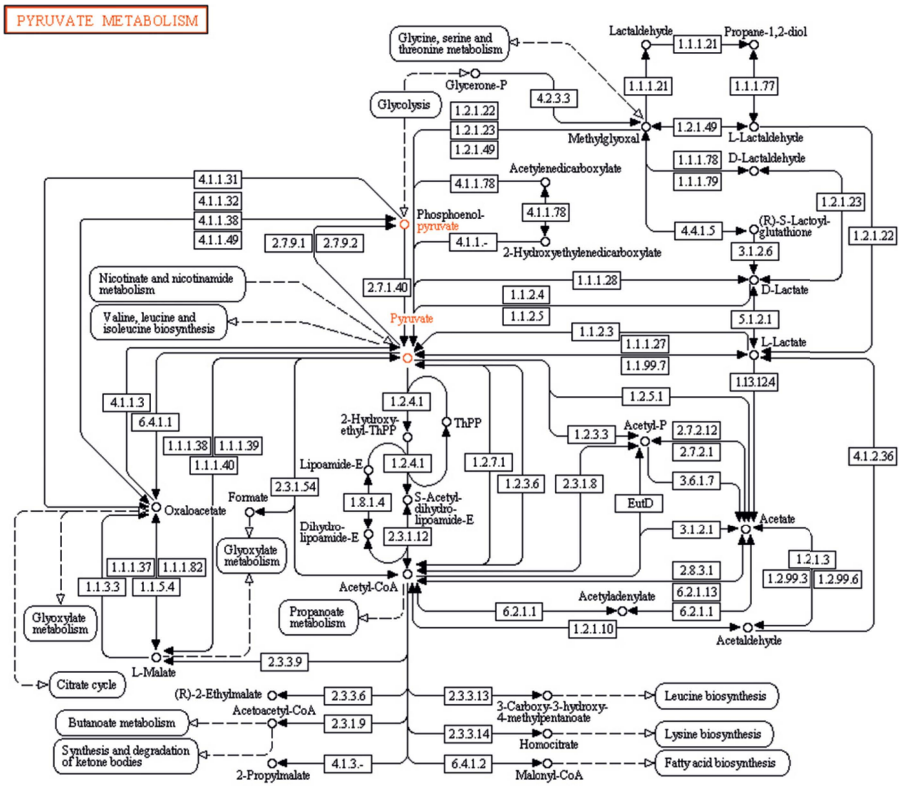


Figure 8.8 The metabolite pyruvate is involved in numerous biochemical reactions. These reactions are catalyzed by enzymes (identified by their Enzyme Commission (EC) numbers in boxes). Such pathways are discussed further in Chapter 8. (Courtesy of Kanehisa Laboratories.)

product attaches to the enzyme in a location different from the active site. This phenomenon leads to a physical deformation of the active site that alters the efficiency of substrate binding and is called **cooperativity**. The second mode of regulation is exerted by other proteins, which can quickly activate or deactivate an enzyme through the attachment or removal of a key molecular grouping, such as a phosphate or glycosyl group. More permanently, the regulatory protein ubiquitin can bind to the enzyme and mark it for degradation by the proteasome, as we discussed before. Finally, the activity of enzymes can be changed over much longer time horizons by means of gene regulation. Namely, the genes coding for the enzyme or its modulators may be up- or down-regulated, which ultimately affects the amount of enzyme present in the cell and therefore its throughput. As one might expect, the different modes of regulation are not mutually exclusive, and situations such as environmental stresses can easily evoke some or all of them simultaneously.

8.3.2 Transporters and Carriers

In addition to enzymes, transport or carrier proteins are crucially important for metabolism. Transport is needed between the inside and outside of a cell, within a cell, and outside cells, for instance, in the bloodstream. Transport across the cell membrane, in either direction, is often facilitated by **transmembrane proteins**, which accomplish their task either through active transport or through facilitated diffusion in the case of small molecules. A very important class of these proteins carries ions across the membrane; this ion transport is central to a vast number of processes, including the action and deactivation of nerve and muscle cells. We will discuss this in greater detail in Chapter 12.

Transport of small molecules within cells often occurs via diffusion. However, larger molecules are too large for this process to be efficient. Instead, they are often packed into vesicles, where they can be stored and eventually expelled somewhere else inside the cell or to the outside. Vesicles typically consist of lipid bilayers, but it is the current understanding that proteins are often involved in their formation and docking to a membrane [63]. An important example of vesicle dynamics is the movement of neurotransmitters, such as dopamine, from the nerve cell into the synaptic cleft between cells, a process that is the basis for signal transduction in

the brain [64]. Another example of vesicle transport is the uptake of lipids from the intestines into the bloodstream via the lymphatic system [65].

A different, very interesting, type of movement within cells is facilitated by kinesin and dynein proteins, which use chemical energy from the hydrolysis of ATP to move along microtubule cables [66]. This movement is involved in a number of processes, including mitosis and the transport of molecules within the axons of nerve cells. The microtubules themselves also consist of proteins.

Transport between neighboring cells can occur through different mechanisms. A particularly prevalent task is cell-to-cell communication, which is often accomplished by sending ions or other small molecules through physical channels. These channels, called **gap junctions**, are formed by proteins that span the intracellular space between neighboring cells. Specifically, these connectors consist of two half-channels that each span the membrane of one of the neighboring cells and are composed of four helices. This arrangement constitutes a positively charged entrance within the cytoplasm, a funnel leading into a negatively charged tunnel, and an extracellular cavity. The funnel itself is formed by six helices that determine the maximal size of molecules entering the channel [21]. Gap junctions can be opened or closed through rotating and sliding motion of their component proteins, called connexins, which thereby create or close a center opening (**Figure 8.9**). Important examples of gap junctions are found in the heart, where they help the cardiac myocytes contract in a coordinated fashion, making up the heartbeat (see Chapter 12), but the examples don't stop there: the same sliding design is used in camera lens apertures and for the roof of Atlanta's multi-purpose Mercedes-Benz stadium, which opened in 2017.

Transport outside the cell includes the delivery of oxygen from the arterial blood to all tissues of the body. Responsible for this process is the iron-containing metalloprotein hemoglobin, which accounts for about 97% of a red blood cell's dry mass (**Figure 8.10**). Like many enzymes, this protein is subject to potential competitive and essentially irreversible inhibition of its binding of oxygen, for instance by carbon monoxide, which is therefore very poisonous. Another carrier protein is serum albumin, which transports water-insoluble lipids through the bloodstream. Albumin is the most abundant protein in blood plasma. It is a tightly packed globular protein with lipophilic groups on the inside and hydrophilic side groups on the outside. The primary task of serum albumin is the regulation of osmotic pressure in the plasma, which is needed for an adequate distribution of fluids between the blood and extracellular spaces.

Another class of carrier proteins contains cytochromes, which operate in the electron transport chain and thus in the respiration process. They receive hydrogen

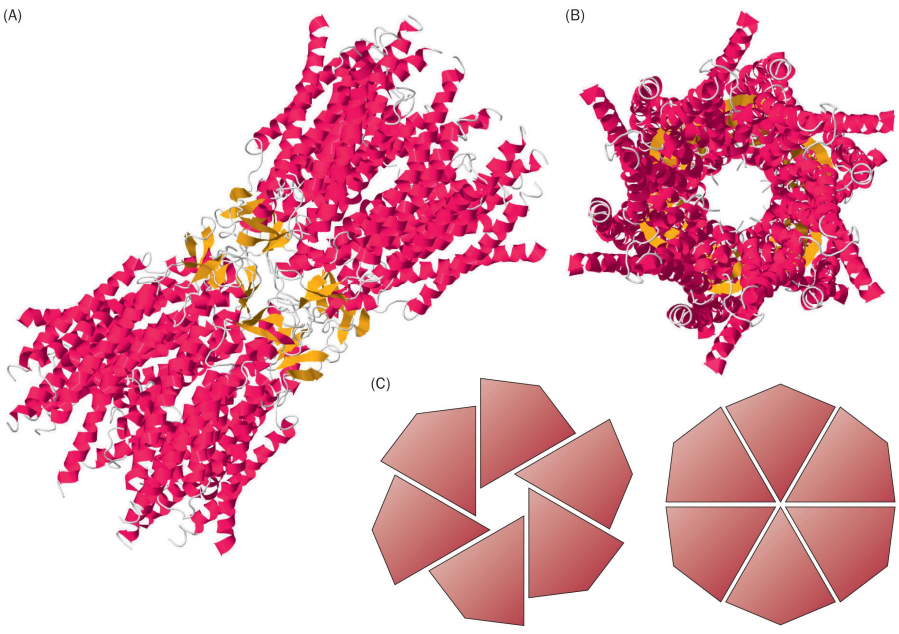


Figure 8.9 Gap junctions are composed of proteins called connexins. (A) Side view of a connexin complex. The red domains span the membranes of two neighboring cells, while the yellow domains bridge the extracellular space. (B) Top view of the same connexin complex. (C) Symbolic representation of the opening and closing of a gap junction by means of rotating connexin subunits. (Protein Data Bank: PDB 2ZW3 [2, 67].)

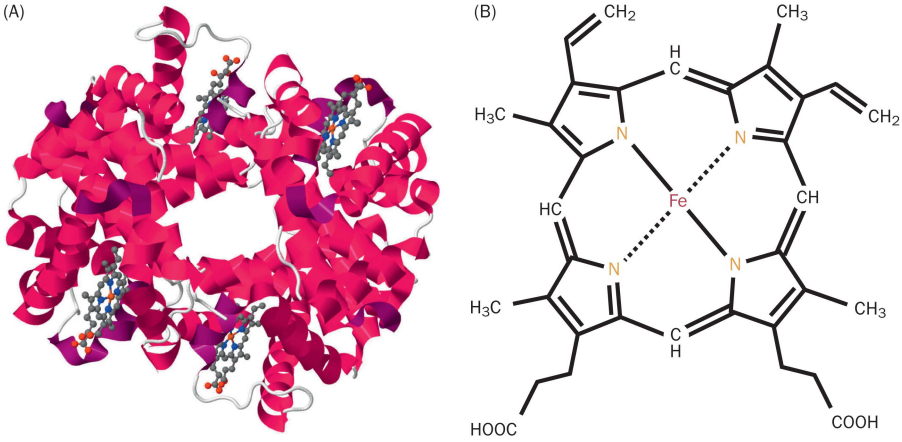


Figure 8.10 Heme and hemoglobin. (A) Human hemoglobin A is a globular metalloprotein (red) composed of four subunits that anchor the planar heme groups (the ball-and-stick assemblies) (Protein Data Bank: PDB 3MMM [2, 68]). Each heme group consists of a complex porphyrin ring with four nitrogen atoms that holds an iron (Fe^{2+}) ion, to which oxygen can bind. (From Brucker EA. *Acta Crystallogr, Sect D.* 560 [2000] 812–816.)

ions from the citric acid cycle and combine them with oxygen to form water. This process releases energy, which is used to generate ATP from ADP. Cytochromes can be inhibited by the poison cyanide, which therefore suppresses cellular respiration. In a different role, cytochromes are responsible for degrading foreign substances ("xenobiotics") in the liver.

Proteins in the extracellular matrix, such as fibronectins, are also involved with the coordinated transport of materials between blood capillaries and cells. This interstitial transport is important for the delivery of substrates. It is also the key mechanism for exposing cancer cells to chemotherapy [69].

A particularly fascinating transport protein complex is the flagellar motor, which allows many bacteria and archaea to move (**Figure 8.11**) [70]. The flagellum itself is a hollow tube that is made from the protein flagellin and contains an internal cytoskeleton composed of microtubules and other proteins that render sliding and rotating motions possible [66]. Its shaft runs through an arrangement of protein bearings in the cell membrane. It is turned by a rotor that is located on the inside of the cell membrane and consists of several distinct proteins. This flagellar motor converts electrochemical energy into torque and is powered by a proton pump in the cell membrane, which moves protons—or in some cases sodium—across the

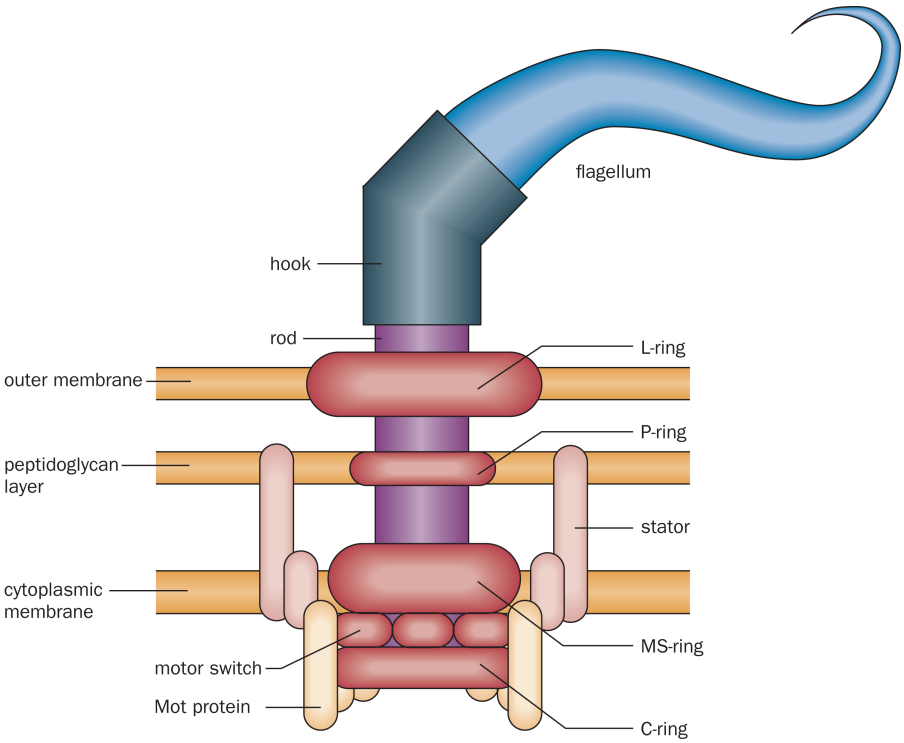


Figure 8.11 Diagram of a flagellar motor in bacteria. This complex protein arrangement allows bacteria to swim in different direction. Similar assemblies are found in sperm cells, in the Fallopian tube (where eggs need to be moved from the ovary to the uterus), and in the respiratory tract (where cilia clear the airways of mucus and debris).

membrane. The flow of protons is possible because of a metabolically generated concentration gradient. The motor can achieve several hundred rotations of the flagellum per minute, which translates into an amazing bacterial movement speed of about 50 cell lengths per second. Furthermore, by slightly changing the positioning of a specific type of motor protein, the organism can switch direction.

Flagella and similarly structured cilia are not only found in prokaryotes but are also crucial in eukaryotes, where they often have a more complex microtubular structure [71]. Prominent examples are sperm cells, cells of the Fallopian tube, which move the egg from the ovary to the uterus and endothelial cells in the respiratory tract, which use cilia to clear the airways of mucus and debris. Mammals often develop immune responses to flagellar antigens such as flagellin, with which they fight bacterial infections. Key components in these immune responses are proteins that recognize certain molecular patterns and are called Toll-like receptors (TLRs) [72], which we will discuss later in the chapter.

8.3.3 Signaling and Messenger Proteins

Cells in higher organisms communicate with other cells and with the outside on a regular basis. They send and receive signals, which can be chemical, electrical, or mechanical and must respond to these signals in a coordinated manner. This process, which is generically called **signal transduction**, is of obvious and utmost importance, and the entire Chapter 10 is devoted to it.

As just one example, let us consider a particularly important class of membrane-spanning signaling proteins, called **receptor tyrosine kinases (RTKs)**. Generically, kinases are enzymes that transfer a phosphate group from a donor such as ATP to a substrate. In the case of RTKs, this enzymatic action amounts to the initiation of signal transduction. Namely, when a ligand, such as an epidermal growth factor, cytokine, or hormone, binds to the receptor on the outside of the cell, its protein domain on the inside (i.e., in the cytoplasm) initiates auto-phosphorylation. This process leads to the recruitment of signaling proteins and thereby starts the intracellular signal transduction process. RTK-based processes are crucial in healthy cells and also seem to be important in the development and progression of cancers, as well as neurodegenerative diseases such as Alzheimer's disease and multiple sclerosis. An example of an epidermal growth factor receptor is shown in **Figure 8.12**.

In addition to receptors and signaling proteins, the signal transduction process is supported by scaffold proteins, which interact with signaling proteins in a fashion that is not entirely understood. It seems that scaffold proteins assist in the appropriate localization of pathway components to the cell membrane, the

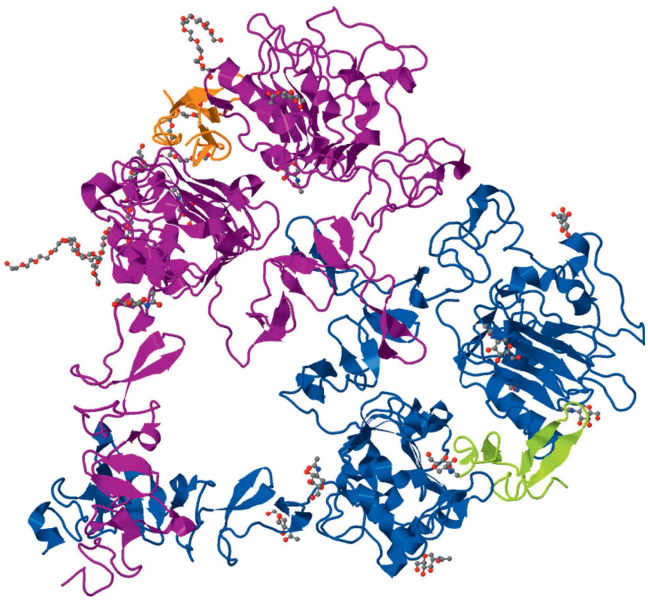


Figure 8.12 Crystal structure of the human epidermal growth factor receptor. Two types of ligands are shown. The stretched-out ligands at the top and center left are nonaethylene glycol molecules. On the right, one can see several *N*-acetyl-D-glucosamine (GlcNAc) molecules, which are components of a biopolymer in bacterial cell walls, called peptidoglycan. Interestingly, GlcNAc is also a neurotransmitter. (Protein Data Bank: PDB 3NJP [2, 73].)

nucleus, the Golgi apparatus, and other organelles. Scaffold proteins furthermore assist the condensation of DNA and are involved with the recognition of surface structures of viruses by antibodies [74].

It should be noted that some, although not all, hormones are proteins or peptides, which act over long distances. Hormones are produced by animal cells in glands, such as the pituitary, thyroid, adrenal glands, and the pancreas, are transported through the bloodstream and bind to specific **receptors** in distant locations of the body, where they signal the need for some action. Examples are the blood-sugar-regulating protein insulin, growth hormone, the blood-vessel-control hormones angiotensin and vasopressin, the exhilaration-causing endorphins, as well as the more recently identified "hunger hormones" orexin and ghrelin. Plants also produce hormones.

8.3.4 Proteins of the Immune System

Proteins are crucial players at many levels of the vertebrate immune system. Antibodies, or **immunoglobulins**, are globular plasma proteins that are produced by white blood cells (**Figure 8.13**). They are either attached to the surface of immune cells—called effector B cells—or circulate with the blood and other fluids. A typical B cell carries tens of thousands of antibodies on its surface. Antibodies bind to neutralize foreign molecules (antigens) as they are found on the surfaces of bacteria, viruses, and cancer cells, as well as nonliving chemicals, such as drugs and toxins. Owing to this binding, the invaders lose their ability to enter their target cells in the host. The antibodies also initiate the destruction of pathogens by macrophages.

Antibodies typically consist of a complex of two heavy-chain and two light-chain proteins, and while these always form the same generic structure, specific genetic mechanisms have evolved to generate tips at the ends of these proteins that contain extremely variable antigen-binding sites [76]. When B cells proliferate in response to an insult, high mutation rates increase the variability of these sites. As a consequence, the antigen-binding sites are capable of distinguishing millions of antigens by their structural features or epitopes. In autoimmune disease, the same antibody responses are directed against the body's own cells. In the biotechnology sector, antibodies represent a class of therapeutic molecules that are often elevated to bind to target proteins such as surface receptors or soluble ligands (see Chapter 13).

A second group of proteins that are crucial for immune responses is the large family of **cytokines**. These signaling molecules are used for cell-to-cell communication and drive the progression of inflammatory processes. Cytokines are produced by T helper cells, which can respond to antigen peptides within a few seconds yet require hours of signal exposure to become fully activated [77, 78].

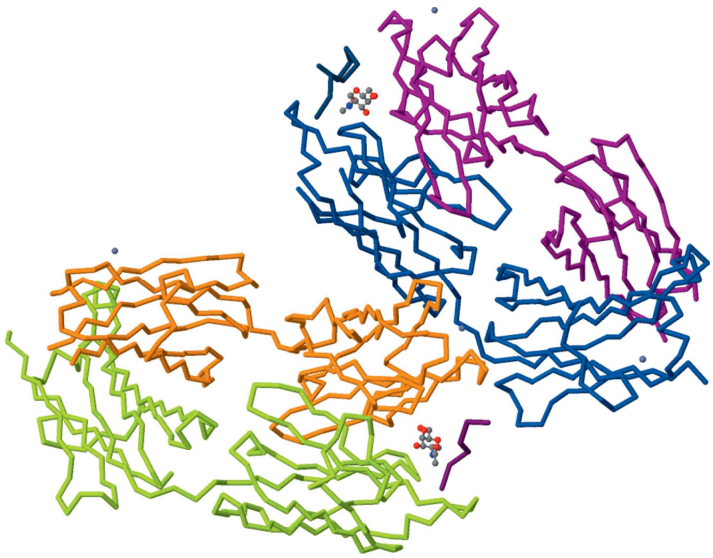


Figure 8.13 A mouse antibody recognizes a so-called Tn antigen (center top and bottom). Individual spheres are zinc molecules. Tn antigens have been associated with many cancers and made them targets for the development of cancer vaccines. Tn stands for threonine/*N*-acetyl-D-galactosamine α-*O*-serine. (Protein Data Bank: PDB 3IET [2, 75].)

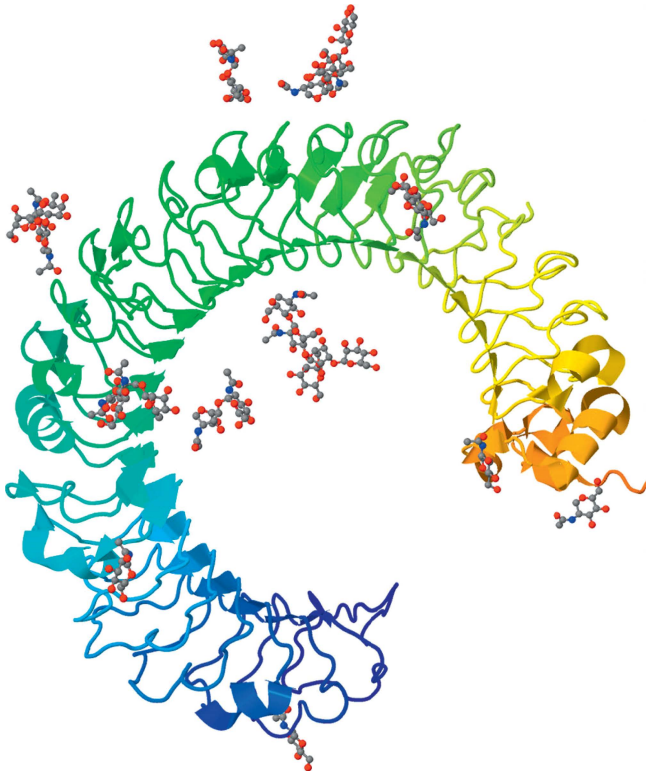


Figure 8.14 Toll-like receptors (TLRs) sense and bind various ligands, a process that initiates a signaling cascade and leads to the production of inflammation mediators. Shown here is the outer domain of the human TLR3, which recognizes double-stranded RNA, a molecular signature of viruses. The horseshoe shape is due to 23 repeated leucine-rich domains and establishes a distinct receptor for pathogen recognition, such as the cell surface sugars fucose, mannose, and *N*-acetyl-D-glucosamine (GlcNAc), which can be seen inside and outside the receptor. (Protein Data Bank: PDB 3CIG [2, 80, 81].)

Cytokines help recruit further immune cells and macrophages to the site of injury or a bacterial or viral infection. Such an infection can trigger 1,000-fold increases in cytokines, and slight imbalances in the numbers and types of cytokines can lead to "cytokine storms," sepsis, multi-organ failure, and death. Well-known examples of cytokines include interleukins and interferons. We discuss some recent modeling approaches to understanding the exceedingly complicated dynamic system of cytokines in Chapter 15. Finally, enzymes in tears and the oils of the skin contribute to the defense against invaders and thus to the innate immunity of the body.

Prominent proteins at the intersection of signaling and immune response are the **Toll-like receptors (TLRs)**—*toll* is German for "Wow!"—which function as sentinels that recognize numerous epitopes, including peptidoglycans and lipopolysaccharides, which are typical components of bacterial cell walls (**Figure 8.14**). Upon recognizing a specific epitope, the TLR activates pro-inflammatory cytokines and signaling pathways and thereby initiates a proper immune response. TLRs come in a number of variants and form a very complex functional interaction network [79].

8.3.5 Structure Proteins

Collagen is the most abundant protein in humans. It is found in many tissues that provide structural support to the body, including bone, cartilage, connective tissue, tendons, ligaments, and the discs in the spine. It also constitutes between 2% and 5% of muscle tissue and is a component of blood vessels and the cornea of the eye. Collagen supports the structure of the extracellular matrix that provides support to cells and is involved in cell adhesion and the transport of molecules within the space between cells.

Collagen consists of three polypeptide strands that contain a large number of repeated patterns of specific amino acids, such as glycine–proline–AA or glycine–AA–hydroxyproline, where "AA" represents some other amino acid. Owing to the presence of these patterns, the amount of glycine is unusually high—the only other protein with this feature seems to be a protein in silk fibers. The abundance of glycine, combined with the relatively high content of proline and hydroxypro-

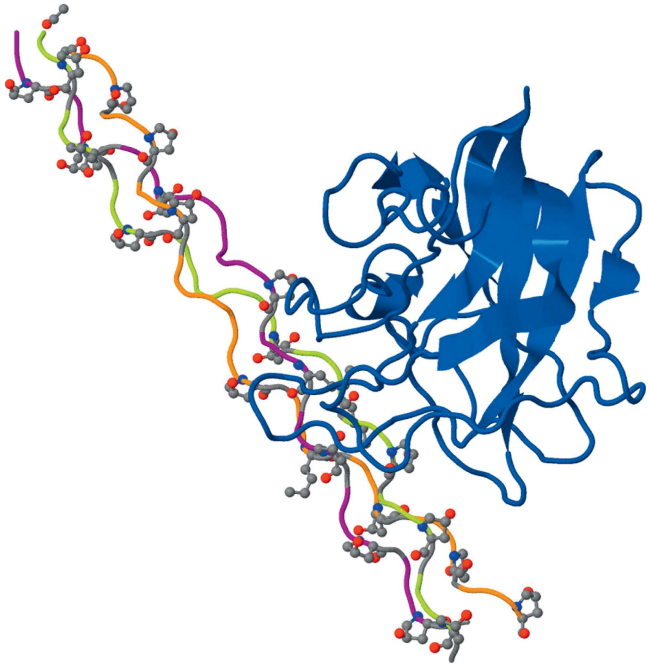


Figure 8.15 Human collagen (diagonal strands) complexed with the discoidin domain receptor DDR2 (to the right). DDR2 is a common receptor tyrosine kinase that is activated by triple-helical collagen. The binding of collagen to DDR2 causes structural changes in the surface loops of the receptor and has been linked to its activation. The receptors control cell behavior and are incorrectly regulated in several human diseases. (Protein Data Bank: PDB 2WUH [2, 82].)

line rings, causes collagen polypeptides to curl into left-handed helices. These helical strands in turn are twisted into a right-handed triple helix, which is thereby a prime example of a quaternary structure that is stabilized by numerous hydrogen bonds. Each triple helix can form a super-coiled collagen microfibril, and many of these microfibrils make up a strong yet flexible macro-structure. When hydrolyzed, collagen becomes gelatin, which has multiple uses in the food industry.

Like other proteins, collagen is involved in control tasks as well, especially in the context of cell growth and differentiation. For instance, it induces the receptor tyrosine kinase DDR2 (member 2 of the discoidin domain receptor family), which triggers autophosphorylation of targets in the cytosol (**Figure 8.15**). To permit these control tasks, some sections of the collagen polypeptide do not show the regular curling patterns described earlier.

Complementing the load-bearing strength of collagen, elastins equip tissues with shape flexibility, which is particularly important in muscle cells, blood vessels, the lungs, and skin. Like collagen, elastins contain high amounts of glycine but also similar amounts of proline, valine, and alanine. Fibroblasts secrete the glycoprotein fibrillin into the extracellular matrix, where it is incorporated into microfibrils that are thought to provide a scaffold for the deposition of elastin.

Another structural protein is keratin, which is the base material for skin, hair, nails, horns, feathers, beaks, and scales. Keratin contains many α-helices, stabilized with disulfide bonds between cysteine residues. Similar to collagen, three α-helices are twisted around each other into a protofibril. Eleven of these protofibrils form a tough microfibril, in which nine protofibrils are wound around two center protofibrils. Hundreds of microfibrils form a macrofibril. Interestingly, the toughness of keratin is only matched by chitin, which is not a protein but a long-chain polymer of *N*-acetylglucosamine that provides strength to the protective armor of crustaceans like crabs and lobsters and is also found on the surfaces of bacteria.

Actin is a globular protein in essentially all eukaryotic cells. It has two important roles. First, it interacts in muscle cells with the protein myosin. Actin polymerizes to form thin filaments, while myosin forms thick filaments. Together they form the contractile unit of the muscle cell (see Chapter 12). Second, actin interacts with cellular membranes. In this role, actin and the proteins cadherin and vinculin form microfilaments. These microfilaments are components of the cytoskeleton and participate in numerous fundamental cellular functions, including cell division, cell motility, vesicle and organelle movement, and the creation and maintenance of cell shape. They are also directly involved in processes such as the phagocytosis of bacteria by macrophages.

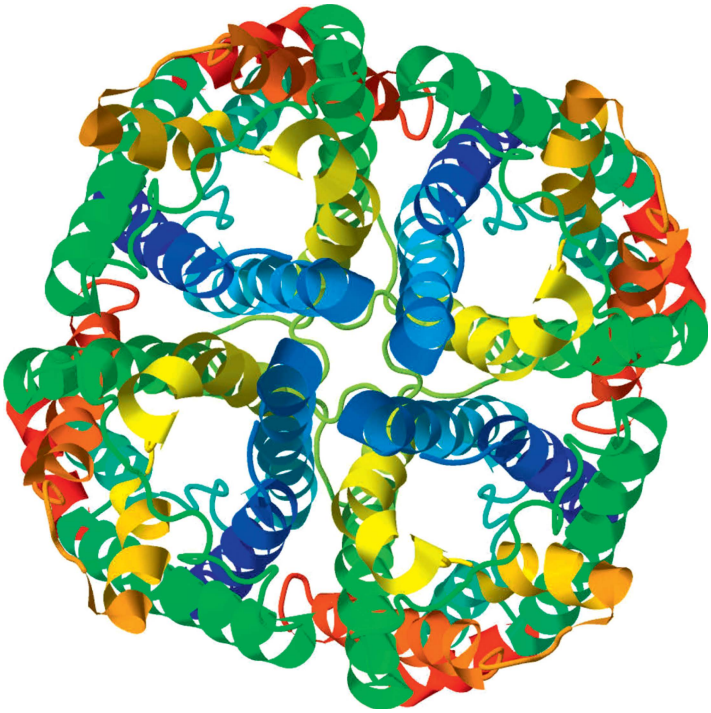


Figure 8.16 The vertebrate eye lens protein aquaporin-0 (AQP0). AQP0 consists of a complex of four subunits, which are arranged in a three-dimensional lattice. According to current understanding, AQP0 proteins mediate the connections among the very tightly packed fiber cells of the lens. (Protein Data Bank: PDB 2C32 [2, 84].)

The lens of the mammalian eye is a complicated structure that contains fibers composed of long transparent cells. More than a third of its weight is due to proteins. These proteins, which have fascinated scientists for over 100 years, include crystallins [83], collagen, and the membrane protein aquaporin [84], which maintains an unusually tight packing between neighboring cells (**Figure 8.16**). Intriguingly, there is very little turnover in the lens, and humans retain their embryonic lens proteins throughout life. Age-related changes in the lens, such as cataracts, are therefore difficult to repair.

The versatility of proteins has resulted in uncounted roles associated with the maintenance of shape and structure. In many cases, these roles are performed with "mixed" molecules that consist of proteins and something else. For instance, apolipoproteins, together with several types of lipids, can form lipoproteins, which serve as carriers for lipids in the blood and also as enzymes and structural proteins. Well-known examples are high-density and low-density lipoproteins (HDLs and LDLs), which, respectively, carry cholesterol from the body's tissues to the liver and vice versa. They are sometimes referred to as "good" and "bad" cholesterols, respectively.

Glycoproteins contain sugars (glycans) that are bound to the side chains of the polypeptides. Glycans are often attached during or after translation of the mRNA into the amino acid chain, in a process called glycosylation. Glycoproteins are often integrated into membranes, where they are used for cell-to-cell communication. Prominent examples are various hormones and mucins, which are contained in the mucus of the respiratory and digestive tracts. Glycoproteins are important for two aspects of immune responses. First, some of these proteins are mammalian immunoglobulins or surface proteins on T cells and platelets. Second, glycoproteins form structures on the surfaces of bacteria such as streptococci, which are recognized by the host's immune system.

8.4 MODELING PROTEIN ACTIVITY AND DYNAMICS

Proteins are drivers of dynamic activities at many temporal and spatial scales. Some proteins interact with other molecules on a timescale of seconds or even faster, as is the case with enzymes, whereas the dynamic changes in some proteins may take many years, as we can see in the very slow bonding of sugars to proteins in the lens of the eye, which clouds the lens in advanced age and can lead to cataracts [85]. Correspondingly, investigations of proteins very often target how

proteins interact with each other and how these interactions or other molecular association/dissociation affect their activity. As is to be expected, these investigations can take very different forms. We already discussed in Chapter 7 the prominent role of proteins as transcription factors that govern the transcription of genes into mRNAs. The docking between small ligands and proteins is the basis for enzyme action. It is of enormous importance, because it permits the control of flux through different pathways diverging from a branch point and thereby the fast adaptation of an organism to changing conditions. This aspect of proteins will be discussed in Chapter 9. Interactions between proteins are often underlying signaling processes; these will be discussed in Chapter 10.

The following sections describe some base models of changes in the state of a protein that correspond to its activation or deactivation. These simple processes are the foundation of much more complicated systems that control much of physiology in the body.

One fundamental process involving proteins is the attachment or detachment of a phosphate group, which switches the activity state of the protein. **Figure 8.17** shows the essential features of this process, where a signal activates a kinase k, which is an enzyme that attaches a phosphate group to the protein P_0, thereby activating it. Hundreds of human kinases have been identified. The phosphorylated protein P_1 can be returned to the unphosphorylated form P_0 by the activity of a phosphatase p. One may account for the fact that P_0 may be synthesized (s_0) and that either form of the protein may be degraded by processes d_0 and d_1.

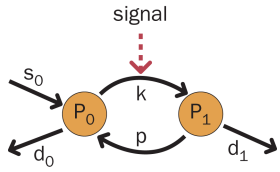


Figure 8.17 Generic activation and deactivation of a protein. In the presence of a signal, a kinase k attaches a phosphate group to inactive protein P_0. The phosphorylated protein P_1 is activated and act as a protein or signaling molecule. It can also be dephosphorylated toward P_0 by a phosphatase p. P_0 may potentially be synthesized and both P_0 and P_1 may be degraded.

It is straightforward to set up a model of the protein activation system, for instance, in the format of mass-action equations, where the signal is implemented as a factor of the form $(S+1)$. Without a signal present, this term is 1, and if a signal is present, the term is set to a much higher value. For this first illustration, let's assume that there is no protein synthesis or degradation (see Exercise 8.30). The equations take the form

$$\begin{aligned}\dot{P}_0 &= p\,P_1 - k\,P_0\,(S+1)\\ \dot{P}_1 &= k\,P_0\,(S+1) - p\,P_1\end{aligned} \tag{8.1}$$

Even without simulations, we can derive a lot of insight from the equations in their symbolic form. One notes that the two right-hand sides are the same, except for a minus sign. What does that mean? If we add the two equations together, the result is $\dot{P}_0 + \dot{P}_1 = 0$, which is equivalent with saying that $P_0 + P_1 =$ constant. In other words, the total concentration of protein, P_T, remains unchanged throughout the dynamics of the phosphorylation-dephosphorylation process. This makes sense in terms of conservation of mass, as there is neither an influx into the system nor an efflux out of the system: the system is closed.

The constancy has an effect on the initial conditions (or vice versa, if you will). Namely, if we choose any positive value for $P_0(t_0)$, we know that $P_1(t_0) = P_T - P_0(t_0)$. What value should we choose for $P_0(t_0)$? From a mathematical point of view, any nonnegative number would do, but from a biological standpoint, most of the protein mass should be in P_0, because if much were in P_1, the system would be active even with a signal. In other words, the system would not really respond to an incoming signal.

We can compute the steady state by setting the two equations equal to zero, which yields

$$P_{1,ss} = \frac{k}{p}(S+1)P_{0,ss} \tag{8.2}$$

The result is the same for both equations, so there is no unique solution. However, if we evoke the fact that the total protein concentration is constant, we may replace $P_{0,ss}$ with $P_T - P_{1,ss}$, which leads to

$$P_{1,ss} = \big(k(S+1)/p\big)\big(P_T - P_{1,ss}\big) \tag{8.3}$$

and

$$P_{1,ss} = \frac{k(S+1)/p}{1+k(S+1)/p}P_T\,, \tag{8.4}$$

from which we obtain

$$P_{0,ss} = P_T - \frac{k\frac{(S+1)}{p}}{1+k\frac{(S+1)}{p}}P_T$$

$$= P_T\left(1-\frac{k\frac{(S+1)}{p}}{1+k\frac{(S+1)}{p}}\right) = P_T\left(\frac{1}{1+k\frac{(S+1)}{p}}\right). \tag{8.5}$$

Suppose that there is initially no signal and that the system is in a steady state. As we discussed earlier, most of the protein mass should initially be in P_0 for a later signal to be effective. If so, we obtain $P_{0,ss} \gg P_{1,ss}$, which yields

$$\frac{1}{1+k(S+1)/p} \gg \frac{k(S+1)/p}{1+k(S+1)/p} \tag{8.6}$$

For $S = 0$, simple rearrangement of terms yields the following condition for a suitable initial steady state: $P \gg k$.

Taking these results together, we still have the choice of numerical values, but we have clear constraints governing the system for effective operation.

Let's set $P_0 = 10$, $P_1 = 1$, $k = 0.01$ and $p = 0.1$. As a base simulation, start with $S = 0$, reset S to a high value later (e.g., $S = 100$ at $t = 5$) and set the signal back to 0 at a later time (e.g., $S = 0$ at $t = 15$). The result is shown in **Figure 8.18**. The initial values determine the total amount of protein, P_T, the magnitudes of k and p characterize the speed of the response, while their ratio determines the initial steady-state distribution of mass between P_0 and P_1 and S affects the steady-state values of P_0 and P_1 once the signal is active. Exercise 8.31 explores these statements further.

Many variations on this theme of activation-deactivation are possible. An example is positive feedback exerted by the activated form on the kinase (see Exercise 8.32). Furthermore, if this feedback is present and the two forms of the protein are allowed to diffuse on a cell surface, the cell can polarize [86]. Slightly more complex feedback combined with diffusion can lead to complicated wave patterns of activity [87].

Many diseases involve the binding of a small molecule, called a ligand, to a cell surface receptor. This binding is the basis of many types of signal transduction where, in uncounted variations, the ligand causes a cellular response, such as the translocation of a transcription factor into the nucleus. A simple, generic scheme is shown in **Figure 8.19**. The cell produces receptors R (with rate k_R) and positions them on its membrane. A specific extracellular ligand L, such as a steroid hormone, can bind to these receptors (with rate k_{on}), forming a complex C in a reversible

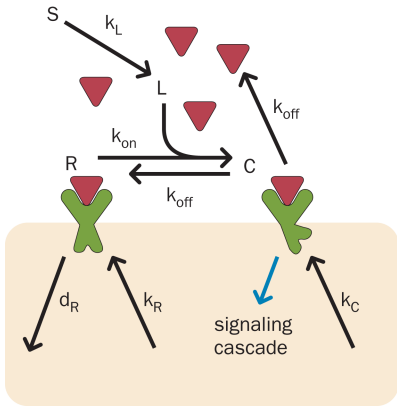


Figure 8.19 Generic receptor-ligand binding system. A source S provides ligand L, which can bind to receptor R, thereby forming a complex C, which activates an intracellular signaling cascade. The lower-case labels identify the rates of processes (see Table 8.3).

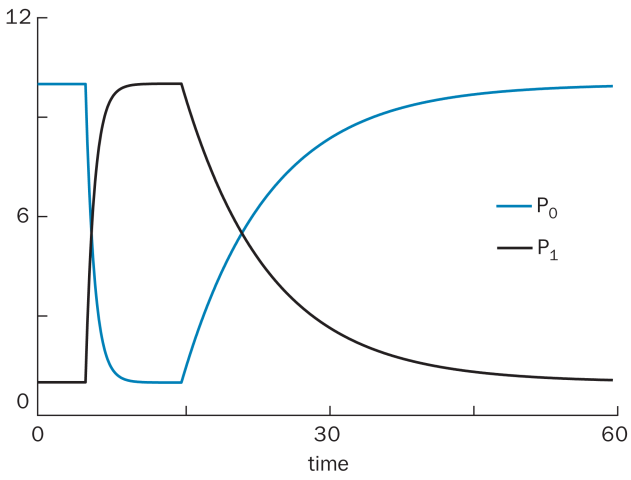


Figure 8.18 Simulation of the phosphorylation-dephosphorylation system in (8.1). The parameter settings are: $P_0 = 10$, $P_1 = 1$, $k = 0.01$, $p = 0.1$, $S = 0$ at $t = 0$, $S = 100$ at $t = 5$, and $S = 0$ at $t = 15$.

process with rate k_{off}. The complex formation often incurs a conformational change in the receptor, which inside the cell activates a signaling cascade (for a review, see [88]). Both the free receptor and the complex may be internalized and degraded in lysosomal compartments (d_R and d_C). The ligand is provided by a source S with rate k_L. Applications of this scheme to various receptors may be found in [89].

The different molecular events constituting the binding process can be formulated directly as a set of mass-action equations (Chapter 5); they take the form

$$\begin{aligned}\dot{R} &= k_R - k_{on} R L + k_{off} C - d_R R \\ \dot{C} &= k_{on} R L - k_{off} C - d_C C \\ \dot{L} &= -k_{on} R L + k_{off} C + k_L S \\ \dot{S} &= -k_L S. \end{aligned} \tag{8.7}$$

We initiate the system at the steady state, where no ligand is present, and represent the situation with the source for the ligand being empty $S(0) = 0$. As a consequence, no complex can be formed. Thus, initially we have $S(0) = L(0) = C(0) = 0$. Because the system is in a steady state, we can compute the number of free receptors, $R(0)$, by setting the first equation equal to zero and eliminating the terms containing L and C. The result is $R(0) = k_R/d_R$. Choosing more or less arbitrary positive values for all rates (Table 8.3) yields a fully defined dynamic system.

Suppose that ligand becomes available at time $t = 10$. We can model this situation by resetting the ligand source S from initially 0 to a value of, say, 200. It is a good idea to follow the logic of the diagram in order to predict the dynamics of the system. We can then look at the simulation result (**Figure 8.20**) for confirmation (or revision of our prediction). Initially, the system is at the "trivial" steady state, where no ligand is present. As soon as S is reset at $t = 10$, it starts delivering L, which quickly increases. Most of L binds to R, thereby forming C. With S decreasing toward zero, the system eventually begins returning to its original trivial steady state. Other simulations are explored in Exercises 8.33–8.34.

Many variations of the activation-deactivation (AD) system in (8.1) and the receptor-ligand (RL) binding system in (8.7) have been proposed and using the two as modules in more complicated combinations have been designed to model complex cellular phenomena. The most prominent example is the MAP-kinase signaling pathway, which is discussed in Chapter 10 at great length, along with several

TABLE 8.3: SETTINGS OF PARAMETER VALUES AND INITIAL VALUES FOR THE MODEL IN (8.7)

k_R	k_{on}	k_{off}	d_R	d_C	k_L	$R(0)$	$L(0)$	$C(0)$	$S(0)$
2	0.005	0.0005	0.04	0.03	0.05	k_R / d_R	0	0	0

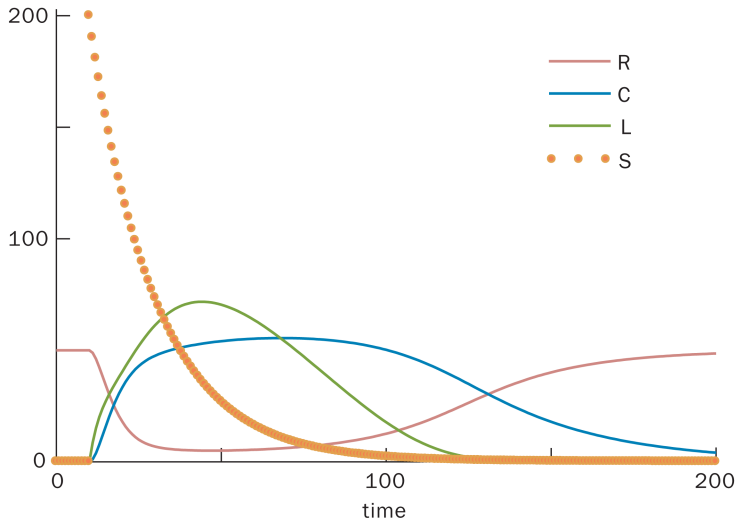


Figure 8.20 Simulation result for setting S = 200 at time t = 10. The availability of *L* leads to complex formation and robust signaling.

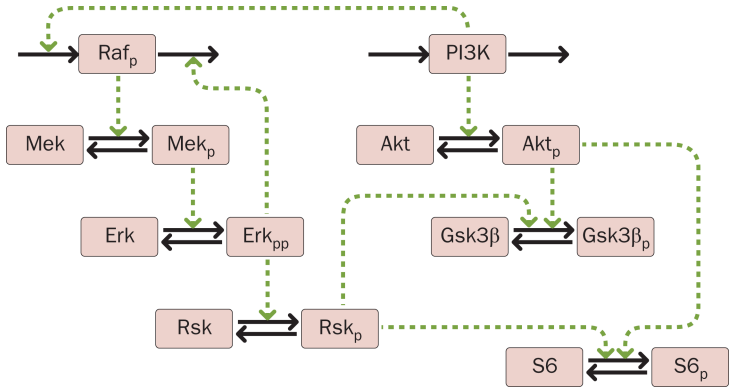


Figure 8.21 Signaling pathway controlling an epithelial-mesenchymal transition. The system consists of two parallel signaling cascades with crosstalk. Adapted from [90].

other signaling systems. As a more complicated example of this type of signaling with crosstalk between two signaling cascades, Wade et al. [90] studied a signaling system controlling the important epithelial-mesenchymal transition (EMT), which often initiates cancer metastasis (**Figure 8.21**). The model was designed based on experimental knowledge regarding the AF-MEK-ERK-RSK-S6 and PI3K-AKT-GSK3β-S6 signaling pathways [91–94] and included known crosstalk between the two pathways at the levels of PI3K to RAC [95] and RSK to GSK3β [92]. Chapter 13 discusses the question of whether a pharmaceutical company should develop an antibody against a receptor or its ligand, in order to disrupt a disease process most effectively [96, 97].

Good reviews of mathematical models of generic intracellular signaling pathways and of pathways involving the important cytokine TNFα and the transcription factor NF-κB are [98] and [99], respectively. More complicated models contain partial differential equations, which permit spatial considerations. Examples of such models for autocrine and paracrine signaling and calcium signaling include [100] and [101].

Chapter 9 analyzes metabolism, which is clearly driven by proteins. However, in metabolic analyses, the focus is almost always on substrates that are converted into products, and although proteins are essential for this conversion, they are usually not discussed much explicitly, except that they can be regulated and thereby permit the rerouting of metabolism from one pathway to another, according to changing cellular demands. This regulation is often exerted by other metabolites, but it may also be governed by physical factors such as heat and osmotic pressure (e.g., [102]).

8.5 EXERCISES

8.1 Search the literature to determine the relative frequency of amino acids in different biological systems.

8.2 Compute the theoretical number of different proteins consisting of 100, 1,000, or 27,000 amino acids.

8.3 Make intuitive predictions of how much the total variability of proteins of a given length would increase if pyrrolysine and selenocysteine were found as often as standard amino acids. Check your predictions with calculations.

8.4 Is the number of possible proteins with a fixed number of amino acids affected by the relative frequencies with which different amino acids are found? Argue for or against a significant effect and/or do some computations.

8.5 Explore the visualization options of the PDB and summarize the highlights in a report.

8.6 Search the literature and the Internet for important physical, chemical, and biological properties of the widely found proteins RuBisCo, collagen, and actin.

8.7 Explore whether plants have proteins—or even classes of proteins—that are not found in animals.

8.8 Search the literature for a visualization of the allosteric inhibition site of an enzyme.

8.9 Retrieve from the literature a picture of an enzyme as it binds to its substrate.

8.10 Discuss differences and similarities between prokaryotic flagella and eukaryotic cilia and their molecular motors.

8.11 In a short report, explain differences between hormones and pheromones. Include a discussion of broad chemical categories that contain hormones and pheromones.

8.12 Discuss in a one-page report how nature accomplishes the enormous variety and adaptability among antibodies.

8.13 Discuss the terms "pro-inflammatory" and "anti-inflammatory" in the context of cytokines.

8.14 Discuss in a report advantages and disadvantages of different separation techniques for proteins.

8.15 Explore the literature and the Internet and describe the concepts of computer algorithms that reconstitute peptides from digestion fragments.

8.16 Determine three reasons why protein structure prediction is challenging.

8.17 Survey the literature and the Internet for different types of membrane proteins. Write a report describing their properties and functions.

8.18 Explain in more detail the "positive feedback between the Cdk1–cyclin B complex and Cdc25 and the double-negative feedback between Cdk1 and Wee1" in Figure 8.6.

8.19 What would be the steps for setting up a dynamic model of the system in Figure 8.6? Once you have determined a strategy, study the article by Zwolak and collaborators [103] and compare your strategy and theirs.

8.20 According to the Central Dogma of molecular biology (see Chapter 7), DNA is transcribed into RNA, which is translated into protein, which in this case is an enzyme that catalyzes a metabolic reaction. Develop a simple cascaded model to describe these processes, as shown in **Figure 8.22**, where X_1, X_2, X_3 are mRNA, protein, and a metabolite, respectively, X_4, X_5, and X_6 are precursors (explain what these should be), and X_7 is a transcription factor or activating metabolite. Set up equations with power-law functions (Chapter 5) and select reasonable parameter values. Analyze what happens if a gene is expressed. Study what happens if X_3 and X_7 are the same metabolite.

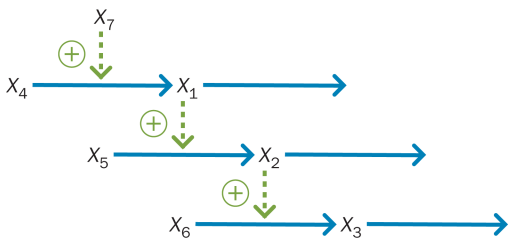


Figure 8.22 Schematic of the Central Dogma. X_1 represents an expressed gene, X_2 a completed mRNA, and X_3 a finished protein. X_4, X_5, and X_6 represent the unexpressed gene, ribonucleotides, and amino acids, respectively. X_7 is a transcription factor or activator.

8.21 How would you model post-translational modifications and alternative splice variants in the model in Exercise 8.20?

8.22 Assume that metabolite X_3 in Exercise 8.20 is the substrate for a linear pathway beginning with Z_1, leading to Z_2, Z_3, Z_4, and so on, until the final product Z_{10} is synthesized. Suppose Z_{10} strongly represses transcription at the top level of the cascade. Compare simulation results with those in Exercise 8.20.

8.23 Suppose a protein can be unphosphorylated, phosphorylated in some position A, phosphorylated in some position B, or phosphorylated in positions A and B. Sketch a diagram of the different phosphorylation states and transitions between them. Design an ordinary differential equation (ODE) model describing the transitions and states. How do you assure that the total amount of protein does not change over time? Suppose an external regulator inhibits one or two particular transitions. What is its effect?

8.24 Consider the same phosphorylation states as in Exercise 8.23. Design a Markov model (Chapter 4) describing the transitions and states. Compare features of the model and of simulations with those in Exercise 8.23. Is it possible to account for external regulators?

8.25 Many signaling cascades are based on the phosphorylation and dephosphorylation of proteins and have a structure as shown in simplified form in **Figure 8.23**. Using methods of systems analysis with differential equations, set up a model for the cascade, select reasonable parameter values, and perform sufficiently many simulations to write a confident report about the dynamical features of the cascade.

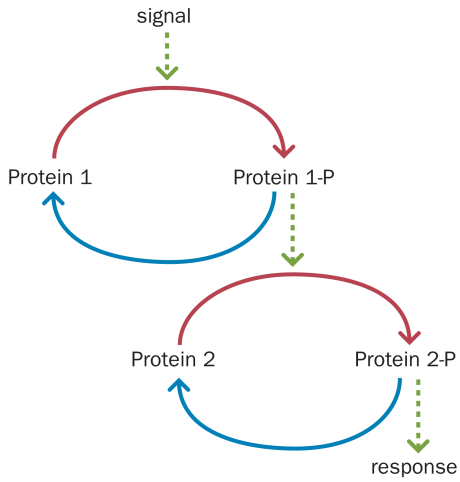


Figure 8.23 Simplified signaling cascade. The protein at either level may be phosphorylated ("-P") or dephosphorylated. Only the phosphorylated form of Protein 1 triggers the second level of the cascade.

8.26 Search the literature and the Internet for a protein–protein interaction network whose structure was inferred with computational means. Write a one-page summary of the study.

8.27 Akama et al. [104] describe the protein–protein interaction network of the cellular retinoic acid binding protein CRABP1, which is involved in the development of the nervous system (**Figure 8.24**). What standard types of numerical features that characterize this network can you extract?

8.28 Predict the role of the value of S in the signaling system (8.1) when the signal is active and test it with simulations in PLAS or some other software.

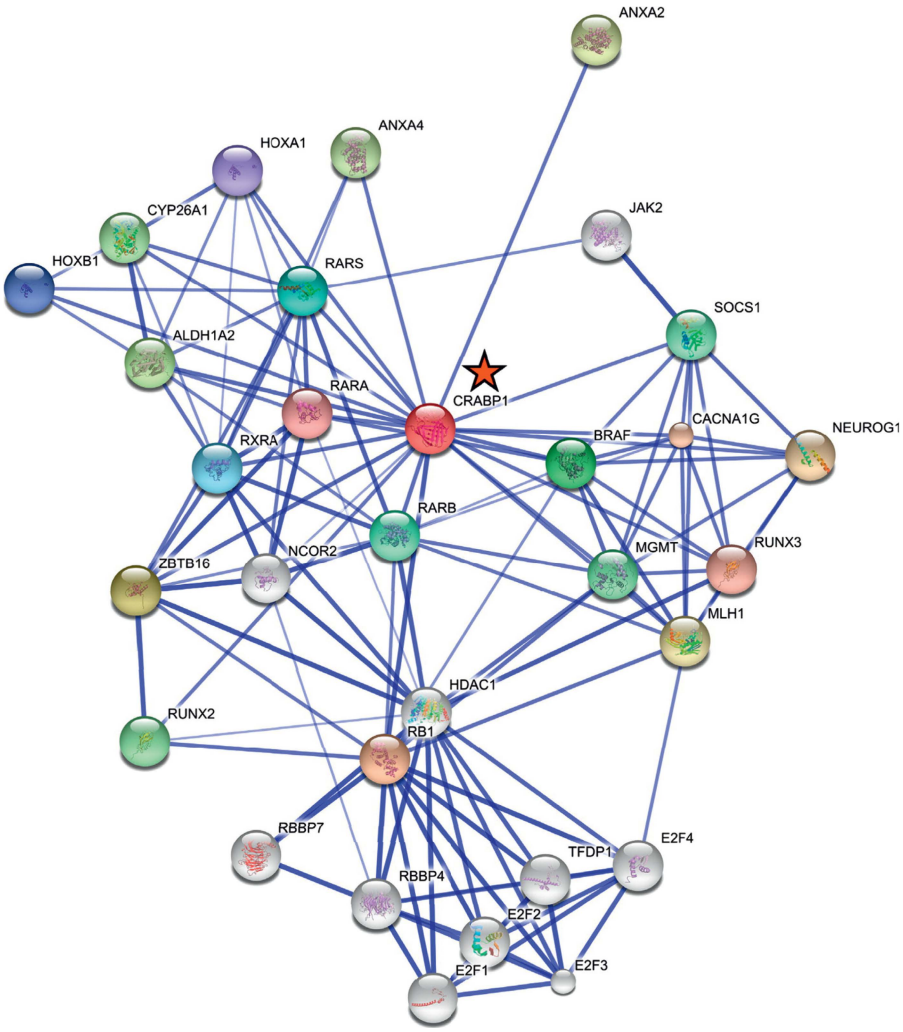


Figure 8.24 Protein–protein interaction network of CRABP1. It is evident that different nodes exhibit distinct connection patterns (see Chapter 3). (From Akama K, Horikoshi T, Nakayama T, et al. [*Macaca fascicularis*] *in vitro Biochim Biophys Acta.* 1814 [2011] 265–276. With permission from Elsevier.)

8.29 Keep $k = 0.01$ in the signaling system (8.1) but increase and decrease p between 0.05 and 5. Predict what the response in each case will be and then test the prediction in PLAS or some other software. Before you change S to a high value, allow the system to reach the initial steady state for $S = 0$.

8.30 Use the baseline settings of the signaling system in (8.1) but account for protein synthesis and degradation. Set $s_0 = 0.011$ and $d_0 = d_1 = 0.001$. The steady state for $S = 0$ should be about the same as before ($P_0 \approx 10.01$, $P_1 \approx 0.99$). Test the response of the system if $S = 0$ at $t = 0$, $S = 100$ at $t = 10$, and $S = 0$ at $t = 100$. Increase s_0, d_0, and d_1 by the same factor f, where f is 10, 100, 1,000, and 10,000. Interpret the results.

8.31 Predict what will happen in the activation-deactivation system (8.1) if:

1. The ratio of k_{on}/k_{off} is "reversed" to 0.0005/0.005.
2. $k_R = 0$
3. k_L is increased by a factor of 10.
4. The internalization rates d_R and d_C are increased by a factor of 10.

Test your predictions with simulations and explain the results.

8.32 Add to the baseline activation-deactivation model in Eqn. (8.1) positive feedback (in the form of a simple power-law) that is exerted by the activated form of the protein and affects the kinase reaction.

1. Show this feedback in a diagram of the system. Ignore protein synthesis and degradation
2. Implement the change in PLAS or some other software and set the power to 1.
3. Explore differences in the responses between this model and the baseline model with respect to difference signal strengths at $t = 5$. Interpret your findings.

8.33 Model the following situation with the receptor-binding model (8.7): At the beginning, $S(0) = 0$, as before. From time $t = 10$ to $t = 30$, S is constant at a value of 200. S subsequently decreases with rate k_L. How do these settings affect the signaling of the system?

8.34 In the receptor-binding model (8.7), make S an independent variable with value of 10 or 100. For each case, compute with algebraic means the steady state of the system in terms of S. Confirm the results in PLAS and test each steady state for stability.

FURTHER READING

Chautard E, Thierry-Mieg N & Ricard-Blum S. Interaction networks: From protein functions to drug discovery. A review. *Pathol Biol (Paris)*. 57 (2009) 324–333.

Latterich M, Abramovitz M & Leyland-Jones B. Proteomics: New technologies and clinical applications. *Eur J Cancer*. 44 (2008) 2737–2741.

Nibbe RK, Chowdhury SA, Koyuturk M, et al. Protein–protein interaction networks and subnetworks in the biology of disease. *Wiley Interdiscip Rev Syst Biol Med.* 3 (2011) 357–367.

Shenoy SR & Jayaram B. Proteins: Sequence to structure and function—current status. *Curr Protein Pept Sci.* 11 (2010) 498–514.

Twyman R. *Principles of Proteomics*. Garland Science, New York, NY, 2004.

Yates JR, Ruse CI & Nakorchevsky A. Proteomics by mass spectrometry: Approaches, advances, and applications. *Annu Rev Biomed Eng.* 11 (2009) 49–79.

Zvelebil M & Baum JO. *Understanding Bioinformatics.* Garland Science, New York, NY, 2008.

REFERENCES

1. BioGRID, http://www.thebiogrid.org/, 2010.
2. PDB, *RSCB Protein Data Bank*, http://www.pdb.org/pdb/home/home.do, 2010.
3. Swiss-Prot, http://ca.expasy.org/sprot/, 2010.
4. UniProt, http://www.uniprot.org/, 2010.
5. Zvelebil, M. and J.O. Baum, *Understanding Bioinformatics*. 2008, New York, London: Garland Science, p. xxiii + 772.
6. HPA, *Human Protein Atlas*, http://www.proteinatlas.org, 2010.
7. Atkins, J.F. and R. Gesteland, The 22nd amino acid. *Science*, 2002. **296**(5572): p. 1409–1410.
8. Stadtman, T.C., Selenocysteine. *Ann Rev Biochem*, 1996. **65**: p. 83–100.
9. Brown, T.A., *Genomes 3*. 3rd ed. 2007, New York, London: Garland Science.
10. Zhou, F., V. Olman, and Y. Xu, Barcodes for genomes and applications. *BMC Bioinform*, 2008. **9**: p. 546.
11. Zhang, F., et al., Differential arginylation of actin isoforms is regulated by coding sequence-dependent degradation. *Science*, 2010. **329**(5998): p. 1534–1537.
12. Seppälä, S., et al., Control of membrane protein topology by a single C-terminal residue. *Science*, 2010. **328**(5986): p. 1698–1700.
13. Gsponer, J., et al., Tight regulation of unstructured proteins: From transcript synthesis to protein degradation. *Science*, 2008. **322**(5906): p. 1365–1368.
14. Anfinsen, C.B., Principles that govern the folding of protein chains. *Science*, 1973. **181**(96): p. 223–230.
15. Levinthal, C., How to fold graciously, in *Mossbauer Spectroscopy in Biological Systems: Proceedings of a Meeting Held at Allerton House, Monticello, Illinois*, J.T.P. DeBrunner and E. Munck, Editors. 1969, Champaign, IL: University of Illinois Press, p. 22–24.
16. Ranson, N.A., et al., ATP-bound states of GroEL captured by cryo-electron microscopy. *Cell*, 2001. **107**: p. 869–879.
17. Clare, D.K., et al., An expanded protein folding cage in the GroEL-gp31 complex. *J Mol Biol*, 2006. **358**(3): p. 905–911.
18. Hatahet, F. and L.W. Ruddock, Protein disulfide isomerase: A critical evaluation of its function in disulfide bond formation. *Antioxid Redox Signal*, 2009. **11**(11): p. 2807–2850.
19. PTM_Stattistics_Curator, *Automated Curation and Population of PTM Statistics from the Swiss-Prot Knowledgebase*, http://selene.princeton.edu/PTMCuration/, 2011.
20. Vellai, T. and K. Takacs-Vellai, Regulation of protein turnover by longevity pathways. *Adv Exp Med Biol*, 2010. **694**: p. 69–80.
21. Shcherbik, N. and D.G. Pestov, Ubiquitin and ubiquitin-like proteins in the nucleolus: Multitasking tools for a ribosome factory. *Genes Cancer*, 2010, July 1. **1**(7): p. 681–689.
22. Sherwood, D. and J. Cooper, *Crystals, X-rays and Proteins: Comprehensive Protein Crystallography*. 2011, Oxford: Oxford University Press, p. 656.
23. Jacobsen, N.E., *NMR Spectroscopy Explained: Simplified Theory, Applications and Examples for Organic Chemistry and Structural Biology*. 2007, Hoboken, NJ: Wiley & Sons, p. 668.
24. Schmidt-Krey, I., Electron crystallography of membrane proteins: Two-dimensional crystallization and screening by electron microscopy. *Methods*, 2007. **41**: p. 417–426.
25. Jmol, *An Open-Source Java Viewer for Chemical Structures in 3D*, https://www.jmol.sourceforge.net/, 2010.
26. Raghavendra, N., V. Pattabhi, and S.S. Rajan, Metal induced structural changes observed in Human insulin. *PDB 3IRO*, 2010, https://doi.org/10.2210/pdb3ir0/pdb.
27. Method of the year 2021: Protein structure prediction. *Nat Methods*, 2022. **19**(1), https://doi.org/10.1038/s41592-021-01380-4.
28. Baek, M., et al., Accurate prediction of protein structures and interactions using a three-track neural network. *Science*, 2021. **373**(6557): p. 871–876.
29. Jumper, J., et al., Highly accurate protein structure prediction with AlphaFold. *Nature*, 2021. **596**(7873): p. 583–589.
30. EMBL-EBI, *DeepMind. AlphaFold Protein Structure Database*, https://alphafold.ebi.ac.uk/, 2024.
31. Expasy, http://www.expasy.ch/, 2010.
32. The Gene Ontology, http://www.geneontology.org/, 2010.
33. Höltje, H.-D., et al., *Molecular Modeling: Basic Principles and Applications*. 3rd ed. 2008, Weinheim, Germany: Wiley-VCH.
34. Nystedt, B., et al., The Norway spruce genome sequence and conifer genome evolution. *Nature*, 2013. **497**(7451): p. 579–584.
35. Minden, J.S., et al., Difference gel electrophoresis. *Electrophoresis*, 2009. **30**(Suppl. 1): p. S156–S161.
36. Rabilloud, T., Variations on a theme: Changes to electrophoretic separations that can make a difference. *J Proteomics*, 2010. **73**(8): p. 1562–1572.
37. Waller, L.N., K. Shores, and D.R. Knapp, Shotgun proteomic analysis of cerebrospinal fluid using off-gel electrophoresis as the first-dimension separation. *J Proteome Res*, 2008. **7**(10): p. 4577–4584.
38. Srinivas, P.R., Introduction to protein electrophoresis. *Methods Mol Biol*, 2019. **1855**: p. 23–29.
39. Latterich, M., M. Abramovitz, and B. Leyland-Jones, Proteomics: New technologies and clinical applications. *Eur J Cancer*, 2008. **44**(18): p. 2737–2741.
40. Yates, J.R., C.I. Ruse, and A. Nakorchevsky, Proteomics by mass spectrometry: Approaches, advances, and applications. *Annu Rev Biomed Eng*, 2009. **11**: p. 49–79.
41. Ali, I., et al., Separation of biological proteins by liquid chromatography. *Saudi Pharm J*, 2010. **18**(2): p. 59–73.
42. Duong, V.A., J.M. Park, and H. Lee, Review of three-dimensional liquid chromatography platforms for bottom-up proteomics. *Int J Mol Sci*, 2020. **21**(4).

43. Grey, A.C., et al., MALDI imaging mass spectrometry of integral membrane proteins from ocular lens and retinal tissue. *J Proteome Res*, 2009. **8**(7): p. 3278–3283.
44. Sen, A.K., J. Darabi, and D.R. Knapp, Design, fabrication and test of a microfluidic nebulizer chip for desorption electrospray ionization mass spectrometry. *Sens Actuators B Chem*, 2009. **137**(2): p. 789–796.
45. Wysocki, V.H., et al., Mass spectrometry of peptides and proteins. *Methods*, 2005. **35**(3): p. 211–222.
46. Sinha, A. and M. Mann, A beginner's guide to mass spectrometry–based proteomics. *Biochemist*, 2020. **42**(5): p. 64–69.
47. Al-Amrani, S., et al., Proteomics: Concepts and applications in human medicine. *World J Biol Chem*, 2021. **12**(5): p. 57–69.
48. Phillips, G.J., Green fluorescent protein—a bright idea for the study of bacterial protein localization. *FEMS Microbiol Lett*, 2001. **204**(1): p. 9–18.
49. Stepanenko, O.V., et al., Fluorescent proteins as biomarkers and biosensors: Throwing color lights on molecular and cellular processes. *Curr Protein Pept Sci*, 2008. **9**(4): p. 338–369.
50. Zimmer, M., Green fluorescent protein (GFP): Applications, structure, and related photophysical behavior. *Chem Rev*, 2002. **102**(3): p. 759–782.
51. Singleton, K.L., et al., Spatiotemporal patterning during T cell activation is highly diverse. *Sci Signal*, 2009. **2**(65): p. ra15.
52. Rodriguez, E.A., et al., The growing and glowing toolbox of fluorescent and photoactive proteins. *Trends Biochem Sci*, 2017. **42**(2): p. 111–129.
53. Fang, C., et al., Editorial: Mechanisms of fluorescent proteins. *Front Mol Biosci*, 2021. **8**.
54. Sinha, T.K., et al., Integrating spatially resolved three-dimensional MALDI IMS with in vivo magnetic resonance imaging. *Nat Methods*, 2008. **5**(1): p. 57–59.
55. Ayalew, M., et al., A new model for xylem loading and long-distance transport of iron explains the effect of kanamycin on metal uptake in Arabidopsis. *Front Plant Sci*, 2023. **14**: p. 1147598.
56. Chautard, E., N. Thierry-Mieg, and S. Ricard-Blum, Interaction networks: From protein functions to drug discovery. A review. *Pathol Biol (Paris)*, 2009. **57**(4): p. 324–333.
57. Fields, S., Interactive learning: Lessons from two hybrids over two decades. *Proteomics*, 2009. **9**(23): p. 5209–5213.
58. Monti, M., et al., Puzzle of protein complexes in vivo: A present and future challenge for functional proteomics. *Expert Rev Proteomics*, 2009. **6**(2): p. 159–169.
59. Tong, A.H., et al., A combined experimental and computational strategy to define protein interaction networks for peptide recognition modules. *Science*, 2002. **295**(5553): p. 321–324.
60. Field, C.B., et al., Primary production of the biosphere: Integrating terrestrial and oceanic components. *Science*, 1998. **281**(5374): p. 237–240.
61. Wu, M., K. Cronin, and J.S. Crane. *Biochemistry, Collagen Synthesis*. 2024, Treasure Island, FL: StatPearls Publishing, https://www.ncbi.nlm.nih.gov/books/NBK507709/.
62. Cooper, G.M. *The Cell: A Molecular Approach*. 2nd ed. 2000, Sunderland, MA: Sinauer Associates, https://www.ncbi.nlm.nih.gov/books/NBK9908/.
63. Ahnert-Hilger, G., et al., Regulation of vesicular neurotransmitter transporters. *Rev Physiol Biochem Pharmacol*, 2003. **150**: p. 140–160.
64. Lohr, K.M., et al., Vesicular monoamine transporter 2 (VMAT2) level regulates MPTP vulnerability and clearance of excess dopamine in mouse striatal terminals. *Toxicol Sci*, 2016. **153**(1): p. 79–88.
65. Dixon, J.B., Lymphatic lipid transport: Sewer or subway? *Trends Endocrinol Metab*, 2010. **21**(8): p. 480–487.
66. Wade, R.H., On and around microtubules: An overview. *Mol Biotechnol*, 2009. **43**(2): p. 177–191.
67. Maeda, S., et al., Structure of the connexin 26 gap junction channel at 3.5 A resolution. *Nature*, 2009. **458**: p. 597–602.
68. Birukou, I., J. Soman, and J.S. Olson, Blocking the gate for ligand entry in human hemoglobin A. unpublished.
69. Pietras, K., Increasing tumor uptake of anticancer drugs with imatinib. *Semin Oncol*, 2004. **31**(2, Suppl. 6): p. 18–23.
70. Kearns, D.B., A field guide to bacterial swarming motility. *Nat Rev Microbiol*, 2010. **8**(9): p. 634–644.
71. Lindemann, C.B. and K.A. Lesich, Flagellar and ciliary beating: The proven and the possible. *J Cell Sci*, 2010. **123**(Pt 4): p. 519–528.
72. Bessa, J. and M.F. Bachmann, T cell-dependent and -independent IgA responses: Role of TLR signalling. *Immunol Invest*, 2010. **39**(4–5): p. 407–428.
73. Lu, C., et al., Structural evidence for loose linkage between ligand binding and kinase activation in the epidermal growth factor receptor. *Mol Cell Biol*, 2010. **22**.
74. Holmes, M.A., Computational design of epitope-scaffolds allows induction of antibodies specific for a poorly immunogenic HIV vaccine epitope. *Structure*, 2010. **18**: p. 1116–1126.
75. Nemazee, D., Receptor editing in lymphocyte development and central tolerance. *Nat Rev Immunol*, 2006. **6**(10): p. 728–740.
76. Brooks, C.L., et al., Antibody recognition of a unique tumor-specific glycopeptide antigen. *Proc Natl Acad Sci USA*, 2010. **107**: p. 10056–10061.
77. Davis, M.M., et al., T cells as a self-referential, sensory organ. *Annu Rev Immunol*, 2007. **25**: p. 681–695.
78. Weiss, A., et al., Ligand-receptor interactions required for commitment to the activation of the interleukin 2 gene. *J Immunol*, 1987. **138**(7): p. 2169–2176.
79. Oda, K. and H. Kitano, A comprehensive map of the Toll-like receptor signaling network. *Mol Syst Biol*, 2006. **2**: p. 2006.0015.
80. Bell, J.K., et al., The molecular structure of the Toll-like receptor 3 ligand-binding domain. *Proc Natl Acad Sci USA*, 2005. **102**: p. 10976–10980.
81. Moreland, J.L., et al., The molecular biology toolkit (MBT): A modular platform for developing molecular visualization applications. *BMC Bioinform*, 2005. **6**: p. 21.
82. Carafoli, F., et al., Crystallographic insight into collagen recognition by discoidin domain receptor 2. *Structure*, 2009. **17**: p. 1573–1581.
83. Bloemendal, H., Lens proteins. *CRC Crit Rev Biochem*, 1982. **12**(1): p. 1–38.
84. Palanivelu, D.V., et al., Co-axial association of recombinant eye lens aquaporin-0 observed in loosely packed 3D crystals. *J Mol Biol*, 2006. **355**(4): p. 605–611.
85. Ulrich, P. and A. Cerami, Protein glycation, diabetes, and aging. *Recent Prog Horm Res*, 2001. **56**: p. 1–21.
86. Chiou, J.G., et al., Principles that govern competition or co-existence in Rho-GTPase driven polarization. *PLoS Comput Biol*, 2018. **14**(4): p. e1006095.
87. Hladyshau, S., et al., Spatiotemporal development of coexisting wave domains of Rho activity in the cell cortex. *Sci Rep*, 2021. **11**(1): p. 19512.
88. Yang, D., et al., G protein-coupled receptors: Structure- and function-based drug discovery. *Signal Transduct Target Ther*, 2021. **6**(1): p. 7.
89. Shankaran, H., H. Resat, and H.S. Wiley, Cell surface receptors for signal transduction and ligand transport: A design principles study. *PLoS Comput Biol*, 2007. **3**(6): p. e101.
90. Wade, J.D., et al., Mechanistic model of signaling dynamics across an epithelial mesenchymal transition. *Front Physiol*, 2020. **11**: p. 579117.

91. Mendoza, M.C., E.E. Er, and J. Blenis, The Ras-ERK and PI3K-mTOR pathways: Cross-talk and compensation. *Trends Biochem Sci*, 2011. **36**(6): p. 320–328.
92. Manning, B.D. and L.C. Cantley, AKT/PKB signaling: Navigating downstream. *Cell*, 2007. **129**(7): p. 1261–1274.
93. Olayioye, M.A., et al., The ErbB signaling network: Receptor heterodimerization in development and cancer. *EMBO J*, 2000. **19**(13): p. 3159–3167.
94. Wee, P. and Z. Wang, Epidermal growth factor receptor cell proliferation signaling pathways. *Cancers (Basel)*, 2017. **9**(5).
95. Ebi, H., et al., PI3K regulates MEK/ERK signaling in breast cancer via the Rac-GEF, P-Rex1. *Proc Natl Acad Sci USA*, 2013. **110**(52): p. 21124–21129.
96. Chabot, J.R. and B. Gomes, Modeling efficacy and safety of engineered biologics, in *Drug Efficacy, Safety, and Biologics Discovery: Emerging Technologies and Tools*, J. Xu and S. Ekins, Editors. 2009, Hoboken, NJ: Wiley InterScience.
97. Mayawala, K. and B. Gomes, Prediction of therapeutic index of antibody-based therapeutics: Mathematical modeling approaches, in *Predictive Toxicology in Drug Safety*, J. Xu and L. Urban, Editors. 2010, Cambridge, UK: Cambridge University Press.
98. Klipp, E. and W. Liebermeister, Mathematical modeling of intracellular signaling pathways. *BMC Neurosci*, 2006. **7**(1): p. S10.
99. Cheong, R., A. Hoffmann, and A. Levchenko, Understanding NF-κB signaling via mathematical modeling. *Mol Syst Biol*, 2008. **4**(1): p. 192.
100. Ghosh, S., S. Elankumaran, and I.K. Puri, Mathematical model of the role of intercellular signalling in intercellular cooperation during tumorigenesis. *Cell Prolif*, 2011. **44**(2): p. 192–203.
101. Dave, D.D. and B.K. Jha, Mathematical modeling of Calcium oscillatory patterns in a neuron. *Interdiscip Sci: Comput Life Sci*, 2021. **13**(1): p. 12–24.
102. Neves, M.J. and J. François, On the mechanism by which a heat shock induces trehalose accumulation in *Saccharomyces cerevisiae*. *Biochem J*, 1992. **288**(Pt 3): p. 859–864.
103. Zwolak, J., et al., A quantitative model of the effect of unreplicated DNA on cell cycle progression in frog egg extracts. *J Theor Biol*, 2009. **260**(1): p. 110–120.
104. Akama, K., et al., Proteomic identification of differentially expressed genes in neural stem cells and neurons differentiated from embryonic stem cells of cynomolgus monkey (*Macaca fascicularis*) in vitro. *Biochim Biophys Acta*, 2011. **1814**(2): p. 265–276.

Taylor & Francis
Taylor & Francis Group
http://taylorandfrancis.com

Metabolic Systems

9

When you have read this chapter, you should be able to:

- Identify and characterize the components of metabolic systems
- Describe conceptually how the components of metabolic systems interact
- Understand the basics of mass action and enzyme kinetics
- Be aware of the various data sources supporting metabolic analyses
- Associate different types of metabolic data with different modeling tasks
- Explain different purposes of metabolic analyses
- Understand the computational concepts of metabolic engineering

Our genome is sometimes called the blueprint of who we are. We have seen in previous chapters that this notion is somewhat simplified, because the macro- and micro-environments have significant roles throughout the biological hierarchy, including the expression of genes. Nonetheless, our genes do contain the code that ultimately determines the machinery of life. The lion's share of this machinery is present in the form of proteins and, like most machines, they take input and convert it into output. In the case of living systems, the vast majority of both inputs and outputs consist of metabolites. Most prominently, proteins convert metabolites, including food or substrate, into energy and into a slew of other metabolites ranging from sugars and amino acids to vitamins and neurotransmitters. Signal transduction is sometimes seen as distinct from metabolic action, but it is also often accomplished through metabolic changes, such as the phosphorylation or glycosylation of proteins or the synthesis or degradation of signaling lipids. Frequently, it is at the level of metabolites that a normal physiological process becomes pathological and health turns into disease. A gene may be mutated but, on its own, this change does not cause problems. However, if the mutation leads to an abnormal protein or regulatory mechanism and thereby affects the dynamics of a metabolic pathway, it may ultimately cause an excess or dearth of specific metabolites, with consequences that in some cases might be tolerable, but in other cases can compromise the affected cell and possibly even kill it.

DNA, proteins, and metabolites are composed of the same "organic" atoms (mainly C, O, N, S, P, and H), and there is really no essential biochemical difference between them, except that DNA and proteins are large molecules with particular chemical structures, while metabolites in the strict sense are much smaller by comparison. For instance, the paradigm metabolite glucose ($C_6H_{12}O_6$) has a molecular mass of $(6 \times 12 + 12 \times 1 + 6 \times 16) = 180$, whereas hexokinase, the enzyme that converts glucose into glucose 6-phosphate, has a molecular mass of roughly 100,000 Daltons (Da) [1] (**Figure 9.1**). The average molecular mass of a nucleotide

DOI: 10.1201/9781003402770-9

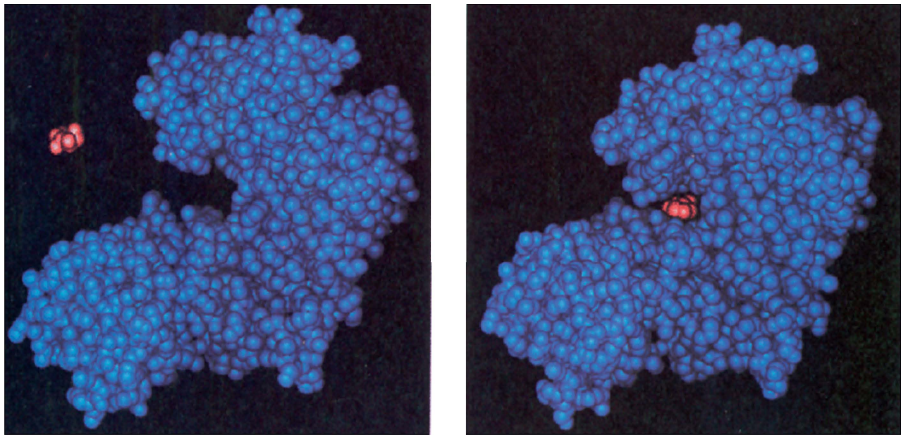


Figure 9.1 Typical metabolites are small in comparison with proteins and DNA. The enzyme hexokinase (blue) binds a molecule of glucose (red). Note the huge difference in sizes of the two molecules. (Courtesy of Juan Cui and Ying Xu, University of Georgia.)

in DNA is about 330 Da, which for the average-sized human gene of roughly 3,000 bases corresponds to a molecular mass of about 1,000,000 Da. The largest known human gene codes for dystrophin, a key protein in muscle, which, if altered, is one of the root causes of muscular dystrophy. This gene consists of 2.4 million bases and thus has a molecular mass of almost 800 million Daltons [2]. Indeed, a typical metabolite is tiny in comparison with proteins and nucleic acids.

The conversion of metabolites into other metabolites has been the bread and butter of traditional biochemistry. This conversion typically occurs in several steps that form metabolic pathways. Many biochemistry books are organized according to these pathways, containing chapters on central metabolism, energy metabolism, amino acid metabolism, fatty acid metabolism, and so on, sometimes including less prevalent pathways, such as the synthesis of poisons in insects and spiders or pathways leading to secondary metabolites, such as color pigments, which may not be absolutely required for survival. In reality, metabolism is not neatly arranged in pathways, but pathways are merely the most obvious contributors to complex regulated pathway systems, just as interstate highways are the most prominent features of a much larger and highly connected network of roads, streets, and alleyways.

Traditionally, biochemistry has mostly focused on the individual steps a cell employs to convert different metabolites into each other. Of particular importance in these discussions are enzymes, which we discussed in Chapter 8. Each enzyme facilitates one or maybe a few of the specific steps in a string of metabolic conversions. In recent years, much of the emphasis on single reaction steps has shifted toward high-throughput methodologies of **metabolomics** that try to characterize the entire metabolic state of a living system (or subsystem) at a given point in time. A good example is a mass spectrum, which is capable of quantitatively assessing the relative amounts of thousands of metabolites in a biological sample [3].

This chapter is very clearly not an attempt to review biochemistry, and it will mention specific pathways only in the context of illustrative examples. Rather, it tries to describe some of the basic and general concepts of biochemistry and metabolomics that are particularly pertinent for computational analyses in systems biology. It begins with the characterization of enzyme-catalyzed reactions, discusses means of controlling the rates of these reactions, establishes the foundation for mathematical methods that capture the dynamics of metabolic systems, and lists some of the methods of data generation and the comprehensive resources of information regarding metabolites that are presently available. It also discusses the principles of metabolic engineering and of methods used to reconstruct metabolic pathway systems from data.

9.1 BIOCHEMICAL REACTIONS

9.1.1 Background

Metabolism is the sequential conversion of organic compounds into other compounds. As chemical reactions, these conversions must of course obey the laws of chemistry and physics. Thermodynamics tells us that for the unaided conversion

of one chemical compound into another to take place, the latter must be in a lower-energy state than the former. Does that mean that biochemistry and metabolism are at odds with one of the fundamental principles of physics by converting food into metabolites of high energy content, such as amino acids or lipids? The situation is particularly puzzling in plants, where the "food" consists merely of carbon dioxide, CO_2, which has very low energy. How is the production of high-energy compounds possible?

The answer, which of course is fully compatible with the laws of thermodynamics, consists of three components:

- Plants are able to utilize sunlight as energy for converting CO_2 into metabolites of higher energy content.
- In all organisms, the metabolic production of a desired high-energy molecule is "paid for" by the conversion of some other highly energetic molecule into a lower-energy molecule. For instance, the conversion of glucose into glucose 6-phosphate is coupled with the conversion of the high-energy molecule adenosine triphosphate (ATP) into the lower-energy molecule adenosine diphosphate (ADP).
- Highly energetic molecules in cells do not simply degrade into lower-energy molecules, as thermodynamics might suggest. Instead, they can often only move to the lower-energy state if they transit an intermediate state of even higher energy. This transition is therefore not automatic but must be facilitated, which is accomplished with enzymes (see Chapter 8 and **Box 9.1**). Indeed, the vast majority of metabolic reactions require an enzyme, and they very often involve a secondary substrate or co-factor that provides the energy for the creation of a higher-energy molecule from a lower-energy molecule. The involvement of an enzyme speeds up the reaction rate enormously and furthermore affords the possibility of regulation, in which one or more metabolites influence the degree of enzyme activity and thereby the rate of the metabolic reaction.

For the purposes of metabolic systems analysis, we do not always need to know much more about the molecular mechanisms associated with an enzyme than that the enzyme enables the transition from substrate to product and that its activity can be regulated with various mechanisms that act on different timescales (see Chapters 8 and 11). What we do need to explore in detail is how enzymatic processes are translated into mathematical models.

BOX 9.1: THE FUNCTION OF AN ENZYME DEMONSTRATED WITH AN ANALOGY

The function on an enzyme is illustrated in **Figure 1A**, where the original metabolite (typically called the substrate) with a high-energy state (red) must be briefly elevated to a transition state of even higher energy (blue), before it moves to the product state with lower energy. The enzyme temporarily provides the necessary activation energy E_A. An intuitive analogy is popcorn (Figure 1B). In its original state, consisting of cold corn kernels, its energy is constrained: although the popcorn is in a pan high above the floor, it cannot fall down to the lower-energy level of the floor. Now we turn on the heat (activation energy), which releases the stored energy and allows each kernel to jump during its pop (the transition state), and it is indeed possible that some kernels will jump over the rim of the pan onto the floor.

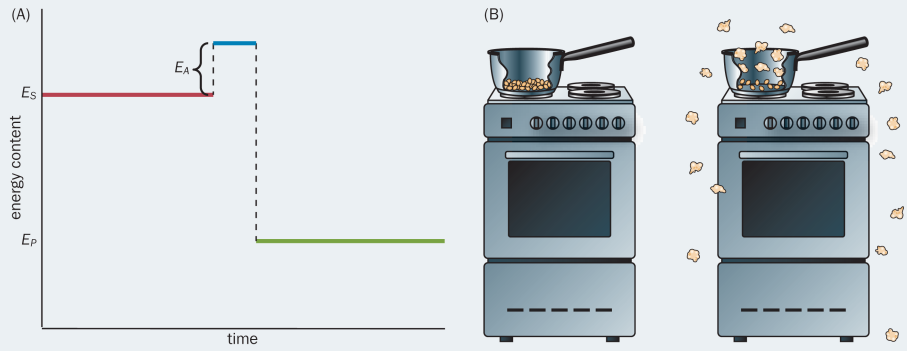


Figure 1 Processes overcoming thresholds of high energy. (A) The conversion of a metabolic substrate with energy E_S into a product with lower energy E_P requires activation energy E_A that exceeds the threshold (blue). (B) The situation in (A) is similar to popping corn, where individual kernels may receive enough (heat) energy to jump out of the pan and onto the floor.

9.1.2 Mathematical Formulation of Elementary Reactions

To ease our way into this translation task, we begin with the simpler situation in which a chemical compound degrades over time without the involvement of an enzyme or cofactor. This situation is not very interesting biochemically but is a good start for model development. It has two prominent applications. One is radioactive decay, in which radionuclides spontaneously disintegrate and the collective disintegration process is proportional to the presently existing amount. The second pertains to diffusion and transport processes, in which the amount of material moving from one state to another is proportional to the current amount.

The proportionality in both cases gives us a direct hint for how to set up a describing equation: namely, the change at time t is proportional to the amount at time t. Recalling Chapters 2 and 5, change is expressed mathematically as the derivative with respect to time, the amount is a function of time, and proportionality implies a linear function. Putting the pieces together into an equation yields

$$\frac{dX}{dt} = \dot{X} = -k \cdot X. \tag{9.1}$$

The right-hand side of the ordinary differential equation (ODE) (9.1) contains three items that are reminiscent of the SIR model in Chapter 2. X is the amount (for example, of the radionuclide), and one often refers to X generically as a pool (of molecules). It might also represent the concentration of a molecule. The second component is the **rate constant** k, which is always positive (or zero) and constant over time. It quantifies how many units of X are changing per time unit. Finally, the minus sign indicates that the change ($\dot{X}$) is in the negative direction: material disappears rather than accumulates. The formulation in (9.1) as a description of a chemical process is based on considerations of statistical thermodynamics and was proposed more than one hundred years ago by the Swedish scientist and Nobel Laureate Svante Arrhenius (1859–1927).

As we discussed in Chapter 5, (9.1) is a linear ODE that describes exponential behavior. Suppose now that X is converted into Y and that the disappearance of X is captured well by (9.1). Because all material leaving pool X moves into pool Y, it is easy to see that the change in Y must equal the change in X, except that the two have opposite signs. The dynamics of the two variables is therefore easily described as

$$\begin{aligned} \dot{X} &= -kX, \\ \dot{Y} &= kX. \end{aligned} \tag{9.2}$$

We are allowed to add the two differential equations, which yields

$$\dot{X} + \dot{Y} = 0, \tag{9.3}$$

and has the following interpretation: The total change in the system, consisting of the change in X plus the change in Y, $\dfrac{d(X+Y)}{dt}$, equals zero. There is no overall change. This makes sense, because material is just flowing from one pool to the other and no material is gained or lost.

Many metabolic reactions involve two substrates and are therefore called bimolecular. Their mathematical description is constructed in analogy to the one-substrate case and leads to a differential equation where the right-hand side involves the product of the two substrates. Specifically, suppose X_1 and X_2 are the substrates of a bimolecular reaction that generates product X_3. Then the increase in the concentration of X_3 is given as

$$\dot{X}_3 = k_3 X_1 X_2. \tag{9.4}$$

Note that the substrates enter the equation as a product and not a sum, even though one might speak of "adding" a second substrate or formulate the reaction as $X_1 + X_2 \rightarrow X_3$. The reason for this formulation can be traced back to thermodynamics and to the fact that the two molecules must come into physical contact within the physical space where the reaction happens, which is a matter of probability and leads to the product.

Because X_3 is the recipient of material and does not affect its own synthesis, the right-hand side is positive and does not depend on X_3, and its concentration continues to increase as long as X_1 and X_2 are available. For every molecule of X_3 produced, one molecule of X_1 and one molecule of X_2 are used up. Therefore, the loss in either one substrate is

$$\dot{X}_1 = \dot{X}_2 = -\dot{X}_3 = -k_3 X_1 X_2. \tag{9.5}$$

It is of course possible that X_3 is produced from two molecules of type X_1 rather than from X_1 and X_2. In this case, the describing equations are

$$\dot{X}_1 = -2k_3 X_1^2,\ \dot{X}_3 = k_3 X_1^2. \tag{9.6}$$

The product of X_1 and X_2 in (9.5) becomes X_1^2 in both equations of (9.6), and one says that the process is of second order with respect to X_1. The first equation in (9.6) furthermore contains the stoichiometric factor 2, which does not enter the second equation. The reason is that X_1 is used up twice as fast as X_3 is produced, because two molecules of X_1 are needed to generate one molecule of X_3. The mathematical formulations discussed here are the foundation of **mass action kinetics**, which was introduced about 150 years ago [4–6]. According to this widely used analytical framework, all substrates enter a reaction term as factors in a product, where powers reflect the number of contributing molecules of each type. The term also contains a rate constant, which is positive (or possibly zero but never negative) and does not change over time.

Mass action formulations implicitly assume that many substrate molecules are available and that they are freely moving about in a homogeneous medium. In many cases, especially in living cells, these assumptions are not always true, but they do provide good approximations to realistic metabolic systems. They are frequently used because more accurate formulations are incomparably more complicated. For instance, one could more realistically consider the reaction between X_1 and X_2 as a random encounter and thus a stochastic process. While intuitively plausible, a model of such a process requires heavy-duty mathematics for further analysis [7, 8].

9.1.3 Rate Laws

Most biochemical reactions in a metabolic system are catalyzed by enzymes. Over a century ago, the biochemists Henri, Michaelis, and Menten proposed a mechanism and a mathematical formula describing this process [9, 10]. They postulated that the substrate S and the catalyzing enzyme E reversibly form an intermediary complex (ES), which subsequently breaks apart and irreversibly yields the reaction product P while releasing the enzyme unchanged and ready for reuse. The diagram of the mechanism with typical rate constants for all steps is shown in **Figure 9.2**. Modern methods of transient and single-molecule kinetics have shown that this simple scheme is somewhat simplistic and that many reactions in fact consist of entire networks of fast sub-reactions with different intermediary complexes that form a multidimensional free-energy surface [11]. Nonetheless, the Michaelis–Menten mechanism is a very useful conceptual framework, and an enormous number of studies have measured its characteristic parameters.

A mathematical model for the mechanism is obtained straightforwardly by setting up mass action functions. Specifically, each process term is formulated by

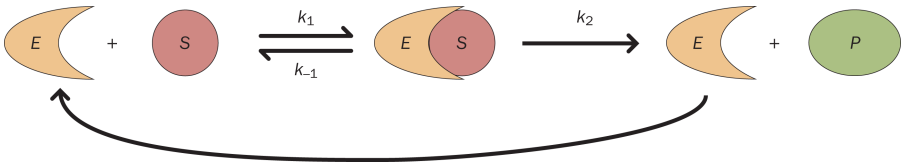


Figure 9.2 Conceptual model of an enzyme-catalyzed reaction mechanism, as proposed by Michaelis and Menten. A substrate *S* and an enzyme *E* form a complex (*ES*), which may either return *S* and *E* or lead to the product *P*, thereby releasing free enzyme.

including all variables that are directly involved, as well as the appropriate rate constant. The result is

$$\dot{S} = -k_1 \cdot S \cdot E + k_{-1} \cdot (ES), \tag{9.7}$$

$$\left(\dot{ES}\right) = k_1 \cdot S \cdot E - \left(k_{-1} + k_2\right) \cdot (ES), \tag{9.8}$$

$$\dot{P} = k_2 \cdot (ES). \tag{9.9}$$

These equations are easy to interpret. For instance, (9.7) says that the change in substrate is governed by two processes. In the first, $-k_1SE$, the substrate and enzyme enter a bimolecular reaction with rate k_1. The process carries a minus sign because substrate is lost. The second process, $k_{-1}(ES)$, describes that some of the complex (ES) reverts to S and E, and this happens with rate k_{-1}. The sign here is positive because the process augments the pool S.

An equation for E is not formulated, because the sum of free enzyme E and enzyme bound in the complex (ES) is assumed to remain constant during the reaction; this total enzyme concentration is denoted E_{tot}. Thus, once we know the dynamics of (ES), we can infer the dynamics of E from the equation $E = E_{tot} - (ES)$. It is usually assumed that the reversible formation of the complex (ES) occurs much faster than the production of P, and that therefore $k_1 >> k_2$ and $k_{-1} >> k_2$. The parameter k_2 is sometimes also called the catalytic constant and denoted as *kcat*. The production of P is assumed to be irreversible.

The differential equations in (9.7)–(9.9) do not have an explicit analytical solution but can of course be solved computationally with a numerical integrator, which is available in many software packages. Since Henri, Michaelis, and Menten did not have computers, they proposed reasonable assumptions that yielded a significant simplification from the ODE system to an algebraic formulation. The assumptions are that substrate is present in excess, relative to the enzyme ($S >> E$), and that the enzyme–substrate complex (ES) is in equilibrium with the free enzyme and substrate. As a more realistic variation of the latter, Briggs and Haldane [12] proposed the **quasi-steady-state assumption** (QSSA), which entails that (ES) does not appreciably change in concentration over time and that the total enzyme concentration E_{tot} is constant. With this assumption, life becomes much simpler. The left-hand side of (9.8) is now equal to zero, because (ES) is assumed not to change. Furthermore, one defines the rate of product formation as $v_P = \dot{P}$ and introduces the maximal velocity of the reaction as $V_{\max} = k_2 E_{tot}$ and the Michaelis constant as $K_M = (k_{-1} + k_2)/k_1$. Using straightforward algebra, one can show that the rate of product formation can then be expressed as

$$v_P = \frac{V_{\max}}{K_M + S} \tag{9.10}$$

(see Exercise 9.5). The function in (9.10) is referred to as the **Michaelis–Menten rate law (MMRL)** or the Henri–Michaelis–Menten rate law (HMMRL). A plot of MMRL shows a monotonically increasing function that approaches $V_{\max}$ (Figure 9.3), which means that $V_{\max}$ quantifies the fastest possible speed with which the reaction can proceed. K_M corresponds to the substrate concentration for which the reaction speed is exactly half the maximum velocity. Its numerical value can vary widely among different enzymes. In many cases, it is close to the natural substrate concentration in vivo. Recall from Chapter 6 how the parameter values of

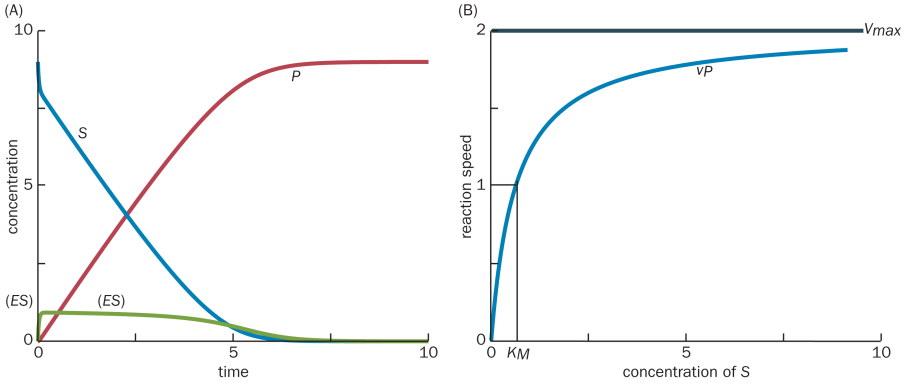


Figure 9.3 Plots of the Michaelis–Menten rate law. (A) Solution of the system of equations (9.7–9.9) in the form of concentrations versus time. (B) The more familiar plot of reaction speed versus substrate concentration, (8.10). Parameters are $k_1 = 4$, $k_{-1} = 0.5$, $k_2 = 2$, $K_M = 0.625$, $E_{\text{total}} = 1$, $V_{\text{max}} = 2$, $S(0) = 9$, $(ES)(0) = P(0) = 0$.

MMRL can be obtained from in vitro measurements with simple linear regression upon plotting $1/v$ against $1/S$.

Instead of using algebra, MMRL can also be deduced from the following biochemical arguments. The reversible reaction among substrate, enzyme, and intermediary complex on the left side of Figure 9.2 is governed by the mass action formulation

$$\frac{k_{-1}}{k_1} = k_d = \frac{E \cdot S}{(ES)} \tag{9.11}$$

where k_d is called the equilibrium or dissociation constant. Because k_2 is traditionally assumed to be small in comparison to k_1 and k_{-1}, k_d is approximately equal to $(k_{-1} + k_2)/k_1$ and thus to K_M.

The occupancy of an enzyme reflects how much of the enzyme is bound to the substrate. In other words, it is the ratio (ES) / E_{tot}. We can use Eqn. (9.11) to rearrange this ratio as

$$\frac{(ES)}{E_{tot}} = \frac{(ES)}{E + (ES)} = \frac{ES/k_d}{E + ES/k_d} = \frac{S/k_d}{1 + S/k_d} = \frac{S}{k_d + S} \approx \frac{S}{K_M + S} \tag{9.12}$$

With this result, we immediately obtain MMRL as

$$v_P = \dot{P} = k_2 \cdot (ES) \approx \frac{k_2 \cdot E_{tot} \cdot S}{K_M + S} = \frac{V_{\max} \cdot S}{K_M + S} \tag{9.13}$$

Note that the solution of an ODE is typically presented as a plot of the dependent variables as functions of time t. In (9.7)–(9.9), these are S, (ES), and P. By contrast, (9.10), which is the form almost always found in biochemistry books and articles, formulates the speed of product formation v_P as a function of the substrate concentration S. The two plots are distinctly different (Figure 9.3).

Thousands of biochemical investigations have used this formulation, and MMRL is even used as a "black box" model in other contexts that have nothing to do with biochemical kinetics, just because its shape fits many observed data. Of interest is that K_M is considered a property of an enzyme–substrate pair, so that K_M values from the literature or a database may be useful for an analysis even if they were measured by different groups and under slightly different conditions. By contrast, $V_{\max}$ very much depends on the experimental conditions and can seldom be assumed to have the same value under different conditions.

Very many reactions in metabolism involve two substrates and two products. A typical example is the first step of glycolysis, where glucose is converted into

glucose 6-phosphate. In humans, the phosphate moiety comes from ATP, which in the process is dephosphorylated to ADP. Thus, the reaction scheme is

$$\text{Glucose} + \text{ATP} \rightleftharpoons \text{Glucose 6-phosphate} + \text{ADP}$$

In many cases, one of the two substrates is so plentiful that its dynamics can be ignored. This is often the case in glycolysis, where the explicit role of ATP is not modeled and the reaction is formulated with MMRL. However, in other cases, both substrates can potentially limit the reaction speed, which has important consequences. Let's study a bi-substrate reaction with the generic notation

$$\text{A} + \text{B} \rightleftharpoons \text{P} + \text{Q}$$

which not only accounts for two substrates and two products but also for the reverse reaction from P + Q to A + B.

It might seem that this step-up from one substrate to two would not require much extra work, but that is not the case. The reason is that the overall process now involves not only one complex (between S and E) but all kinds of intermediate complexes, which may be short-lived but may have to be considered for a realistic assessment. Various mechanisms have been proposed, but in a typical, relatively simple model of enzyme catalysis, we find the following reactions

$$\begin{aligned}
\text{E} + \text{A} &\rightleftarrows \text{EA} \\
\text{EA} + \text{B} &\rightleftarrows \text{EAB} \\
\text{EAB} &\rightleftarrows \text{EPQ} \\
\text{EPQ} &\rightleftarrows \text{EQ} + \text{P} \\
\text{EQ} &\rightleftarrows \text{E} + \text{Q}
\end{aligned}$$

[13, 14].

It is furthermore possible that other complexes of three components are formed, for instance, between A, B, and P. Each of these quantities changes in amount or concentration throughout the course of the reaction. Depending on the specifics of the model, subsuming, as before, the constant enzyme concentration into quantities like $V_{\max}$, here called V_1 and V_{-1}, and defining Michaelis constants as combinations of reaction rates, the simplified rate law may read

$$v = \frac{\dfrac{V_1 AB}{K_{iA} K_{mB}} - \dfrac{V_{-1} PQ}{K_{mP} K_{iQ}}}{Denominator} \tag{9.14}$$

where

$$\begin{aligned}
Denominator = 1 &+ \frac{A}{K_{iA}} + \frac{K_{mA} B}{K_{iA} K_{mB}} + \frac{K_{mQ} P}{K_{mP} K_{iQ}} + \frac{Q}{K_{iQ}} + \frac{AB}{K_{iA} K_{mB}} + \frac{K_{mQ} AP}{K_{iA} K_{mP} K_{iQ}} \\
&+ \frac{K_{mA} BQ}{K_{iA} K_{mB} K_{iQ}} + \frac{PQ}{K_{mP} K_{iQ}} + \frac{ABP}{K_{iA} K_{mB} K_{iP}} + \frac{BPQ}{K_{iB} K_{mP} K_{iQ}}
\end{aligned}$$

[13]; it may become even more complicated [14, 15]. Pondering this complexity for almost the simplest of the simplest, very common reaction scheme, and then considering the fact that systems biology usually wants to look at a pathway system consisting of many reactions, we are encouraged—or rather forced—to look for alternatives. A particularly useful, streamlined approach is the use of generic power-law functions that we studied in Chapter 5.

Many enzymes respond to changes in metabolite concentrations with exceptional sensitivity, a phenomenon called cooperativity because it involves cooperation among several active sites or subunits of an enzyme [13]. The observed consequence

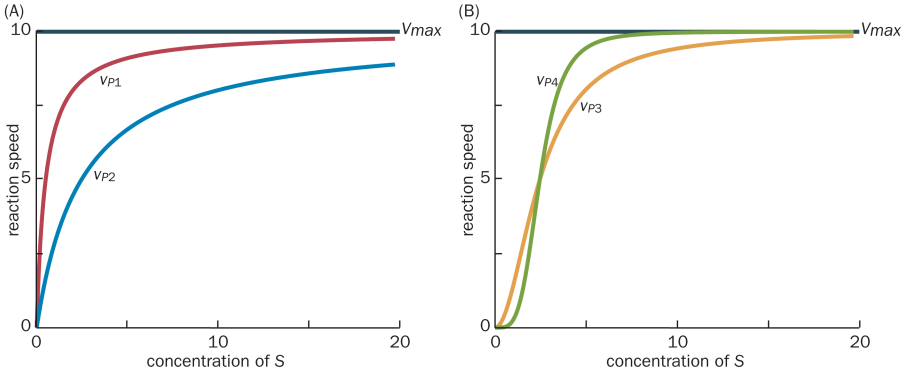


Figure 9.4 Plots of Michaelis–Menten (MMRL) and Hill rate laws (HRL). (A) Two MMRLs, v_{P1} and v_{P2}, with the same $V_{max} = 10$ and $K_{M1} = 0.5$ and $K_{M2} = 2.5$, respectively. (B) Two HRLs, v_{P3} and v_{P4}, with the same $V_{max} = 10$ and $K_M = 2.5$ and with Hill coefficients $n = 2$ and $n = 4$, respectively. Note that v_{P2} in (A) corresponds to an HRL with $V_{max} = 10$, $K_M = 2.5$, and $n = 1$.

of cooperativity is a rate function that has a sigmoid shape (Figure 9.4) and therefore differs notably from MMRL. Archibald Hill [16] developed a heuristic description for this effect in his studies of oxygen binding to hemoglobin and proposed a mathematical function that is a generalization of MMRL and known today as **Hill rate law (HRL)** or Hill–Langmuir equation. In this generalization, the substrate appears with an exponent called the Hill coefficient, h. Typically, h is an integer power that reflects the number of subunits or active sites of the enzyme molecule. However, from a mathematical point of view, h could assume any positive real value. HRL reads

$$v_{P,Hill} = \frac{V_{max}\, S^h}{K_M^h + S^h}. \tag{9.15}$$

For $h = 1$, HRL is the same as MMRL, and the S-shape is reduced to a simple shoulder curve. The derivation of HRL follows exactly the same steps as in the case of MMRL, except that one accounts for h substrate molecules and h product molecules. For $h = 2$, for instance, (9.7) becomes

$$\dot{S}_H = -2k_1 S_H^2 E_H + 2k_{-1}(ES)_H, \tag{9.16}$$

where the index H simply distinguishes HRL from MMRL (9.7). The algebraic form (9.15) of the rate law can be derived again from the corresponding differential equations, either with arguments from algebra or biochemistry, as we discussed it for MMRL. Similar to what was shown in Figure 6.5 for MMRL, HRL can be put into a linear format, for instance, to estimate its parameters from experimental data (see Exercises).

Several authors addressed cooperativity with other mathematical forms than the HRL, in an attempt to represent the actual biological mechanisms. Even with several simplifying assumptions, this approach often leads to rather complicated systems of equations [13, 17–22].

As described so far, the speed of an enzyme-catalyzed reaction depends exclusively on the substrate concentration, the parameters K_M and V_{max}, and possibly h. In living systems, many reactions are controllable by inhibitors and activators that sense the demand for a product and adjust the turnover rate of the reaction. Modulation of the reaction speed by an inhibitor or activator may occur at several locations in the Michaelis–Menten mechanism shown in Figure 9.2. For instance, an inhibitor may compete with the substrate molecules for an active site on the enzyme. This may happen if the inhibitor molecule is structurally similar to the substrate and therefore fits into the same binding site on the enzyme. It may also happen that the inhibitor forms a rather stable complex with the enzyme–substrate complex (ES), which partially prevents the reaction toward the product from occurring. If the inhibitor and the substrate are structurally quite different, inhibition of the enzyme activity is still possible if the inhibitor binds to the enzyme but

not at the substrate binding site. This docking of the inhibitor at a different location on the surface of the protein affects the folding of the enzyme molecule and thereby alters the substrate binding site. The result is a reduced (or increased) affinity between enzyme and inhibitor (activator). This type of **allosteric** modulation is encountered frequently. Because different binding sites are now involved, the general mathematical formulation of allosteric regulation is very complicated [17, 20–22].

Inhibition can be captured mathematically much more easily if the inhibitor competes with the substrate for the biding side on the enzyme or if it reduces the maximal speed of the process, $V_{\max}$. Thousands of articles and many books have discussed enzyme regulation and described biochemical techniques for characterizing the regulatory mechanisms. Since these investigations are firmly grounded within the realm of biochemistry, we will not discuss them much further, but we will state the most important results. Detailed representations of mathematical models of enzyme modulation can be found in [13, 14, 17].

Three prominent types of inhibition are called competitive, uncompetitive (or anti-competitive), and noncompetitive [13, 17]. The terminology of the latter two types is confusing but has become the standard in the field. Mathematically, the three types are often modeled as extensions of MMRL that modulate K_M, $V_{\max}$, or both.

In competitive inhibition, an inhibitor with concentration I competes with the substrate for the binding site on the enzyme. The corresponding rate law is

$$v_{P,CI} = \frac{V_{\max} S}{K_M \cdot \left(1 + \frac{I}{K_I}\right) + S}. \tag{9.17}$$

The new parameter K_I is called the inhibition constant.

In uncompetitive, or anti-competitive inhibition, the inhibitor binds to the complex (ES). The corresponding rate law is

$$v_{P,UI} = \frac{V_{\max} S}{K_M + S \cdot \left(1 + \frac{I}{K_I}\right)}. \tag{9.18}$$

In the case of noncompetitive inhibition, the inhibitor reduces the binding affinity between enzyme and substrate without directly affecting the substrate binding site. The corresponding rate law is

$$v_{P,NI} = \frac{V_{\max} S}{K_M + S} \cdot \left(1 + \frac{I}{K_I}\right)^{-1}. \tag{9.19}$$

In addition to these three types of inhibition and allosteric regulation, the modulation of enzyme activity may be a mixture of several mechanisms, which unsurprisingly increases the complexity of a mathematical representation, especially for pathways that consist of numerous processes.

Circumventing this issue of ever-increasing complexity, biochemical systems theory (BST) treats an inhibitor like any other variable and represents it with a power-law approximation (see Chapter 5). As an example, if the degradation reaction of X_1 is inhibited by X_j, one sets up the model for X_1 as

$$\dot{X}_1 = production - \beta_1 X_1^{h_{11}} X_j^{h_{1j}} \tag{9.20}$$

and theory asserts that $h_1 j$ is a negative quantity. If the inhibitor X_j is small in quantity, it is useful to replace X_j with $(1 + X_j)$, so that Eq. (9.20) reads

$$\dot{X}_1 = production - \beta_1 X_1^{h_{11}} \left(1 + X_j\right)^{h_{1j}} \tag{9.21a}$$

[23]. One may also explicitly account for an inhibition constant K_I and formulate the inhibition as

$$\dot{X}_1 = production - \beta_1 X_1^{h_{11}} \left(1 + \frac{X_j}{K_I}\right)^{h_{1I}} \tag{9.21b}$$

Whichever format is chosen, the unmodulated power-law function is regained, if the inhibitor concentration is reduced to 0. Note that the result of this representation is similar to the noncompetitive rate law, although the power-law term is derived purely through linearization in logarithmic space (Chapter 5).

Data directly related to a traditional enzyme kinetic investigation are typically presented as quantitative features of enzymes that are characterized by parameters such as K_M, inhibition and dissociation constants K_I and K_D, and sometimes $V_{\max}$ values. Often, co-factors and modulators of reactions are listed as well. Originally, these features were described in a huge body of literature, but nowadays databases such as BRENDA [24] (Box 9.2) and ENZYME [25] offer collections of many properties of enzymes and reactions, along with pertinent literature references.

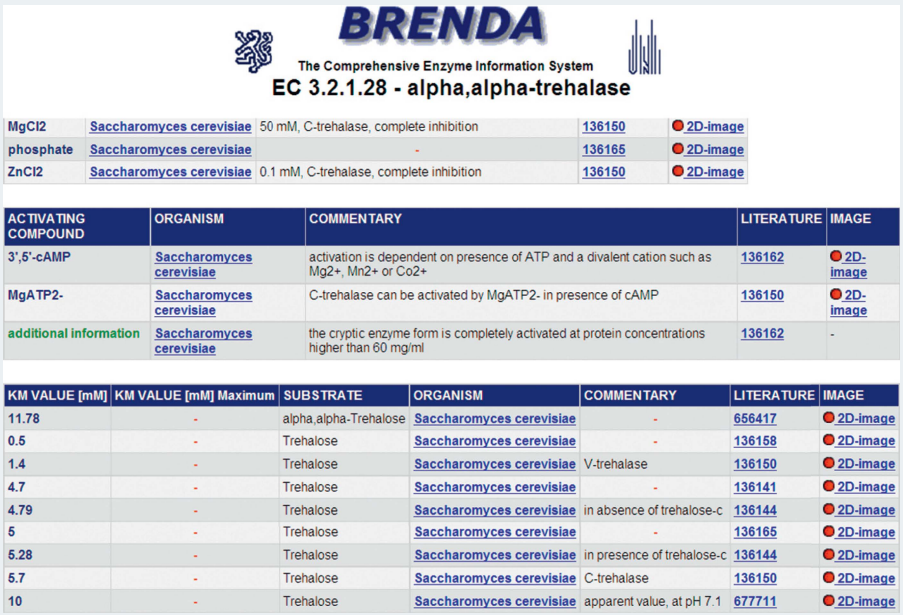
Few functions in biology have received as much attention as MMRL and HRL, but that does not mean that they are perfect. In fact, like all mathematical models in biology, they are based on simplifying assumptions, which may or may not be justified in a living organism (see, for example, [14, 26, 27]). First and foremost, these functions, just like mass action kinetics, assume that the reactions occur in well-stirred, homogeneous media, which obviously is not the case inside a living cell that is full of materials and organelles and where reactions often happen at surfaces. Second, in some realistic situations, the amount of enzyme is not constant, and it may be much smaller than the substrate concentration, as MMRL assumes it to be. Third, it cannot be said in general to what degree the quasi-steady-state assumption (QSSA) is truly valid within a living cell. QSSA was made for in vitro experiments, where indeed the solution can be well stirred and the substrate concentration exceeds the enzyme concentration many times. Even there, it is clear from Figure 9.3 that QSSA is not true at the very beginning of an experiment, when substrate and enzyme encounter each other for the first time and that it becomes inaccurate at the end when almost all substrate is used up (see Figure 9.3; however, also see Exercises 9.3–9.6). Nonetheless, a lot of experience has demonstrated that the benefits of QSSA often outweigh these inaccuracies (for further discussion, see [26–28]). Finally, theoretical considerations demonstrate that continuous kinetic functions are not appropriate for small numbers of reactants and may have to be replaced with much more complicated stochastic processes for realistic assessments [7, 8].

An alternative to MMRL, HRL, and their extensions toward inhibition and other types of modulation is the use of generic approximations. Linear approximations are not suitable in this case, due to the strongly nonlinear nature of enzyme kinetics, but power-law representations (Chapter 5) have proven very successful and do not require many assumptions. As a drawback, power-law models do not mimic the actual physicochemical mechanisms with which enzymes catalyze the conversion of substrates into products.

No matter which mathematical format one chooses, its practical use requires information regarding metabolites, enzymes, and modulators. Using painstaking analysis, traditional biochemistry has yielded a solid understanding of most reactions in all major and many minor metabolic pathways, and huge amounts of data related to enzyme kinetics, such as K_M and $V_{\max}$ values, have been measured and are available under their Enzyme Commission (EC) number in databases such as BRENDA [24] (**Box 9.2**).

BOX 9.2: METABOLIC PATHWAY INFORMATION DISPLAYED IN BRENDA

The focus of BRENDA [24] is quantitative information on enzymes, which can be searched by name or Enzyme Commission (EC) number, as well as by species. **Figure 1** is a screenshot showing the results of a typical search. BRENDA offers values for important quantities such as K_M values, substrates, co-factors, inhibitors, and other features that can be very important for modeling purposes. All information is curated and backed up with references.



BRENDA

The Comprehensive Enzyme Information System

EC 3.2.1.28 - alpha,alpha-trehalase

MgCl2	Saccharomyces cerevisiae	50 mM, C-trehalase, complete inhibition	136150	2D-image
phosphate	Saccharomyces cerevisiae	-	136165	2D-image
ZnCl2	Saccharomyces cerevisiae	0.1 mM, C-trehalase, complete inhibition	136150	2D-image

ACTIVATING COMPOUND	ORGANISM	COMMENTARY	LITERATURE	IMAGE
3',5'-cAMP	Saccharomyces cerevisiae	activation is dependent on presence of ATP and a divalent cation such as Mg2+, Mn2+ or Co2+	136162	2D-image
MgATP2-	Saccharomyces cerevisiae	C-trehalase can be activated by MgATP2- in presence of cAMP	136150	2D-image
additional information	Saccharomyces cerevisiae	the cryptic enzyme form is completely activated at protein concentrations higher than 60 mg/ml	136162	-

KM VALUE [mM]	KM VALUE [mM] Maximum	SUBSTRATE	ORGANISM	COMMENTARY	LITERATURE	IMAGE
11.78	-	alpha,alpha-Trehalose	Saccharomyces cerevisiae	-	656417	2D-image
0.5	-	Trehalose	Saccharomyces cerevisiae	-	136158	2D-image
1.4	-	Trehalose	Saccharomyces cerevisiae	V-trehalase	136150	2D-image
4.7	-	Trehalose	Saccharomyces cerevisiae	-	136141	2D-image
4.79	-	Trehalose	Saccharomyces cerevisiae	in absence of trehalose-c	136144	2D-image
5	-	Trehalose	Saccharomyces cerevisiae	-	136165	2D-image
5.28	-	Trehalose	Saccharomyces cerevisiae	in presence of trehalose-c	136144	2D-image
5.7	-	Trehalose	Saccharomyces cerevisiae	C-trehalase	136150	2D-image
10	-	Trehalose	Saccharomyces cerevisiae	apparent value, at pH 7.1	677711	2D-image

Figure 1 Typical information provided in BRENDA. Specifying the EC number or name of trehalase offers a lot of quantitative information that can be very useful for pathway modeling. The screenshot here shows only a selection of results related to *KM* values obtained under various conditions. BRENDA also has direct links to KEGG (see Box 9.3), BioCyc (see Box 9.4), and other databases. (Courtesy of Dietmar Schomburg, Braunschweig University of Technology, Germany.)

9.2 METABOLIC PATHWAYS AND PATHWAY SYSTEMS

9.2.1 Biochemistry and Metabolomics

If one had to describe the difference between traditional enzyme kinetics and metabolomics, one might simplistically say that the former focuses primarily on individual reactions and pathways, along with factors influencing them, while the latter is primarily interested in (large) assemblies of reactions that collectively form pathway systems. Expressed differently, the main focus of traditional biochemistry and enzyme kinetics can be seen as targeted and local, while the focus of metabolomics is comprehensive and global. This simplified distinction may suggest that biochemistry is a subset of metabolomics, but this is not entirely so, because metabolomics looks at so many reactions simultaneously that details at the local level are often only of secondary interest. As a case in point, an important component of enzyme kinetic investigations is a detailed protocol for the purification of an enzyme and the characterization of its affinity to different substrates. By contrast, typical metabolomic studies attempt to determine two features. The first is the **metabolic profile** of a biological sample, which consists of the relative (or absolute) amounts or concentrations of all metabolites. The second feature is the **flux profile**, which consists of the quantification and distribution of the relative (or absolute) magnitudes of all fluxes throughout the system. Here, each flux is defined as the rate of turnover of molecules in a reaction step or pathway or, expressed differently, as the amount of material flowing through a metabolite pool or pathway at a given time. Some authors reserve the term flux for pathways, while calling essentially the same quantity a rate when a single reaction is being studied [29].

Studying hundreds of metabolites or fluxes simultaneously usually mandates that details of individual reaction steps or mechanisms of inhibition or activation be ignored. This situation is comparable to the different foci in zoology and ecology, where the former primarily targets the unique features of specific animals or animal species, while the latter tries to understand the interactions among (many) species, including animals, plants, and sometimes even bacteria and viruses.

Of course, there is plenty of overlap between enzyme kinetics and metabolomics. For instance, the flux through a pathway system is related to the enzyme activities and V_{max} values of the individual reaction steps. One could surmise that the analysis of the dynamics of a metabolic system might not require very detailed information regarding each and every reaction step, as long as we know the turnover rate of the reaction and its modulators. However, once we start extrapolating

results from the analysis of one set of conditions to another, these details can become very important. Thus, traditional biochemistry and metabolomics have their own foci but also much common ground.

While a little naïve, the previous distinction between traditional enzyme kinetics and metabolomics is useful for now, because it provides rationale and explanations for the different classes of analytical methods in the two fields, and for the types of data that are associated with them. Traditional biochemistry has revealed an enormous amount of information regarding biochemical reactions, whereas modern metabolomic data consist of comprehensive metabolic concentration or flux profiles, from which, ideally, information can be extracted regarding both steady states and the dynamics of metabolites. Often, methods of machine learning are applied toward this extraction [30, 31].

9.2.2 A Very Brief Synopsis of Methods for the Generation of Metabolome Data

Throughout the history of biochemistry, precise measurements of concentrations and fluxes have been as important as the characterization of enzymes, modulators, and turnover rates. A distinguishing feature of modern high-throughput metabolomics is the simultaneous generation of large metabolite or flux profiles, which sometimes consist of hundreds of data points obtained in a single experiment [32, 33]. Most of these methods measure systems at a steady state, but some techniques also allow characterization of changing profiles over a sequence of time points, following some stimulus. It is worth noting in this context that there is a genuine difference in complexity between metabolomics on one hand and genomics and proteomics on the other. The genome consists of only four bases and the proteome of 20 amino acids, and these are rather similar. By contrast, the metabolome consists of thousands of metabolites, which are chemically very diverse and come in all sizes, chemical families, and affinities to water or lipids.

Uncounted methods have been developed for measuring metabolites, and a comprehensive account is far beyond the scope of this book. Some of these are quite generic whereas others are specific for the metabolites in question. Generically, metabolome analysis normally entails a sequence of processes [33, 34]:

1. Sample preparation and quenching of metabolic activity. The key to this task is efficient rapid sampling [35]. This step is crucial, because any changes induced by a stimulus or any intended or unintended variation in the environment can directly affect the metabolite concentrations in the cell, and these often have a turnover rate at the order of a mole per second, and many metabolites respond to new conditions within the millisecond range [33]. Thus, to yield valid snapshots of a metabolic system, any rapid-sampling method must instantaneously quench all metabolic activity. This quenching is typically achieved through very fast changes in temperature, for instance, by injecting the sample into liquid nitrogen, methanol, or a solution containing ethanol and sodium chloride [36].
2. Extraction procedures. Intracellular metabolites must be separated from metabolites of the external medium and extracted with a reagent that disrupts the cell, releases the intracellular metabolites, and denatures the enzymes, thereby preventing further degradation and biochemical conversions [34]. In exceptional cases, measurements in living cells are feasible, for instance, with fluorescence techniques or in vivo NMR.
3. Separation of metabolites. High-throughput metabolome studies often use gas chromatography (GC), combined with mass spectrometry (GC-MS) or high-performance liquid chromatography (HPLC). Capillary electrophoresis (CE) encompasses a family of methods for separating ionized molecules based on their charge and has become a valuable tool for metabolic fingerprinting [37].
4. Quantification and/or metabolic profiling. Among various methods, mass spectrometry (MS) and nuclear magnetic resonance (NMR) spectroscopy stand out. MS may be used as a stand-alone technique without prior separation, or in combination with a separation technique such as GC, HPLC, or CE [38].

5. Flux measurements. Several computational analyses require the experimental determination of metabolic fluxes. Fluxes entering or leaving the biological system are comparatively easy to measure with methods of analytical chemistry. By contrast, it is difficult to quantify internal fluxes, but these are often of greatest interest. One method that addresses some of these issues is called **metabolic flux analysis** and is based on isotopomers, which are isomers with isotopic atoms in different positions [39]. For instance, in a typical isotopomer experiment with ^{13}C, a type of carbon that is much rarer than the common ^{12}C, a metabolite with three carbons can at most have 2^3 isotopomers, because each carbon may independently be labeled or unlabeled. The various isotopomers are usually not equally likely and instead are produced with a frequency distribution that contains clues regarding the internal fluxes of a system that need to be deciphered with sophisticated computational methods.

9.3 DATABASES FOR MODELING METABOLIC SYSTEMS

Of enormous help for the construction of a metabolic pathway model are freely accessible databases such as the Kyoto Encyclopedia of Genes and Genomes (KEGG) [40] and BioCyc [41] (**Boxes 9.3** and **9.4**), which contain collections of pathways, along with information regarding their enzymes, genes, and plenty of useful comments, references, and other pieces of information, along with links to other databases like BRENDA. The databases furthermore provide numerous tools for exploring pathways in well-characterized organisms such as *Escherichia coli*, the baker's yeast *Saccharomyces cerevisiae*, mice and humans, and also for inferring the composition of ill-characterized pathways, for instance, in lesser-known microbial organisms.

BOX 9.3: METABOLIC PATHWAY REPRESENTATION IN KEGG

Chapter 11 will analyze a small metabolic pathway in yeast that is very important for the organism's response to stresses such as heat. The main compound of interest within this pathway is the sugar trehalose. **Figures 1–3** are screenshots from KEGG—the Kyoto Encyclopedia of Genes and Genomes [40]. KEGG consists of many databases with information on genomes, pathways, functional aspects of biological systems, chemicals, drugs, and enzymes as well as links to other pertinent databases.

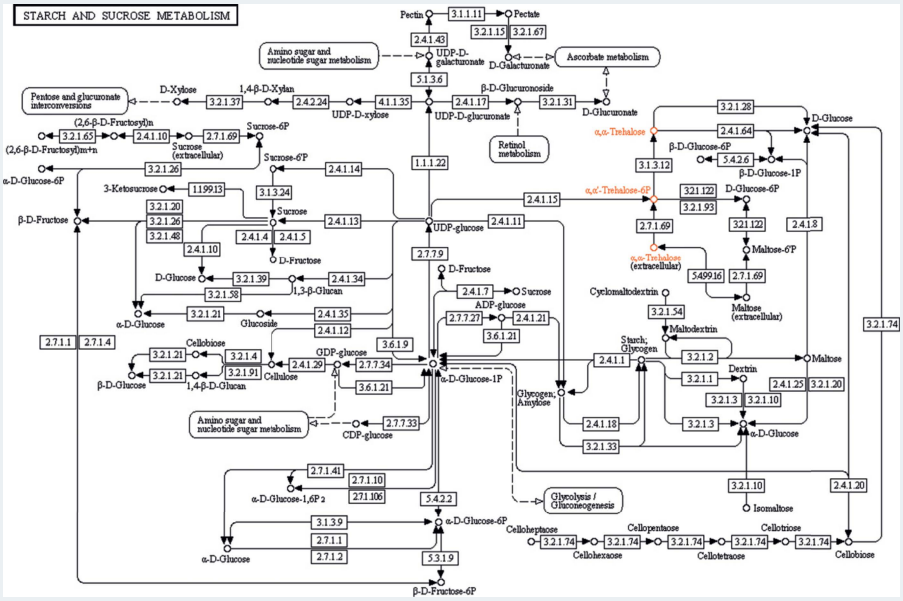


Figure 1 Metabolic network associated with trehalose dynamics. Trehalose is a carbohydrate and can be found in KEGG's starch and sucrose pathway system. In this representation, the metabolites are the nodes. Associated with the reactions (edges) are numbers in boxes, which are the EC numbers that uniquely identify the catalyzing enzymes. (Courtesy of Kanehisa Laboratories.)

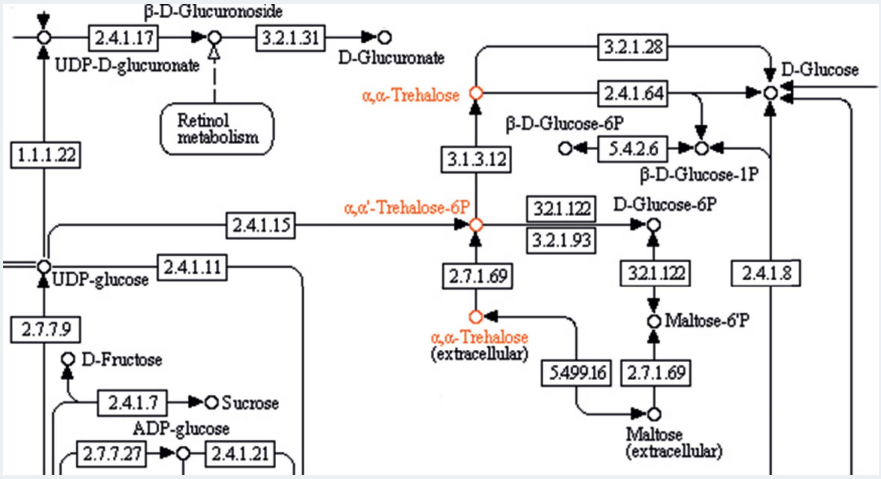


Figure 2 A more detailed view of the synthetic and degradation pathways of trehalose in KEGG. Clicking on an item in a box leads to further information (see Figure 3). (Courtesy of Kanehisa Laboratories.)

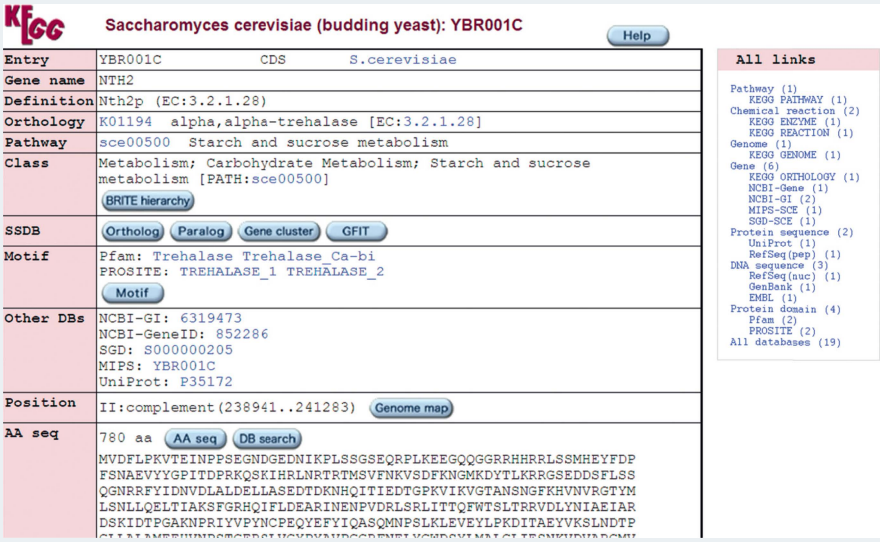


Figure 3 Details provided by KEGG. Clicking on an EC number in a pathway representation as in Figure 2 leads to biochemical and genome-related features of the corresponding enzyme. The information displayed here describes the enzyme trehalase, which degrades trehalose into glucose (see Chapter 11). (Courtesy of Kanehisa Laboratories.)

BOX 9.4: METABOLIC PATHWAY REPRESENTATION IN BIOCYC

As mentioned in Box 9.3, Chapter 11 will require information on the sugar trehalose in yeast. **Figures 1** and **2** are screenshots from BioCyc. Running the cursor over any of the entities shown on the website generates names, synonyms, reactions, and

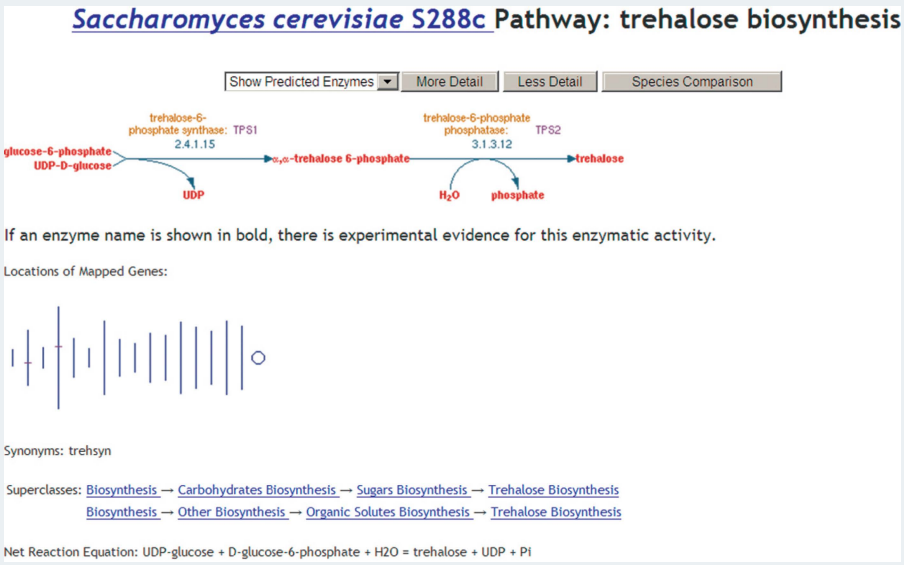


Figure 1 Screenshot from BioCyc. Specifying the yeast *Saccharomyces cerevisiae* as the organism of interest and selecting trehalose biosynthesis as target pathway leads to a display of the pathway. All pertinent items can be clicked and guide the user to further information. (Courtesy of Peter Karp, SRI International.)

other useful information. BioCyc also provides information on pertinent genes, a summary of pathway features, references, and comparisons with other organisms. BioCyc furthermore offers sophisticated tools for mining **curated** databases.

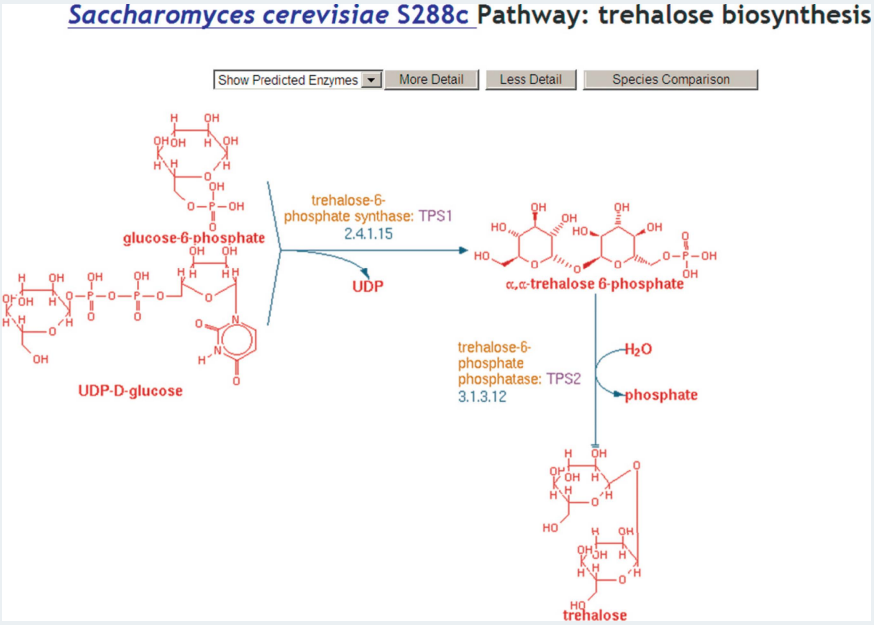


Figure 2 Details available in Biocyc. The simplified pathway representation in Figure 1 allows the user to request more detail, which is shown here. (Courtesy of Peter Karp, SRI International.)

In addition to KEGG and BioCyc, other tools are available for metabolic pathway analysis. The commercial software package ERGO [42] offers a bioinformatics suite with tools supporting comparative analyses of genomes and metabolic pathways. It also assigns functions to genes and helps shed light on ill-characterized genes and gene products. Users of ERGO can combine various sources of information, including sequence similarity, protein and gene context, and regulatory and expression data for optimal functional predictions and for computational reconstructions of large parts of metabolism.

Another interesting database of biomolecular pathways is Reactome [43–45], which contains not only metabolic pathways but also information regarding DNA replication, transcription, translation, the cell cycle, and signaling cascades. The platform is open source, allows open access, and is linked to numerous databases. Its graphical user interface provides ready access to bioinformatics tools for the visualization, interpretation, and analysis of genomic, metabolic, and signaling systems. The Reactome platform can be a very useful initial resource for exploring a pathway before one starts to design a model, as it has an efficient pathway browser that displays all relevant components and interactions, as well as links to pertinent databases containing information about gene expression in different tissues, molecular properties, and disease associations. COSBI [46] offers computational tools for establishing data analysis pipelines leading to pathway analysis and simulation.

The Lipid Maps project [47] provides information and databases specifically for lipids and lipidomics, a term that was chosen to indicate that the totality of lipids may be comparable to genomics and proteomics in scope and importance. Indeed, the more we learn about lipids, the more we must marvel at the variety of functions that they serve—from energy storage to membranes and to genuine mechanisms of signal transduction. And not to forget: 60% of our brain is fat.

9.4 CONTROL OF PATHWAY SYSTEMS

A key feature of metabolic systems is the regulation and control of the flow of material through the various involved pathways. Regulation allows the system to

respond to changing environments and changing demands where some metabolite is needed in greater amounts than normally. For instance, by changing the activity of an enzyme at a branch point where two pathways diverge, more material can be channeled into one or other pathway, depending on demand. We can easily see the importance of controllability by studying what would happen without control. Imagine a water distribution system with pipes of fixed widths, where at each node the collective cross section of all input pipes equals the collective cross section of all output pipes. If the system is running at maximal capacity, the amount of water at each point is proportional to the pipe size. If less water is entering the system, the amounts decrease correspondingly, but without means of intervention it is impossible to steer the water flow selectively into a direction where it is most needed for a particular purpose.

The control of metabolic pathway systems occurs at several levels and on multiple timescales (see Chapter 11). The fastest is a (temporary) change in enzyme activity, which is achieved at the metabolic level itself. A premier example is feedback inhibition by the end product of a pathway: the more product is generated, the more the enzyme at the beginning of the pathway is inhibited (**Figure 9.5**). Similarly, by inhibiting one pathway exiting a branch point, the other branch receives more substrate. In many cases, the end product inhibits not just its own branch but activates the competing branch and/or inhibits a reaction step upstream of the branch. In a nested feedback design, for instance, either end product inhibits its own branch and also the initial substrate production (**Figure 9.6**). These types of feedback patterns have been analyzed in detail with regard to the efficiency of alternative designs (see, for example, [17, 48]). The methods described in Chapters 2 and 5 allow us to set up models for these types of pathways, and Chapter 6 provides recipes for finding appropriate parameter values if the right data are available.

The second manner of control happens at the proteomic level, where enzymes can be activated or de-activated just like other proteins—for instance, by covalent modification in the form of phosphorylation, ubiquitination, or glycosylation; by S–S bonding; or through protein–protein interactions. Of course, it is also possible to degrade enzymes permanently by means of proteases.

The third option for controlling metabolism occurs at the level of gene expression, where transcription, translation, or both can be affected. At the level of transcription, the gene coding for an enzyme may be upregulated and lead to a larger amount of enzyme and therefore to an increased turnover capacity. This mode of control, which often relies on the mobilization or translocation of transcription

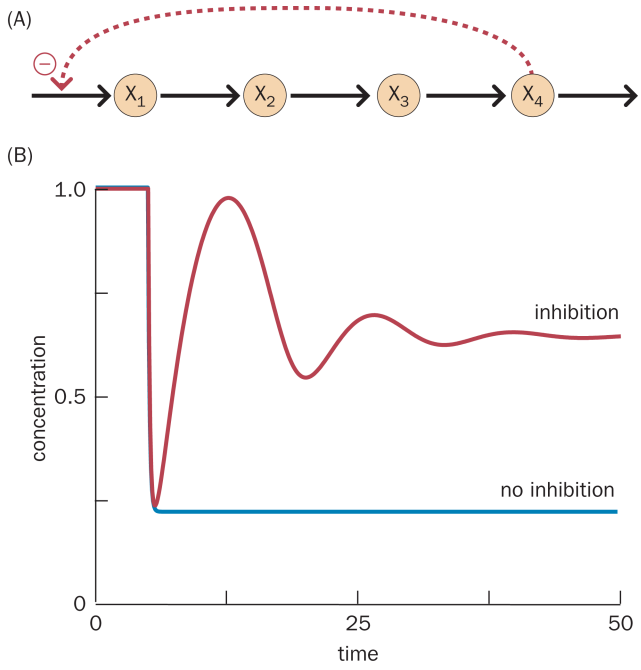


Figure 9.5 Linear pathway with feedback. (A) Reaction scheme with feedback inhibition of the initial step by the end product. (B) Comparison of responses to a sudden and persistent demand of metabolite X_4, starting at time $t = 8$. With feedback, the concentration of X_4 oscillates and converges to a level of about two-thirds of the initial value. Without feedback, the concentration of X_4 sinks to less than one-quarter.

factors, is very effective but much slower than metabolic regulation such as end-product inhibition, because it involves transcription and translation, which typically occur on a timescale of the order of tens of minutes if not hours. It is also possible to exert regulation at the level of translating RNA into proteins.

Finally, metabolic control can be implemented through the design of the pathway and, in particular, with isozymes, that is, with distinct but similar enzymes catalyzing the same reaction. For instance, in a situation as shown in Figure 9.6, it is not uncommon to find two isozymes for the production of A, each of which is inhibited by only one of the end products, C or D. In addition to feedback inhibition, these isozymes may also be differently regulated at the genetic level. The two isozymes may furthermore be physically separated by forming different supramolecular complexes or through their location in different compartments.

In complicated situations such as stress responses, the regulation of metabolism is the result of a combination of several modes of action, including direct metabolic adjustments, the activation of transcriptional regulators, post-translational modifications, and various signaling events [49]. The immediate response may also occur directly at the molecular level. For instance, it is known that temperature can change the activity of enzymes that are involved in heat-stress response [50], and it has been shown that this mechanism is sufficient for mounting an immediate cellular response, whereas heat-induced changes in gene expression govern heat-stress responses over longer time horizons (Chapter 11 and [51]).

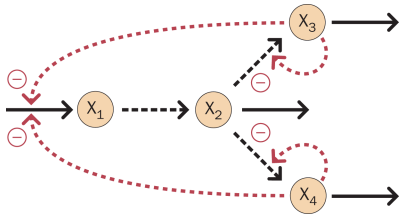


Figure 9.6 Branched pathway with several feedback inhibition signals. Depending on the strengths of the feedback signals, material may be channeled preferentially into either one of the two branches. (Adapted from Savageau MA. Biochemical Systems Analysis: A Study of Function and Design in Molecular Biology. Addison-Wesley, 1976. With permission from John Wiley & Sons.)

9.4.1 Metabolic Control Analysis

Metabolic control analysis (**MCA**) was devised to quantify the impact any of the fluxes within a system has on the overall response of a pathway. At least the original versions of MCA were concerned exclusively with pathway operation close to a stable steady state and very small perturbations [29, 52–55]. MCA was conceived to replace the formerly widespread notion that every pathway has one rate-limiting step, which is a slow reaction that by itself is credited with determining the magnitude of flux through the pathway. Such a rate-limiting step could be compared to the neck of a funnel, which controls how much liquid can run through the funnel per second, independent of how much liquid has accumulated in the funnel. In linear pathway sections without branches, the rate-limiting step was traditionally thought to be positioned at the first reaction, where it was possibly inhibited through feedback exerted by the end-product. In MCA, this concept of a rate-limiting step was supplanted with the concept of shared or distributed control, which posits that every step in a pathway contributes, to some degree, to the control of the steady-state flux. MCA formalizes this concept primarily with quantities called control coefficients and elasticities.

The control coefficients measure the relative change in a flux or substrate concentration at steady state that results from a relative change in a key parameter, such as an enzyme activity. In other words, one asks a question like: "If one alters the activity of one of the enzymes, E_i, by a small amount, say 2%, how much will the overall flux J through the pathway change?" It is assumed that each reaction R_i is directly proportional to the catalyzing E_i so that the control coefficients may be equivalently expressed in terms of R_i or E_i. This assumption holds in most cases, although one should be aware that there are exceptions [56]. Furthermore, it is mathematically convenient to replace the relative change with a change in the logarithm of the quantity in question, which is valid if the changes are small. Each elasticity coefficient measures how a reaction rate R_i changes in response to a perturbation in a metabolite S_k or some other parameter, much like the strength of a kinetic order in biochemical systems theory (Chapter 5). In this sense, control coefficients and elasticities expand the sensitivity analysis from the simpler metrics of pairwise parameter-output relationships described in Chapter 5 to a whole-system evaluation.

The main insights provided by MCA are gained from relationships among the control and elasticity coefficients. Specifically, it was shown that all effects of perturbing enzyme levels, in the form of flux control coefficients, sum to 1 and that all concentration control coefficients with respect to a given substrate S sum to 0. This summation relationship with respect to flux control has two interesting implications. First, one sees directly that the control of metabolic flux is shared by all

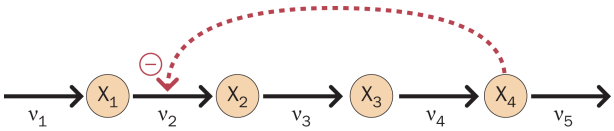


Figure 9.7 Assessment of metabolic control over a linear pathway flux J. Under steady-state conditions, MCA evaluates the contribution of each reaction rate v_i to the overall flux J as flux control coefficients $\mathbf{C}^J_{vi}$. The contributions of each metabolite X_i to a reaction rate v_i are quantified as elasticities ε^{vi}_{Xi}. Connectivity relations impose the constraint $\sum_{i=1}^{n} \mathbf{C}^J_{vi} \cdot \varepsilon^{vi}_{Xi} = 0$.

reactions in the system; this global aspect identifies the control coefficients as systemic properties. And second, if a single reaction is altered and its contribution to the control of the flux changes, the effect is compensated by changes in flux control by the remaining reactions. The summation relationship with respect to flux control has often been used experimentally in the following fashion: one measures in the laboratory the effects of changes in various reactions of a pathway, and if the sum of flux control coefficients is below 1, then one knows that one or more contributions to the control structure are missing.

A second type of insight comes from connectivity relationships, which establish constraints between control coefficients and elasticities. These relationships have been used to characterize the close connection between the kinetic features of individual reactions and the overall responses of a pathway to perturbations.

An application is the following. Suppose there is reason to believe that the reaction step v_2 is a bottleneck and that the goal of the analysis is to propose strategies for increasing the flux through the pathway in **Figure 9.7**. A reasonable strategy might be to increase the amount of the enzyme in v_2, for instance, using modern tools of genetic engineering. But it could instead be better to modify the enzyme in such a manner that it is less affected by the inhibition exerted by X_4. Which alternative is better? MCA helps answer questions of this type.

For small pathways, MCA computations are quite simple. For larger systems, matrix methods have been developed that permit more structured assessments of the control within pathways [57, 58]. It has also been shown that the fundamental features of MCA are directly reflected and generalized in the steady-state properties of fully dynamic models in biochemical systems theory (BST) [59, 60] and in the lin–log formalism [61, 62]. Thus, MCA can be seen as a direct transition from static networks to dynamic systems analyses. Many variations of the original quantities in MCA have been defined over the years (e.g., see [29, 63]).

9.5 COMPUTATIONAL ANALYSIS OF DYNAMIC PATHWAY SYSTEMS

The analysis of metabolic pathway systems typically follows one of two routes, which are distinct but directly complement each other. The first is stoichiometric analysis. It focuses on the connectivity within the pathway system and addresses the question of which metabolite can be converted directly into which of the other metabolites. Such a stoichiometric analysis begins with collecting information regarding known or alleged reactions producing or degrading the metabolites in the pathway system of interest. These reactions form a stoichiometric map consisting of nodes (metabolites) and directed vertices (reactions). In mathematical terms, this map is a directed graph, for which many methods of analysis are available (Chapter 3). The second route of pathway analysis expands stoichiometric network models toward fully regulated dynamic systems that can be analyzed with methods discussed in Chapter 5. We begin with the second route and then discuss the first.

9.5.1 The General Stoichiometric Model

The connectivity of a stoichiometric network is captured in the stoichiometric matrix, which shows directly which metabolite is associated with one or more reactions (see [64, 65] and Chapter 3). This concept is readily expanded to a dynamic representation. Namely, the general stoichiometric model is formulated as

$$\dot{\mathbf{X}} = \boldsymbol{N}\,\boldsymbol{R}. \tag{9.22}$$

Here, $\dot{\mathbf{X}}$ is the vector of derivatives of all involved metabolites, $\boldsymbol{N}$ is the stoichiometric matrix, and $\boldsymbol{R}$ is a vector of reactions.

Let us illustrate the process of setting up the model with the example shown in **Figure 9.8**. The pathway contains four metabolites, seven reaction steps, and one inhibition signal.

The first task is the construction of the stoichiometric matrix, which displays the associations between all metabolites (rows) and all reactions (columns). For the example, **N** has the format

$$\mathbf{N} = \begin{array}{c} \begin{matrix} R_1 & R_2 & R_3 & R_4 & R_5 & R_6 & R_7 \end{matrix} \\ \begin{pmatrix} 1 & -1 & 0 & -1 & 0 & 0 & 0 \\ 0 & 1 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & -1 & -1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 & -1 \end{pmatrix} \end{array} \begin{matrix} X_1 \\ X_2 \\ X_3 \\ X_4 \end{matrix}$$

Note that the inhibition signal is not presented in this matrix at all; the vector of reactions will account for it. As discussed in Chapter 3, a value of + 1 is assigned to element (i, j) of **N** if a reaction j points toward variable i, and a value of –1 is assigned if it points away from it. A value of 0 reflects no association. Other integer values would be needed for merging or splitting reactions (Chapter 3), which are not present here.

Next is the vector **R**. The stoichiometric matrix provides no information about the reactions, except for their association with particular metabolites. This information must come from other sources or rely on assumptions. Suppose the following:

R_1: this input to the system is constant: $R_1 = Input$

R_2: this reaction can be modeled as MMRL with competitive inhibition by compound X_3:

$$R_2 = \frac{V_{\max 2} X_1}{K_{M2}(1 + X_3 / K_I) + X_1}$$

R_3: an appropriate representation of this reaction is HRL. Thus,

$$R_3 = \frac{V_{\max 3} X_2^h}{K_{M3}^h + X_2^h}$$

R_4: this reaction is modeled as a power-law function:

$$R_4 = \gamma_4 X_1^{f_4}$$

R_5: this reaction is a simple MMRL:

$$R_5 = \frac{V_{\max 5} X_3}{K_{M5} + X_3}$$

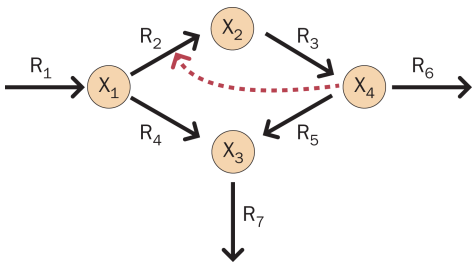


Figure 9.8 Illustration pathway for setting up a stoichiometric model. The dashed red arrow represents feedback inhibition.

R_6: this transport out of the system is linear:

$$R_6 = c_6 X_3$$

R_7: this consumption process is a power-law:

$$R_7 = \gamma_7 X_4^{f_7}$$

These assignments make it evident that any constant, linear, or nonlinear process description may be used. The nonlinear representations of the reactions may involve regulation, as it is the case for R_2. Plugging **N** and $R_1, \ldots, R_7$ into (9.22) yields a fully parameterized ODE model of the pathway.

The stoichiometric equation is a generic format for any dynamic metabolic systems model in ODE format. Importantly, this equation separates linear components (the system's connectivity structure) from nonlinear components (specific process descriptions) of the system. For instance, a Michaelis–Menten model and a GMA model of the same pathway have the same stoichiometric matrix but different reaction vectors.

As an example, let's look again at the pathway diagram in Figure 9.8. Both the original and the GMA model have the same stoichiometric matrix **N**. For the reaction vector, we enter parameter values into the reaction vector of the original model. Furthermore, we compute the power-law approximation with the steady state of the system as operating point, with methods discussed in Chapter 5; the result is displayed in **Table 9.1**. Starting the two models (original, GMA) at the steady state and perturbing the input at $t = 15$ and $t = 25$ by setting $R_1(0) = 20$, $R_1(15) = 25$, $R_1(25) = 20$ yields the output shown in **Figure 9.9**. We can see that the two models differ slightly but not all that much in their responses, especially if we imagine that they would model actual data, which always have some noise (see Chapter 6).

Metabolic analyses of the resulting dynamic models follow the methods presented in Chapters 2 and 5. They typically include studies of the following types:

1. Test whether information on metabolites, enzymes, and modulators, when assembled in a dynamic model, yields reasonable results. If not, important information is missing.
2. Unlimited types of what-if scenarios are possible: if one feature of the model (a parameter, initial value, process representation) is changed, what are the consequences for the dynamics and the steady state of the system?

TABLE 9.1: COMPARISON OF FORMATS OF REACTIONS CORRESPONDING TO THE DIAGRAM IN FIGURE 9.8

Reaction (see Figure 9.8)	*Original Format*	*GMA Format*
R_1	20	20
R_2	$\frac{12 X_1}{4(1+X_3/2.5)+X_1}$	$1.554 X_1^{0.690} X_3^{-0.092}$
R_3	$\frac{5 X_2^2}{0.5^2 + X_2^2}$	$4.043 X_2^{0.408}$
R_4	$8 X_1^{0.5}$	$8 X_1^{0.5}$
R_5	$\frac{4 X_3}{3+X_3}$	$1.125 X_3^{0.521}$
R_6	$0.5 X_3$	$0.5 X_3$
R_7	$2 X_4^{1.2}$	$2 X_4^{1.2}$

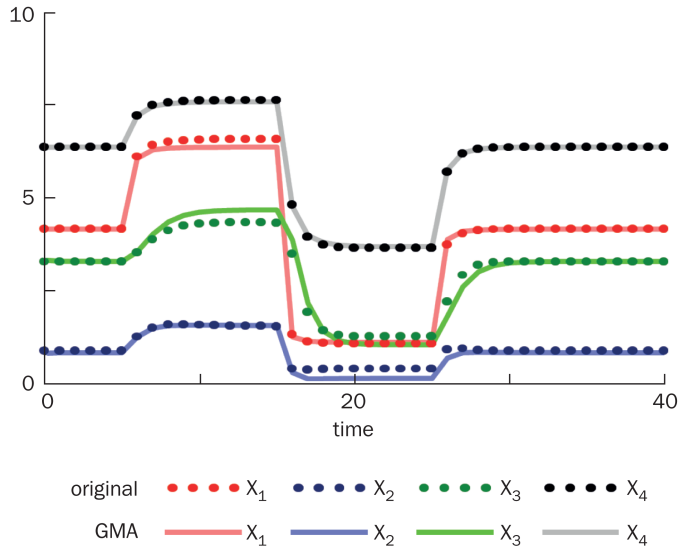


Figure 9.9 Comparison of different model formulation. The responses of the original model and the corresponding GMA model, representing the pathway in Figure 9.8 are superimposed. The models start at the steady state and the input R_1 is perturbed at times 5, 15, and 25. See text for details.

3. Is it possible to "correct" a system that is malfunctioning, for instance, due to disease, by altering some of the parameter values? Such alterations might correspond to a candidate for a drug treatment (see Chapter 13).
4. Is it possible to explain system responses that are not intuitive?
5. Is it possible to explain from observation data how a system responds to a mutation or other alteration?
6. Can the model be used to make predictions regarding changes in parameter values that would lead to the increased production of a desirable metabolite or flux? This task is at the core of metabolic engineering.

Biochemical systems theory (BST), which assigns power-law functions to all reactions and was introduced in Chapter 5, is a very effective framework for all these types of analyses, as models can be set up for given pathway diagrams without detailed knowledge of molecular mechanisms. The intentional limitation of the approach to power-laws may seem to be an overly harsh restriction, but embedded within a whole pathway system, the system responses, for instance of systems modeled either within BST or with MMRL, HRL, and extensions toward inhibition and other types of modulation, are often surprisingly minor (see Figure 9.9 and e.g., [66–68]).

In a totally different application, the stoichiometric formulation of a dynamic system may be used to identify features of the fluxes within the system from experimental data [69, 70].

9.5.2 Analyses at the Steady State

Important analyses of a metabolic system become much simpler if the system operates at a steady state, where all changes in metabolite concentrations are zero ($\dot{\mathbf{X}} = 0$) and the stoichiometric equation (9.22) therefore reduces to a system of algebraic equation

$$\mathbf{N}\mathbf{R} = 0. \tag{9.23}$$

Although each of the reactions in the vector $\mathbf{R}$ may have about any mathematical format, it has a single value at a given steady. This is important because the equations in (9.23) are therefore linear and can be analyzed with the enormous repertoire of available tools from linear algebra and linear operations research. The stoichiometric equation is thus a natural bridge between network models (Chapter 3) and dynamic systems models (Chapter 5).

The main application of stoichiometric analysis is **metabolic engineering**, which has the overall task of producing organic compounds that either are very

valuable or must be generated cheaply in large quantities. The former group includes insulin and amino acids, while the latter includes industrial alcohol and citric acid.

The overall concept is the following. Microorganisms like the baker's yeast *S. cerevisiae* and the bacterium *E. coli* can produce valuable organic compounds with exquisite purity. For instance, yeast cells have been known for a long time to convert sugars into alcohol. The issue is that wild-type strains produce these compounds only in moderate quantities. Thus, it is the task of the metabolic engineer to improve yield, often by a factor of 1,000 or more (recall the example of artemisinin from Chapter 8). A second task might be to splice genes into well-known organisms that code for desirable compounds and are taken from other organisms. Traditionally, metabolic engineers pursued these goals with numerous iterations of genetic manipulations, repeated fine-tuning of the growth medium, and selection of the best strains. As an emerging alternative, modern methods of metabolic systems analysis may be pursued that use computational means [27, 71, 72].

In relatively straightforward cases, it is clear which enzyme is to be changed to improve the yield of a metabolic system. However, this is not so in complicated pathway systems. A good example is citric acid production by microbes, which has been gradually improving over a span of over 100 years [73]. As a consequence of this repeated trial-and-error, simple changes in one or two genes or enzymes can no longer improve yield further, because they have already been detected during the iterative improvement of strains and media. It has therefore been proposed to set up stoichiometric or dynamic models of pathway systems and to optimize them toward a desired goal, while taking into account physiological constraints on metabolite concentrations, fluxes, and metabolic burden [27, 71, 74]. The latter means that it is costly for the organism to maintain a lot of mRNAs, proteins, and metabolic products, and if these are expendable, the bacterium has an advantage if they are eliminated. Many articles have been written about this approach, and we will not pursue it further here, because its goals and concepts are intuitive, but the techniques of analysis and implementation are rather difficult.

Two specific approaches to computational metabolic engineering have received enormous attention over the past decades: flux balance analysis (FBA) and elementary mode analysis (EMA).

9.5.3 Flux Balance Analysis (FBA)

The goal of **flux balance analysis** (**FBA**) [74, 75] is the rerouting of material through the fluxes (reactions) in a metabolic pathway system so that as much as possible of a desired output metabolite is produced. Usually, such a task is performed with bacteria or fungi, but it has also become feasible in recent years to use mammalian cells. Of course, there are biological and other limitations to each flux and metabolite concentration, which may be thermodynamic or of some other physicochemical nature. FBA accounts for these **constraints**, using **constrained optimization** as its core method of analysis.

Let's see how this method works by revisiting a simple generic system of fluxes that is similar to the pathway in Figure 3.14 in Chapter 3. The diagram is shown in **Figure 9.10**.

Suppose that the fluxes in the wildtype strain have the values shown in Panel B, expressed in arbitrary units of material moved per some time unit. Importantly, this combination of values is not the only solution that balances all fluxes. For instance, balance at every pool is also achieved in panel C, and indeed there are infinitely many other solutions.

As an example, the task of FBA may be to maximize the efflux from N_3, which in the original system (B) is 20. It would be easy to cut off all material flowing out of N_4, but these types of optimization problems usually come with additional constraints. For instance, it could be that the influxes into N_1 and N_2 cannot be increased and that every internal flux must at least be 10 units, because otherwise the microbes producing these metabolites would not survive. One solution of this constrained optimization task is visualized in panel D, but it is not the only solution. Exercises 9.29 and 9.30 ask you to compute a different optimal solution for similar scenarios.

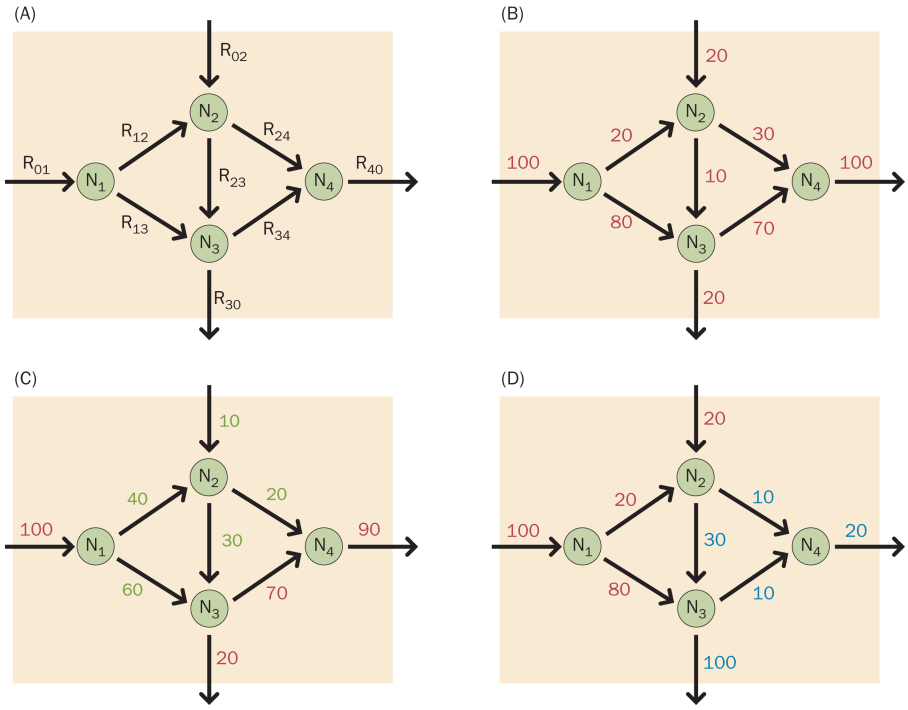


Figure 9.10 Illustration pathway for flux balance analysis. (A) Generic pathway system. (B) System with fluxes at the baseline. (C) Example of flux rerouting. (D) Constrained alterations in flux magnitudes leading to maximal efflux from N_3.

For simple networks, it is not too difficult to obtain a solution by hand. The task becomes more difficult, if not impossible, for more complicated scenarios, but software packages come to the rescue. If the objective function and the constraints are linear, the problem of this **linear programming** task may even be solved in Excel with the *Solver Add-In* within the *Data Analysis* package. As a specific example, suppose the same constraints on fluxes apply as before, but now both effluxes are important, with the efflux from N_3 being 50% more valuable than the efflux from N_4. Thus, the task is to compute the optimal output $3 \times R_{30} + 2 \times R_{40}$; one could also state, e.g., $1.5 \times R_{30} + R_{40}$, which would result in the same optimal flux distribution but half the value of the objective function. **Figure 9.11A** shows the Excel sheet, where the objective function in Cell B5 is defined as **=H8*H9+J8*J9** and the entry in Cell K12 is defined as **=SUMPRODUCT(B12:J12,$B$9:$J$9)**; Cells K13–K22 are to be entered analogously. The constraints in Cells A12–A22 must be entered into the *Solver* window (Figure 9.11B). Not knowing what the optimal flux distribution might be, we arbitrarily enter 10 for all flux values (Cells B9 to J9) as initial settings. Running *Solver* returns the fluxes as

Flux	**R_{01}**	**R_{12}**	**R_{13}**	**R_{02}**	**R_{23}**	**R_{24}**	**R_{30}**	**R_{34}**	**R_{40}**
Value	100	10	90	20	20	10	100	10	20

with a total of 340 (Cell B5). As before, the solution is not unique, and other start settings for the fluxes may yield, e.g.:

Flux	**R_{01}**	**R_{12}**	**R_{13}**	**R_{02}**	**R_{23}**	**R_{24}**	**R_{30}**	**R_{34}**	**R_{40}**
Value	100	36.6667	63.3333	20	46.6667	10	100	10	20
Value	100	90	10	20	100	10	100	10	20

Realistic metabolic pathway systems are much larger, possibly with hundreds of different compounds, and there might be other constraints that must be satisfied. Nonetheless, methods of constrained optimization often generate optimal solutions relatively quickly, even for very large networks. As in the previous example, the structure of the flux network is entered into an algorithm, together with the objective function and with the biological constraints and the algorithm computes

(A)

	A	B	C	D	E	F	G	H	I	J	K	L	M
1	**Linear Programming for Metabolic Engineering**												
2													
3													
4		**Objective Function**											
5	Maximum Value:	50											
6													
7	Fluxes:	R01	R12	R13	R02	R23	R24	R30	R34	R40			
8	Coefficients in Obj. Fct:							3		2			
9	Values:	10	10	10	10	10	10	10	10	10			
10													
11	***Subject to:***										Total		Limit
12	Constraint 1	1	-1	-1							-10	<>	0
13	Constraint 2		1		1	-1	-1				0	<>	0
14	Constraint 3			1		1		-1	-1		0	<>	0
15	Constraint 4						1		1	-1	10	<>	0
16	Constraint 5	1									10	<=	100
17	Constraint 6				1						10	<=	20
18	Constraint 7		1								10	>=	10
19	Constraint 8			1							10	>=	10
20	Constraint 9					1					10	>=	10
21	Constraint 10						1				10	>=	10
22	Constraint 11								1		10	>=	10

(B)

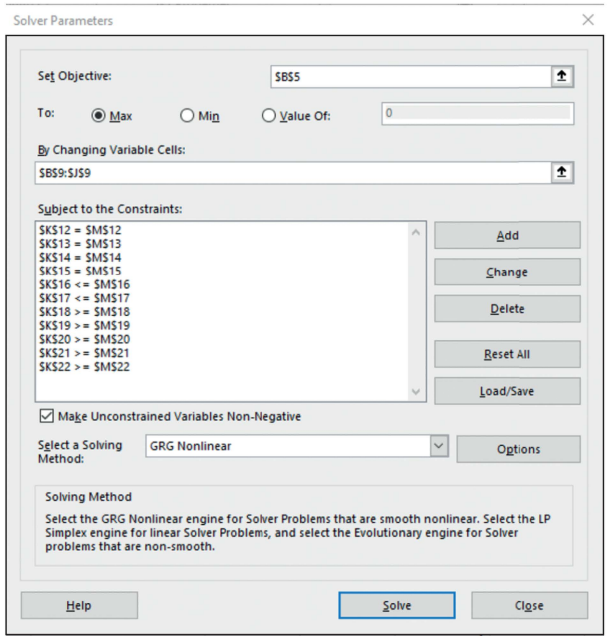


Figure 9.11A Excel sheet for flux balance analysis. Details are described in the Text.
Figure 9.11B Solver window of the Excel optimizer. In this window, the value of the objective function, the parameters available for optimization, and the constraints are entered.

the optimal solution, which identifies the highest value of the objective function satisfying all constraints. If all functions and constraints are linear, the solution is usually obtained without problems; if it is nonlinear, the solution in drastically slower and an optimal solution is not always guaranteed. These optimization methods have been applied uncounted times, usually to microbial populations that are manipulated to produce compounds like industrial alcohol, citric acid, or amino acids.

In an actual metabolic engineering experiment, one would ideally be able to measure many fluxes within a network, because it would facilitate the computation of the remaining fluxes and help with the determination of likely bounds on at least some of the fluxes. In most realistic cases, however, only a few influxes and effluxes can be measured, and they are almost never sufficient to identify the entire network. Nonetheless, with the exclusive focus on the steady state, linear algebra and linear and nonlinear optimization methods offer many tools for a great variety of tasks.

An interesting example is the industrial production of the amino acid lysine, using the microorganism *Corynebacterium glutamicum*. Lysine is a valuable additive for animal feeds, and many studies have therefore worked on improving its yield. Most of these studies used stoichiometric and flux balance analysis (see [72, 76, 77]), but some groups also designed fully kinetic models of the pathway [78, 79]. This option was feasible because the pathway structure of lysine biosynthesis

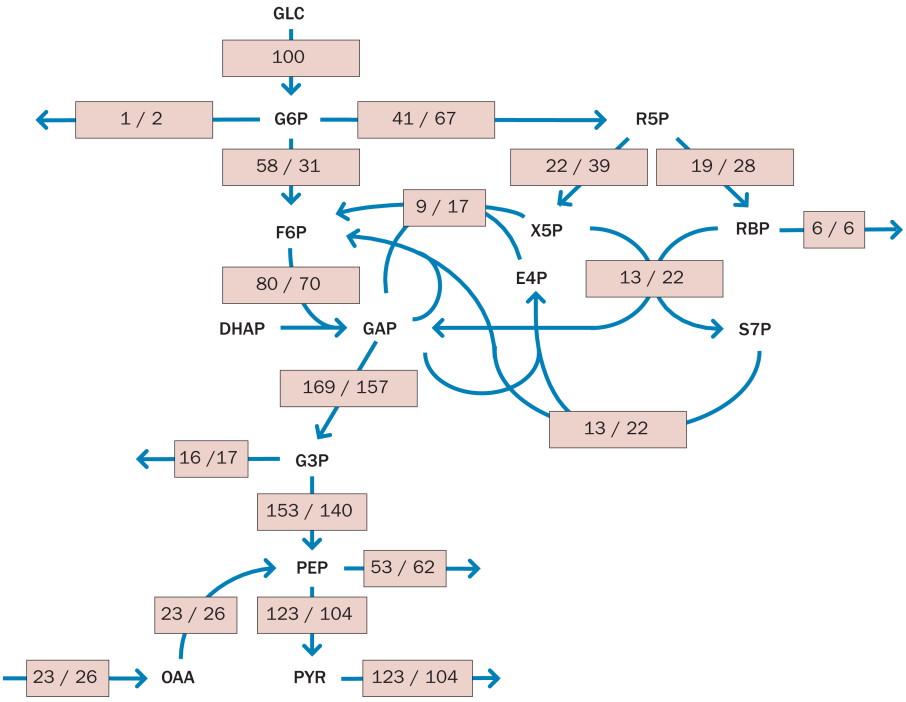


Figure 9.12 Flux distributions in cultures of *E. coli* growing under different conditions. The blue arrows represent enzyme-catalyzed reactions, while the boxes contain flux measurements for two different growth conditions. All magnitudes of fluxes are represented in relationship to the glucose consumption step, which is set as 100. Abbreviations: DHAP, dihydroxyacetone phosphate; E4P, erythrose 4-phosphate; F6P, fructose 6-phosphate; GAP, glyceraldehyde 3-phosphate; GLC, glucose; G6P, glucose 6-phosphate; G3P, 3-phosphoglycerate; OAA, oxaloacetate; PEP, phosphoenolpyruvate; PYR, pyruvate; R5P, ribulose 5-phosphate; RBP, ribose 5-phosphate; S7P, sedoheptulose 7-phosphate; X5P, xylulose 5-phosphate. (Data from Kayser A, Weber J, Hecht V & Rinas U. *Microbiology* 151 [2005] 693–706.)

is well known, and databases such as KEGG, BioCyc, and BRENDA contain information on all the enzymes involved.

As another real-world example, Kayser and co-workers [80] analyzed the metabolic flux distribution in *Escherichia coli* under different glucose-limited growth conditions (**Figure 9.12**). It is easy to check that the incoming and outgoing fluxes balance exactly for each metabolite pool and for both steady-state growth conditions (left and right numbers in beige boxes). For instance, 100 units flow from glucose into glucose 6-phosphate (G6P), and these are forwarded in three directions: depending on the experimental conditions, either 58 or 31 units flow toward fructose 6-phosphate (F6P), 41 or 67 units are used for the production of ribulose 5-phosphate (R5P), and 1 or 2 units move toward unspecified uses, respectively. Note the apparent doubling between F6P and glyceraldehyde 3-phosphate (GAP), which is due to the fact that F6P contains 6 carbons and GAP only 3. The system also produces carbon dioxide and uses other compounds, such as ATP, which are not explicitly listed.

The balances at each metabolite pool can be formulated as a set of simple (stoichiometric) equations. Even if not all fluxes can be measured, these equations can be used to infer them. For instance, the efflux of 16 or 17 units of 3-phosphoglycerate (G3P) in Figure 9.12 simply balances the known influx from GAP and the efflux toward phosphoenolpyruvate (PEP).

Due to its linear nature, FBA is mathematically straightforward but has some significant biological shortcomings that are briefly discussed in Section 9.11, along with partial solutions.

9.5.4 Elementary Mode Analysis in Metabolic Engineering

Elementary modes (EMs) are thermodynamically feasible pathways that form an essentially unique, minimal set of irreversible reactions taking place at a steady state. The set is minimal in a sense that the removal of any reaction will disrupt the functioning of the entire pathway system. Because of this feature, elementary modes can be seen as the inherent building blocks of a metabolic pathway system. Expressed mathematically, any steady-state flux distribution can be decomposed into a linear combination of elementary modes, with, say, 20% EM1, 50% EM4, 30% EM5, and 0% of all other EMs. The concept of EMA was introduced by Schuster and Hilgetag [81] as an alternative to simple stoichiometric network analysis or

FBA, with the specific goal of extracting minimal sets of most relevant pathways from complicated metabolic networks. Liao and co-workers [82] were the first to demonstrate the applicability of the method by optimizing the biosynthesis of 3-deoxy-d-arabino-heptulosonate-7-phosphate (DAHP) in *E. coli*.

EMA begins with a stoichiometric system at steady state (9.23). This equation usually has infinitely many solutions because realistic systems tend to have more fluxes than metabolites. Within this space of solutions, an elementary mode is a subset of reactions that satisfy: (A) the steady-state equation in (9.23) with non-zero fluxes for reactions within the subset and zero fluxes for reactions outside the subset; (B) the necessary thermodynamic constraints of all reactions such that the signs of the non-zero reactions are consistent with the directions of the arrows in the network diagram; and (C) a non-decomposability constraint, which means that no smaller subset of reactions in an elementary node can itself be an elementary mode. The latter condition ensures that no elementary mode can be removed or validly split into simpler modes without rendering some connections between two nodes impossible.

The first step of an EMA consists of the construction of a network describing the pathway system of interest and its stoichiometric matrix. In the second step, one screens the reactions and their directionalities and identifies all non-decomposable pathways within this network. Mathematically speaking, one determines all flux vectors that contain a minimum of active reactions and otherwise contain as many flux values of zero as possible, while still satisfying (9.23). Under different experimental conditions, cells may use any combination of these elementary modes. The next step is EM selection, where one only retains those pathways that connect a desired input with a desired output. Other pathways are subject to deletion, for instance, through knock-out mutations, which tend to increase the efficiency of the organism with respect to the desired pathway. In an optional final step, the chosen pathway may be further subjected to targeted evolution. It is easy to imagine that a realistically sized pathway system contains very many elementary modes. Fortunately, algorithms have been developed to automate this analysis [83–85].

As an illustration, consider the metabolic network shown in **Figure 9.13A**. It is adapted from [86] and characterized by the following stoichiometric matrix:

r_1	r_2	r_3	r_4	r_5	r_{6r}	r_7	r_{8r}	r_9	
1	−1	0	0	−1	0	0	0	0	***A***
0	0	0	0	1	−1	−1	−1	0	***B***
0	1	−1	0	0	1	0	0	0	***C***
0	0	1	−1	0	0	2	0	0	***D***
0	0	1	0	0	0	0	0	−1	***E***

The goal is the use of substrate A_e to produce as much D_e as possible.

Two comments are in order. First, reaction r_7 entering node D, has a coefficient of 2, but a coefficient of –1 with respect to B. The reason is the following: each molecule of type B can be converted into C, which then splits into two molecules of types D and E. Naively speaking, B and C molecules are twice as big as D or E molecules. At the same time, r_7 converts a B molecule directly into type D. To be compatible with the path through C, each B molecule must result in two D molecules. Second, two reactions are shown as reversible. For simplicity, the stoichiometric matrix only shows the dominant reaction. For r_{6r}, the main direction is from B to C, while the minus sign associated with r_{8r} in the row of B indicates a flux out of the system. Nonetheless, r_{8r} may also be an influx, in which case the corresponding flux value is set to –1, and r_{6r} may be used from C to B, again by setting the corresponding flux value to –1.

Either by hand or using an algorithm like [83, 84] or [85, 87], one now enumerates all non-decomposable connections between an input and an output. For example, starting at A_e, a mode may run through r_1, r_2, and r_{6r} (in reverse) and leave the system through r_{8r} toward B_e. The corresponding mode (flux vector) is $(1\ 1\ 0\ 0\ 0\ -1\ 0\ 1\ 0)^{tr}$; it is shown as EM1 in Figure 9.13B. As a second example, consider EM2, which starts at B_e, runs through r_{8r} and r_{6r} to C, where r_3 splits toward D and

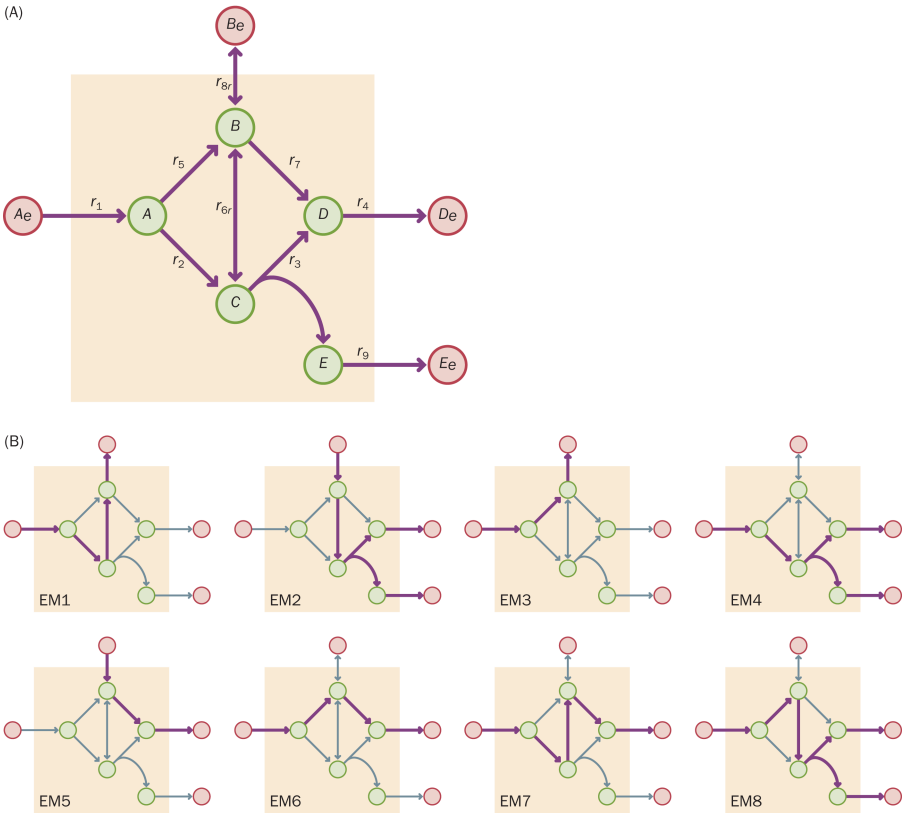


Figure 9.13 Principles of elementary mode analysis (EMA). (A) Illustrative example of a pathway system with four external pools (red, subscript *e*), five internal metabolite pools (green), and several irreversible and two reversible reactions (r_6 and r_8). (B) The eight elementary modes are shown in purple. The most efficient pathways for producing D_e from A_e are EM6 and EM7. EM4 and EM8 are suboptimal, because they also yield E_e. (Adapted from Trinh CT, Wlaschin A & Srienc F. *Appl Microbiol Biotechnol.* 81 [2009] 813–826. With permission from Springer Science and Business Media.)

E, from where material leaves the system through r_4 toward D_e and r_9 toward E_e. Analysis of this type shows that the network contains eight elementary modes, which are shown in Figure 9.13B. If the goal is to convert A_e into D_e, some modes are not useful and should be directly eliminated from further consideration. In this case, EM1 and EM3 are excluded, because they produce B_e instead of D_e. Moreover, EM2 and EM5 are not pursued further, because they use B_e as substrate. The most efficient pathways are EM6 and EM7. EM4 and EM8 are also admissible but suboptimal, because they yield E_e in addition to D_e and thereby convert valuable substrate into an undesired by-product.

Case study. Many laboratories around the world have become involved in new strategies for improving the yield of biofuels and, in particular, the creation of microbial strains that convert non-edible plant biomass into ethanol, butanol, fatty acids, or hydrogen. Friedrich Srienc and his laboratory set out to develop a high-efficiency *E. coli* strain that would convert hexoses and pentoses into ethanol [88]. Instead of over-expressing promising genes or knocking out genes that were assumed unimportant, as had been done in the past, they used a computationally based, systematic, top-down approach that started with a wild-type strain and eventually led to a highly efficient production strain. The guiding concept was that cells specialized for a purpose such as ethanol production could otherwise afford minimized functionality when living under controlled, favorable conditions. The goal was to create a strain that would, under optimized conditions, grow, replicate, and produce economically feasible amounts of ethanol but not necessarily do much else. While obviously intriguing, this goal was certainly not trivial to accomplish. For instance, how could one predict which reaction steps are truly necessary for survival? The investigators approached and solved the challenge with a combined computational and molecular strategy.

The first task was to identify redundancies within the metabolic pathway structure. These redundancies are beneficial for survival in uncertain environments but presumably are not needed under ideal laboratory conditions. In other words, if the bacterium had two ways of producing the same metabolite, one pathway could possibly be eliminated, thereby reducing the organism's physiological machinery

and decreasing its metabolic burden. The tool for this identification of absolutely necessary pathways was elementary mode analysis (EMA) [86].

The investigators constructed and experimentally validated a comprehensive model of the intermediary metabolism of *E. coli*, which included reactions involved in the dynamics of pentoses and hexoses, including glucose, galactose, mannose, xylose, and arabinose. Wild-type *E. coli* does not possess the enzyme pyruvate decarboxylase, which converts pyruvate to acetaldehyde, from which ethanol is produced, and the enzyme was therefore provided through a plasmid. This process generated an ethanol production system similar to that in the bacterium *Zymomonas mobilis* and the yeast *Saccharomyces cerevisiae*.

The next step was the calculation of all elementary modes (EMs) with a computer algorithm that analyzed all reactions and their directionalities within this network and identified all unique pathways [86]. While the details are too complicated for this case description, the results are in principle the same as those shown in Figure 9.13. In the actual case of ethanol production in *E. coli*, the investigators identified over 15,000 EMs with which the bacterium can metabolize xylose and arabinose, the most prevalent pentoses found in biomass.

The next step was the selection of desirable EMs and the elimination of inefficient EMs. Among the 15,000 EMs identified, about 1,000 consumed xylose and arabinose under the desired anaerobic conditions. Amazingly, the deletion of just seven reactions reduced this number to only 12 ethanol- and biomass-generating EMs. The seven reactions for disruption were identified by application of basic EMA rules. Namely, the consequences of eliminating a reaction were studied with respect to the remaining EMs. Then, the maximal yields in biomass and ethanol were assessed for the remaining pathways. Finally, the reaction with the least number of EMs for ethanol and biomass production was eliminated. The 12 EMs were further reduced to six, because the other six EMs made use of the pyruvate dehydrogenase complex, which is not active under anaerobic conditions. Only two of the remaining six EMs co-produced a sufficient amount of ethanol and biomass.

The investigators used the same rationale and methods for ethanol production based on hexose consumption. Furthermore, it was possible to study the simultaneous consumption of both pentoses and hexoses. In this case, almost 100,000 EMs were identified, of which only 18 remained for anaerobic conditions. Of these, 12 utilized either glucose or xylose individually, and six consumed both. The set of EMs using hexoses and pentoses could be further reduced by blocking glucose transport into the cell or glucose phosphorylation within the cell.

Intriguingly, all previous inferences were based on purely theoretical considerations. However, they were subsequently validated by the actual genetic engineering of a deletion strain as suggested by the computational analysis, and the agreement with the prediction was excellent, with a yield of about 0.4–0.5 gram of ethanol per gram of sugars.

9.5.5 Optimization of Dynamic Systems in Metabolic Engineering

FBA and EMA focus strictly on metabolic networks operating at a steady state. This restriction has important mathematical advantages, because it permits optimization with comparatively straightforward methods of linear programming within the field of operations research. From a biological point of view, however, this strict focus on steady-state operation raises some questions. First, the steady-state approach does not really allow an account of regulatory responses to introduced alterations: FBA prescribes changes in a flux distribution, caused by alterations the metabolic engineer introduces, while pretending that everything else in the system remains the same. Of course, one must expect that organisms respond to such perturbations through regulatory processes, which FBA ignores. Second, FBA requires a specific objective function. For microbes, this function is often assumed to be the maximization of biomass for growth. However, for experiments with mammalian cells, for instance, such an assumption would be problematic. Third, since FBA only considers systems of fluxes, no statements can be made regarding the metabolite concentrations.

In response to these issues, different approaches have been proposed. However, they are mathematically challenging and therefore beyond the scope of this introduction. They are briefly discussed here, and the interested reader is encouraged to follow some of the key references provided later.

Various direct extensions of FBA have attempted to permit analyses of metabolic systems that change over time. Mahadevan and colleagues [89] described a type of dynamic flux balance analysis (DFBA) that incorporates flux constraints that change over time. It is also possible to include kinetic information if it is available [90, 91]. DFBA is not only applicable to microbial optimization tasks in metabolic engineering but was even used to investigate whole-organism metabolism of plants [92] and humans with diabetes [93]. DFBA is technically challenging because it requires nonlinear optimization, whereas FBA uses linear optimization, which is fairly straightforward.

As an alternative to DFBA, the group of Torres [27, 94, 95] developed strategies of optimizing targeted metabolite concentrations or fluxes in fully defined dynamical systems of metabolic pathways represented with power-law functions. A different but less-used alternative is cybernetic modeling, where a microorganism is considered a strategist and its regulatory processes are assumed to be optimal. The organism attempts to optimize the use of available resources. The assumptions lead to the task of optimally allocating resources within dynamic model of the network [96].

9.6 INFERENCE OF METABOLIC SYSTEMS

9.6.1 Metabolic Network Reconstruction

So far, we assumed that we knew the structure of the metabolic network. What needs to be done if we don't know it? The answer is that the scope of the problem depends on the availability and type of data. Traditionally, biochemists have been using wet-lab experiments, trying to identify reactions, regulators, and kinetic properties associated with a particular metabolite. These studies are very labor-intensive and slow but have collectively produced an enormous body of knowledge, which was originally depicted in the Boehringer Map that adorned many hallways of biochemistry departments. This map is now available electronically [97], along with other databases such as KEGG [40] and MetaCyc [98].

In more recent times, high-throughput methods of molecular biology have become available for metabolic network characterization. Many of these focus on the actual existence of specific metabolites in the target network and use methods such as mass spectrometry and nuclear magnetic resonance (see, for example, [32]). Since genome data are usually easier to obtain than information on the connectivity of metabolic pathway systems, methods have also been developed to infer pathways from the presence of genes, homologies with other organisms, and the genomic organization into operons and other regulatory structures (see, for example, Chapter 7). For instance, consider the situation in which there is a gap in an incomplete metabolic network of species X, where some metabolite apparently cannot be produced with any of the enzymes known in X (**Figure 9.14**). Suppose that a related species, Y, possesses a gene G_Y that codes for an enzyme E_Y catalyzing the needed reaction and that could potentially bridge the gap in X. If so, one uses homology methods of **bioinformatics** in an attempt to determine whether X's genome contains a gene G_X that has sufficient similarity with G_Y. If such a gene can be found, one infers that G_X might code for an enzyme E_X that is similar to E_Y and fills the gap [99]. Free software associated with BioCyc and MetaCyc, such as BioCyc's *PathoLogic* and KEGG's KAAS [100], serve this specific purpose [101–103]. An intriguing case study using this approach is [49].

A different situation occurs when the connectivity and regulation of a metabolic map is known, but specific kinetic parameters are not. Many methods for estimating such parameters have been developed in recent years (see, for example, [104]). None of these methods are foolproof, however, and all require quite a bit of computing experience.

Case Study. It is obvious that some organisms are better understood than others. Much research has focused on model organisms such as the bacterium

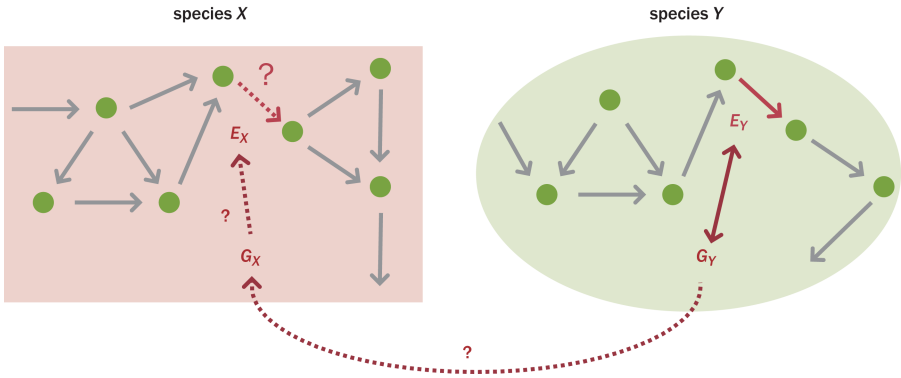


Figure 9.14 Filling gaps in a metabolic map through genome comparison. Suppose that metabolic and proteomic analysis has so far not revealed an enzyme E_X that would be able to connect two separate parts of the metabolic network of species *X*. Suppose that an enzyme E_Y catalyzing the missing reaction is known in the related species *Y*, and that its gene G_Y has been identified. Sequence comparisons may reveal a homologous gene G_X in species *X* and identify it as a candidate for coding for the unknown enzyme E_X.

Escherichia coli, the fruit fly *Drosophila melanogaster*, and the mouse *Mus musculus*, with the expectation that insights from these representatives are applicable to lesser-known organisms. In many cases, such inferences are valid, but nature also has a way of varying details so that, for instance, high percentages of genes still have unknown functions, even in moderately well-characterized organisms.

A beautiful example of characterizing metabolism with a variety of modern methods has been published in *Science* [49]. In this study, an international group of researchers established a complete metabolic map of the small bacterium *Mycoplasma pneumoniae*, which is the culprit in some types of human pneumonia. They selected this particular organism mainly because of its comparatively small size and simple organization: *M. pneumoniae* contains fewer than 700 protein-coding genes, of which the authors, however, found only 231 to be annotated. Earlier studies had characterized a number of metabolic aspects in this bacterium with biochemical and computational methods [105, 106], and the results of these studies were integrated here with new analyses.

The authors of [49] began by assembling a catalog of known reactions, using KEGG as a starting point and assessing the activities of the various reactions with information from the literature and some new annotations. Complementing this information, they collected genome information, such as the co-occurrence of genes in the same operon and the homology of sequences with known enzymes in related organisms and evaluated the likely functionality of inferred enzymes using structural information regarding pertinent catalytic residues. This combined metabolomic and genomic **data mining** led to apparently disconnected pathways but also to suggestions for completing the metabolic map. As an example, the authors used methods discussed earlier to fill a critical gap in the seemingly disconnected ascorbate pathway. Through inferences from sequence homology, they succeeded in predicting activity and the position within the responsible operon that the organism is likely to possess the critical enzyme L-ascorbate-6-phosphate lactonase, without which the ascorbate pathway would not be functional. In other cases, the structural analysis permitted the elimination of putative enzymes that lacked the catalytic residues necessary for substrate turnover. Finally, putative enzyme-catalyzed reactions were eliminated from the metabolic map if their substrates were not produced by any of the other pathways. The result of this information mining and manual curation was a metabolic map without gaps, isolated reactions, or open metabolic loops. The map was then further refined with targeted laboratory experiments in order to validate the presence of alternative pathways and to determine the primary direction in reversible pathways. These experiments included growth on different substrates and ^{13}C-labeled glucose.

Further analysis of the refined metabolic map, supported by additional wet experiments, led to interesting insights. One finding confirmed the intuitive assumption that genes are up- or down-regulated in functional, pathway-associated clusters and in a dynamic fashion. Moreover, there appeared to be good concordance among gene expression, protein abundance, and metabolic turnover. For instance, the authors observed concomitant increases in mRNA, protein expression, and turnover rate associated with glycolysis, following medium acidification. Closer inspection of the results indicated, not surprisingly, a definite time delay between gene regulation and changes in metabolite concentrations.

Experiments with ^{13}C-labeled glucose showed that 95% of all glucose in *M. pneumoniae* ultimately becomes lactate and acetate, which implies that the lion's share of glycolysis is used for energy generation. Another interesting finding was that this simple organism contains many more linear pathways and fewer diverging and converging branch points than more complicated organisms and is therefore less redundant. This decreased redundancy has direct implications for the robustness of the organism because some metabolites can only be produced in a single manner. As a consequence, 60% of the metabolic enzymes in *M. pneumoniae* were found to be essential, which is about four times as many as in *E. coli*. Finally, the authors concluded that *M. pneumoniae* is not optimized for fast growth and biomass production, which is often assumed in flux balance analyses as the main objective of a microbe but possibly for survival strategies that suppress the growth of competitor populations and interactions with the host organism.

9.6.2 Extraction of Dynamic Models from Experimental Metabolic Data

The vast majority of dynamic models of metabolic systems has been constructed from the bottom up. In the first step, each reaction or process within the system is assigned a mathematical representation, such as a mass action term, MMRL, or a power-law function. Second, data from the literature are used to estimate the parameters of each rate function. Third, all representations are merged into a model of the entire system. Fourth, with high hopes, one tests the integrated model against validation data. Unfortunately, this test fails quite often, and several rounds of refinements and searches for better parameter values ensue. Even for moderately sized systems, the process often takes months, if not years. However, the result is a fully dynamic model that has a much higher potential than a static model. Examples are manifold and include [67, 68, 107].

A distinct alternative is model development from the top down, using time-series data. The group of Santos and Neves used in vivo ^{13}C- and ^{31}P-NMR spectroscopy to measure metabolite concentrations in living populations of the bacterium *Lactococcus lactis* [108, 109]. Because NMR is non-destructive, the group was able to measure dense time series of the decreasing concentration of labeled substrate, as well as the time-dependent concentrations of metabolic intermediates and end products, all within the same cell culture. By fine-tuning their methods, they were able to measure the key metabolites of glycolysis in intervals of 30 seconds or less, thereby creating hour-long time series of metabolite profiles. The resulting data were successfully converted into computational models (see, for example, [110, 111]).

Kinoshita and co-workers [112] measured time-course profiles of over 100 metabolites in human red blood cells, using capillary electrophoresis with subsequent mass spectrometry (CE-MS). Using these data, the authors developed a mathematical pathway model, consisting of about 50 enzyme kinetic rate equations, whose formats were extracted from the literature. The model allowed manipulations mimicking the condition of hypoxia, where the organism is deprived of oxygen.

9.7 EXERCISES

9.1 Screen the literature or Internet for large metabolites that are not proteins, peptides, RNA, or DNA. Compare their sizes with those of proteins.

9.2 Describe (9.8) and (9.9) in words.

9.3 Confirm with mathematical means that no material is lost or gained in the Michaelis–Menten process.

9.4 Formulate a differential equation for E in the Michaelis–Menten process. Compare this equation with (9.8) and interpret the result.

9.5 Derive (9.10) from the ODE system (9.7)–(9.9) or find the derivation in the literature or Internet. For the derivation, make use of the quasi-steady-state assumption and of the fact that the sum $E + (ES)$ is constant.

9.6 Simulate (9.7)–(9.10) and compare $\dot{P}$ with v_p in (9.10) for different initial values and parameter values. Why are no initial values needed in (9.10)?

9.7 Redo the analysis of Exercise 9.6 but assume that S is 1,000 times larger than E, which might mimic in vitro conditions. Evaluate the validity of QSSA in comparison to Figure 9.3.

9.8 Explain the details of the dynamics of substrate in the Hill model (9.15). Formulate equations for (ES) and P.

9.9 For a reaction with competitive inhibition, plot $v_{P,CI}$ against S, and $1/v_{P,CI}$ against $1/S$, for several inhibitor concentrations and compare the plots with the corresponding plot from the uninhibited reaction.

9.10 For a reaction with uncompetitive inhibition, plot $v_{P,UI}$ against S, and $1/v_{P,UI}$ against $1/S$, for several inhibitor concentrations and compare the plots with the corresponding plot from the uninhibited reaction. If you have done Exercise (9.9), compare the two types of inhibition.

9.11 For a reaction with noncompetitive inhibition, plot $v_{P,NI}$ against S, and $1/v_{P,NI}$ against $1/S$, for several inhibitor concentrations and compare the plots with the corresponding plot from the uninhibited reaction. If you have done Exercise (9.9) or (9.10), compare your results.

9.12 Transform the Hill rate law (9.14) into a linear format by considering the logarithm of $v_{P,\,Hill}/(V_{\max} - v_{P,\,Hill})$.

9.13 Extract from KEGG and BioCyc all pathways using glucose 6-phosphate as a substrate. Are these pathways the same in *E. coli*, baker's yeast, and humans?

9.14 Mine databases and the literature to reconstruct a metabolic pathway system describing the production of lysine. Search for kinetic parameter values for as many steps as possible. *Note:* this is an open-ended problem with many possible solutions.

9.15 Given the known influxes (blue) and effluxes (green) in **Figure 9.15**, compute the internal fluxes (red) of the system. Is the solution unique? Discuss issues of uniqueness generically and provide examples and counterexamples. Is the system in Figure 9.11 necessarily in a steady state? Is the flux information sufficient to determine the concentrations of the metabolites $A, \ldots, H$?

9.16 Design models for the pathway in Figure 9.5. Represent the production of A as $1 \times D^{-0.5}$. For all simulations, start at the steady state (which you must compute first) and double the efflux from D at $t = 10$. In the first set of simulations, use the MMRL for the reactions that use A, B, C, or D as substrate. Choose different combinations of $V_{\max}$ and K_M values as you like. Repeat the simulations with power-law rate laws. Summarize your findings in a report.

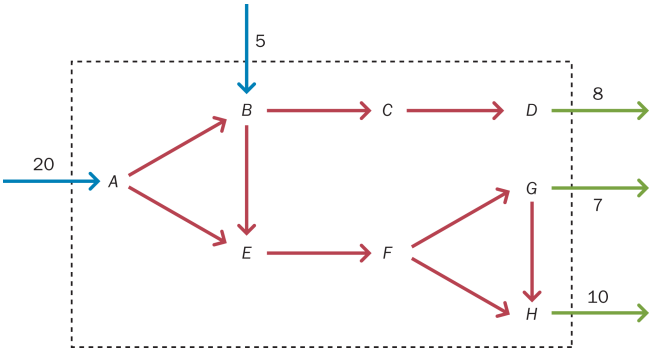


Figure 9.15 Generic stoichiometric map. The influxes (blue) and effluxes (green) are given in units of mmol L^{-1} min^{-1}.

9.17 Design a power-law model for the pathway in Figure 9.6. Ignore intermediates that are suggested by dotted lines and set all rate constants equal to 1. For all substrate degradation processes, use kinetic orders of 0.5. For inhibition effects, start with kinetic orders of –0.5. Subsequently, change the inhibition effects one by one—or in combination—to other negative values. Always start your simulations at the steady state. Summarize your findings in a report.

9.18 How could one model the effect of altered gene expression on a metabolic pathway? Discuss different options.

9.19 Compare the ascorbate pathway in *M. pneumoniae* with that in *E. coli*. Report your findings.

9.20 Find five pathways in *E. coli* that do not exist in *M. pneumoniae*.

9.21 Explore how much citric acid is being produced annually and which microorganism is primarily used for its large-scale industrial production. Search for models that have addressed citric acid production and review their key features.

9.22 Compare kinetic parameters of the fermentation pathway in yeast that were presented in the work of Galazzo and Bailey [113] and Curto et al. [68] with those that are listed in BRENDA and ExPASy. Summarize your findings in a spreadsheet or a report.

9.23 Consider the model in **Figure 9.16**, which we have already studied in Chapter 3 (Exercise 3.32). It describes the pentose pathway, which exchanges carbohydrates with different numbers of carbon atoms, as shown in **Table 9.2**. For instance, two units of X_1 are needed to produce one unit of X_2 and X_3, and each reaction $V_{5,3}$ uses one unit of X_5 to generate two units of X_3. Note that X_3 appears in two locations but represents the same pool. Assume that all processes follow mass action kinetics and that the influx ($V_{7,1}$) and the effluxes ($V_{3,0}$ and CO_2) are 20 mmol L^{-1} min^{-1} and that all other fluxes have a value of 10 mmol L^{-1} min^{-1}. Confirm that these settings correspond to a consistent steady-state flux distribution. Assume the following steady-state concentrations. $X_1 = 10$, $X_2 = 6$, $X_3 = 12$, $X_4 = 8$, $X_5 = 8$, $X_6 = 4$, $X_7 = 20$. Compute the rate constants of all processes.

9.24 Using the model in Exercise 9.23 with the same flux distribution, is it possible that the system can have different steady-state concentrations? If not, explain. If the answer is yes, construct one or more examples. Write a brief report on your findings.

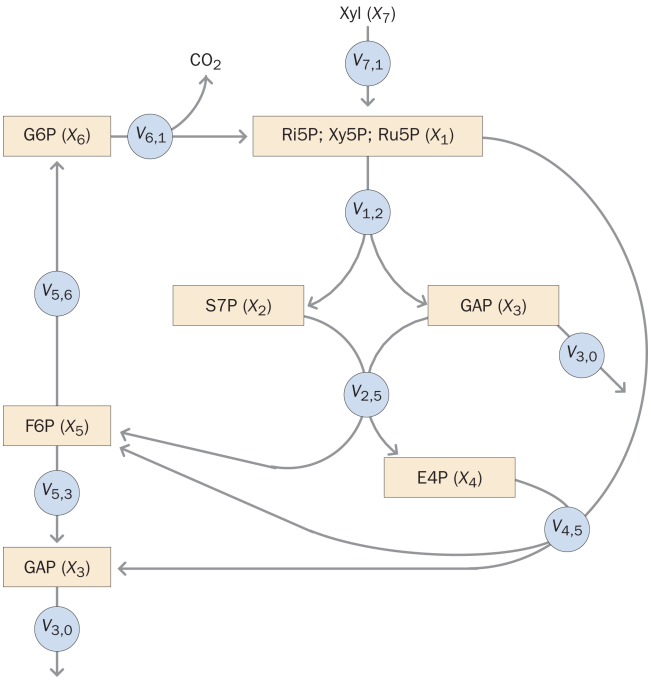


Figure 9.16 Pentose phosphate pathway. Information on influxes and effluxes permits the computation of internal fluxes. (From Wiechert W & de Graaf AA. *Biotechnol Bioeng.* 55 [1997] 101–117. With permission from John Wiley & Sons.)

TABLE 9.2: NUMBER OF CARBON ATOMS IN EACH METABOLITE IN THE PATHWAY IN FIGURE 9.16

Compound Pool	*Number of Carbon Atoms*
X_1	5
X_2	7
X_3	3
X_4	4
X_5	6
X_6	6
X_7	5

9.25 Using the results from Exercise 9.23, perform a simulation that starts at the steady state and models a bolus perturbation to X_1. Make a prediction of the model responses before you simulate and summarize your findings in a brief report.

9.26 Using the results from Exercises 9.23–9.25, perform two simulations that start at the steady state and model either a persistent or a bolus perturbation to X_7. Make a prediction of the model responses before you simulate and write a brief report on your findings.

9.27 Formulate the following pathway model as a stoichiometric equation. The abbreviations along the reactions refer to a linear process (L), power-law process (PL), or Hill rate law (HRL).

9.28 Draw a diagram that is consistent with the following GMA-system equations. Formulate the model as a stoichiometric equation.

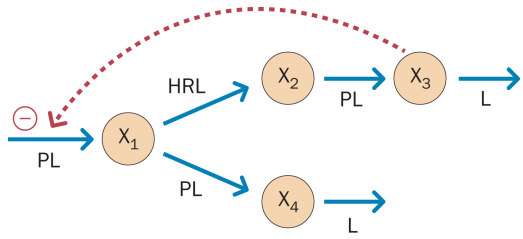


Figure 9.17 Branched pathway with feedback. PL = power-law, HRL = Hill rate law, L = linear function.

$$\dot{X}_1 = 5X_4^{-4} - 0.2X_1 - 3X_1^2X_4^2$$

$$\dot{X}_2 = 0.2X_1 - 2X_2^{0.4}$$

$$\dot{X}_3 = 2X_2^{0.4} - 2X_3^{0.7}$$

$$\dot{X}_4 = 2X_3^{0.7} - 2X_4^{0.8}$$

$$\dot{X}_5 = 3X_1^2X_4^2 - 0.5X_5$$

$$\dot{X}_6 = 0.5X_5 - 0.5X_6$$

9.29 Compute a flux distribution for the system on Figure 9.10 that is different from the one in panel D of the figure but generates an efflux out of N_3 that is 100. The solution must satisfy the conditions that the influx into N_1 and N_2 cannot be greater than 20 units and that every internal flux must at least be 10 units. Is the solution unique? Is it possible to achieve a higher efflux out of N_3 under the same conditions?

9.30 Using Excel or other software, optimize the flux distribution for the network in Figure 9.10 for the case that R_{40} is three times as valuable as R_{30}. Interpret the result.

9.31 Determine the elementary modes in the simple reaction system in **Figure 9.18A**, either by hand or with software like [83–85, 87]. Assume that the nodes inside the shaded area are internal and that the dominant direction for r_1 is toward A. Define the stoichiometric matrix. What is/are the optimal elementary mode(s) if the goal is optimal production of B_e?

9.32 Formulate the stoichiometric matrix of the simple reaction network in **Figure 9.18B**. Determine the elementary modes of the network.

9.33 Formulate the stoichiometric matrix of the simple reaction network in **Figure 9.18C**. Determine the elementary modes of the network.

9.34 Identify the abbreviations in the simplified yeast fermentation pathway in **Figure 9.19**. Assume that reactions outside the shaded area are external. Formulate the stoichiometric matrix. Determine the elementary modes of the network, either by hand or with the software [83–85, 87].

9.35 The text states that elementary modes are pathways that form an essentially unique, minimal set of irreversible reactions taking place at steady state. The inclusion of the word "essentially"

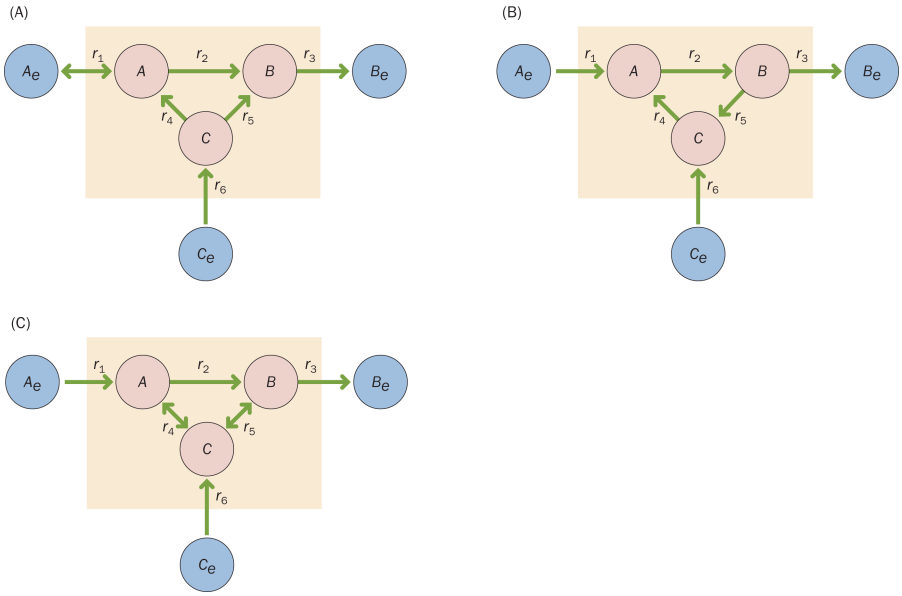


Figure 9.18 Generic metabolic reaction system. In (A) the pathway contains two inputs and two outputs. (B) and (C) represent slight variations. (Adapted from Palsson BØ. Systems Biology: Properties of Reconstructed Networks. Cambridge University Press, 2006. With permission from Cambridge University Press.)

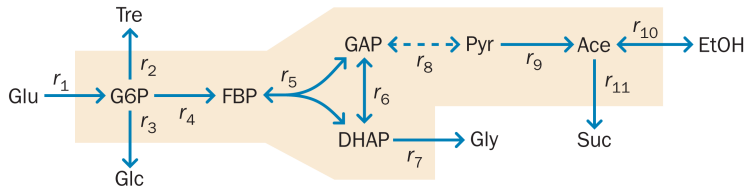


Figure 9.19 Simplified diagram of the yeast fermentation system. The shaded area is considered inside the system, while EtOH, Glu, Glc, Gly, Suc, and Tre are external. The dashed arrow indicates a sequence of several reactions that are not explicitly shown. (Adapted from Schwartz J-M & Kanehisa M. *BMC Bioinform* 7 [2006] 186. With permission from Bio Med Central Ltd.)

means that the sets are unique except for simple scaling (i.e., multiplication by the same constant). Argue why this type of scalability is always present in stoichiometric, flux balance, and elementary mode analyses. Is the same true for dynamic simulations?

FURTHER READING

Cornish-Bowden A. *Fundamentals of Enzyme Kinetics*, 3rd ed. Portland Press, London, UK, 2004.

Edelstein-Keshet L. *Mathematical Models in Biology*. McGraw Hill, Irwin, CA, 1988.

Fell DA. *Understanding the Control of Metabolism*. Portland Press, London, UK, 1997.

Harrigan GG & Goodacre R (eds). *Metabolic Profiling*. Kluwer, Boston, MA, 2003.

Koffas M & Stephanopoulos G. Strain improvement by metabolic engineering: Lysine production as a case study for systems biology. *Curr Opin Biotechnol.* 16 (2005) 361–366.

Savageau MA. *Biochemical Systems Analysis: A Study of Function and Design in Molecular Biology*. Addison-Wesley, Reading, MA, 1976.

Schultz AR. *Enzyme Kinetics: From Diastase to Multi-Enzyme Systems*. Cambridge University Press, Cambridge, UK, 1994.

Torres NV & Voit EO. *Pathway Analysis and Optimization in Metabolic Engineering*. Cambridge University Press, Cambridge, UK, 2002.

Voit EO. *Computational Analysis of Biochemical Systems: A Practical Guide for Biochemists and Molecular Biologists*. Cambridge University Press, Cambridge, UK, 2000.

Voit EO. Biochemical systems theory: A review. *Int Scholarly Res Network (ISRN—Biomathematics)*. (2013) 1–53. Article 897658.

REFERENCES

1. Derechin, M., et al., Yeast hexokinase. II. Molecular weight and dissociation behavior. *Biochemistry*, 1966. **5**(12): p. 4017–4025.
2. HPG, *Human Genome Project Information*, http://genomics.energy.gov/, 2010.
3. Johnson, J.M., et al., A practical approach to detect unique metabolic patterns for personalized medicine. *Analyst*, 2010. **135**(11): p. 2864–2870.
4. Guldberg, C.M. and P. Waage, Studier i affiniteten. *Forhandlinger i Videnskabs-Selskabet i Christiania*, 1864. **35**.
5. Guldberg, C.M. and P. Waage, *Études sur les affinites chimiques*. 1867, Christiania: Brøgger & Christie.
6. Voit, E.O., H.A. Martens, and S.W. Omholt, 150 years of the mass action law. *PLoS Comput Biol*, 2015. **11**(1): p. e1004012.
7. Gillespie, D.T., Stochastic simulation of chemical kinetics. *Annu Rev Phys Chem*, 2007. **58**: p. 35–55.

8. Wolkenhauer, O., et al., Modelling and simulation of intracellular dynamics: Choosing an appropriate framework. *IEEE Trans Nanobiosci*, 2004. **3**: p. 200–207.
9. Henri, M.V., *Lois générales de l'action des diastases*. 1903, Paris: Hermann.
10. Michaelis, L. and M.L. Menten, Die Kinetik der Invertinwirkung. *Biochemische Zeitschrift*, 1913. **49**: p. 333–369.
11. Benkovic, S.J., G.G. Hammes, and S. Hammes-Schiffer, Free-energy landscape of enzyme catalysis. *Biochemistry*, 2008. **47**: p. 3317–3321.
12. Briggs, G.E. and J.B.S. Haldane, A note on the kinetics of enzyme action. *Biochem J*, 1925. **19**: p. 338–339.
13. Cornish-Bowden, A., *Fundamentals of Enzyme Kinetics*. 3rd ed. 2004, London: Portland Press Ltd.
14. Schultz, A.R., *Enzyme Kinetics: From Diastase to Multi-Enzyme Systems*. 1994, Cambridge, New York: Cambridge University Press, p. x, 246.
15. Voit, E.O., The best models of metabolism. *WIREs Syst Biol Med*, 2017. 9(6): p. e1391.
16. Hill, A.V., The possible effects of the aggregation of the molecules of hæmoglobin on its dissociation curves. *J Physiol*, 1910. **40**: p. iv–vii.
17. Savageau, M.A., *Biochemical Systems Analysis: A Study of Function and Design in Molecular Biology*. 1976, Reading, MA: Addison-Wesley Pub. Co. Advanced Book Program. p. xvii, 379.
18. Taketa, K. and B.M. Pogell, Allosteric inhibition of rat liver fructose 1,6-diphosphatase by adenosine 5′-monophosphate. *J Biol Chem*, 1965. **240**: p. 651–662.
19. Martin, R.G., The phosphorolysis of nucleosides by rabbit bone marrow: The nature of feedback inhibition by histidine. *J Biol Chem*, 1963. **238**: p. 257–268.
20. Monod, J., J. Wyman, and J.P. Changeux, On the nature of allosteric transitions: A plausible model. *J Mol Biol*, 1965. **12**: p. 88–118.
21. Monod, J., J.P. Changeux, and F. Jacob, Allosteric proteins and cellular control systems. *J Mol Biol*, 1963. **6**: p. 306–329.
22. Koshland, D.E., Jr., G. Nemethy, and D. Filmer, Comparison of experimental binding data and theoretical models in proteins containing subunits. *Biochemistry*, 1966. **5**(1): p. 365–385.
23. Berg, P.H., E.O. Voit, and R.L. White, A pharmacodynamic model for the action of the antibiotic imipenem on *Pseudomonas aeruginosa* populations *in vitro*. *Bull Math Biol*, 1996. **58**(5): p. 923–938.
24. BRENDA, http://www.brenda-enzymes.org/.
25. ENZYME, *Enzyme Nomenclature Database*, http://enzyme.expasy.org/, accessed 2023.
26. Savageau, M.A., Enzyme kinetics in vitro and in vivo: Michaelis-Menten revisited, in *Principles of Medical Biology*, E.E. Bittar, Editor. 1995, Greenwich, CT: JAI Press Inc., p. 93–146.
27. Torres, N.V. and E.O. Voit, *Pathway Analysis and Optimization in Metabolic Engineering*. 2002, Cambridge, UK: Cambridge University Press.
28. Eilertsen, J., et al., On the quasi-steady-state approximation in an open Michaelis-Menten reaction mechanism. *AIMS Math*, 2021. **6**(7): p. 6781–6814.
29. Fell, D.A., *Understanding the Control of Metabolism*. 1997, London, UK: Portland Press.
30. Lecca, P., Machine learning for causal inference in biological networks: Perspectives of this challenge. *Front Bioinform,* 2021. **1**: p. 746712.
31. Yazdani, A., et al., Systems biology informed deep learning for inferring parameters and hidden dynamics. *PLoS Comput Biol*, 2020. **16**(11): p. e1007575.
32. Harrigan, G.G. and R. Goodacre, eds. *Metabolic Profiling*. 2003, Boston: Kluwer Academic Publishers.
33. Oldiges, M. and R. Takors, Applying metabolic profiling techniques for stimulus-response experiments: chances and pitfalls. *Adv Biochem Eng Biotechnol*, 2005. **92**: p. 173–196.
34. Villas-Bôas, S.G., et al., Mass spectrometry in metabolome analysis. *Mass Spectrom Rev*, 2005. **24**: p. 613–646.
35. Theobald, U., et al., *In vivo* analysis of metabolic dynamics in *Saccharomyces cerevisiae*: I. Experimental observations. *Biotechnol Bioeng*, 1997. **55**: p. 305–316.
36. Spura, J., et al., A method for enzyme quenching in microbial metabolome analysis successfully applied to gram-positive and gram-negative bacteria and yeast. *Anal Biochem*, 2009. **394**(2): p. 192–201.
37. Garcia-Perez, I., et al., Metabolic fingerprinting with capillary electrophoresis. *J Chromatogr A*, 2008. **1204**(2): p. 130–139.
38. Scriba, G.K., Nonaqueous capillary electrophoresis-mass spectrometry. *J Chromatogr A*, 2007. **1159**(1–2): p. 28–41.
39. Wiechert, W., 13-C metabolic flux analysis. *Metab Eng*, 2001. **3**: p. 195–206.
40. KEGG, *The Kyoto Encyclopedia of Genes and Genomes*, http://www.genome.jp/kegg/pathway.html.
41. BioCyc, http://biocyc.org/.
42. ERGO, *Integrated Genomics Inc.*, http://ergo.integratedgenomics.com/ERGO/.
43. Joshi-Tope, G., et al., Reactome: A knowledgebase of biological pathways. *Nucleic Acids Res*, 2005. **33**(Database issue): p. D428–D432.
44. Reactome, http://www.reactome.org, 2010.
45. Fabregat, A., et al., Reactome pathway analysis: A high-performance in-memory approach. *BMC Bioinform*, 2017. **18**(1): p. 142.
46. COSBI, *The Microsoft Research–University of Trento Centre for Computational and Systems Biology*, www.cosbi.eu/index.php/research/prototypes/kinfer, accessed 2023.
47. Lipid-Map, http://www.lipidmaps.org/, 2010.
48. Alves, R. and M.A. Savageau, Effect of overall feedback inhibition in unbranched biosynthetic pathways. *Biophys. J*, 2000. **79**(5): p. 2290–2304.
49. Yus, E., et al., Impact of genome reduction on bacterial metabolism and its regulation. *Science*, 2009. **326**(5957): p. 1263–1268.
50. Neves, M.-J. and J. François, On the mechanism by which a heat shock induces trehalose accumulation in *Saccharomyces cerevisiae*. *Biochem J*, 1992. **288**: p. 859–864.
51. Fonseca, L.L., et al., Complex coordination of multi-scale cellular responses to environmental stress. *Mol BioSyst*, 2011. **7**: p. 731–741, https://doi.org/10.1039/C0MB00102C.
52. Heinrich, R. and T.A. Rapoport, A linear steady-state treatment of enzymatic chains. Critique of the crossover theorem and a general procedure to identify interaction sites with an effector. *Eur J Biochem*, 1974. **42**(1): p. 97–105.
53. Kacser, H. and J.A. Burns, The control of flux. *Symp Soc Exp Biol*, 1973. **27**: p. 65–104.
54. Moreno-Sánchez, R., et al., Metabolic control analysis: A tool for designing strategies to manipulate metabolic pathways. *J Biomed Biotechnol*, 2008. **2008**: p. 597913.
55. Burns, J.A., et al., Control analysis of metabolic systems. *Trends Biochem Sci*, 1985. **10**(1): p. 16.
56. Savageau, M.A., Dominance according to metabolic control analysis: Major achievement or house of cards? *J Theor Biol*, 1992. **154**: p. 131–136.
57. Reder, C., Metabolic control theory: A structural approach. *J Theor Biol*, 1988. **135**: p. 175–201.
58. Visser, D. and J.J. Heijnen, The mathematics of metabolic control analysis revisited. *Metab Eng*, 2002. **4**(2): p. 114–123.
59. Savageau, M.A., E.O. Voit, and D.H. Irvine, Biochemical systems theory and metabolic control theory. I. Fundamental similarities and differences. *Mathem Biosci*, 1987. **86**: p. 127–145.

60. Savageau, M.A., E.O. Voit, and D.H. Irvine, Biochemical systems theory and metabolic control theory. II. The role of summation and connectivity relationships. *Mathem Biosci,* 1987. **86**: p. 147–169.
61. Hatzimanikatis, V. and J.E. Bailey, MCA has more to say. *J Theor Biol*, 1996. **182**(3): p. 233–242.
62. Heijnen, J.J., New experimental and theoretical tools for metabolic engineering of micro-organisms. *Meded Rijksuniv Gent Fak Landbouwkd Toegep Biol Wet*, 2001. **66**(3a): p. 11–30.
63. Moreno-Sánchez, R., et al., Metabolic control analysis: A tool for designing strategies to manipulate metabolic pathways. *J Biomed Biotechnol*, 2008: p. 597913, https://doi.org/10.1155/2008/597913.
64. Gavalas, G.R., *Nonlinear Differential Equations of Chemically Reacting Systems*. 1968, Berlin: Springer-Verlag.
65. Heinrich, R. and S. Schuster, *The Regulation of Cellular Systems*. 1996, New York: Chapman and Hall.
66. Voit, E.O. and M.A. Savageau, Accuracy of alternative representations for integrated biochemical systems. *Biochemistry*, 1987. **26**(21): p. 6869–6880.
67. Alvarez-Vasquez, F., et al., Integration of kinetic information on yeast sphingolipid metabolism in dynamical pathway models. *J Theor Biol*, 2004. **226**(3): p. 265–291.
68. Curto, R., A. Sorribas, and M. Cascante, Comparative characterization of the fermentation pathway of *Saccharomyces cerevisiae* using biochemical systems theory and metabolic control analysis: Model definition and nomenclature. *Math Biosc*, 1995. **130**: p. 25–50.
69. Goel, G., I.-C. Chou, and E.O. Voit, System estimation from metabolic time series data. *Bioinformatics*, 2008. 24: p. 2505–2511.
70. Faraji, M. and E.O. Voit, *Stepwise Inference of Likely Dynamic Flux Distributions from Metabolic Time Series Data* (in review), 2017.
71. Stephanopoulos, G.N., A.A. Aristidou, and J. Nielsen, *Metabolic Engineering. Principles and Methodologies*. 1998, San Diego, CA: Academic Press.
72. Koffas, M. and G. Stephanopoulos, Strain improvement by metabolic engineering: lysine production as a case study for systems biology. *Curr Opin Biotechnol*, 2005. **16**(3): p. 361–366.
73. Currie, J.N., On the citric acid production of *Aspergillus niger*. *Science*, 1916. **44**: p. 215–216.
74. Palsson, B.Ø., *Systems Biology: Properties of Reconstructed Networks*. 2006, New York: Cambridge University Press.
75. Varma, A. and B.Ø. Palsson, Metabolic flux balancing: Basic concepts, scientific and practical use. *Bio/Technology*, 1994. **12**: p. 994–998.
76. Kjeldsen, K.R. and J. Nielsen, In silico genome-scale reconstruction and validation of the *Corynebacterium glutamicum* metabolic network. *Biotechnol Bioeng*, 2009. **102**(2): p. 583–597.
77. Wittmann, C., H.M. Kim, and E. Heinzle, Metabolic network analysis of lysine producing *Corynebacterium glutamicum* at a miniaturized scale. *Biotechnol Bioeng*, 2004. **87**(1): p. 1–6.
78. Dräger, A., et al., Modeling metabolic networks in *C. glutamicum*: A comparison of rate laws in combination with various parameter optimization strategies. *BMC Syst Biol*, 2009. **3**: p. 5.
79. Yang, C., Q. Hua, and K. Shimizu, Development of a kinetic model for L-lysine biosynthesis in *Corynebacterium glutamicum* and its application to metabolic control analysis. *J Biosci Bioeng*, 1999. **88**(4): p. 393–403.
80. Kayser, A., et al., Metabolic flux analysis of *Escherichia coli* in glucose-limited continuous culture. I. Growth-rate-dependent metabolic efficiency at steady state. *Microbiology* 2005. **151**: p. 693–706.
81. Schuster, S. and S. Hilgetag, On elementary flux modes in biochemical reaction systems at steady state. *J Biol Syst*, 1994. **2**: p. 165–182.
82. Liao, J.C., S.Y. Hou, and Y.P. Chao, Pathway analysis, engineering, and physiological considerations for redirecting central metabolism. *Biotechnol Bioeng*, 1996. **52**: p. 129–140.
83. Poolman, M.G., ScrumPy: Metabolic modelling with Python. *Syst Biol (Stevenage)*, 2006. **153**(5): p. 375–378.
84. Poolman, M.G., et al., A method for the determination of flux in elementary modes, and its application to *Lactobacillus rhamnosus*. *Biotechnol Bioeng,* 2004. **88**(5): p. 601–612.
85. Schuster, S., T. Dandekar, and D.A. Fell, Detection of elementary flux modes in biochemical networks: a promising tool for pathway analysis and metabolic engineering. *Trends Biotechnol*, 1999. **17**(2): p. 53–60.
86. Trinh, C.T., A. Wlaschin, and F. Srienc, Elementary mode analysis: A useful metabolic pathway analysis tool for characterizing cellular metabolism. *Appl Microbiol Biotechnol*, 2009. **81**: p. 813–826.
87. von Kamp, A. and S. Schuster, Metatool 5.0: Fast and flexible elementary modes analysis. *Bioinformatics*, 2006. **22**(15): p. 1930–1931.
88. Trinh, C.T., P. Unrean, and F. Srienc, Minimal *Escherichia coli* cell for the most efficient production of ethanol from hexoses and pentoses. *Appl Env Microbiol*, 2008. **74**(12): p. 3634–3643.
89. Mahadevan, R., J.S. Edwards, and F.J. Doyle, 3rd, Dynamic flux balance analysis of diauxic growth in *Escherichia coli*. *Biophys J*, 2002. **83**(3): p. 1331–1340.
90. Antoniewicz, M.R., Dynamic metabolic flux analysis—tools for probing transient states of metabolic networks. *Curr Opin Biotechnol*, 2013. **24**(6): p. 973–978.
91. Yugi, K., et al., Hybrid dynamic/static method for large-scale simulation of metabolism. *Theor Biol Med Model*, 2005. **2**: p. 42.
92. Grafahrend-Belau, E., et al., Multiscale metabolic modeling: dynamic flux balance analysis on a whole-plant scale. *Plant Physiol*, 2013. **163**(2): p. 637–647.
93. Ben Guebila, M., Thiele, I., Dynamic flux balance analysis of whole-body metabolism for type 1 diabetes. *Nat Comput Sci*, 2021. **1**: p. 348–361.
94. Garcia, J. and N. Torres, Mathematical modelling and assessment of the pH homeostasis mechanisms in *Aspergillus niger* while in citric acid producing conditions. *J Theor Biol*, 2011. **282**(1): p. 23–35.
95. Alvarez-Vasquez, F., C. Gonzalez-Alcon, and N.V. Torres, Metabolism of citric acid production by *Aspergillus niger*: Model definition, steady-state analysis and constrained optimization of citric acid production rate. *Biotechnol Bioeng*, 2000. **70**(1): p. 82–108.
96. Ramakrishna, R., D. Ramkrishna, and A.E. Konopka, Cybernetic modeling of growth in mixed, substitutable substrate environments: Preferential and simultaneous utilization. *Biotechnol Bioeng*, 1996. **52**(1): p. 141–151.
97. ExPASy, http://ca.expasy.org/enzyme/.
98. MetaCyc, http://biocyc.org/metacyc/index.shtml, accessed 2023.
99. Reed, J.L., et al., An expanded genome-scale model of *Escherichia coli* K-12 (iJR904 GSM/GPR). *Genome Biol*, 2003. **4**(9): p. R54.
100. KAAS, http://www.genome.jp/tools/kaas/.
101. Karp, P.D., et al., Pathway tools version 13.0: Integrated software for pathway/genome informatics and systems biology. *Brief Bioinform*, 2010. **11**(1): p. 40–79.
102. Paley, S.M. and P.D. Karp, Evaluation of computational metabolic-pathway predictions for *Helicobacter pylori*. *Bioinformatics*, 2002. **18**(5): p. 715–724.
103. Pathologic, http://biocyc.org/intro.shtml#pathologic.
104. Chou, I.C. and E.O. Voit, Recent developments in parameter estimation and structure identification of biochemical and genomic systems. *Math Biosci*, 2009. **219**(2): p. 57–83.

105. Pachkov, M., et al., Use of pathway analysis and genome context methods for functional genomics of *Mycoplasma pneumoniae* nucleotide metabolism. *Gene*, 2007. **396**(2): p. 215–225.
106. Pollack, J.D., Mycoplasma genes: A case for reflective annotation. *Trends Microbiol*, 1997. **5**(10): p. 413–419.
107. Hynne, F., S. Danø, and P.G. Sørensen, Full-scale model of glycolysis in *Saccharomyces cerevisiae*. *Biophys Chem*, 2001. **94**: p. 121–163.
108. Neves, A.R., et al., Overview on sugar metabolism and its control in *Lactococcus lactis*—the input from in vivo NMR. *FEMS Microbiol Rev*, 2005. **29**(3): p. 531–554.
109. Neves, A.R., et al., Is the glycolytic flux in Lactococcus lactis primarily controlled by the redox charge? Kinetics of NAD(+) and NADH pools determined in vivo by 13C NMR. *J Biol Chem*, 2002. **277**(31): p. 28088–28098.
110. Voit, E.O., A.R. Neves, and H. Santos, The intricate side of systems biology. *Proc Natl Acad Sci U S A*, 2006. **103**(25): p. 9452–9457.
111. Voit, E.O., et al., Regulation of glycolysis in *Lactococcus lactis*: An unfinished systems biological case study. *IEE Proc Systems Biol*, 2006. **153**: p. 286–298.
112. Kinoshita, A., et al., Roles of hemoglobin Allostery in hypoxia-induced metabolic alterations in erythrocytes: Simulation and its verification by metabolome analysis. *J Biol Chem*, 2007. **282**(14): p. 10731–10741.
113. Galazzo, J.L. and J.E. Bailey, Fermentation pathway kinetics and metabolic flux control in suspended and immobilized *S. cerevisiae*. *Enzyme Microbiol Technol*, 1990. **12**: p. 162–172.

Signaling Systems

10

When you have read this chapter, you should be able to:

- Understand the fundamentals of signal transduction systems
- Discuss the advantages and disadvantages of discrete and continuous models of signal transduction
- Describe the role of G-proteins and receptor tyrosine kinases
- Understand bistability and hysteresis
- Discuss quorum sensing
- Analyze differential equation models of two-component signal transduction systems
- Analyze differential equation models of the MAPK signal transduction system

In cell biology, **signal transduction** describes a class of mechanisms through which a cell receives internal or external **signals**, processes them, and triggers appropriate responses. In many cases, the signals are chemical, but cells can also respond to different kinds of physical signals, such as light, electrical stimuli, and mechanical stresses. In fact, different signals are often processed at the same time [1]. If the cell receives an external chemical signal, the transduction process usually begins with the binding of specific ligands to receptors on the outer surface of the cell membrane. These receptors are transmembrane proteins, and binding of a ligand to the extracellular domain of the receptor triggers a change in the conformation or function of the receptor that propagates to the intracellular domain of the receptor. This change mediates other changes within the cell that ultimately result in an appropriate response to the original signal. As a specific example, integrins are transmembrane proteins that transduce information from the extracellular matrix or the surrounding tissue to the inside of the cell and are involved in cell survival, apoptosis, proliferation, and differentiation [2].

Good examples of signal transduction circuits can be found in an important family of signaling molecules called **G-proteins**. These are located on the inside of the cell membrane and are coupled to integrin receptors that reach from the inside to the outer surface of the cell and whose conformation changes when a ligand binds to them. Specifically, upon binding to the extracellular domain of such a **G-protein-coupled receptor (GPCR)** (**Figure 10.1**), the change in the intracellular domain of the GPCR causes a subunit of the G-protein to be released and to bind to a molecule of guanosine triphosphate (GTP), which leads to the dissociation of two molecular subunits. The importance of this process led to the prefix "G," which stands for "guanine-nucleotide-binding." The activated G-protein subunits detach from the receptor and initiate internal signaling processes that are specific

DOI: 10.1201/9781003402770-10

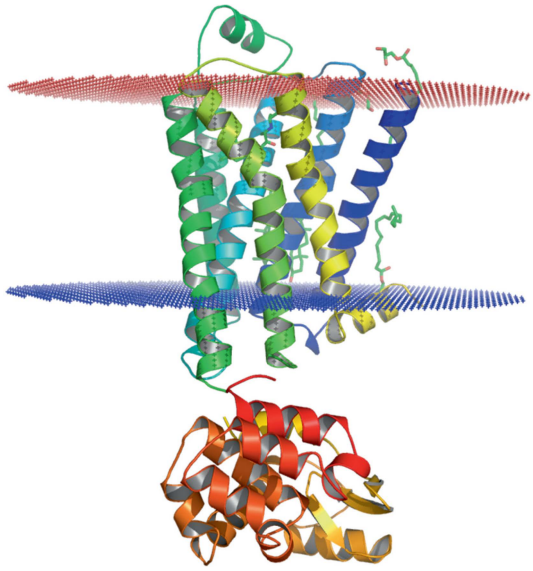


Figure 10.1 Example of a membrane-spanning G-protein-coupled receptor (GPCR_3ny8). The human β_2-adrenergic receptor (β_2-AR) consists of a seven-helical-bundle membrane protein. The two planes represent the lipid bilayer. GPCRs represent a large fraction of current pharmaceutical targets, and β_2-AR is one of the most extensively studied. (Courtesy of Huiling Chen and Ying Xu, University of Georgia.)

to the particular type of G-protein. For instance, the process may trigger the production of cyclic adenosine monophosphate (cAMP), which can activate protein kinase A, which in turn can phosphorylate and thereby activate many possible downstream targets. Other G-proteins stimulate the generation of signaling compounds such as diacylglycerol and inositol trisphosphate, which control the release of intracellular calcium from storage compartments into the cytoplasm. After some recovery period, hydrolysis removes the third phosphate group from GTP and readies the G-protein for transduction of a new signal. The functionality of very many drugs is associated with GPCRs. For their "discovery of G-proteins and the role of these proteins in signal transduction in cells," Alfred Gilman and Martin Rodbell received the 1994 Nobel Prize in Physiology or Medicine.

Another large class of transmembrane proteins that induce intracellular signaling cascades is the **receptor tyrosine kinase (RTK)** family, which encompasses many cytokine and growth factor receptors and was already introduced in Chapter 8. RTKs differ from GPCRs in that they consist of two molecular components that form a dimer protein. Binding of ligand on the extracellular domain initiates conformational changes in the intracellular domain that stabilizes the dimer form such that each half can enzymatically modify the other partner through the addition of multiple phosphate groups to tyrosine residues in the cytosolic region, using ATP as phosphate donor. This process is called autophosphorylation and activates a number of downstream signaling events. The phosphorylated RTK allows many adaptor proteins to attach to the receptor, which leads to a macromolecular complex where signaling proteins can bind and phosphorylate one another. The initiation of the MAPK signaling cascades, discussed later in this chapter, are a frequent occurrence of RTK activation. The linking of extracellular molecules to dynamic intracellular signaling via RTK phosphorylation has historically been one of the most studied application areas for modeling in systems biology, due to the importance of this class of proteins to cancer therapeutics and developmental biology. The two-component signaling systems (later in this chapter) uses a similar mechanism.

Regardless of how the cell first senses its environment and transmits this information via membrane receptors, the signal transduction processes inside the cell are mediated by chains of biochemical reactions that are catalyzed by enzymes and often modulated by **second messengers**, which are water-soluble or -insoluble molecules or gases such as nitric oxide. Typical water-soluble, hydrophilic messengers are calcium and cAMP. They are usually located in the cytosol and may be activated or deactivated by specific enzymes. Hydrophobic signaling molecules are often associated with the plasma membrane. Examples include different phosphatidylinositols, inositol trisphosphate, and the sphingolipid ceramide. The steps between receiving a signal at the membrane and triggering a response in the cytosol or nucleus are often organized as a cascade of events that amplify the

incoming signal and filter out noise. The speed of signal transduction depends strongly on the mechanism. Calcium-triggered electrical signals occur at the order of milliseconds, while protein- and lipid-based cascades respond at the order of minutes. Signals involving genomic responses happen within tens of minutes, hours, or even days. Chapter 12 discusses the complex, calcium-based events occurring during every beat of the heart.

Because signaling processes are ubiquitous in biology, an enormous literature has been amassed over the past decades and even a cursory review of all signal transduction systems is far beyond the scope of this chapter. In contrast to the wealth of experimental information, modeling studies are much fewer, although their number has also been rising quickly in recent years. Typical models fall into two distinct classes with only scant overlap. The first class considers signaling networks as **graphs** and tries, on the one hand, to understand their functionality and, on the other hand, to deduce their causal connectivity from experimental observations with sophisticated statistical methods. We discussed these systems in Chapter 3. In this chapter, we discuss representative examples of the second class of signal transduction systems, which attempts to capture the signaling dynamics with ordinary—or sometimes stochastic—differential equation models.

10.1 DIFFERENTIAL EQUATION MODELS OF SIGNAL TRANSDUCTION SYSTEMS

10.1.1 Bistability

The construction of systems of differential equations that are geared toward capturing the dynamics of signal transduction systems can take many forms. While the sky is the limit in this approach, the key component is usually a module that permits **bistability**. As the name suggests, such a module possesses two stable states, which are in most cases separated by an unstable state. If a signal is low, the system assumes one stable steady state, and if the signal exceeds some threshold, which is related to the unstable state, the system moves to the other stable steady state. The simplest example of such a system is a switch that can be triggered by a signal. If the signal is present, the system is on and when the signal disappears, the system turns off. Bistable systems of different types are found in many areas of biology, and examples include classical genetic toggle switches [3, 4], cell cycle control, cellular signal transduction pathways, switches to new programs during development, and the switch between lysis and lysogeny in bacteriophages [5].

Bistable systems are easy to construct. For instance, in Chapter 5 we used a polynomial of the format

$$\dot{X} = c \times (SS_1 - X) \cdot (SS_2 - X) \cdot (SS_3 - X) \tag{10.1}$$

Here, the parameter c just determines the speed of a response, while SS_1, SS_2, and SS_3 are three steady states. They are easy to confirm, because setting X to any of the steady-state values renders $\dot{X}$ to become zero. For instance, for $(SS_1, SS_2, SS_3) = (10, 40, 100)$, and assuming that c is positive, $SS_1 = 10$ and $SS_3 = 100$ are stable steady states, while $SS_2 = 40$ is unstable.

A different approach to designing a system with a bistable switch is the combination of a sigmoidal trigger with a linear or power-law relaxation function that lets the response variable return to its off-state. A minimalistic model consists of the single ODE, such as

$$\dot{X} = 10 + \frac{20\,X^4}{100^4 + X^4} - 2\,X^{0.5} \tag{10.2}$$

It barely qualifies as a "system," as it only has one variable but, notwithstanding semantics, we may interpret the equation as the description of a quantity X that is produced with a constant input of magnitude 10 and replicates or activates its own production by means of a Hill term. Furthermore, X is consumed or degraded according to a power-law function.

As always, the steady state(s) of the system can be assessed by setting Eqn. (10.2) equal to zero. We could try to solve the equation algebraically, but that is a bit messy. However, we can easily convince ourselves that a steady state is achieved if the two positive terms exactly balance the negative term. Thus, we can plot the two functions and study where they intersect. The graph (**Figure 10.2**) indicates that there are actually three intersection points, which directly correspond to the steady states. One occurs for $X \approx 25$, the second for $X \approx 100$, and the third for $X \approx 210$. Implementing the system in software that solves ODEs and starting the solution with different initial values quickly shows that the true steady states for $X \approx 25.41736$ and $X \approx 210.7623$ are stable. The third steady state is actually at exactly $X = 100$, and it is unstable: starting slightly higher, even at 100.1, the system diverges toward the higher stable steady state, whereas it approaches the lower state for smaller initial values, even if they are as close as 99.9. An interpretation could be that the system models the body temperature and is either in a healthy (low steady) state or a diseased (high) state. Both are stable, and small perturbations are easily tolerated in each state, whereas large perturbations may shift the system from health to disease or vice versa. Biologically, the unstable state is never realized, at least not for long, because it does not tolerate even the smallest disturbances.

As an illustration, suppose the system resides at the low steady state ($X \approx 25$), but external signals reset the value of X at a series of time points, as indicated in **Table 10.1**. **Figure 10.3** illustrates the responses of the system. Upon each resetting event, the system approaches the stable steady state in whose **basin of attraction** it is. The unstable state does not have such a basin, because it is repelling: solutions starting even slightly higher than $X = 100$ wander toward the high steady state, and solutions starting lower than the unstable state approach the low steady state.

With a little bit more effort, ODEs with sigmoidal components can be made more interesting. For instance, consider a small two-variable system, which could be interpreted as consisting of two genes that affect each other's expression. A high signal S triggers the expression of Y, Y affects X, and X affects Y. The system is shown in **Figure 10.4**.

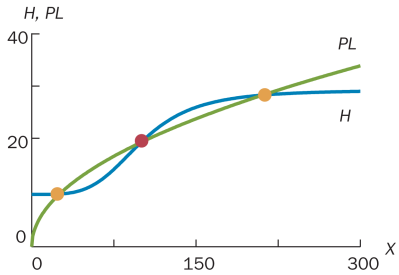


Figure 10.2 Bistability is often implemented as a differential equation containing the difference between an s-shaped function and a linear or power-law term. Shown here are the graph of a typical signal–response relationship in the form of a shifted sigmoidal Hill function $H(S) = 20 + 100S^4/(3^4 + S^4)$ as in (10.2) (blue) and a linear degradation function (green). Their intersections indicate steady-state points. The most pertinent cases contain two stable states (yellow) and one unstable state (red). Note that the green function is shown for general illustration only and does not correspond to any of the terms in (10.2).

TABLE 10.1 SCHEDULE OF RESETTING *X* IN (10.2) AT SEVERAL TIME POINTS *T* (SEE FIGURE 10.3)

t	*X*
0	25
20	10
40	90
100	60
180	120
300	280
400	150
480	50
520	80
550	10

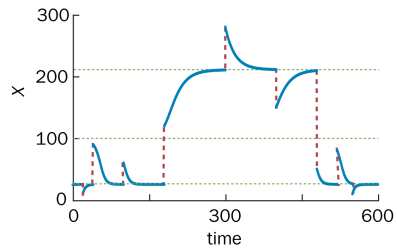


Figure 10.3 Response of *X* in (10.2) to external signals that reset the value of *X* at the time points identified with dashed red lines (see Table 10.1). The three dotted lines indicate the three steady states. The central steady state at $X = 100$ is unstable and separates the two basins of attraction of the stable steady states at about 25 and 211. If reset above 100, the system approaches the higher steady state, whereas resetting below 100 causes the system to approach the low steady state.

10.1.2 Hysteresis

As we have seen in Chapters 2 and 5, it is not too difficult to translate a diagram like the one in Figure 10.4 into differential equations. Of course, there are many choices for an implementation, and we select one that contains a constant input and a sigmoidal Hill function representing the effect of Y on the expression of X. Specifically, let us consider the following system:

$$\dot{X} = 200 + \frac{800\,Y^4}{16^4 + Y^4} - 50\,X^{0.5} \tag{10.3}$$

$$\dot{Y} = 50\,X^{0.5}S - 100\,Y \tag{10.4}$$

The signal S is an independent variable that affects the system from the outside. Suppose the signal changes during our simulation experiment as shown in **Table 10.2**. Starting close to the steady state $(X, Y) \approx (20.114, 6.727)$ with $S = 3$, and following the signal resetting schedule in the table, we obtain the response in **Figure 10.5**.

Of greatest interest in comparison to the bistable system is that something truly new and intriguing is happening here. The system starts with $S = 3$, which corresponds to the low steady-state value $X \approx 20$ and is to be interpreted as *off*. When the signal is increased to 10, X shoots up to about 400 and is *on*. Fifteen time units later, the signal goes back to $S = 3$, and we would expect the response to shut off again. However, X does not return to its *off* state of about 20 but instead drops only modestly to a value of about 337, which is still to be interpreted as *on*. At the end of the experiment, the signal is back to $S = 3$ again, and X is *off* again, but this happens only after the signal had been reduced temporarily to an even smaller value ($S = 2.5$). Thus, for a signal strength of $S = 3$, the system can be either *on* or *off*! Further analysis of this curious behavior implies that the system has built-in memory: the state of the response variable depends not only on the current value of the external signal but also on the recent history of the signal and the system. Mathematically, this history is stored in the dependent variables. Indeed, if we look at signal S and

response X, it appears that there is memory—or **hysteresis**—as it is called in the jargon of the field. In conveniently structured models, conditions can be derived for the existence or absence of bistability and hysteretic effects (e.g., [6–8]). An example of a relatively simple real-world system is a so-called two-component signaling system in bacteria, which we will discuss in the next section of this chapter.

Hysteresis can be explored in greater detail with a two-phase simulation study (see Figure 10.6). In the first phase, we start with a steady state resulting from a low value of S. We increase S slowly, in a stepwise fashion, let the system come sufficiently close to the steady state, and then increase S again. For this illustration, let's concentrate on the variable X; Y behaves similarly. For values of S between 0 and 2.5, the steady-state value of X, X_{ss}, is close to 16 and changes very little, and further raising S to about 3.3, X_{ss} increases gradually to about 30. Raising S further, to about 3.4, X_{ss} all of the sudden jumps to about 365. Checking signal strength in the vicinity of 3.4 demonstrates that this is a real jump and not a very fast but gradual increase in X_{ss}. Further raises in S, above 3.4, increase X_{ss} a little more, but not much.

Now we enter the second phase, where we start with a high value of S and slowly decrease it in a stepwise manner, every time recording X_{ss}. At first the steady-state values exactly trace those obtained from the first phase, and we would probably expect a jump back for $S \approx 3.4$. Surprisingly, this jump does not happen, and X_{ss} decreases only slowly. As a second surprise, X_{ss} crashes from about 230 to about 17 once S reaches a value close to 2.5. Again, this is not a fast, gradual decrease but a real jump. Further decreases in S cause X_{ss} to trace the early results of the first phase of the simulation.

The important conclusion for this system is the following: for a signal with a strength in the range between about 2.6 and 3.3, the value of X_{ss} depends on the history of the system. The situation is depicted in Figure 10.6B, which enlarges the range of different steady states of X depending on whether S is increased from low values or decreased from high values. Note that there are no steady-state values on the dashed arrows. The response of Y is similar, although of lower magnitude, and we will return to it later in the chapter, when we discuss clocks.

A simple bistable system models a "hard" switch: a signal just a tiny bit above the threshold will always trigger an *on* response, while a signal just below results in an *off* response. These response patterns are entirely deterministic. Suppose now that the system operates close to the threshold value for X_{ss}, that is, the unstable steady-state point. This is a reasonable assumption, because otherwise the capability of bistability of the system would be wasted. Let's suppose further that the environment is mildly stochastic. If so, every change in S across the threshold, up or down, will make the system jump. Hysteresis softens this situation and in essence buffers the system response to noise in the signal input. In the range between 2.6 and 3.3, the illustration system tends to remain on or off, depending where it was just before, and noise in the signal is therefore easily tolerated. Expressed differently, it takes a true, substantial change in signal to make the system jump. Section 10.7 and Box 10.1 discuss further analyses and variants of this basic hysteretic system.

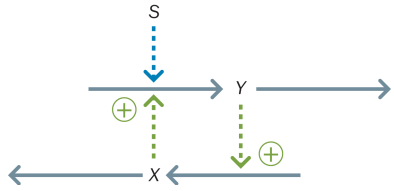


Figure 10.4 Artificial system with two components and an external signal. X and Y might be interpreted as the expression states of two genes. The system is affected by signal S, which could be interpreted as a stressor or a transcription factor.

TABLE 10.2 SUSTAINED SIGNAL STRENGTHS S BEGINNING AT TIME POINTS T

t	S
0	3
5	10
20	3
40	2.5
50	3

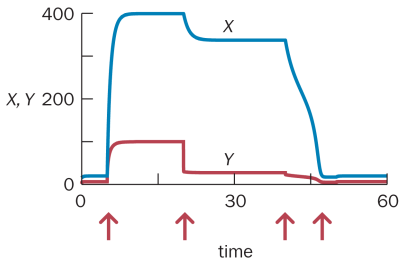


Figure 10.5 The two-variable signaling system in (10.3–10.4) shows hysteresis. The response variable assumes different values for the same input signal, depending on the history of the system. See the text for values of the signal, which changes at the times indicated by the arrows.

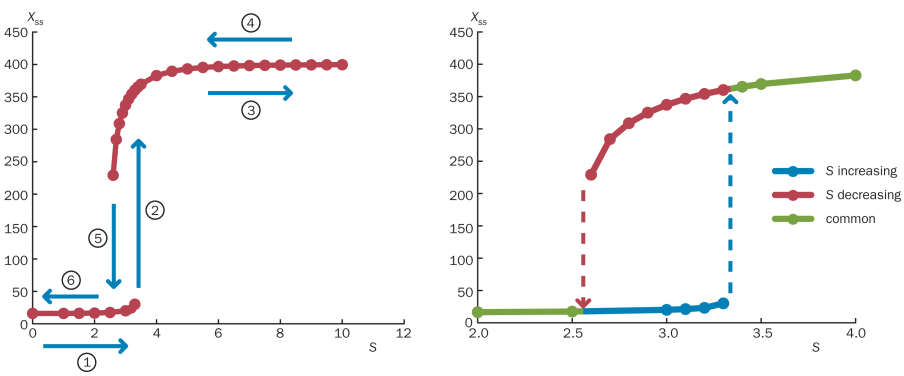


Figure 10.6 Demonstration of hysteresis. (A) indicates a series of alterations in S and the corresponding responses in X_{ss}. Increasing S from 0 to about 3 affects X_{ss} only slightly. However, changing S from about 3.3 to 3.4 results in an enormous jump in X_{ss}. Further increases in S do not change the new steady state much. Decreasing S from 10 to 3.4 traces the same steady states. However, there is no jump at 3.4. Surprisingly, the jump happens between about 2.6 and 2.5. (B) shows a magnification of the system response plot for signal strengths around 3, where X_{ss} depends on the recent history of the system. The dashed arrows indicate jumps and do not reflect steady states.

Why would nature evolve hysteretic behavior in cellular signal transduction? One answer is that many responses to environmental cues are energetically expensive; for example, it takes a lot of ATP to divide a cell into two cells. Furthermore, a spurious extracellular signal could initiate a process like mitosis or major increases in transcription or translation that would be wasteful of resources. Creating a high threshold for turning on a pathway is therefore desirable. Likewise, once a commitment is made for pathway activation, building a buffer against turning off the signal—or even making the commitment irreversible (see Box 10.1)—prevents rapid flip-flopping of cell state that could conflict with processes downstream of transcriptional activation. Hysteresis has been observed in T cells, where sustaining the activation of a signaling pathway after the stimulus is withdrawn helps the immune cells exhibit a form of short-term memory of past encounters with antigen [9]. In another example, the entry into cell cycle has been characterized as containing one-way hysteresis with properties similar to the scheme in Figure 2 of Box 10.1 [10]. In the presence of growth signals from RTK activation, DNA replication is initiated by a series of signaling events as they were briefly mentioned in Chapter 8 (Figure 8.12). Researchers discovered that removal of the growth stimulus before a critical point in the cell cycle shifts cells back to a quiescent, resting state. Once cells progress past this point in the cell cycle, mitosis will continue independently of growth stimulation, even if the starting signal is completely removed. In this case, a high threshold for pushing forward in the cell cycle ensures that all resources are in place for full execution of cell division.

In recent years, many authors have studied what happens if a signal and/or a system are corrupted by noise. Such investigations require methods that allow stochastic input, such as stochastic modeling, differential equations with stochastic components, or stochastic **Petri nets** [5, 11, 12].

BOX 10.1: ONE-WAY HYSTERESIS

An important step in the mammalian immune response is the differentiation of B cells into antibody-secreting plasma cells. This step is triggered by an antigen that the host had not encountered before. The signaling pathway controlling this differentiation process is governed by a gene regulatory network, where several genes and transcription factors play critical roles. Focusing on the three key transcription factors Bcl-6, Blimp-1 and Pax5, Bhattacharya and collaborators [13] developed a mathematical model capturing the dynamics of this differentiation system. They were specifically interested in understanding how dioxin (TCDD; 2,3,7,8-tetrachlorodibenzo-p-dioxin) can interrupt the healthy functioning of this signaling pathway. Dioxin is a toxic environmental contaminant that is produced in incineration processes. It is also infamous as a toxic trace component of Agent Orange, which was used in the Vietnam War to defoliate crops and forests. A simplified diagram describing the functional interactions among the genes and transcription factors in the model is shown in **Figure 1A**. The model the authors proposed is actually more complicated, as they explicitly considered gene expression, transcription and translation. Of interest is that the multiple repression signals form a dynamic switch that, upon stimulation with an antigen, ultimately directs a B cell to differentiate into a plasma cell. The antigen considered here is a lipopolysaccharide (LPS), a compound found on the outer membrane of many bacteria and a known trigger of strong immune responses in animals.

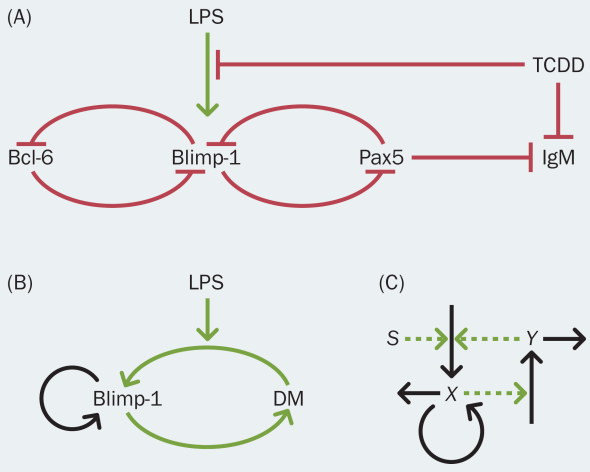


Figure 1 Simplified diagrams of the B-cell differentiation pathway. (A) A slightly simplified version of the model proposed by Bhattacharya et al. Lipopolysaccharide (LPS) stimulates this system. Blimp-1 is a master regulator that represses the transcription factors Bcl-6 and Pax5, which are typical for the B-cell phenotype. Pax5 inhibits the expression of immunoglobulin M (IgM), which mediates the immune response and can lead to the differentiation of the B cell into an antigen-secreting plasma cell. The environmental pollutant dioxin (TCDD) interferes with the signaling pathway. (B) A drastic reduction of the pathway, where TCDD is left out and Pax5 and IgM are combined into a differentiation marker, DM. (C) The same diagram in our typical notation, with X representing Blimp-1, Y the differentiation marker, and S the lipopolysaccharide. (From Bhattacharya S, Conolly RB, Kaminski NE et al. [2010] *Toxicol. Sci.* **115**: 51–65. With permission from Oxford University Press.)

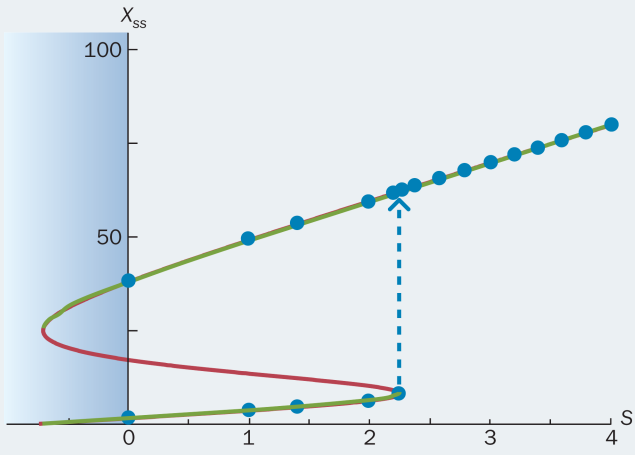


Figure 2 The B-cell differentiation pathway exhibits "one-way hysteresis." For small signals, the system has a steady state on the lower branch of the green hysteresis curve, which corresponds to the phenotype of a B cell. When the signal exceeds 2.25 in magnitude, the system jumps (dashed arrow) to a steady state on the upper branch, which corresponds to the phenotype of an antibody-secreting plasma cell. However, once the system has entered a steady state on the upper branch, it can never return to a steady state on the lower branch, even if the signal disappears entirely. Thus, the differentiation step is irreversible. The red component depicts unstable steady states. The dots indicate some simulation results.

The authors analyzed in detail how this pathway responds to deterministic and stochastic noise in the LPS signal [13, 14].

To study the switching behavior of the deterministic system, we make further drastic simplifications. In particular, we omit TCDD and its effect on the pathway. Furthermore, we leave out Bcl-6 and replace the associated dual inhibition with an auto-activation process for Blimp-1, because two serial inhibitions correspond to activation. Finally, we create a differentiation marker DM by combining Pax5 with IgM, with the understanding that IgM is high when Pax5 is low and vice versa. The simplified signaling pathway is shown in Figure 1B. Defining X as Blimp-1, Y as the DM, and S as LPS, it is easy to transcribe the system into our typical notation (Figure 1C). Note that this diagram is actually quite similar to the small system in Figure 10.4, which we studied earlier.

We formulate the system with the model in Eqn. (1), where the Hill function H is a function of X.

$$\dot{X} = 3.2H^2 + Y^{0.5}S - 2X \qquad (1)$$

$$\dot{Y} = 4H - Y^{0.5},$$

where

$$H = 1 + \frac{4X^4}{16^4 + X^4}.$$

To examine the responses of the system to different magnitudes of the stimulus, we perform exactly the same analysis as for the system in Eqns. (10.3–10.4). Namely, we repeatedly increase S in small steps from a very low value to a high value, every time computing the stable steady state of the system and focusing in particular on the steady-state values X_{SS}. The results are shown in **Figure 2**.

For small signal strengths, X_{SS} is only mildly affected. For $S = 0$, it has a value of about 1.6, which slowly increases to about 8 with higher values of S. Because we have some experience with these types of systems, we are not too surprised that there is a critical point, here for about $S \approx 2.25$, where the response jumps to a high level of about 62 (dashed arrow). For higher signal strengths the response increases further, but at a relatively slow pace. Now we go backwards, decreasing S at every step a little bit. Again, we are not surprised that the system does not jump back down to the lower branch at $S \approx 2.25$, because we expect it to take this jump for a smaller value. However, it never does. Even reducing the signal to 0 does not move the system from the upper branch to the lower branch. Mathematically, such a jump would happen for about $S \approx -0.8$, but that is of course biologically not possible. In other words, the system has similar "off" steady states for low signal strengths, jumps to "on" steady states at about 2.25, but now is stuck in these "on" steady states, even if the signal disappears entirely. Interpreting this observation for a B cell, we are witnessing its irreversible differentiation into an antibody-secreting plasma cell. The unstable steady states (red) cannot be simulated. However, they can be computed if we express S as a function of X_{SS} (see Exercises).

10.2 TWO-COMPONENT SIGNALING SYSTEMS

10.2.1 Background

Two-component signaling (TCS) systems occur widely in archaea and bacteria, as well as in some fungi and plants. They are absent in animals, which rather transduce signals with three-component systems, protein kinase cascades, and/or lipid-based systems. TCS systems allow organisms to sense changes in environmental factors such as temperature and osmolarity and to discern the direction of chemical gradients. Once the signal is received, it is processed and ultimately allows the organism to respond in an appropriate fashion. In many cases, the TCS system is integrated with other regulatory mechanisms, such as allosteric enzyme regulation in the nitrogen assimilation of enteric bacteria [15]. For a good collection of articles on TCS systems, see [16].

A relatively well-understood variant of a bacterial TCS system is the signal transduction protein CheY in *Escherichia coli*, which reacts to nutrients and other

chemical attractants in the environment and affects the direction of the rotation of the organism's flagellar motor [17] (**Figure 10.7**). Binding of an attractant lowers the activity of the receptor-coupled kinase CheA, which decreases the level of phosphorylated CheY. This CheY-P in turn binds to a component of the motor (see Chapter 9). The motor usually rotates counterclockwise, but changes direction when CheY is phosphorylated. As a result, the bacterium tumbles and changes direction. Thus, mediated by CheY, the cell controls its swimming behavior.

The first component of a TCS system acts as the sensor or transmitter and consists of an unphosphorylated membrane-bound histidine kinase (**Figure 10.8**). If a suitable signal is present, a phosphate group is transferred from ATP to a histidine residue on the sensor, resulting in ADP and a phosphohistidine residue. Because all this happens on the same molecule, the kinase is sometimes called an

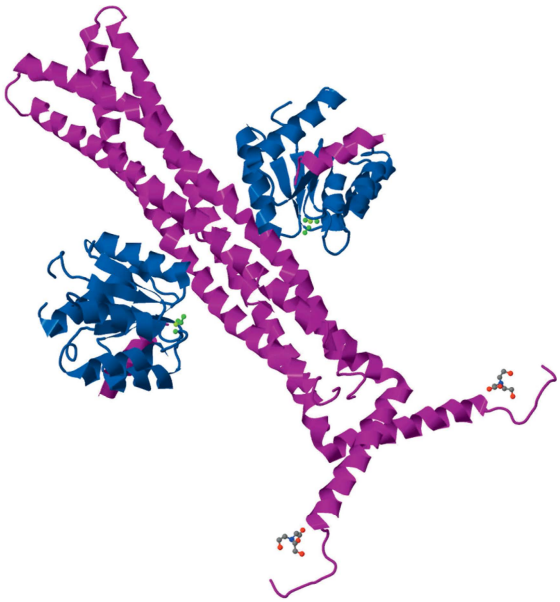


Figure 10.7 Structure of a CheZ dimer, which is instrumental in chemotaxis in *E. coli*. The phosphatase CheZ stimulates the dephosphorylation of the response regulator CheY by a so-far unknown mechanism. The most prominent structural feature of the CheZ dimer is a long four-helix bundle that consists of two helices from each monomer. (Protein Data Bank: PDB 1KMI [18].)

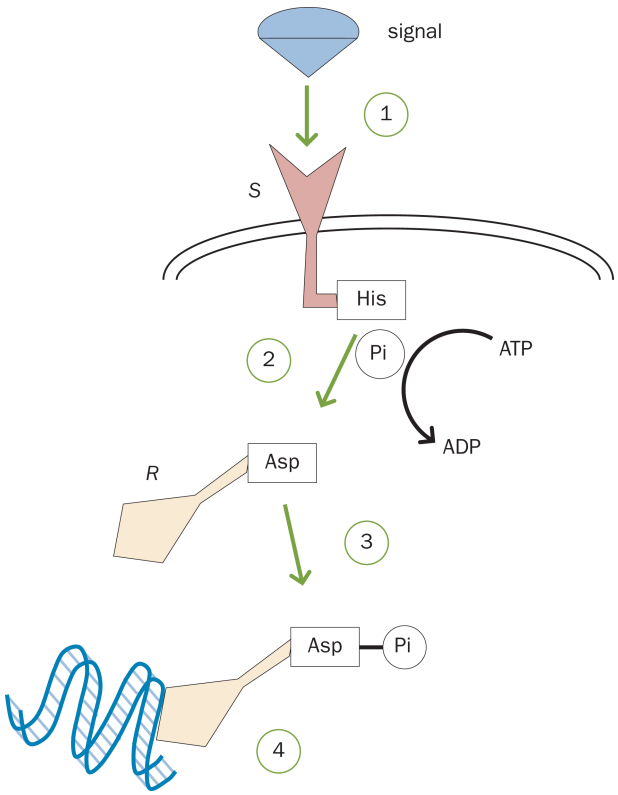


Figure 10.8 Signal transduction in a two-component system. The environmental signal binds to the sensor *S*. A histidine residue (His) on *S* receives a phosphate group (Pi), which is quickly transferred to an aspartic acid (Asp) residue on the response regulator *R*, which binds to DNA and effects a change in gene expression.

autokinase and the process is called autophosphorylation. The bond between the residue and the phosphate group is rather unstable, which allows the histidine kinase to transfer the phosphate group to an aspartic acid residue on the second component, the response regulator or receiver. The phosphate transfer causes a conformational change in the receiver, which changes its affinity to a target protein or more typically to a specific DNA sequence, causing differential expression of one or more target genes. The ultimate physiological effects can be manifold.

Once the task of the TCS system has been accomplished, the response regulator must be dephosphorylated in order to return it to its receptive state. The rate at which the aspartyl phosphate is released as inorganic phosphate is well controlled and results in half-lives of the phosphorylated form of the regulator that may vary from seconds to hours. Interestingly, two slightly different variants have been observed in the dephosphorylation mechanism of the response regulator. In the simpler case, the mechanism is independent of the state of the sensor, and the sensor is called monofunctional. In the more complicated variant, the dephosphorylation of the response regulator is enhanced by the unphosphorylated form of the sensor protein, which implies that the sensor has two roles and is therefore bifunctional. The two variants are shown in **Figure 10.9**. In the following, we perform simulations with both of them and refer the reader to the literature [6, 19] for a further discussion of their respective relevance.

It should be noted that the phosphorylation state of the regulator in both variants of the design may change in the absence of a signal, which introduces a certain level of noise in the system [6, 19]. Other variations of TCS are coupled TCS systems and analogous systems containing more components. An example of the latter is the sporulation system Spo in organisms like *Bacillus anthracis*, which consists of nine possible histidine sensors and two response regulators [20]. Other systems of a similar type are found in phosphorelays that regulate virulence [16]. Signaling systems in plants also often contain more than two components.

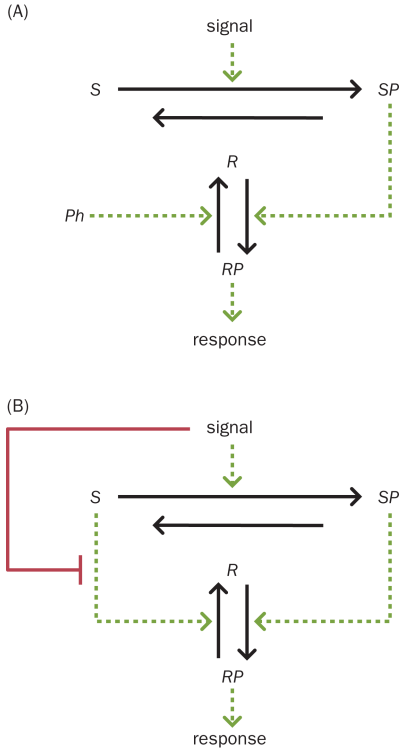


Figure 10.9 Two variants of a two-component signaling (TCS) system. The signal triggers the phosphorylation of a sensor *S*, which leads to phosphorylation of a response regulator or receiver *R*, leading to a response. (A) The monofunctional variant includes a phosphatase (*Ph*) which removes the phosphate from the phosphorylated receiver *RP*. (B) In the bifunctional variant, the sensor facilitates dephosphorylation of the phosphorylated receiver, a process that is inhibited by the signal.

10.2.2 Mathematical Models of TCS Systems

For a model analysis that is representative of these types of studies, we follow the work of Igoshin et al. [6], which built upon earlier analyses by Alves and Savageau [19] and Batchelor and Goulian [21]. Of the many questions that could be analyzed, we focus on the shape of the signal–response curves and ask whether the TCS system can be bistable and show hysteresis. Diagrams of mono- and bifunctional TCS systems are shown. **Figure 10.10** shows a more detailed diagram that contains both variants as special cases. It also contains intermediate complexes and shows the rate constants used in the model; their names are taken directly from Igoshin et al. [6]. By setting some parameters equal either to zero or to a different value, the mono- or bifunctional models are obtained.

Following Igoshin and collaborators, we set the model up as a mass-action model, so that the only parameters are rate constants, while all kinetic orders are 1 or 0. Translation of the diagram in Figure 10.10 leads directly to the following model:

$$
\begin{aligned}
S' &= -k_{ap}(\mathit{signal})\cdot S + k_{ad}(SP) + k_{d3}(S{\sim}R) - k_{b3}S\cdot R + k_{d2}(S{\sim}RP) - k_{b2}S\cdot(RP),\\
(SP)' &= k_{ap}S - k_{ad}(SP) - k_{b1}(SP)\cdot R + k_{d1}(SP{\sim}R),\\
(SP{\sim}R)' &= k_{b1}(SP)\cdot R - k_{d1}(SP{\sim}R) - k_{pt}(SP{\sim}R),\\
(S{\sim}RP)' &= k_{pt}(SP{\sim}R) + k_{b2}S\cdot(RP) - k_{d2}(S{\sim}RP) - k_{ph}(S{\sim}RP),\\
(S{\sim}R)' &= k_{b3}S\cdot R - k_{d3}(S{\sim}R) + k_{ph}(S{\sim}RP),\\
R' &= -k_{b1}(SP)\cdot R + k_{d1}(SP{\sim}R) + k_{d3}(S{\sim}R) - k_{b3}S\cdot R + k_{cat}(Ph{\sim}RP),\\
(RP)' &= k_{d2}(S{\sim}RP) - k_{b2}S\cdot(RP) + k_{d4}(Ph{\sim}RP) - k_{b4}(Ph)\cdot(RP),\\
(Ph{\sim}RP)' &= k_{b4}(Ph)\cdot(RP) - k_{d4}(Ph{\sim}RP) - k_{cat}(Ph{\sim}RP),\\
(Ph)' &= k_{d4}(Ph{\sim}RP) - k_{b4}(Ph)\cdot(RP) + k_{cat}(Ph{\sim}RP),
\end{aligned}
\qquad (10.5)
$$

where, for notational convenience, in this and the following equations, time derivatives are denoted by a prime rather than a dot. Igoshin and co-workers obtained most of the equations and parameter values from Batchelor and Goulian [21] but added some reactions and the corresponding parameters. A list of rate constants

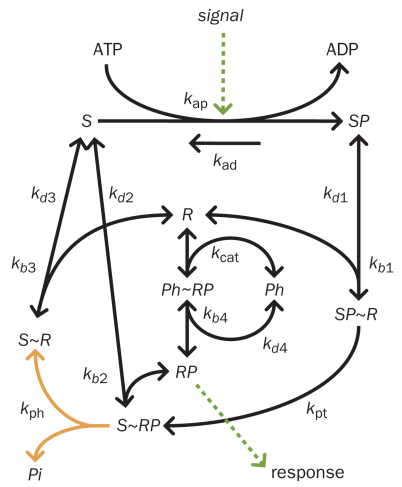


Figure 10.10 Detailed model diagram of the TCS system. By setting $k_{ph} = 0$, one obtains the monofunctional variant.

TABLE 10.3 RATE CONSTANTS AND INITIAL VALUES FOR MONOFUNCTIONAL (MF) AND BIFUNCTIONAL (BF) TCS SYSTEMS (DIFFERENCES ARE SHADED)

	k_{ap}	k_{ad}	k_{b1}	k_{d1}	k_{b2}	k_{d2}	k_{b3}	k_{d3}	k_{b4}	k_{d4}	k_{pt}	k_{ph}	k_{cat}	$S(0)$	$R(0)$	$Ph(0)$
MF	0.1	0.001	0.5	0.5	0.05	0.5	0.5	0.5	0.5	0.5	1.5	0	0.005	0.17	6.0	0.17
BF	0.1	0.001	0.5	0.5	0.05	0.5	0.5	0.5	0.5	0.5	1.5	0.05	0.025	0.17	6.0	0.17

and initial values is given in Table 10.3; other initial values are zero. These parameter values were originally determined for the EnvZ/OmpR TCS system. EnvZ is an inner membrane histidine kinase/phosphatase, for instance in *E. coli*, that senses changes in the osmolarity of the surrounding medium. EnvZ thus corresponds to *S* in our model. It regulates the phosphorylation state of the transcription factor OmpR, which corresponds to *R* in our notation. OmpR controls the expression of genes that code for membrane channel proteins, called porins, that are relatively abundant and permit the diffusion of molecules across the membrane.

We begin our analysis with a typical simulation experiment, namely, we start the system at its steady state and send a signal at time $t = 100$. Mechanistically, we just change the independent variable *signal* in (10.5) from 1 to 3. Since we are not really interested in intermediate complexes, we only show the responses in *S*, *SP*, *R*, and *RP*, among which *RP* is the most important output to be recorded, because it is an indicator of the response. The mono- and bifunctional systems show similar yet distinctly different responses. In particular, the bifunctional system responds noticeably faster (Figure 10.11). If the signal is stronger, (for example, *signal* = 20), the two systems show very similar responses. Note that the two systems start at different values, which correspond to their (different) steady states.

As a second test, we model a short signal. Specifically, we set *signal* = 20 at time $t = 100$ and stop the signal at time $t = 200$ by resetting it to 1. Interestingly, the two TCS systems now respond in a qualitatively different fashion (Figure 10.12). The monofunctional system responds to the signal and returns to its steady state as soon as the signal stops. By contrast, the bifunctional system seems to assume a new steady state. Will it eventually return to the initial state? The answer is difficult to predict but easy to check with a simulation: we simply extend the simulation to an arbitrarily large *t* value like $t = 12{,}000$. Surprisingly, the system does not stay at this apparent steady state (which really is not a steady state); neither does it return to the initial state. Instead, after a while, it starts rising again (without any intervention from the outside) and assumes a new steady state (Figure 10.13).

The short-signal experiment suggests at the very least that the bifunctional TCS system permits bistability. But does it also permit hysteresis? Moreover, can we

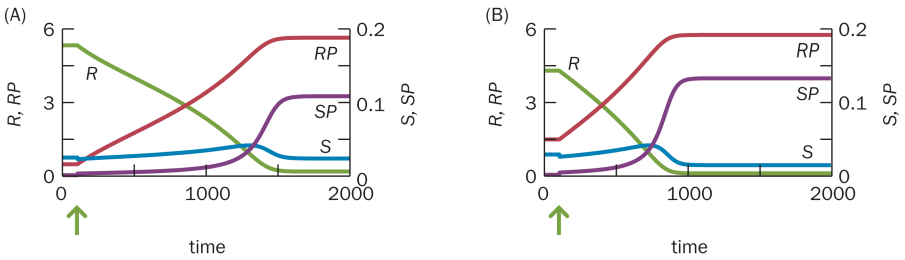


Figure 10.11 Simulations with the two variants of the TCS system. (A) Responses of the monofunctional variant to a signal at time $t = 100$ (arrow). (B) Corresponding responses of the bifunctional design.

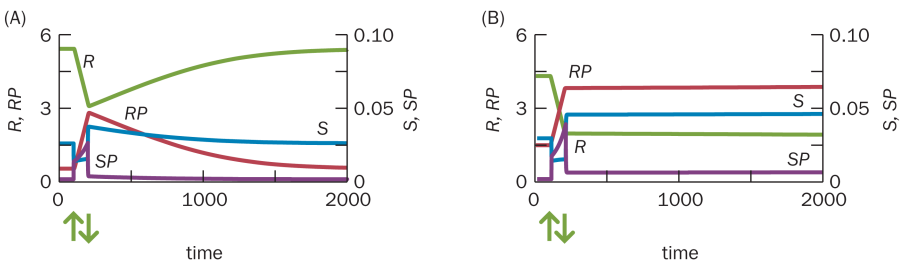


Figure 10.12 Differences in responses of the two variants of the TCS system. The monofunctional (A) and bifunctional (B) TCS systems show qualitatively different responses to a short, transient signal (arrows), with the former returning to its steady state and the latter reaching a new state that appears to be a steady state; however, see Figure 10.13.

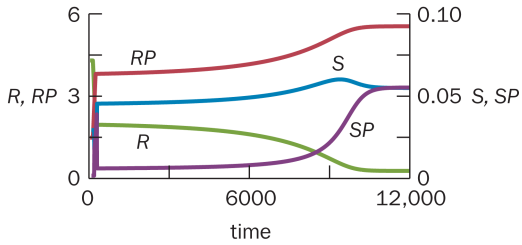


Figure 10.13 Continuation of the simulation of the bifunctional system in Figure 10.12. The presumed steady state is only a transient state, and the system shows a late response to the signal by approaching the true steady state.

test whether the monofunctional system exhibits similar responses? To answer these questions, we execute a series of experiments with varying signal strengths and study the response in *RP*. We begin with the monofunctional variant, by running it from the initial values in Table 10.3 (with all other initial values at zero) with a low signal of 1 and computing its steady state. Now the real series of experiments begins. We initiate the system at the steady state and reset the signal to a different value between 0 and 4 at time $t = 100$. For signal strengths between 0 and about 1.795, the corresponding values of the response variable *RP* increase slowly from 0 to about 2. Intriguingly, a tiny further increase in the signal to 1.8 evokes a response of 5.57. Further increases in signal do not change this response much. It is pretty clear that the system is bistable.

Does the system also exhibit hysteresis? To find out, we perform a second set of experiments, this time starting with the high steady state, which is characterized by values that we obtain by running the original system with a high signal towards its steady state. Beginning with a high signal of 4, we again see a response of about 5.6. Now we slowly lower the signal from experiment to experiment, every time recording the value of *RP*. Nothing much happens until 1.8, and the results are essentially identical to those in the previous series of experiments. However, lowering the signal to 1.795 does not cause a jump. In fact, the response is high until the signal strength is reduced to about 1.23, where the response value suddenly drops to about 0.65. These results are a clear indication of hysteresis. The signal–response relationship is shown in **Figure 10.14**.

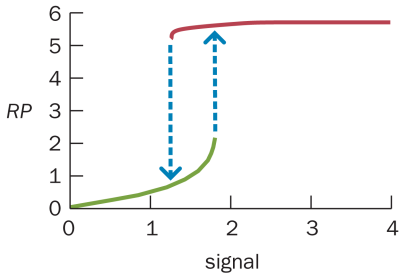


Figure 10.14 The signal–response relationship of the monofunctional TCS model shows clear hysteresis. For signal strengths between about 1.23 and 1.8, the response depends on whether the system was in a low or high state when the signal hit. This type of phenomenon shows that the monofunctional TCS system has memory.

Igoshin et al. [6] and Alves and Savageau [19] derived conditions for when the different TCS systems exhibit bistability and hysteresis and studied the effect of noise in the signal, which in the case of TCS systems appears to be of minor significance. They also discussed the advantages and limitations of the mono- and bifunctional variants in detail. As in many cases of this nature, one variant is more efficient with respect to some physiologically relevant aspects of the response but less efficient with respect to other aspects, and the overall superiority of one over the other depends on the environment in which the organism lives.

10.3 MITOGEN-ACTIVATED PROTEIN KINASE (MAPK) CASCADES

10.3.1 Features of MAPK Cascades

In contrast to the TCS system in bacteria, most higher organisms use a more complicated signal transduction system whose key component is the mitogen-activated protein kinase (**MAPK**) **signaling cascade**. MAPK (usually pronounced Map-kinase) systems appear to be present in all eukaryotes and also in a few prokaryotes, such as the biofilm forming bacterium *Myxococcus xanthus*. Intriguingly, MAPK cascades have a highly conserved architecture. The MAPK signaling cascade receives signals from cell surface receptors or cytosolic events, processes them by filtering out noise, usually amplifies them, and ultimately affects various downstream targets, such as cytosolic proteins and nuclear transcription factors. The external signal may consist of a mitogen or inflammatory cytokine, a growth factor, or some physiological stress. This signal is transduced, for example, by a G-protein-coupled receptor that activates the MAPK cascade. The ultimate result of the MAPK system may be as different as inflammation, differentiation, apoptosis, or initiation of the cell cycle.

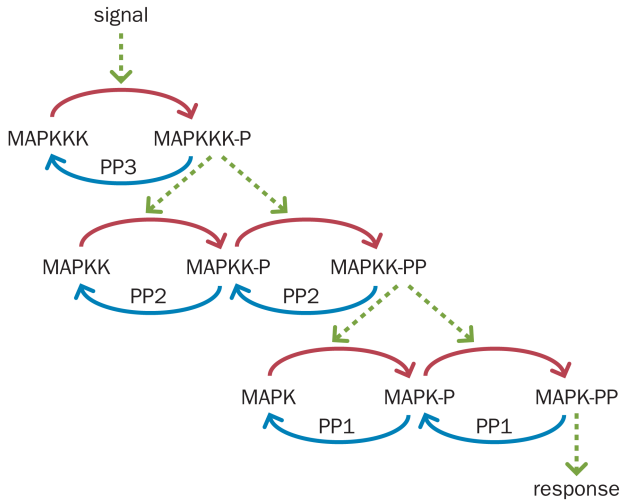


Figure 10.15 Typical three-layer MAPK cascade. At each level, the inactive form on the left is phosphorylated (once or twice) into the active form, which serves as enzyme at the next layer. Phosphatases (PP1, PP2, PP3) return the active forms to their inactive states by removing phosphate.

The typical MAPK cascade is shown in **Figure 10.15**. It consists of three layers where proteins are phosphorylated or dephosphorylated. The kinase nearest to the signal source is generically called MAP kinase kinase kinase (MAPKKK). If it is activated by a cytosolic or external signal, MAPKKK is phosphorylated to MAPKKK-P. Being a kinase itself, MAPKKK-P activates phosphorylation of the MAP kinase kinase (MAPKK) in the second layer. In fact, full activation of MAPKK requires sequential phosphorylation at two sites: a tyrosine site and a threonine site. The resulting MAPKK-PP in turn phosphorylates and thereby activates the kinase of the third layer, the MAP kinase (MAPK). Again, full activation requires two phosphorylation steps. The activated MAPK-PP can phosphorylate cytosolic targets or translocate to the nucleus, where it activates specific transcriptional programs. At each layer, a phosphatase can dephosphorylate the active forms into the corresponding inactive forms. Two prominent examples of MAP kinases are the extracellular-signal-regulated kinase (ERK) and the c-Jun N-terminal kinase (JNK), where Jun refers to a family of transcription factors. The responses of the cascades are quite fast: within the first 5 minutes of stimulation, ERK is activated up to 70% [22], and within 10 minutes, significant amounts of activated ERK are translocated to the nucleus [23]. A mutated form of MAPKKK in the ERK pathway has often been found in malignant melanomas and other cancers.

An obvious question is this: the three-layer cascade has been conserved through evolution, so it is fair to assume that this design has advantages, but what are these, for instance, in comparison with a simple or double phosphorylation at a single layer? We use a model analysis to shed light on this intriguing question.

The key module of the cascade is the sequential phosphorylation at two sites, which is catalyzed by the same enzyme from the next higher layer. Simplified and detailed diagrams of this module are presented in **Figures 10.16** and **10.17**, where *Y* stands for MAPK, *E* is the upstream catalyzing enzyme MAPKK-PP, *YP* and *YPP* are singly or doubly phosphorylated forms, and *YE* and *YPE* are complexes of *Y* and *YP* with the enzyme. The same signaling motif exists for the MAPK layer, which is the final two-step kinase activation and is activated by MAPK-PP as E.

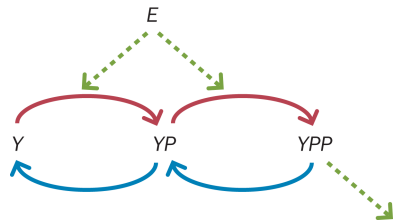


Figure 10.16 Key module of the three-layer MAPK cascade in Figure 10.15. It is tempting to use Michaelis–Menten functions to model the two phosphorylation steps, but the two steps compete for the same enzyme, whose concentration is not constant.

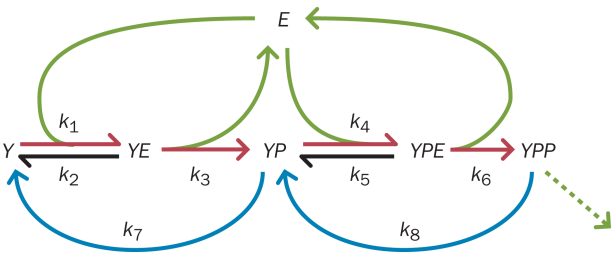


Figure 10.17 Detailed representation of the module in Figure 10.16. Accounting for the two substrate–enzyme complexes *YE* and *YPE* shows the dual role of enzyme *E*.

10.3.2 Mathematical Models of MAPK Cascades

It is tempting to set up the two phosphorylation steps with Michaelis–Menten rate functions, but such a strategy is not the best option, because (A) the enzyme concentration is not constant, (B) the enzyme concentration is not necessarily smaller than the substrate concentration, and (C) the two reaction steps are competing for the same enzyme. Instead, it is useful to retain the mechanistic ideas of the Michaelis–Menten mechanism—which postulates the formation of a substrate–enzyme complex—and to formulate this mechanism in the basic format of mass-action kinetics (see Chapters 2, 5, and 9). What we do *not* want to do is to rely on the quasi-steady-state assumption that would simplify this system toward the well-known Michaelis–Menten function, because in this format the enzyme concentration is no longer explicit, let alone dynamic. A direct translation of the diagram in Figure 10.17 into mass-action equations is straightforward:

$$\begin{aligned}
\dot{Y} &= -k_1 Y \cdot E + k_2(YE) + k_7(YP) \\
(\dot{YE}) &= k_1 Y \cdot E - (k_2 + k_3)(YE) \\
(\dot{YP}) &= k_3(YE) - k_7(YP) - k_4(YP) \cdot E + k_5(YPE) + k_8(YPP) \\
(\dot{YPE}) &= k_4(YP) \cdot E - (k_5 + k_6)(YPE) \\
(\dot{YPP}) &= k_6(YPE) - k_8(YPP)
\end{aligned} \tag{10.6}$$

To perform simulations, we specify more-or-less arbitrarily chosen values for the parameters and initial concentrations in **Table 10.4**. It is now easy to study the effects of changes in the enzyme E on the variables of the module. At the beginning of the simulation, $E = 0.01$, and the initial conditions for the various forms of Y are set such that the system is more or less in a steady state. Suppose now that at time $t = 1$ the signal E is increased to 10. This is a strong signal, and the system responds with phosphorylation on both sites. After a brief transition, the balance between Y, YP, and YPP is entirely switched, with very little unphosphorylated Y left (**Figure 10.18**). Notably, the amount of YPE is relatively large, because quite a lot of enzyme is available. If the signal is reset to 0.01, the module returns to its initial state (**Figure 10.19**).

It is also easy to study how strong the signal must be to trigger a response. If E is set to 3, rather than 10, the response is rather similar to the previous scenario, although the response is not as pronounced (results not shown). If E is set to 1 instead, the response is much slower. Y is reduced only to about 13.6, while YPP assumes a value of about 3.2. In other words, no real switch is triggered. For even weaker signals, the response is insignificant (**Figure 10.20**).

Now that we have a bit of a feel for the module, we can use it to implement a complete MAPK cascade (see Figure 10.15). Specifically, we set up the module in triplicate, with additional variable names X, XP, Z, ZP, and ZPP, which represent the top and bottom layers of the cascade, respectively and use the same parameter values and initial conditions as before. Regarding the first layer, which involves only one phosphorylation of MAPKKK, we use a single Michaelis–Menten mechanism, again formulated in mass-action representation, with $X(0) = 28$ and $XE(0) = XP(0) = 0.1$. The input to the first module is signal S, and the output of this module is the singly phosphorylated form XP, which catalyzes the two phosphorylation steps $Y \rightarrow YP$ and $YP \rightarrow YPP$ at the next layer. Similarly, the output of the second layer, YPP, catalyzes the reactions $Z \rightarrow ZP$ and $ZP \rightarrow ZPP$ at the bottom layer. ZPP triggers the actual physiological response, but we simply use ZPP as output indicator of the cascaded system. The model implementation is shown in **Figure 10.21**, and we set the kinetic parameters to exactly the same values as before (see Table 10.4).

TABLE 10.4 KINETIC PARAMETER VALUES AND INITIAL CONDITIONS FOR THE MAPK SYSTEM (10.6)

$k_1 = k_4$	$k_2 = k_5$	$k_3 = k_6$	$k_7 = k_8$	$Y(0)$	$YE(0)$	$YP(0)$	$YPE(0)$	$YPP(0)$
1	0.1	4	2	28	0.1	0.2	0.01	0.02

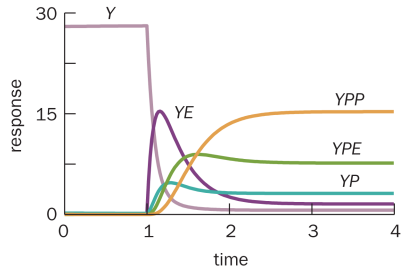


Figure 10.18 Response of the MAPK module in Figure 10.17. At time $t = 1$, the signal is increased. The result is single and double phosphorylation of Y.

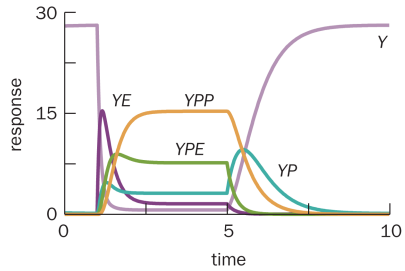


Figure 10.19 Continuation of the response in Figure 10.18. At time $t = 5$, the signal is decreased to its original low state, and X returns to its unphosphorylated state.

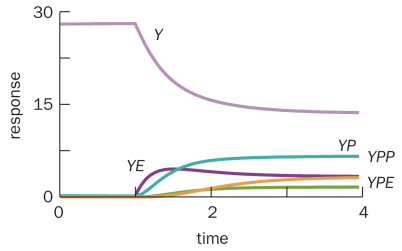


Figure 10.20 Response of the MAPK module in Figure 10.17 to a weak signal at t = 1. While Y responds to the signal with a decrease, the double-phosphorylated form YPP remains low.

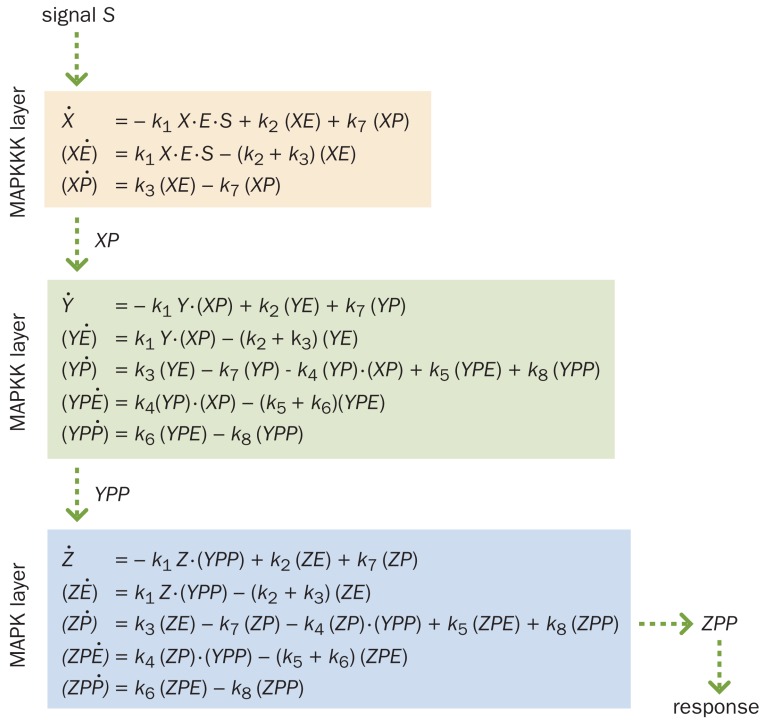


Figure 10.21 Implementation of a model of the MAPK cascade in Figure 10.15. The MAPK and MAPKK layers are analogous, while the MAPKKK layer is simpler since it contains only one phosphorylation step.

Strictly speaking, this implementation is a simplification that assumes that the "enzymes" *XP* and *YPP*, as well as *ZPP*, remain constant, which is not really the case, since *XP* undergoes reactions with *Y* and *YP* and *YPP* reacts with *Z* and *ZP*; the role of *ZPP* is harder to quantify in general. Thus, for a more accurate representation, all reactions utilizing or releasing *XP*, *YPP*, or *ZPP* should be included in the equations governing these variables. For instance, the equation of *XP* should read

$$\dot{XP} = k_3 XE - k_7 XP - k_1 \cdot Y \cdot XP + k_2 YE + k_3 YE - k_4 \cdot YP \cdot XP + k_5 YPE + k_6 YPE \qquad (10.7)$$

If the signal is off, *XP* and *YPP* have rather small values, but in the case of greater interest, namely if the signal is on and stays on, *XP* and *YPP* are constant, with values of about 14.3 and 12.7 for the given parameters k_i, and simulations show that accounting explicitly for interactions with the lower levels slows down the responses of the cascade a little bit but that ignoring them does not affect the results qualitatively. We therefore ignore these processes for simplicity. An Exercise assesses the differences between models.

As a first simulation, suppose the external signal *S* starts at 0.01, switches to 50 at $t = 2$, and switches back to 0.01 at $t = 6$. The responses of the most important variables are shown in **Figure 10.22**. The strong signal causes rapid phosphorylation at all layers. If the signal is not as strong ($S = 5$ at $t = 2$), *X* does not dip as close to zero as before but only to about 6. Otherwise the responses are similar (results not shown). For a relatively weak signal ($S = 0.5$ at $t = 2$), we see the benefit of the cascaded system. While *XP* only rises to about 5, the ultimate response variable *ZPP* shows a robust response (**Figure 10.23**). In other words, we have solid signal amplification in addition to a steeper response and clear signal transduction. If the signal rises to only two or three times its initial value ($S = 0.02$ or $S = 0.03$ at $t = 2$), the response in *ZPP* is very weak. The results give us a first answer to our earlier question of why there are three layers in the cascade: three layers permit improved signal amplification. At the same time, they effectively reduce noise [24]. It has also been shown that the three-layer design improves the robustness of the signal transduction system against variations in parameter values [25].

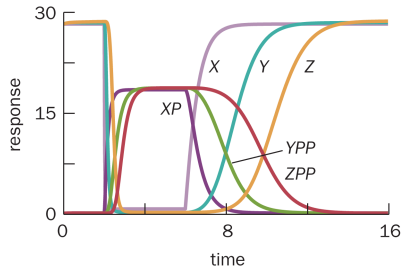


Figure 10.22 Responses of the main variables of the MAPK model to switches in signal. The *X*, *Y*, and *Z* variables respond to an increase and decrease in signal in a staggered manner (see the text for details).

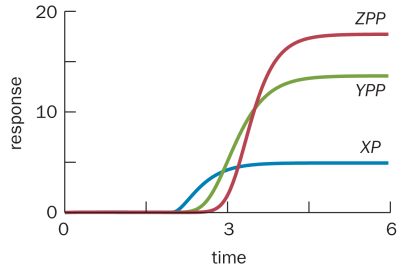


Figure 10.23 Signal amplification by the MAPK cascade. Even for a relatively weak signal, the responses at the three layers of the MAPK cascade are successively stronger, indicating amplification of the signal by the cascaded structure.

It is easy to investigate how the cascade responds to brief repeated signals. We perform a simulation as before but create an *off–on* sequence of signals. As an example, if the signal switches between 0.01 and 0.5 every half time unit, beginning with $S = 0.5$ at $t = 2$, we obtain the result shown in Figure 10.24. We can see that the responses at the different layers are quite different. The first layer exhibits fast, but very weak, responses in *XP*, whereas the ultimate, amplified response *ZPP* at the third layer in some sense smooths over the individual short signals.

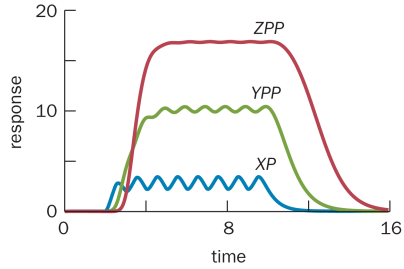


Figure 10.24 Responses of different layers of the MAPK cascade to brief, repeated signals. The signal at the output layer smooths the responses at the MAPKK and MAPKKK layers into a solid and sustained, amplified signal.

It should be clear that no attempt was made in the implementation of the cascade to estimate biologically relevant parameter values; our purpose was simply to demonstrate the concepts of the dual-phosphorylation module and its role in the MAPK cascade. Huang and Ferrell [26] and Bhalla and Iyengar [27] discuss parameter values and features of realistic MAPK cascades in some detail, and Schwacke [28, 29] provides a computational rationale for parameter values of the cascade that are actually observed in nature and that lead to stronger amplification than our ad hoc parameter set.

It should be noted that all signaling systems obviously have a spatial component, which typical ODE models ignore. In fact, it seems that some cascades are organized along protein scaffolds, which can significantly affect their efficiency [30–32]. Furthermore, MAPK cascades typically do not work in isolation. Instead, the cell contains several parallel cascades, which communicate with each other via **crosstalk** [33–35]. For instance, the output of the top layer in one cascade might activate or inhibit the center layer in another. Crosstalk is thought to increase the reliability and fidelity of signal transduction. It furthermore creates options for very complex signaling tasks, such as an in-band detector, where the cascade only fires if the signal is within a certain range but neither weaker nor stronger. Crosstalk also renders all logic functions (AND, OR, NOT gates) including the so-called EXCLUSIVE OR (XOR) mechanism possible. In the latter case, the cascade fires only if either one of two signals is present, but it does not fire if both signals are on or if both signals are off [27] (see also the discussion of bi-fans in Chapter 3 and the Exercises). Detailed experimental and theoretical work seems to indicate that normal and cancer cells may differ not so much in their signaling components but rather in the way in which these components are causally connected within complex signaling networks [36].

It is interesting to note that cells often use the same or similar components of a signaling cascade for different purposes. For instance, yeast cells employ Ste20p, Ste11p, and Ste7p for a number of different stress responses, including pheromone exposure, osmotic stress, and starvation [32].

10.4 ADAPTATION

The human body responds to many signals from the outside world on a daily basis. Our pupils constrict in bright sunlight and dilate when it is dark, we sweat to cool the body, and we pull our hand back when we touch something hot. Many of these responses are driven by the body itself, independent of our conscious control. Intriguingly, we often get used to a signal, if it is repeated in short order or if it persists for some time. We start breathing normally again a few days after moving to high altitudes, and after a while we don't really notice some noises anymore, unless we specifically focus on them. This phenomenon of a diminished biological response to the same, repeated signal is called **adaptation**. In the language of systems biology, the organism is initially at a homeostatic (steady) state and responds to occasional signals by moving from an *off* state to an *on* state but then returns to the *off* state, when the signal has vanished. However, if the same type of signal persists for some while, the system returns from the *on* state to the *off* state anyway, even though the *off* state normally corresponds to the absence of the signal.

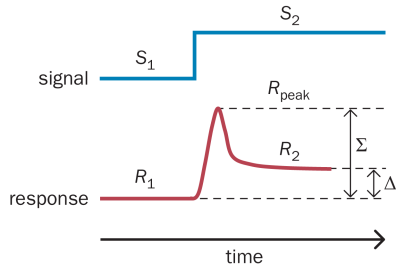


Figure 10.25 Features of adaptation. For a low signal S_1, the typical response is an *off* state, characterized by the response R_1. For short spikes in the signal, the response is transient, and the system returns to R_1 (not shown). In the case of adaptation, the system responds to the signal S_2 by reaching R_{peak} but then approaches a value R_2 that is close to R_1, even though the signal S_2 persists. The response can be characterized by its sensitivity Σ and precision Π; the latter is inversely related to the relative difference Δ between R_1 and R_2.

According to Ma and coworkers [37], adaptation can be characterized by two features, which they call *sensitivity* and *precision* (Figure 10.25). The sensitivity, ∑, measures the normal strength of the response to a signal, that is, the relative difference between the output values of the *on* and *off* states in relation to the relative signal strength. The precision, ∏, is defined as the inverse of the difference Δ

between the original homeostatic state and state the system attains through adaptation. Specifically,

$$\Sigma = \frac{\left|R_{peak} - R_1\right| / \left|R_1\right|}{\left|S_2 - S_1\right| / \left|S_1\right|} \tag{10.8}$$

$$\Pi = \Delta^{-1} = \left(\frac{\left|R_2 - R_1\right| / \left|R_1\right|}{\left|S_2 - S_1\right| / \left|S_1\right|} \right)^{-1} \tag{10.9}$$

Ma's group explored the question of what types of systems with three nodes are capable of displaying adaptation. One such motif is shown in **Figure 10.26** using our notation. Whereas Ma used a Michaelis–Menten representation, we use an S-system model with simplified notation, which here has the intriguing advantage that one can actually compute conditions under which the system displays adaptation (see Exercises). A symbolic S-system model with positive variables A, B, and C can be written down immediately:

$$\begin{aligned} \dot{X}_1 &= \alpha_1 S X_2^{g_1} - \beta_1 X_1^{h_1} \\ \dot{X}_2 &= \alpha_2 X_1^{g_2} - \beta_2 X_2^{h_2} \\ \dot{X}_3 &= \alpha_3 X_1^{g_3} - \beta_3 X_3^{h_3} \end{aligned} \tag{10.10}$$

Without loss of generality, we may suppose that the normal steady state is (1, 1, 1) for a signal $S = 1$, which immediately means that $\alpha_i = \beta_i$ for $i = 1, 2, 3$; convince yourself that this is generally true. **Table 10.5** provides typical parameter values.

To explore the responses of the system, we start at the steady state and send signals as brief boluses at times 5, 30, and 45. In PLAS, this is easily done with commands like @ 5 $S = 4$, @ 5.1 $S = 1$. In all three cases, the output variable X_3 shoots up, then undershoots, before returning toward the homeostatic value of 1 (**Figure 10.27**). The variables X_1 and X_2 are not of prime interest, but they return to 1 as well. These responses are not surprising, because the steady state (1, 1, 1) is stable. Now the real experiment begins. At $t = 75$, we reset the signal permanently to $S = 1.2$, which corresponds to the relative change Ma and collaborators used in their paper. Although the signal stays *on* at this value throughout the rest of the experiment, the system adapts, and X_3 approaches a value of 1.013, which corresponds to a deviation of 1.3% from the original value. Ma and colleagues consider a difference below 2% as sufficiently precise, so our model is indeed both sensitive and precise.

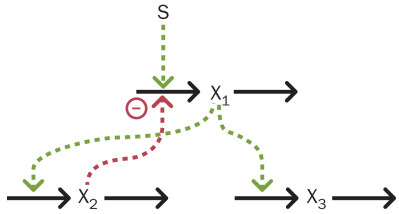


Figure 10.26 Three-node system for studying adaptation. The system is adapted from Ma et al. [37] but represented in the format we have been using throughout the book.

TABLE 10.5 PARAMETER VALUES FOR THE ADAPTATION SYSTEM (10.10)

i	$\alpha_i = \beta_i$	g	h
1	2	−1	0.4
2	0.1	1	0.05
3	4	0.75	0.5

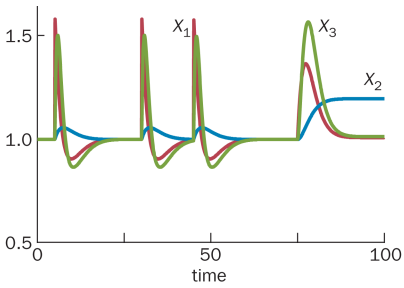


Figure 10.27 Responses of the three-node system. Following transient signals at times $t = 5$, 30, and 45, the variable of interest, C, shoots up and then returns to homeostasis. This response is to be expected, because the homeostatic steady state is stable. However, in response to a persistent signal, here starting at $t = 75$, the system adapts, and C approaches a new state that is very close to its homeostatic value. The time trends of variables A and B are shown but are not of primary interest here.

10.5 OTHER SIGNALING SYSTEMS

The differential equation models discussed so far are paradigm signal transduction systems in the narrowest sense. Many other model approaches have been proposed under the same rubric of signaling systems. For instance, gene regulatory networks are sometimes considered signaling systems, because the expression of one gene provides the signal for other genes to be expressed or repressed, which is mediated through transcription factors, repressors, or small RNAs (see [38] and Chapter 7).

A specific example is the genomic regulation of the cell cycle, which we have already discussed in the context of Boolean network analysis [39] in Chapter 3. For many years, Tyson, Novak, and others have been using ordinary differential equations to study cell cycles and their control. In the case of yeast, these models have become so good that they capture essentially all alterations in cell cycle dynamics that stem from mutants in any of the involved genes [40, 41]. Other approaches have been based on stochastic differential equations [42], recursive models [43], hybrid Petri nets [44], or machine-learning methods [45]. The juxtaposition of these diverse models of gene regulatory networks demonstrates again that modeling can take many shapes and forms and that the choice of a particular model depends greatly on the purpose of the model and the questions that are to be answered.

10.5.1 Circadian Clocks

Another fascinating class of signaling systems concerns circadian clocks. These molecular clocks sense and interpret light or other environmental conditions and allow organisms, ranging from prokaryotes to humans, to be active at appropriate times during the day–night cycle. The oscillations are essentially self-sustained, because they continue even if the organism is held in continuous light or continuous darkness. However, they usually run out of phase after a while, indicating that humans, for instance, have an intrinsic cycle duration that is usually slightly longer than 24 hours. Because the clocks are based on biochemical events, one might surmise that they would depend on the environmental temperature, but, interestingly, this is not the case. An immense body of literature documents the experimental and computational research that has been done on circadian rhythms over the past 100 years, since J. S. Szymanski observed that animals could maintain a 24-hour rhythm without external cues [46]. The relatively new field of chronobiology focuses on issues of circadian clocks and, for instance, tries to understand the effects on night shifts and jet lags on human health [47, 48].

It is surprisingly simple to construct clocks from systems with hysteresis. As a demonstration, let's return to the system in Eqn. (10.7) and study the variable Y. Its hysteresis plot is similar to that of X, although its numerical values are an order of magnitude smaller. Importantly, we see the same type of jumping behavior for the same values of S. This is no coincidence, because X and Y are tightly interconnected.

In the earlier demonstration of hysteresis, we artificially raised S from low values below 2 to high values above 4 and then lowered it back. This procedure suggests a thought experiment where some force within or outside the system would do this raising and lowering repeatedly. If so, we can quite easily convince ourselves that X and Y should start to oscillate in response.

But we can do even better, namely, by formulating a model. So far, the steady states of X and Y are functions of the independent variable S. Now, by making S a dependent variable, we can have its dynamics driven by X or Y so that S becomes high if X or Y is low, and low if X or Y is high.

To implement this plan, let's express S as a function of Y_{ss}, which here is feasible with a little bit of algebra. For Y to be at a steady state, Eqn. (10.4) must equal 0. Thus, we obtain

$$S = Y_{ss} \,/\left(0.5\, X_{ss}^{0.5}\right) \tag{10.11}$$

For X to be at a steady state, the first equation of (10.7) must equal 0, which yields

$$0.5\, X_{ss}^{0.5} = 2 + \frac{8\, Y_{ss}^4}{16^4 + Y_{ss}^4} \tag{10.12}$$

Plugging Eqn. (10.12) into (10.11), we obtain S as a function of Y_{ss}:

$$S = Y_{ss} \,/\left(2 + \frac{8\, Y_{ss}^4}{16^4 + Y_{ss}^4}\right) \tag{10.13}$$

This function is shown in **Figure 10.28A**. Figure 10.28B flips this plot over, so that Y_{ss} depends on S. Note, however, that Y_{ss} is not a function of S, because for values around $S = 3$, Y_{ss} has two values. Nonetheless, this mapping is very useful, because it shows the relationship between Y_{ss} and S. In fact, we can superimpose this plot on the left panel of **Figure 10.29A** and obtain Figure 10.29B. Two critical observations can be made. First, the low and high branches of the plot in Figure 10.29A are exactly recouped by the mapping between Y_{ss} and S. Second, this mapping adds a piece to the curve, connecting the jump-off points of Y_{ss} as a function of S. This new piece, curving from $(S,\, Y_{ss}) \approx (3.3,\, 9.1)$ upward to $(S,\, Y_{ss}) \approx (2.6,\, 19.7)$, exhibits the unstable steady states between the high and low stable steady states. All values on this piece are repelling, and unless one starts exactly at such a point, the solution will go either to the upper or to the lower steady state, which are both stable.

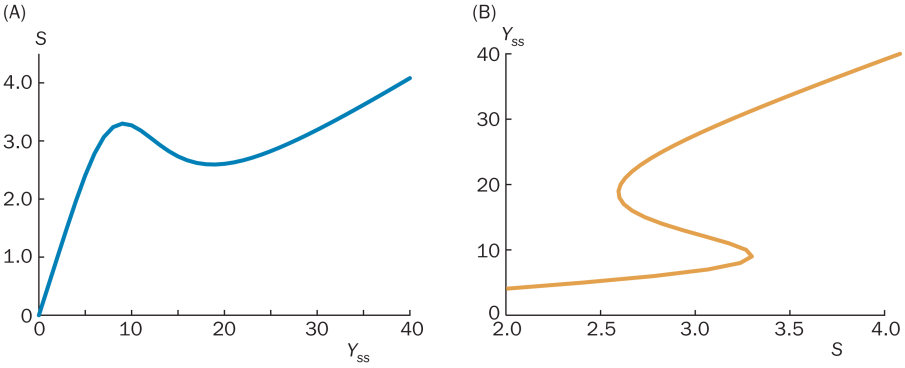


Figure 10.28 Relationship between Y_{ss} and the signal *S*. (A) A plot of (10.13), which results from computing *S* from the steady-state equations of (10.3–10.4). (B) The same plot flipped over for *S* between 2 and 4. Although *S* is a function of Y_{ss}, Y_{ss} is not a function of *S*, because, for a value of *S* around 3, Y_{ss} can have three different values.

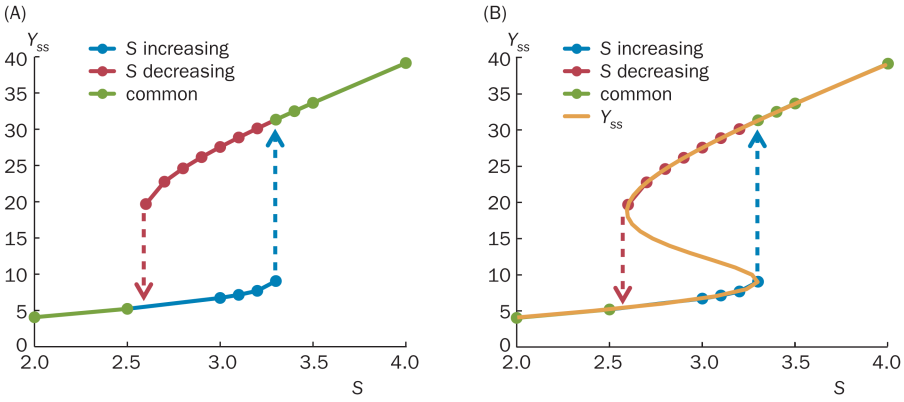


Figure 10.29 Hysteretic behavior of *Yss* in response to the signal *S*. (A) The hysteresis diagram of (10.3–10.4) with respect to Y_{ss}, which is qualitatively similar to the diagram for X_{ss} (Figure 10.6). (B) Superimposing Figure 10.28B on this plot "fills in the blanks" by showing the unstable steady states for signals between about 2.6 and 3.3.

Now let's return to the task of creating a clock. In order to minimize confusion, we replace *S* with the dependent variable *Z*, which we define as

$$\dot{Z} = 5Y^{-1} - 0.25Z^{0.5} \tag{10.14}$$

Other than exchanging *Z* for *S*, we retain the system in (10.3–10.4) without change. The exact format of *Z* is actually not all that critical but affects the shape of the oscillations. The important feature on which we must insist is that large values of *Y* strongly reduce the new signal and that low values strongly increase it. This requirement is met by *Y*'s exponent of –1. Also, *Z* should not wander off to infinity or zero, and should instead have the potential for a steady state, which is accomplished with the degradation term. With these settings, the signal becomes small as soon as *Y* becomes big and vice versa. A small signal causes *Y* to approach the low steady state, but as soon as *Y* becomes small enough, the signal increases, ordering *Y* to approach the high steady state. As a consequence, *X*, *Y*, and *Z* never actually reach their steady states and instead oscillate in the manner of a limit cycle (**Figure 10.30**), which we discussed in Chapter 5 and which is an excellent model for a clock. One easily checks that the system is indeed a limit cycle by perturbing one of

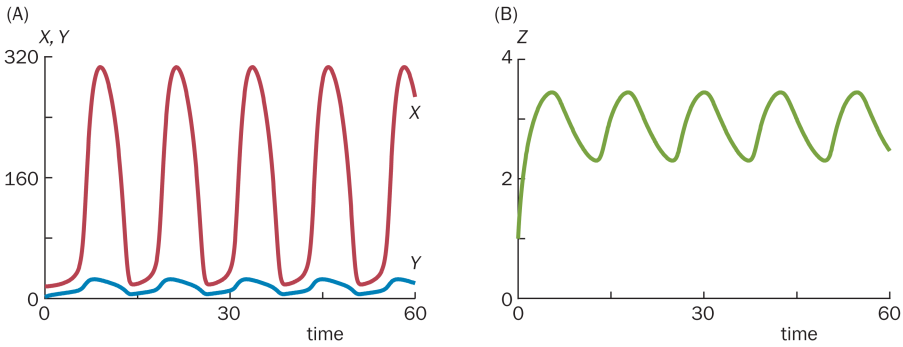


Figure 10.30 Oscillations resulting from making the signal dependent on *Y*. Owing to the construction where high values of *Y* lead to low values of *Z* and low values of *Y* lead to high values of *Z*, the system never reaches a steady state and instead oscillates in a sustained manner.

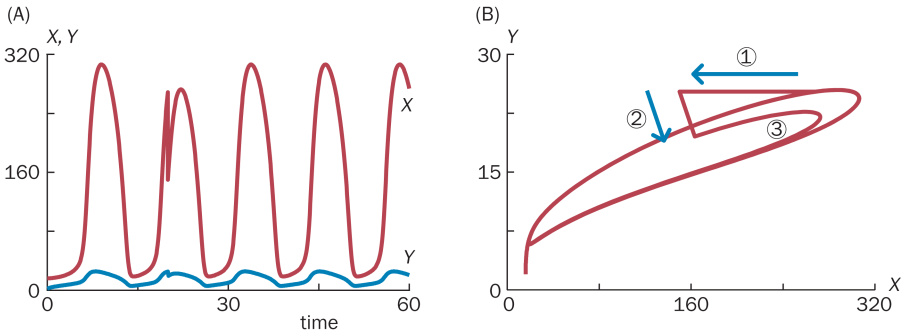


Figure 10.31 The oscillations resulting from making the signal dependent on *Y* form a limit cycle. If one of the variables is perturbed, the system quickly regains its limit-cycle behavior. Here, *X* was reduced at time $t = 20$ from about 270 to 150. The perturbation leads to a slight phase shift, but the original oscillation is regained. (A) shows the oscillation in the time domain, while (B) shows the corresponding phase-plane plot of *Y* versus *X*.

the variables. For instance, even if *X* is externally reduced a lot (here, from about 270 to 150 at time 20, the system recovers very quickly (**Figure 10.31**).

The frequency of the oscillation can be tuned by multiplying all three equations (Eqns. 10.3, 10.4, and 10.14) with the same positive factor. If the factor is less than 1, the oscillation is slower; if it is greater than 1, the oscillation is faster.

This transition from bistability to limit cycles has been studied in a model of an embryonic cell cycle oscillator in the frog *Xenopus* [49] and also with the so-called repressilator system [50] that we discussed in Chapter 7.

10.5.2 Quorum Sensing

A widespread example of cell-to-cell signal transduction occurs in the form of **quorum sensing** [51]. This environmental sensing of a critical mass—such as cell density or a threshold concentration of a signaling molecule—is used by multicellular microbial systems or ecosystems to perform a collective function. The phenomenon allows groups of bacteria or fungi to coordinate their responses, which may manifest in phenomena like swarming, aggregation, the assembly of fruiting bodies, or the formation of biofilms [52]. The latter leads to layers of microorganisms of the same or different species that are almost impenetrable to antibiotics and are therefore of great importance and concern for the treatment of virulent pathogens, such as *Pseudomonas aeruginosa* and *Staphylococcus aureus* [53]. Understanding the mechanisms and molecular details of quorum sensing is not only a prerequisite for controlling these pathogens but may also offer new avenues of synthetic biology, with the goal of engineering cells with rewired metabolic pathways or other desirable properties [54]. For example, the principle of quorum sensing has been harnessed by synthetic biologists to trigger the activation of gene circuits for desired outcomes (see Chapter 7).

Mathematically, the spatial distribution of bacteria or cells and the resultant buildup of a secreted communication molecule appear to be a typical application for complex partial differential equations (PDEs) that factor reaction-diffusion properties into a set of field equations [55, 56]. For a minimal model that aligns with the analyses in this introductory book, however, we can consider a much simpler ODE model in which the cell density in the environment is reflected in a parameter assignment. An example is quorum sensing in *Pseudomonas aeruginosa*, where a simplified system consists of a protein *R*, the freely diffusing autoinducer molecule *A*, and the protein-autoinducer bound complex, *P*, which is the activated form of *R* [57]. Ultimately, the levels of *P* in the system will lead to more *R* and *A* and to upregulated levels of elastase that destroy elastin-containing lung tissue, a common consequence of *P. aeruginosa* infections in cystic fibrosis patients. The model equations can be reduced into expressions for $\dot{R}$ and $\dot{A}$ as follows:

$$\dot{R} = V_R \frac{P}{K_R + P} + R_0 - k_R R$$

$$\dot{A} = V_A \frac{P}{K_A + P} + A_0 - d(\rho) A$$

$$P = k_p R A .$$

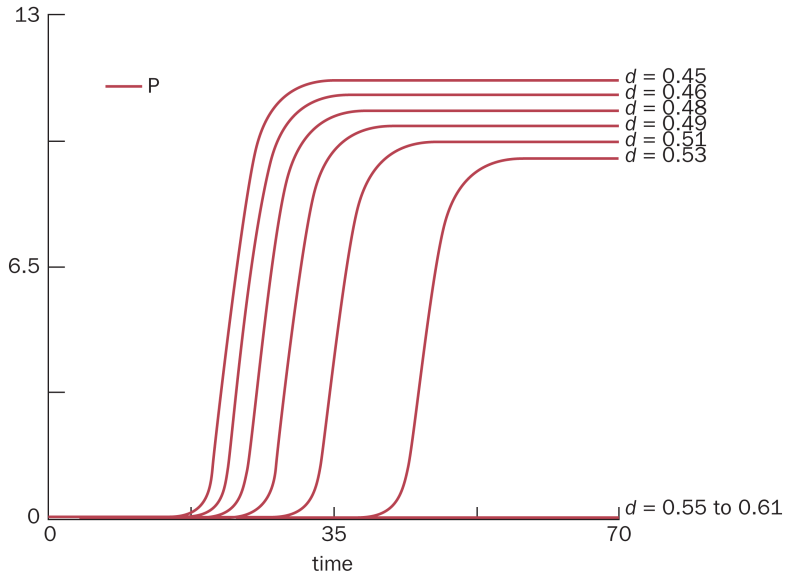


Figure 10.32 Output from a simple quorum sensing model [57] reflecting ultrasensitive behavior and bistability as a function of a parameter d that reflects an inverse relationship to cell density ρ and the extracellular space available for diffusion.

Of special note is the decay rate of A, $d(\rho)$, which is a complicated function that is inversely related to the cell density ρ and the extracellular space available for diffusion.

We could simulate the system with some set of representative parameter values, but it is more interesting to visualize the switch in quorum sensing by examining P as a function of cell density ρ, that is, by stepping through values of $d(\rho)$ and examining the outcome. Specifically, using the authors' parameter values and gradually raising cell density values from initially 0.1 to 0.115 results in a corresponding decrease in d from 0.61 to 0.53. At first, we do not observe major changes in the steady-state level of the active complex P, and it hovers at 0.02. However, at cell density fractions higher than 0.115 ($d < 0.53$) an abrupt switch occurs with P jumping to a high value close to 10. This so-called **ultrasensitivity** to the parameter $d(\rho)$ occurs because the system has become bistable. In the terminology of Chapter 14, the nullclines of the system, which are lines representing the special situations of $\dot{R} = 0$ or $\dot{A} = 0$, now intersect three times, instead of two times for lower densities. Furthermore, the initial conditions used by the authors lie within a regime repelling the system away from the unstable steady state, and the system gravitates toward the high steady state. Thus, although lacking the spatial properties of a partial differential equation model, the much simpler ODE model is still capable of capturing the key feature of quorum sensing, namely that the accumulation of a critical mass of signal eventually triggers a new behavior in the system. The following case study illustrates this key message with a more detailed example.

Case Study: Bioluminescence

A fascinating example of quorum sensing is **bioluminescence** in the form of light emission by microbial populations once they reach a certain density. The phenomenon is driven by specific signaling molecules that can diffuse in and out of microbial cells. These molecules can bind to dedicated receptors, and this binding event in turn leads to changes in the expression of target genes that include, in particular, genes responsible for the production of the signaling molecule. Starting this positive feedback loop requires a certain concentration of the signaling molecule, which is only achieved if a quorum of the microorganisms is excreting the molecule.

A specific example of bioluminescence is the symbiosis between the bacterium *Vibrio fischeri* and cephalopods like the Hawaiian Bobtail Squid (*Euprymna scolopes*; **Figure 10.33**). *V. fischeri* lives in marine water but can enter the so-called light organ in the squid. The symbiosis suits both partners. The bacterium is protected from the elements and predators, while the squid uses the bacterium for camouflage. Namely, *Euprymna* is active at night, swimming close to the surface of

Figure 10.33 Hawaiian Bobtail Squid (Euprymna scolopes). As camouflage, the squid emits bioluminescence that is generated by the marine bacterium *Vibrio fischeri*. (Courtesy of Margaret McFall-Ngai.)

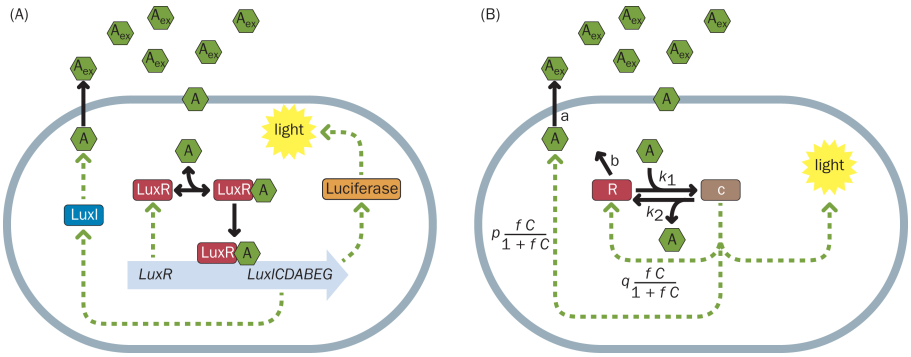


Figure 10.34 Molecular events in V. fischeri leading to fluorescence. (A) Typical biological notation (see Text for details). (B) Simplification for modeling with notation, as used by James and colleagues [60].

the water searching for prey. To counteract the shadow of its body from moon light onto the water below, a specific light organ, illuminated by *Vibrio*, emits light, thereby blending the view of the squid from below with the moonlit night.

How exactly does this bioluminescence work? Like everything in nature, the process is complicated in detail [52–58]. Nonetheless, skipping many of these details, there is a relatively simple chain of molecular events (**Figure 10.34A**). At the core are the *luxICDABEG* genes, which belong to an operon encoding the proteins necessary for light generation. The key protein is the enzyme luciferase, which releases light during the oxidation of a long-chain aldehyde (R-COH) and a reduced flavin mononucleotide ($FMNH_2$) [59]. At low cell densities, the *luxICDEBAG* operon is not active because it is not activated by the activated protein LuxR. At high cell density, however, the genes become expressed. Among them, the *luxI* gene encodes the synthase for an "autoinducer" molecule, OHHL (3-oxohexanoyl-L-homoserine lactone), which can diffuse in and out of the cell. At sufficient concentrations, OHHL forms a complex with LuxR, and this complex serves as a transcription factor for the *lux* operon. Expression of the *lux* genes leads to the generation of luminescence and also to the production of more OHHL.

To facilitate the modeling of this system within an individual *V. fischeri* cell, the diagram may be further simplified (Figure 10.34B), as proposed by James and colleagues [60]. With this simplification, we convert the diagram in Figure 10.34B into ODEs, supposing for now that the concentration of the external autoinducer OHHL, here called A_{ex}, is constant. According to James, the equations for the autoinducer A, receptor R and complex C read

$$\begin{aligned}\dot{A} &= k_2 C - k_1 A R - a(A - A_{ex}) + p\frac{fC}{1+fC}\\ \dot{R} &= k_2 C - k_1 A R - bR + q\frac{fC}{1+fC}\\ \dot{C} &= k_1 A R - k_2 C\end{aligned} \tag{10.15}$$

All lower-case letters are parameters. Note that the equation for A contains the term $-a(A - A_{ex})$, which is negative for values of A_{ex} that are small in comparison to A, and positive for large values. Thus, for low values, the autoinducer A primarily diffuses out, whereas it increases inside the cell for large values of A_{ex}, that is, if the outside concentration of the autoinducer is high because many bacteria are producing it. In addition to the complex association and disassociation events, which are modeled with the first two terms in each equation, the first two equations contain terms containing $\frac{fC}{1+fC}$. These collectively encompass the entire process of transcription and translation and thereby represent the overall indirect effect of C on A and R. This omission of important biology renders the model much simpler than it would otherwise be and appears to be legitimate as long as these processes are operating "normally."

Using the parameter values in **Table 10.6**, which were loosely adapted from [60], and starting the system with $A(0) = R(0) = C(0) = A_{ex} = 0.1$ shows that the system quickly settles down at a stable steady state of $(A, R, C)_{SS} = (0.1, 0, 0)$. In biological terms, there is not much happening if A_{ex} is small, even the small initial amounts of R and C decay. The result becomes more interesting if we increase the concentration of autoinducer in the medium, A_{ex}, for some time period, such $A_{ex} = 0.1$ (for $t < 10$), $A_{ex} = 1$ (for $10 \le t \le 50$), $A_{ex} = 0.1$ (for $t > 50$). The simulation result is shown in **Figure 10.35A**. After the switch at time 10, A equilibrates very quickly with the new value of A_{ex}, reaching a value of 1, but otherwise nothing much seems to be happening: R and C seem to stay at 0. However, closer inspection reveals that all three variables are creeping up very slowly until, without any further intervention, all three jump to much higher levels, before approaching a new stable steady state of about $(A, R, C)_{SS} = (2.33, 0.93, 2.88)$ around time $t = 40$. Biologically this means that, triggered by the increase in external autoinducer, the system ramps up for about 30 time units and then, all of the sudden, starts to luminesce.

Interestingly, if we later reset $A_{ex} = 0.1$, the system does not return to the initial steady state (0.1, 0, 0) even though the external concentration has returned to the original value. The model proposed by James [60] does not account for "turning off the lights." However, this stop is easy to capture in the model if we shut off transcription and translation by setting $p = q = 0$, for example, at time $t = 100$. With this hard switch, the system very quickly returns to the low steady state of $(A, R, C)_{SS} = (0.1, 0, 0)$ (Figure 10.35B).

These results clearly demonstrate that the system has two stable steady states for $A_{ex} = 0.1$, and theory tells us that there must be something unstable in between. We were able to explore the stable states with numerical simulations, but such an exploration is not so easy for unstable states. Can we instead compute the unstable state(s) analytically? The answer is yes. We set the three ODEs equal to zero and

TABLE 10.6 PARAMETER VALUES FOR THE ODE MODEL IN (10.15)

A_{ex}	k_1	k_2	a	b	f	p	q
0.1	16	12	10	4	1	30	5

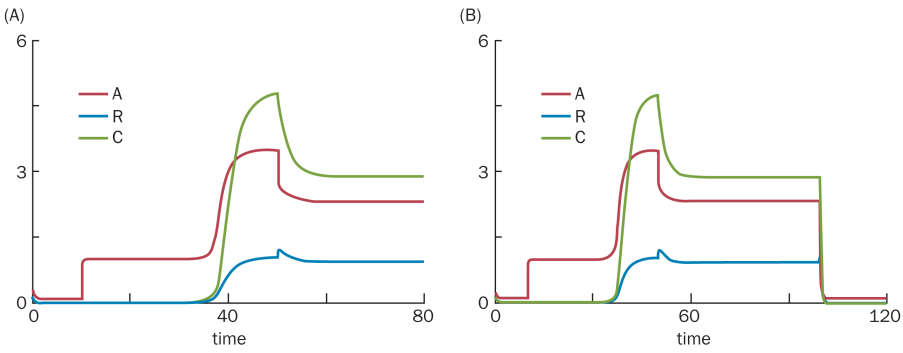


Figure 10.35 Virtual bioluminescence generated by the model in (10.15). (A) The system starts at $t = 0$ with A_{ex}, A, R, $C = 0.1$. For $t \in [10, 50]$, the external autoinducer concentration is increased to $A_{ex} = 1$. A adjusts quickly to a new value of 1, while R and C appear to remain constant. However, all three internal variables start increasing, barely noticeably, until they jump, overshoot, and then settle into a new steady state. Even if A_{ex} is subsequently returned to its original low value, the model keeps on bioluminsescing. (B) If transcription and translation are shut off, here at $t = 100$, bioluminescence stops, and the system quickly returns to its low, non-bioluminescing steady state.

solve for A, R, and C, which is possible in this case with simple algebra even though the system is nonlinear (see Exercises). This analysis yields all three steady states, including the unstable one. For $A_{ex} = 0.1$, they are:

$$
\begin{aligned}
(A, R, C)_{SS_1} &= (0.1, 0, 0) && \text{(stable)}\\
(A, R, C)_{SS_2} &= (2.326208734813, 0.92758697283875, 2.87701455802167) && \text{(stable)}\\
(A, R, C)_{SS_3} &= (0.773791265186999, 0.280746360494583, \\
&\quad 0.289652108644998) && \text{(unstable).}
\end{aligned}
$$

The many decimal places are given because the unstable state is rather fidgety: starting at any of these three states, the system remains there. The two stable states tolerate deviations. However, for the unstable state, any ever so slight perturbation causes the system to move to one of the stable states.

In the simulations so far, we switched A_{ex} from 0.1 to a higher value, quasi by fiat, which is not very natural. In reality, A_{ex} depends on many bacteria releasing the autoinducer into the medium. Can we capture this mechanism with the model? Yes, we need to make A_{ex} a dependent variable. For instance, we add to the earlier ODEs (10.15) the statement

$$\dot{A}_{ex} = 0.2\, A - 0.2\, A_{ex}^2 \tag{10.16}$$

which describes the collective autoinducer production by the population as well as decay or uptake by individual bacteria. Initiating this expanded system as before with $A(0) = R(0) = C(0) = A_{ex} = 0.1$, the external autoinducer concentration increases slowly, and at about $t = 30$, all variables quickly rise to an activated state of bioluminescence (**Figure 10.36**).

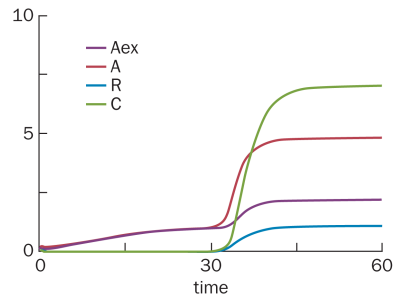


Figure 10.36 Model results for dynamic A_{ex}. A_{ex} is low at the beginning but accumulates due to release of autoinducer from the population of bacteria. Once a certain concentration has been reached, all variables quickly rise to a state corresponding to bioluminescence.

10.5.3 Cell-Free Systems

While it is not surprising that clocks in mammals are quite complex, it is intriguing that it has been possible to create simple circadian rhythms even in vitro, without transcription or translation [61]. All the self-sustained oscillator requires is a set of three proteins, called KaiA, KaiB, and KaiC, from the cyanobacterium *Synechococcus elongatus* and an energy source in the form of ATP. The simple system is even temperature-compensated, exhibiting the same oscillation period under temperature variations. In vivo, the Kai system is of course more intricate, and it has been demonstrated how very slow but ordered and fine-tuned enzymatic phosphorylation of the Kai proteins allows the cyanobacterium to sustain oscillations but also to respond appropriately to *zeitgebers* (timing signals) in the environment [62]. Specifically, KaiC has two phosphorylation sites and can therefore be present in four forms: unphosphorylated (U), phosphorylated only on a serine residue (S), phosphorylated only on a threonine residue (T), or phosphorylated on both sites (ST). These different forms cycle in different phases. KaiA enhances the autophosphorylation of KaiC, and KaiC dephosphorylates in the absence of KaiA. KaiB in turn impedes the activity of KaiA (**Figure 10.37**).

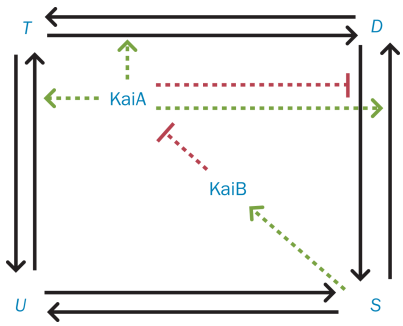


Figure 10.37 Schematic of the Kai oscillation system that is associated with circadian rhythms. Green arrows indicate activation and red lines with bar show inhibition. See the text for details. (Data from Rust MJ, Markson JS, Lane WS, et al. *Science* 318 [2007] 809–812. With permission from the American Association for the Advancement of Science.)

A conversion of the diagram into a kinetic model allows further analyses of the oscillator and of various perturbations and mutations [62]. Under baseline conditions, the model oscillates as shown in **Figure 10.38**. Using first-order kinetics but with a nonlinear influence of the KaiA concentration, the model of Rust and collaborators [62] reads

$$
\begin{aligned}
T' &= k_{UT}(S)U + k_{DT}(S)D - k_{TU}(S)T - k_{TD}(S)T, && T(0) = 0.68,\\
D' &= k_{TD}(S)T + k_{SD}(S)S - k_{DT}(S)D - k_{DS}(S)D, && D(0) = 1.36,\\
S' &= k_{US}(S)U + k_{DS}(S)D - k_{SU}(S)S - k_{SD}(S)S, && S(0) = 0.34,\\
A &= \max\{0, [\text{KaiA}] - 2S\},\\
k_{XY}(S) &= k_{XY}^0 + \frac{k_{XY}^A A(S)}{K + A(S)}.
\end{aligned}
\tag{10.17}
$$

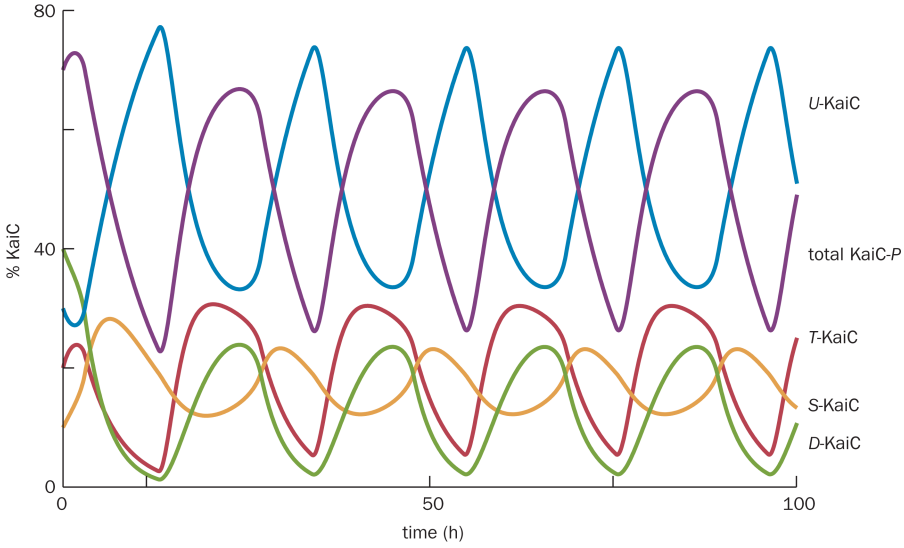


Figure 10.38 Oscillations of the Kai system in Figure 10.37. (From Rust MJ, Markson JS, Lane WS, et al. *Science* 318 [2007] 809–812. With permission from the American Association for the Advancement of Science.)

TABLE 10.7 RATE PARAMETER VALUES FOR THE CLOCK SYSTEM (10.17)

	UT	*TD*	*SD*	*US*	*TU*	*DT*	*DS*	*SU*
k^0	0	0	0	0	0.21	0	0.31	0.11
k^A	0.479077	0.212923	0.505692	0.0532308	0.0798462	0.173	−0.319385	−0.133077

Here, *D* refers to the double-phosphorylation state *ST*. Suitable kinetic parameter values are given in **Table 10.7**. The total concentration was taken as 3.4, and the amount of unphosphorylated KaiC was thus $U = 3.4 - T - D - S$. Furthermore, [KaiA] = 1.3 and $K = 0.43$. [62].

Maybe most intriguing is the precision of this relatively simple oscillator in the actual organism. In spite of asynchronous cell division, the clocks of a cell and its offspring run with very little deviation over several weeks, even in the absence of external cues. Many other processes in the organism are driven by the Kai oscillations, by virtue of rhythmic changes in the supercoiling state of the bacterial DNA and subsequent global changes in gene expression [63].

10.6 EXERCISES

10.1 A Simple Signaling System

State in words the functioning of the following signaling system, in which P_0 and P_1 are unphosphorylated and phosphorylated forms of a protein, respectively. Implement the systems (from scratch) in PLAS or some other software.

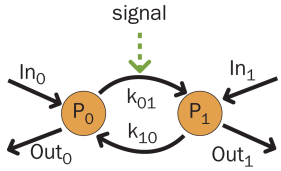


Figure 10.39 Simple signal transduction system. A protein P_0 is phosphorylated by a knase into P_1 if a strong enough signal is present. A phosphatase can dephosphrolyate P_1 back into P_0. Both forms of the protein may be generated (or transported into the system) and degraded (or transported out of the system).

10.1.1 The conversion of P_0 and P_1 is an enzymatic reaction catalyzed by a kinase. The signal is triggering the activity of this enzyme. Adjust the diagram to show these facts.

10.1.2 Initially set $In_0 = Out_0 = In_1 = Out_1 = 0$. What does this setting mean biologically?

10.1.3 Use simple mass action functions to model the above diagram. Model the effect of the signal as a multiplicative factor of the process $P_0 \rightarrow P_1$.

10.1.4 Pick parameters k_{01} and k_{10} such that the system is in a steady state for Signal = 1.

10.1.5 Determine, based on biological considerations, whether $P_0 \approx P_1$, $P_0 >> P_1$ or $P_1 >> P_0$ at the beginning of the experiment (initial values).

10.1.6 Start at the steady state with Signal = 1 and reset it at some time point to a much higher value, such as 50 or 100; solve the equations and interpret the results.

10.1.7 Set Signal back at a later time. What do you expect to happen? Test your expectation by solving the system.

10.1.8 Study different values of the Signal and their effect on P_0 and P_1 when the signal switches to a high value.

10.1.9 Divide k_{01} and k_{01} both by 100. Interpret the results.

10.1.10 Study what happens if the In and/or Out processes are not 0. Can the system still have a steady state? If so, under what conditions?

10.1.11 Assume that only P_0 is produced de novo and that P_0 and P_1 are degraded at the same rate. Model these assumptions with constants or power-law functions, as appropriate. Select values such that the system is in steady state if S = 1. Compare the signaling dynamics of the system with the case when $In_0 = Out_0 = In_1 = Out_1 = 0$ by exploring values for the rates of these processes that are low or relatively high with respect to k_{01} and k_{10}.

10.2 A Seemingly Weird Mechanism

The pathway below is called "incoherent," because following a signal (increase in S), the variable X_6 is affected both positively and negatively (check why that is).

10.2.1 Suppose the system is in a steady state (with $S = 1$; corresponding to no signal). After some while it receives a persistent positive signal S (e.g., $S = 10$). Predict what will happen to X_6 and to v_{out}. Write down your prediction.

10.2.2 Implement the pathway in PLAS as a dynamic model using mass action kinetics (see Chapters 5 and 9). Specifically, set all rate constants to 0.4 and all kinetic orders equal to 1 or 0, depending on the role of the variable of interest, with two exceptions: set the rate constants for the production of X_1 equal to 0.8 and choose –1 for the kinetic order representing the inhibition signal. Formulate v_{out} with the simple setting $v_{out} = 0.4\, X_6$. Test your prediction by running the code. If your prediction was wrong, try to explain why.

10.2.3 Suppose you continue the above simulation by resetting S to 1, maybe 100 time units after you had reset it to 10. Predict what will happen. Simulate and compare.

10.2.4 Change the value of the inhibition parameter (currently –1) first to 0 and then to +1 and study the responses. What do these changes mean biologically? Summarize your findings in a report that includes some simulation output.

10.3 A Bi-Fan Motif

Formulate the bifan below (see also Figure 3.15 in Chapter 3) as a set of ODEs. Model S_1 and S_2 as independent variables and T_1 and T_2 as dependent variables. Include in the equations of T_1 and T_2 a constant influx α_i and an efflux of the format $\beta_i T_i$ so that T_1 and T_2 without any signal have a steady-state value of 1. Model the combined input from S_1 and S_2 into T_1 and into T_2 as Hill functions and choose the model settings so that the bi-fan operates for T_1 as an AND-gate and for T_2 as an OR-gate. Initially set S_1 and $S_2 = 0$ and later change one or both to a value like 10. Confirm

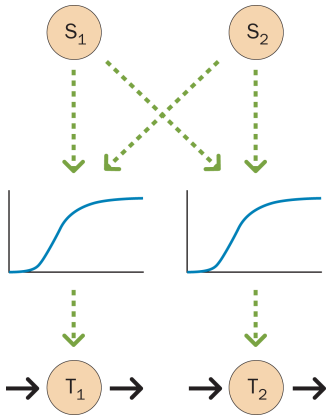


Figure 10.41 Bifan motif. The signals S_1 and S_2 go through a Hill-function transformation before affecting targets T_1 and T_2.

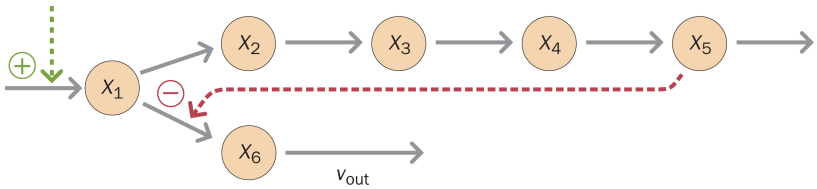


Figure 10.40 Incoherent pathway. If signal S is increased, X_6 eventually shows increase, due to an increased amount of X_1 and also decrease, due to inhibition of its production by X_4. How does that make sense?

with the responses that your gates operate appropriately. Why is it useful to formulate the signals with Hill functions?

10.4 Paveliev and collaborators [64] studied the effect of two signaling pathways on axonal growth (G). One is mediated through Cdk5 (C). It is triggered by laminin (L) but inactive if inhibited by roscovitine (R) The second pathway is mediated through SFK and requires both laminin and neurotrophic factors (NTFs; N); it is inactive if inhibited by the SFK inhibitor SU6656 (SU). The authors proposed a Boolean model and dynamic model, without actually showing their implementation. The two models are shown in **Figure 10.42**.

10.4.1 Develop Boolean tables for the activation status of Cdk5 and SFK. Combine the tables into a single overall table for G. Recall Chapter 3 for inspiration.

10.4.2 Develop a dynamic model of the system. Note that there are many different, correct implementations. The only criterion is whether G is activated under the correct conditions.

10.4.2.1 Which variables should be dependent and which independent?

10.4.2.2 Formulate symbolic equations for the dependent variables.

10.4.2.3 In order to minimize the effects of the numerical values of the various input variables, replace L, R, N, and SU with Hill terms of the type

$$HL = \frac{V L^4}{2^4 + L^4}.$$

In order to avoid problems with division by zero, add 0.01 to HR and HSU.

10.4.2.4 Select values for all unknown parameters. You may adjust these as you wish, as long as the signaling cascade "fires" under the correct conditions.

10.4.2.5 Simulate the equations with different combinations of L, N, R, and SU.

10.4.2.6 Compare the results for G with the results of the Boolean model.

10.5 Revisit Exercise 7.36, which asked you to model a logic circuit with Boolean gates. In the exercise, a synthetic biologist was designing a logic gate circuit consisting of two OR, AND, NOR, and/or NAND gates, G_1 and G_2 (recall Figure 7.42). The system was supposed to generate the following input–output behavior: If the two inputs S_1 and S_2 were *on* but the third input S_3 was *off*, there should be no output. Similarly, if S_3 was *on* and both S_1 or S_2 were *off*, there should be no output. Now, instead of using Boolean methods, construct a model of differential equations yielding this behavior. You do not have to consider transcription or translation, but you should attempt switch-like properties using the tools developed in this chapter. Determine what happens if S_1 and S_3 are *off* but S_2 is *on*? What happens if S_1 is *off* but S_2 and S_3 are *on*?

10.6 Review what is known about switches between lysis and lysogeny in bacteriophages. Write a one-page summary report.

10.7 Discuss the stability of all steady states of the differential equation

$$\dot{X} = 2X \times \left(4 - X^2\right)$$

from a mathematical and a biological point of view, assuming that X is some biological entity. Discuss what happens if the multiplier 2 is replaced with –2.

10.8 Perform simulations with the bistable system in (10.2) to find out in which range(s) the system is most sensitive to noise. Interpret your findings.

10.9 Construct a new "bi-unstable" system with two unstable states and a stable state in between. Demonstrate the responses of the system to perturbations similar to those shown in Table 10.1 and Figure 10.3.

10.10 Demonstrate with simulations that the model in Eqn. (10.3) is bistable.

10.11 Implement the two-variable hysteretic system in (10.3–10.4) and explore how strong a perturbation must be to move the response incorrectly from off to on. Study persistent and temporary perturbations. Explain your findings.

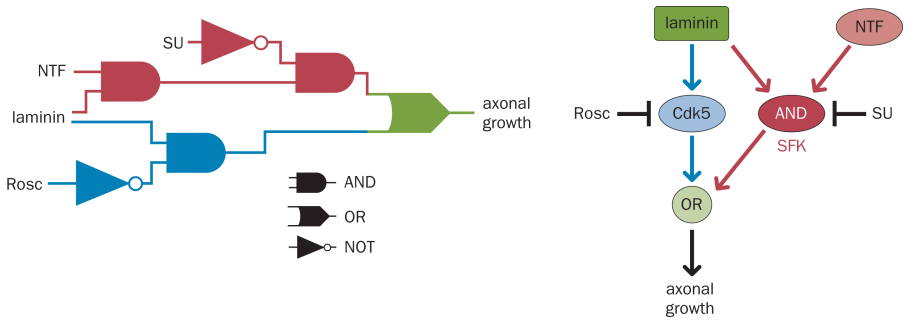


Figure 10.42 Boolean and dynamic models of axonal growth activation. The pathways are triggered by laminin and NTF and may be inhibited by R and SU, as explained in the text.

10.12 Create a sequence of signals that are corrupted by noise. Study its effects on the bistable system in (10.3–10.4).

10.13 Test whether the bifunctional variant of (10.5) also exhibits bistability and hysteresis.

10.14 Igoshin and collaborators [15] emphasized the importance of the phosphatase *Ph* in the bifunctional model. Explore bistability and hysteresis if there is no phosphatase *Ph*. For this purpose, use the same model as in the text, but set $k_{b2} = 0.5$ and set the initial value of *Ph* equal to zero. Repeat the experiment with values of *Ph* between 0 and 0.17. Summarize your findings in a report.

10.15 Study the responses of the two TCS systems (mono- and bifunctional) to repeated brief signals.

10.16 Implement the MAPK cascade shown in Figure 10.21 and study the effects of noise on the signal–response relationship.

10.17 Explore the responses of a MAPK cascade with parameter values that are closer to reality (*cf*.[30, 32, 33]), namely $k_1 = k_4 = 0.034$, $k_2 = k_5 = 7.75$, $k_3 = k_6 = 2.5$, $k_7 = k_8 = 1$; $X(0) = 800$, $Y(0) = 1000$, $Z(0) = 10{,}000$, and $E = 1$; set all other initial values to 0.

10.18 Explore the effects of increasing the rate of phosphorylation at the three layers. Use either the model in the text or the model in Exercise 10.14.

10.19 Study bistability and hysteresis in the MAPK cascade. Can a single module, as shown in Figure 10.17, be bistable and/or hysteretic? Is the complete MAPK cascade bistable? Is it hysteretic? Investigate these questions with simulations. Summarize your answers in a report.

10.20 Create a small pathway model with two sequential reactions that are each modeled as regular Michaelis–Menten rate laws. Compare its responses with those of the double-phosphorylation system at the center and bottom layers of the MAPK cascade, where the two phosphorylation steps compete for the same, limited amount of enzyme.

10.21 Investigate what happens if the MAPK cascade receives repeated brief signals. Discuss your findings for different sizes and spacings of signals.

10.22 Investigate what happens if the phosphatases in the MAPK cascade have very low activity.

10.23 Kholodenko [61] discussed the consequences of potential negative feedback, in which MAPK inhibits the activation of MAPKKK. Implement such a feedback mechanism in the model of the MAPK cascade discussed in the text and study the implications of different strengths of this feedback.

10.24 The MAPK model, as used in the text, assumes that *XP* and *YPP* are constant. Study the consequences of making them dependent on their next lower level in the cascade.

10.25 Is three the magic number for MAPK cascades? Perform simulations comparing the physiological three-layer cascade with (A) a two-layer cascade in which XP activates YPP only and (B) a four-layer cascade in which XP → YPP → ZPP → APP. In your comparison, evaluate signal amplification and robustness against noise as the features of interest.

10.26 Derive conditions on the parameter values in (10.10) under which the system exhibits adaptation. Assume that the initial steady state is (1, 1, 1), that the signal is permanently increased from 1 to 1.2, and that the output variable C approaches the steady-state value of 1.02. Solve the system for the steady state and derive the conditions.

10.27 Explore the numerical features and functions in Eqn. (10.14) and their effect on the limit-cycle clock.

10.28 Use Eqn. (10.15) to analyze biologically relevant scenarios. To get started, consult the original literature. Document your findings in a report.

10.29 Compute the red portion of the hysteresis curve in Figure 2 of Box 10.1 by formulating S as a function of X_{SS}.

10.30 Compute the steady states of the quorum sensing system in 10.15. Assume that all parameters are strictly positive.

10.30.1 Compute the steady state for arbitrary, non-negative A_{ex}.

$$\dot{A} = k_2 C - k_1 A R - a(A - A_{ex}) + p\frac{fC}{1+fC}$$

$$\dot{R} = k_2 C - k_1 A R - bR + q\frac{fC}{1+fC}$$

$$\dot{C} = k_1 A R - k_2 C$$

Check the results against simulations for $A_{ex} = 0.1$.

10.30.2 Is the system bistable for any positive value of A_{ex}?

10.30.3 Is there bistability for $A_{ex} = 0$?

FURTHER READING

Bhalla US & Iyengar R. Emergent properties of networks of biological signaling pathways. *Science.* 283 (1999) 381–387.

Ferrell Jr, JE. Self-perpetuating states in signal transduction: Positive feedback, double-negative feedback and bistability. *Curr Opin Cell Biol.* 14(2) (2002) 140–148.

Hasty J, Pradines J, Dolnik M & Collins JJ. Noise-based switches and amplifiers for gene expression. *Proc Natl Acad Sci USA.* 97 (2000) 2075–2080.

Huang C-YF & Ferrell JEJ. Ultrasensitivity in the mitogen-activated protein kinase cascade. *Proc Natl Acad Sci USA.* 93 (1996) 10078–10083.

Igoshin OA, Alves R & Savageau MA. Hysteretic and graded responses in bacterial two-component signal transduction. *Mol Microbiol.* 68 (2008) 1196–1215.
Jordan JD, Landau EM & Iyengar R. Signaling networks: The origins of cellular multitasking. *Cell.* 103 (2000) 193–200.
Kauffman SA. *Origins of Order: Self-Organization and Selection in Evolution*. Oxford University Press, Oxford, UK, 1993.
Kholodenko BN, Hancock JF & Kolch W. Signalling ballet in space and time. *Nat Rev Mol Cell Biol.* 11 (2010) 414–426.
Klingmüller U, Bauer A, Bohl S, et al. Primary mouse hepatocytes for systems biology approaches: A standardized in vitro system for modelling of signal transduction pathways. *Syst Biol (Stevenage).* 153(6) (2006) 433–447.
Ma'ayan A, Blitzer RD & Iyengar R. Toward predictive models of mammalian cells. *Annu Rev Biophys Biomol Struct.* 34 (2005) 319–349.
Sachs K, Perez O, Pe'er S, et al. Causal protein-signaling networks derived from multiparameter single-cell data. *Science.* 308 (2005) 523–529.
Schwacke JH & Voit EO. Concentration-dependent effects on the rapid and efficient activation of MAPK. *Proteomics.* 7 (2007) 890–899.
Tyson JJ, Csikasz-Nagy A & Novak B. The dynamics of cell cycle regulation. *Bioessays.* 24 (2002) 1095–1109.
Utsumi RE. *Bacterial Signal Transduction: Networks and Drug Targets*. Landes Bioscience, Austin, TX, 2008.

REFERENCES

1. Ma'ayan, A., R.D. Blitzer, and R. Iyengar, Toward predictive models of mammalian cells. *Annu Rev Biophys Biomol Struct*, 2005. **34**: p. 319–349.
2. Cheresh, D.E., *Integrins*. Methods in Enzymology. Vol. 426. 2007, New York: Academic Press.
3. Monod, J. and F. Jacob, General conclusions—teleonomic mechanisms in cellular metabolism, growth, and differentiation. *Cold Spring Harbor Symp Quant Biol*, 1961. **28**: p. 389–401.
4. Novick, A. and M. Weiner, Enzyme induction as an all-or-none phenomenon. *Proc. Nat. Acad. Sci. USA*, 1957. **43**: p. 553–566.
5. Tian, T. and K. Burrage, Stochastic models for regulatory networks of the genetic toggle switch. *Proc Nat Acad Sci USA*, 2006. **103**(22): p. 8372–8377.
6. Igoshin, O.A., R. Alves, and M.A. Savageau, Hysteretic and graded responses in bacterial two-component signal transduction. *Mol Microbiol*, 2008. **68**(5).
7. Savageau, M.A., Design of gene circuitry by natural selection: Analysis of the lactose catabolic system in *Escherichia coli*. *Biochem Soc Trans*, 1999. **27**: p. 264–270.
8. Tan, C., P. Marguet, and L. You, Emergent bistability by a growth-modulating positive feedback circuit. *Nature Chem Biol*, 2009. **5**(11): p. 942–948.
9. Das, J., et al., Digital signaling and hysteresis characterize Ras activation in lymphoid cells. *Cell*, 2009. **136**(2): p. 337–351.
10. Yao, G., et al., A bistable Rb-E2F switch underlies the restriction point. *Nat Cell Biol*, 2008. **10**(4): p. 476–482.
11. Hasty, J., et al., Noise-based switches and amplifiers for gene expression. *Proc Natl Acad Sci U S A*, 2000. **97**(5): p. 2075–2080.
12. Wu, J. and E.O. Voit, Hybrid modeling in biochemical systems theory by means of functional petri nets. *J Bioinf Comp Biol*, 2009. **7**(1): p. 107–134.
13. Bhattacharya, S., et al., A bistable switch underlying B cell differentiation and its disruption by the environmental contaminant 2, 3, 7, 8-tetrachlorodibenzo-p-dioxin. *Toxicol Sci*, 2010. **115**(1): p. 51–65.
14. Zhang, Q., et al., Stochastic modeling of B lymphocyte terminal differentiation and its suppression by dioxin. *BMC Syst. Biol.*, 2010. **4**(1): p. 40.
15. Jiang, P. and A.J. Ninfa, Regulation of autophosphorylation of *Escherichia coli* nitrogen regulator II by the PII signal transduction protein. *J Bacteriol*, 1999. **81**(6): p. 1906–1911.
16. Utsumi, R.E., *Bacterial Signal Transduction: Networks and Drug Targets*. Advances in Experimental Medicine and Biology. Vol. 631. 2008, Austin, TX: Landes Bioscience.
17. Bourret, R.B. and A.M. Stock, Molecular information processing: Lessons from bacterial chemotaxis. *J Biol Chem*, 2002. **277**(12): p. 9625–9628.
18. Zhao, R., et al., Structure and catalytic mechanism of the *E. coli* chemotaxis phosphatase CheZ. *Nat Struct Biol*, 2002. **9**(8): p. 570–575.
19. Alves, R. and M.A. Savageau, Comparative analysis of prototype two-component systems with either bifunctional or monofunctional sensors: Differences in molecular structure and physiological function. *Mol Microbiol*, 2003. **48**: p. 25–51.
20. Bongiorni, C., R. Stoessel, and M. Perego, Negative regulation of *Bacillus anthracis* sporulation by the Spo0E family of phosphatases. *J Bacteriol*, 2007. **189**: p. 2637–2645.
21. Batchelor, E. and M. Goulian, Robustness and the cycle of phosphorylation and dephosphorylation in a two-component system. *Proc Natl Acad Sci U S A*, 2003. **100**: p. 691–696.
22. Schoeberl, B., et al., Computational modeling of the dynamics of the MAP kinase cascade activated by surface and internalized EGF receptors. *Nat Biotechnol*, 2002. **20**: p. 370–375.
23. Pouyssegur, J., V. Volmat, and P. Lenormand, Fidelity and spatio-temporal control in MAP kinase (ERKs) signalling. *Biochem Pharmacol* 2002. **64**: p. 755–763.
24. Thattai, M. and A. van Oudenaarden, Attenuation of noise in ultrasensitive signaling cascades. *Biophys J*, 2002. **82**(6): p. 2943–2950.
25. Bluthgen, N. and H. Herzel, How robust are switches in intracellular signaling cascades? *J Theor Biol*, 2003. **225**(3): p. 293–300.
26. Huang, C.-Y.F. and J.E.J. Ferrell, Ultrasensitivity in the mitogen-activated protein kinase cascade. *Proc Natl Acad Sci U S A*, 1996. **93**: p. 10078–10083.
27. Bhalla, U.S. and R. Iyengar, Emergent properties of networks of biological signaling pathways. *Science*, 1999. **283**(5400): p. 381–387.
28. Schwacke, J.H., *The Potential for Signal Processing in Interacting Mitogen Activated Protein Kinase Cascades*. 2004, Doctoral Dissertation. Charleston, SC: Medical University of South Carolina.
29. Schwacke, J.H. and E.O. Voit, Concentration-dependent effects on the rapid and efficient activation of MAPK. *Proteomics*, 2007. **7**(6): p. 890–899.
30. Kholodenko, B.N. and M.R. Birtwistle, Four-dimensional dynamics of MAPK information processing systems. *Wiley Interdiscip Rev Syst Biol Med*, 2009. **1**(1): p. 28–44.
31. Muñoz-García, J. and B.N. Kholodenko, Signalling over a distance: Gradient patterns and phosphorylation waves within single cells. *Biochem Soc Trans*, 2010. **38**(5): p. 1235–1241.

32. Whitmarsh, A.J. and R.J. Davis, Structural organization of MAP-kinase signaling modules by scaffold proteins in yeast and mammals. *Trends Biochem Sci*, 1998. **23**(12): p. 481–485.
33. Hanahan, D. and R.A. Weinberg, The hallmarks of cancer. *Cell*, 2000. **100**(1): p. 57–70.
34. Jordan, J.D., E.M. Landau, and R. Iyengar, Signaling networks: The origins of cellular multitasking. *Cell*, 2000. **103**: p. 193–200.
35. Kumar, N., et al., Multipathway model enables prediction of kinase inhibitor cross-talk effects on migration of Her2-overexpressing mammary epithelial cells. *Mol Pharmacol*, 2008. **73**(6): p. 1668–1678.
36. Pritchard, J.R., et al., Three-kinase inhibitor combination recreates multipathway effects of a geldanamycin analogue on hepatocellular carcinoma cell death. *Mol Cancer Ther*, 2009. **8**(8): p. 2183–2192.
37. Ma, W., et al., Defining network topologies that can achieve biochemical adaptation. *Cell*, 2009. **138**(4): p. 760–773.
38. Nakatsui, M., et al., Method for inferring and extracting reliable genetic interactions from time-series profile of gene expression. *Math Biosc*, 2008. **215**: p. 105–114.
39. Davidich, M.I. and S. Bornholdt, Boolean network model predicts cell cycle sequence of fission yeast. *PLoS One*, 2008. **3**(2).
40. Tyson, J.J., A. Csikasz-Nagy, and B. Novak, The dynamics of cell cycle regulation. *Bioessays*, 2002. **24**: p. 1095–1109.
41. Tyson, J.J. and B. Novak, Temporal organization of the cell cycle. *Curr Biol*, 2008. **18**(17): p. R759–R768.
42. Sveiczer, A., J.J. Tyson, and B. Novak, A stochastic, molecular model of the fission yeast cell cycle: Role of the nucleocytoplasmic ratio in cycle time regulation. *Biophys Chem*, 2001. **92**(1–2): p. 1–15.
43. Vu, T.T. and J. Vohradsky, Inference of active transcriptional networks by integration of gene expression kinetics modeling and multisource data. *Genomics*, 2009. **93**(5): p. 426–433.
44. Matsuno, H., et al., Hybrid petri net representation of gene regulatory network. *Pac Symp Biocomput*, 2000. **5**: p. 341–352.
45. To, C.C. and J. Vohradsky, Supervised inference of gene-regulatory networks. *BMC Bioinform*, 2008. **9**.
46. Js, S., Aktivität und Ruhe bei Tieren und Menschen. *Zeitschrift für allgemeine Physiologie*, 1919. **18**: p. 509–517.
47. Cermakian, N. and D. Bolvin, The regulation of central and peripheral circadian clocks in humans. *Obes Rev*, 2009. **10**: p. 25–36.
48. Fuhr, L., et al., Circadian systems biology: When time matters. *Comput Struct Biotechnol J*, 2015. **13**: p. 417–426.
49. Ferrell, J.J., T.Y.C. Tsai, and Q. Yang, Modeling the cell cycle: Why do certain circuits oscillate? *Cell*, 2011. **144**: p. 874–885.
50. Elowitz, M.B. and S. Leibler, A synthetic oscillatory network of transcriptional regulators. *Nature Chem Biol*, 2000. **403**: p. 335–338.
51. Fuqua, W.C., S.C. Winans, and E.P. Greenberg, Quorum sensing in bacteria: The LuxR-LuxI family of cell density-responsive transcriptional regulators. *J Bacteriol*, 1994. **176**(2): p. 269–275.
52. Norsworthy, A.N. and K.L. Visick, Gimme shelter: How *Vibrio fischeri* successfully navigates an animal's multiple environments. *Frontiers Microbiol*, 2013. **4**(Article 356): p. 1.
53. Bramhachari, P.V., ed. *Implication of Quorum Sensing System in Biofilm, Formation and Virulence*. 2018, Singapore: Springer Nature.
54. Hooshangi, S. and W.E. Bentley, From unicellular properties to multicellular behavior: bacteria quorum sensing circuitry and applications. *Curr Opin Biotechnol*, 2008. **19**(6): p. 550–555.
55. Kuttler, C. and A.G. Maslovskaya, Wave effects in stochastic time lagging reaction-diffusion model of quorum-sensing in bacterial populations, in *2020 Days on Diffraction (DD)*. 2020, New York, NY: IEEE Publishing, p. 62–67.
56. Pérez-Velázquez, J., M. Gölgeli, and R. García-Contreras, Mathematical modelling of bacterial quorum sensing: A review. *Bull Math Biol*, 2016. **78**: p. 1585–1639.
57. Dockery, J.D. and J.P. Keener, A mathematical model for quorum sensing in *Pseudomonas aeruginosa*. *Bull Math Biol*, 2001. **63**(1): p. 95–116.
58. Miyashiro, T. and E.G. Ruby, Shedding light on bioluminescence regulation in *Vibrio fischeri*. *Mol Microbiol*, 2012. **84**(5): p. 795–806.
59. Schaefer, A.L., D.L. Val, and B.L. Hanzelka, Generation of cell-to-cell signals in quorum sensing: acyl homoserine lactone synthase activity of a purified *Vibrio fischeri* LuxI protein. *Proc Natl Acad Sci U S A*, 1996. **93**(18): p. 9505–9509.
60. James, S., et al., Luminescence control in the marine bacterium *Vibrio fischeri*: An analysis of the dynamics of lux regulation. *J Mol Biol*, 2000. **296**(4): p. 1127–1137.
61. Nakajima, M., et al., Reconstitution of circadian oscillation of cyanobacterial KaiC phosphorylation in vitro. *Science*, 2005. **308**(5720): p. 414–415.
62. Rust, M.J., et al., Ordered phosphorylation governs oscillation of a three-protein circadian clock. *Science*, 2007. **318**(5851): p. 809–812.
63. Woelfle, M.A., et al., Circadian rhythms of superhelical status of DNA in cyanobacteria. *Proc Natl Acad Sci U S A*, 2007. **104**(47): p. 18819–18824.
64. Paveliev, M., et al., Neurotrophic factors switch between two signaling pathways that trigger axonal growth. *J Cell Sci*, 2007. **120**(Pt 15): p. 2507–2516.

Taylor & Francis
Taylor & Francis Group
http://taylorandfrancis.com

Multiscale Modeling and Agent-Based Simulations

11

When you have read this chapter, you should be able to:

- Describe responses of biological systems at different timescales and their importance for modeling
- Discuss the challenges of bridging spatial or temporal scales in computational models
- Describe methodologies for simulating complex multi-scale systems
- Provide rationale for representing spatial and stochastic phenomena with either discrete or continuous systems models
- Understand the fundamentals of agent-based modeling strategies
- Set up models that span cellular-level mechanisms to larger scales, such as tissue-, organ- or population-level behavior
- Implement and interpret simple spatial and stochastic models generated with agent-based platforms

In the model examples provided throughout this text so far, we largely examined biological systems in which all processes are occurring at a single organizational level, such as the conversion of metabolites within an organelle or the aging of individuals within a population. This implicit restriction was in part due to the convenience of keeping illustrative examples simple but, more importantly, it reflects a major challenge for setting up mathematical models in systems biology. Namely, many biological systems involve phenomena at different scales in time and space. As a consequence, processes that change very rapidly may need to have a very fine temporal resolution to capture their dynamic behavior adequately, whereas slower events can be validly approximated by longer time steps. Similarly, events that happen at the molecular level will often need a higher granularity of description compared to events that occur at the organismal level. How do we simultaneously capture different time and length scales that span more than one or two orders of magnitude?

11.1 THE CHALLENGE OF MULTIPLE SCALES

In Chapter 2, we first introduced the concept of multi-scale behavior. Figure 2.3 described different spatial scales in biology that range from nanometers to kilometers, reflecting 12 orders of magnitude. Along the y-axis, the figure showed vastly different timescales that often correspond with the spatial scales. **Figure 11.1** shows another example focusing on the bridging between molecular and cellular scales.

DOI: 10.1201/9781003402770-11

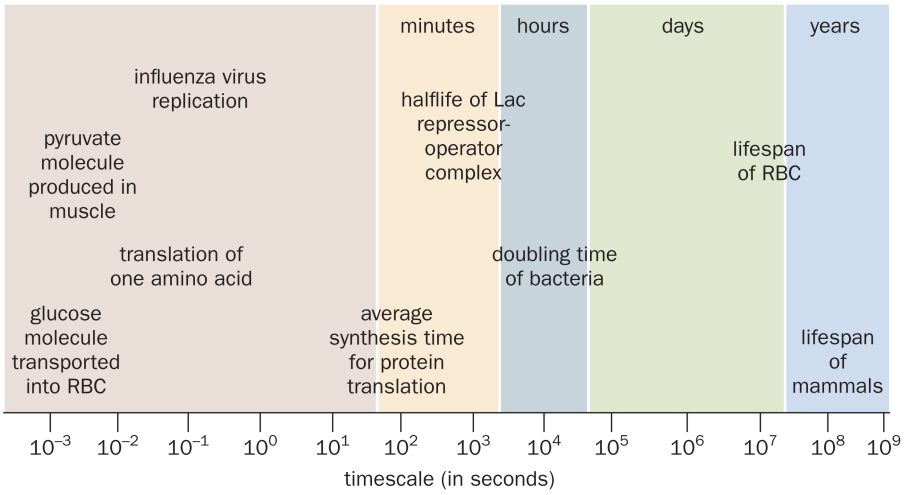


Figure 11.1 Timescales of typical molecular and cellular events. Values were obtained or calculated from bionumbers.org.

While most molecular activities used as examples in this book fall in the subsecond range, the time it takes for macromolecular, cellular, physiological, or disease processes to complete can span hours, if not days or years. For instance, human red blood cells take up and release oxygen very quickly, but they live for about 120 days. White blood cells live more than one year on average, and some neurons last an entire human lifetime. The function and failing of the human heart span many scales, both in space and time (see Chapter 12).

Developing models for interacting processes on different time scales is a true challenge. One approach is the **separation of time scales** which, if feasible, simplifies modeling tremendously. The argument is that processes occurring over very long time scales often do not change much during the action of a fast process and that very fast processes are essentially always in a steady state (*e.g.*, [1–3]). Both situations suggest that all involved differential equations approximately equal zero and therefore become algebraic equations that are much easier to address. A feasible example of time scale separation is the clouding of the lens in the human eye. This clouding is caused by biochemical reactions that proceed quickly in vitro, but their cumulative effect only manifests over a lifetime [4]. In other cases, such as the function of the heart, biochemical and macroscopic events cannot really be separated by time scales (see Chapter 12). Numerous articles have discussed this issue, but there is no silver-bullet solution yet [5–7].

When we evaluate potential computational descriptions of multiscale properties, whether for time and/or space, the first step is often an identification of common processes (functions) or species (variables) that may serve as linkages between scales. This identification helps us refine the level of detail and minimize computational costs for simulation. As an illustration, we begin with a small case study that demonstrates that it is often problematic to slice biological systems for analysis into isolated levels of organization. For instance, we will see that focusing solely on the genomic or the metabolic level masks important clues as to how the system as a whole is organized. The case study also shows that even apparently simple systems require a lot of information regarding their components, processes, and other aspects, and that it is difficult to discern a priori whether particular details can be omitted without much loss or whether they provide clues for important features of the function and regulation of the system. Putting the simplicity of this case study and others in this chapter into perspective, one will easily imagine how any multi-level system becomes much more complicated when the range of scales is widened, and we will discuss several approaches to handling this situation; see also Chapter 12.

11.1.1 Case Study: How Yeast Deals with Heat

The typical baker's yeast, *Saccharomyces cerevisiae*, loves a temperature between 25°C and 35°C. If the temperature of the growth medium is changed to somewhere between 35°C and 45°C, the cells need to respond, because some of their internal

structures are at risk of breaking. To avoid this problem, several molecular events are triggered (for reviews, see [8–11]). These events are very well organized and interconnected, but it is not obvious what controls their coordination. A quick foray into the literature shows that there is a fine but important distinction between heat stress (about 35°C) and heat shock (over 45°C), with the first usually referring to increased heat conditions that can still be considered physiological and the latter referring to temperatures in which the cells barely survive.

To study the heat stress response in greater detail, we start with the most visible events and then discuss how these are triggered and coordinated. Maybe the most astounding response happens at the metabolic level: it is the production of the disaccharide trehalose, which essentially consists of two glucose molecules chemically bound together (see later). Under normal physiological growth conditions, yeast cells contain merely trace concentrations of trehalose and its immediate precursor trehalose 6-phosphate (T6P). However, once the temperature rises above 37°C, the amount of trehalose may rise to an amazing concentration of up to 1 gram of trehalose per gram protein or 40% of the entire cell biomass [12]. Intriguingly, three enzymes in the pathway seem to exist just for the response to heat and some other stresses: those producing trehalose (trehalose 6-phosphate synthase and trehalose 6-phosphate phosphatase) have their optimal catalytic activity at temperatures greater than 37°C, while the trehalose-degrading enzyme trehalase is most active at cooler temperatures (**Figure 11.2**). Mutants lacking the ability to form trehalose or overproducing trehalase are very sensitive to environmental stresses, including heat, which indicates that the dramatic trehalose production is apparently vital for an effective response. So, what does trehalose actually do for the cell? Research has revealed that it protects proteins against denaturation and reduces the formation of unnatural protein aggregates. It also preserves the integrity of DNA, lipids, and membranes. Imagine packing all valuables into Styrofoam during times of trouble. Beyond protecting macromolecules and membranes, the trehalose pathway is involved in the control of glucose utilization by counteracting the accumulation of free cytoplasmic glucose and sugar phosphates, which would otherwise result in the depletion of cytoplasmic phosphate and ATP.

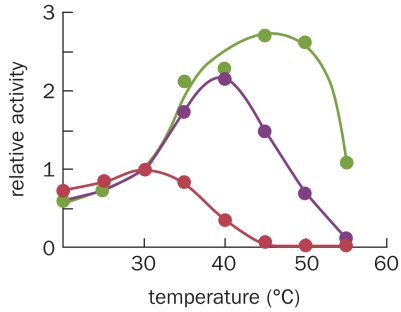


Figure 11.2 Effect of temperature on activities of enzymes producing (green) and degrading (red) trehalose. Trehalose 6-phosphate synthase: light green; trehalose 6-phosphate phosphatase: dark green; trehalase: red. Symbols represent experimental data reproduced from [13]; lines are smoothed, interpolating trends. (Data from Neves MJ and Francois J. *Biochem. J.* 288 [1992] 859–864.)

An entirely different metabolic response is the dramatic activation of specific lipids, called sphingolipids, which have been implicated in a variety of fundamental control tasks at the cellular level, including differentiation, apoptosis, and, most notable here, the response to various stresses including heat [14, 15]. Within a few minutes of heat stress, key enzymes of the pathway of sphingolipid biosynthesis exhibit increased activity. Several intermediates of the pathway increase several-fold in quantity, and some remain elevated for two hours or more. The role of sphingolipids in the response to heat stress appears to be twofold. First, sphingolipids are important components of plasma membranes, and because the stress response requires the transport and mobilization of proteins and metabolites, the additional quantities of sphingolipids might be required to sustain membrane integrity and facilitate increased transport. Second, sphingolipids serve as second messengers in signal transduction processes. In the example of heat stress, sphingolipid metabolites appear to be the first responders: they sense the increasing temperature and trigger an appropriate response by activating the transcription of a number of genes, including some of those associated with trehalose synthesis [16].

At the genome level, the response to heat is widespread, and many gene products are upregulated within a few minutes [17]. Not surprisingly, almost all genes coding for enzymes associated with the trehalose pathway are upregulated, but two details are intriguing. First, the degree of up-regulation differs dramatically, spanning a range from essentially unchanged to over 100-fold. Second, the genes coding for the trehalose-degrading enzymes, called *trehalases*, are robustly upregulated as well (Figure 11.3). Why would that happen if the cell needs large amounts of trehalose? We will return to this puzzling question later.

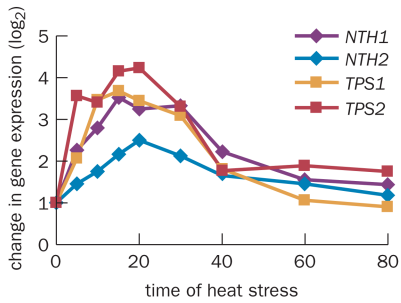


Figure 11.3 Heat stress induced changes in the expression of genes directly associated with trehalose. *TPS1*: T6P synthase; *TPS2*: T6P phosphatase; and *NTH1/2*: trehalase. *TPS2* rises to a level of $2^{4.24}$ =18.9, which means an almost 20-fold increase in gene expression over normal temperature conditions about 20 minutes after heat stress begins.

One should also note that some genes are strongly downregulated. Examples are ribosomal protein transcripts. In fact, there is a transient decrease in the initiation of translation from mRNAs to proteins, which is coupled with transient cell cycle and growth arrest. These findings suggest that the combined effects of the decrease in transcript and protein synthesis reduce biosynthetic costs while the cell adapts to stress [8, 18]. In other words, the cells prioritize heat survival to population growth.

Another response is the mobilization of transcription factors, such as the zinc finger proteins Msn2p and Msn4p, which control the expression of most of the enzymes of carbon metabolism, as well as antioxidant defense proteins of the heat stress response. These responses appear to be obligatory, because *MSN*2 / *MSN*4 double mutants are unable to accumulate trehalose under heat conditions and must suffer the consequences. Interestingly, the importance of this mobilization mechanism is underscored by in-built redundancy: failure in one of the transcription factors is compensated by the other.

The translocation of Msn2p and Msn4p from the cytosol to the nucleus occurs within minutes of heat stress and has been considered the rate-determining step in the activation of the heat stress response. In the nucleus, Msn2p and Msn4p interact with stress response elements (STREs), which are short stretches of DNA found in some of the genes associated with the trehalose cycle, among others and are needed for transcription.

The resulting upregulated proteins are involved in stress responses as well [19]. Several dozen yeast proteins are transiently induced by heat stress. Roughly a third of them are heat shock proteins, which mediate protein-protein interactions and the transport of proteins through membranes. They are apparently involved in the recovery from stress and, in particular, in the correct refolding of proteins that had unfolded due to the heat. Some of these proteins respond very strongly. For instance, heat shock protein Hsp90 is induced 10- to 15-fold. Of particular importance is also Hsp104, which seems to be associated with accumulating trehalose and with the removal of denatured and unsalvageable protein fractions, for instance through the proteasome (see [20] and Chapter 8). In addition to heat shock proteins, several enzymes are activated. This activation appears to be crucial in the metabolic heat stress response, which includes—but is not limited to—the huge increase in trehalose in response to strong rises in temperature.

This brief summary indicates that the heat stress response is very important to the cell and that its control spans all hierarchical levels of biological organization. As reflected in Figure 11.1, metabolic fluxes are capable of changing within milliseconds whereas transcriptional and translational changes will take on the order of half an hour. A strategy to model the properties of this system could consist of just adding the observed temperature-dependent changes in key enzymes to model the trehalose synthesis pathway; however, this approach would ignore the fact that a significant component of the heat stress response, including the production of enzymes, is controlled at the genomic level. In contrast, concentrating on gene expression alone would lead to some insights in regulation but suffer from

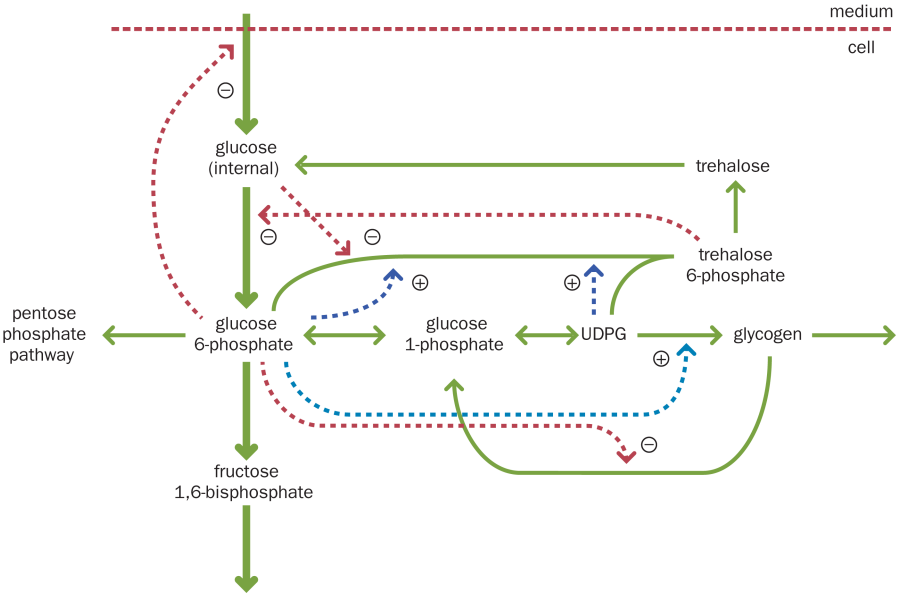


Figure 11.4 Regulation of the trehalose cycle. Green arrows show flow of material; thicker arrows indicate the main flux. Red and blue arrows indicate inhibition or activation, respectively. (Adapted from Voit EO. *J. Theor. Biol.* 223 [2003] 55–78. With permission from Elsevier.)

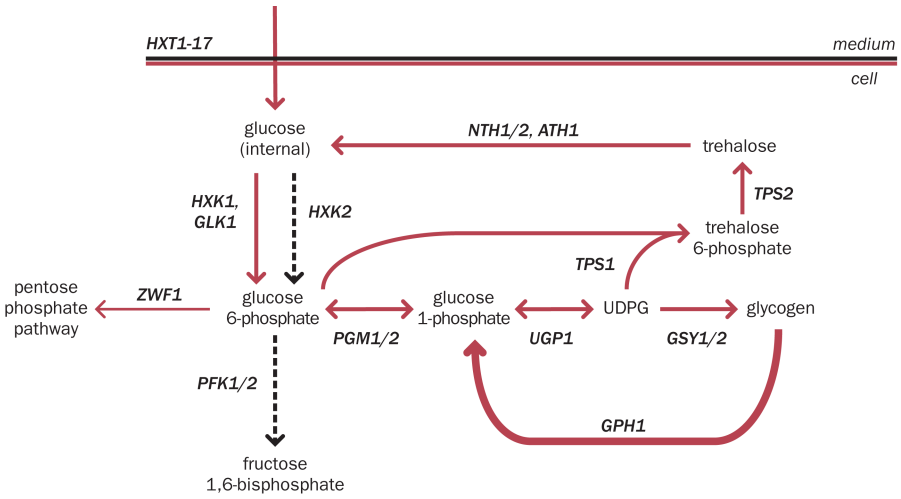


Figure 11.5 Changes in the expression of genes associated with the trehalose cycle in response to heat stress. Different thicknesses indicate degrees of gene expression, while striped arrows represent no change. (Adapted from Voit EO. *J. Theor. Biol.* 223 [2003] 55–78. With permission from Elsevier.)

the uncertainties surrounding the assumption that gene expression can be converted directly and quantitatively into enzyme activities, and flux measurements over time are not easily available. Ideally, we would set up a multi-scale model that would incorporate the most critical features at each level of genetic, proteomic, and metabolic control to capture the mechanisms that are intertwined at distinct timescales.

11.1.2 Models of Some Aspects of the Heat Stress Response

Let's design a model that integrates these findings and possibly explains them. We choose the generalized mass action (GMA) format because it is straightforward to set up and essentially unbiased, yet very flexible (see Chapter 5). For our purposes here, we will make the separate roles of genes and enzymes explicit. The model should address four scenarios. The dynamic responses to heat stress should reflect the history of the cells, because it was found that their metabolic responses were much stronger if the yeast had been preconditioned by heat stress earlier in life, in comparison to cells that were "naïve." We will model control (C) and preconditioned (P) scenarios, both at 30°C and 39°C. Our hypothesis is that the switch from 30°C to NTH1/2, TPS1, TPS2 for either C or P is due to heat-induced changes in the activities of some of the enzymes. A sub-hypothesis is that changes in the three temperature-sensitive enzymes (NTH1/2, TPS1, TPS2; Fig 11.2 and [13]) are sufficient, while the alternative sub-hypothesis postulates that more than these three enzymes must change in activity to launch the observed response.

The change from C to P is complicated, because the governing mechanisms have not been identified experimentally. Nonetheless, we can hypothesize that heat preconditioning affects the changes in the regulation of pertinent genes, which in turn have a longer-term effect on the amounts or activities of the corresponding enzymes. To test this hypothesis with the model, we introduce into each power-law term of the model a factor representing the amount of enzyme. Under normal conditions, this factor is set to 1, while it takes a so-far unknown value for heat-preconditioned cells. Thus, each term in the model equations contains, as always, a rate constant and the contributing variables, raised to appropriate kinetic orders (see Chapter 5), and in addition a factor for heat-induced changes in enzyme activity, as well as a factor for preconditioning.

The estimation procedure for obtaining parameter values is not at all trivial but can be derived from experimental time-series (see Chapter 6). Because we are at this point not interested in algorithmic aspects, we just skip these issues and focus on the results (see [21] for details). The model has a GMA structure with the symbolic formulation in Eqn. (11.1).

$$\frac{dX}{dt} = \begin{cases} -F_1/V_{ext} \\ \left(F_1 + 2\cdot F_{11} - F_2\right)/V_{\text{int}} \\ \left(F_2 + F_4 - F_3 - F_9 - F_{12} - F_{15}\right)/V_{\text{int}} \\ \left(F_3 - F_4 + F_6 - F_5 + F_8\right)/V_{\text{int}} \\ \left(F_5 - F_6 - F_7 - F_9\right)/V_{\text{int}} \\ \left(F_7 - F_8\right)/V_{\text{int}} \\ \left(F_9 - F_{10}\right)/V_{\text{int}} \\ \left(F_{10} - F_{11}\right)/V_{\text{int}} \\ \left(F_{12} - F_{13} - F_{14}\right)/V_{\text{int}} \\ 2\cdot F_{13}/V_{ext} \\ F_{14}/V_{\text{int}} \\ F_{15}/V_{\text{int}} \end{cases} \qquad F = \begin{cases} B\cdot\gamma_1\cdot\tau^{P_1}\cdot X_1^{h_1}\cdot X_3^{hr_1}\cdot Q_1^{\frac{T-30}{10}} \\ B\cdot\gamma_2\cdot\tau^{P_2}\cdot X_2^{h_2}\cdot X_7^{hr_2} \\ B\cdot\gamma_3\cdot\tau^{P_3}\cdot X_3^{h_3}\cdot Q_3^{\frac{T-30}{10}} \\ B\cdot\gamma_4\cdot\tau^{P_4}\cdot X_4^{h_4} \\ B\cdot\gamma_5\cdot\tau^{P_5}\cdot X_4^{h_5} \\ B\cdot\gamma_6\cdot\tau^{P_6}\cdot X_5^{h_6} \\ B\cdot\gamma_7\cdot\tau^{P_7}\cdot X_5^{h_7}\cdot X_3^{hr_3} \\ B\cdot\gamma_8\cdot\tau^{P_8}\cdot X_6^{h_8}\cdot X_3^{hr_4}\cdot X_5^{hr_5} \\ B\cdot\gamma_9\cdot\tau^{P_9}\cdot X_3^{h_9}\cdot X_5^{h_{10}}\cdot X_2^{hr_6}\cdot Q_9^{\frac{T-30}{10}} \\ B\cdot\gamma_{10}\cdot\tau^{P_{10}}\cdot X_7^{h_{11}}\cdot Q_{10}^{\frac{T-30}{10}} \\ B\cdot\gamma_{11}\cdot\tau^{P_{11}}\cdot X_8^{h_{12}}\cdot Q_{11}^{\frac{T-30}{10}} \\ B\cdot\gamma_{12}\cdot\tau^{P_{12}}\cdot X_3^{h_{13}} \\ B\cdot\gamma_{13}\cdot\tau^{P_{13}}\cdot X_9^{h_{14}}\cdot Q_{13}^{\frac{T-30}{10}} \\ B\cdot\gamma_{14}\cdot\tau^{P_{14}}\cdot X_9^{h_{15}} \\ 0.05\cdot F_{12} \end{cases} \tag{11.1}$$

The left side of this formulation shows the differential equations, where, for instance, the first row is to be read as $\frac{dX_1}{dt} = -F_1 / V_{ext}$. The right side shows all fluxes that contain the usual rate constants γ and the dependent variables X_i with their kinetic orders, h and hr, that characterize substrate dependence and regulatory influences, respectively. The equations furthermore contain several other quantities, namely: V_{ext} is the volume of the suspension, in which the yeast cells live (50 ml), V_{int} is the total intracellular volume (7.17 ml), and B is the biomass in the reactor (3,013 mg and 2,410 mg of dry weight under normal temperature and heat, respectively). The factors τ^{P_i} represent changes in protein amounts due to altered gene expression. We set $\tau = 1$ for cells grown under control conditions, while we set $\tau = 2$ for heat-preconditioned cells, so that all τ^{P_i} correspond to powers of 2, as it is typical in gene expression studies. The values of the different P_i were estimated from the measured time series together with the other parameters. The temperature dependence of each affected enzyme is explicitly modeled as $Q_i^{\frac{T-30}{10}}$. Each Q_i is a typical temperature coefficient (Q_{10}) for enzymatic reaction i, which depends on the actual ambient temperature T (either 30°C or 39°C) and represents the change in enzymatic activity brought about by a 10°C increase in temperature [22].

The model with parameters presented in Table 11.1 fit the data well, under both normal temperature and upon heat preconditioning (Figure 11.6). In addition to this validation, we can use the model to explore the questions we posed. First, are the heat-induced changes in enzyme activities, which we discussed earlier, sufficient for an adequate heat stress response? The answer is *yes* and *no*. It is *no* in a sense that the observed changes in the trehalose associated enzymes (TPS1, TPS2, and NTH1/2) alone are not sufficient to explain the observed trehalose response, because the model indicates a much weaker response than observed (**Figure 11.7**). However, the answer is a conditional *yes* if the measured changes in these enzymes are accompanied by a moderate reduction in glucose uptake (to about 60% hexose transporter activity), an (~25%) increase in phosphoglucomutase and a slighter (~25%) collective increase in glycolytic activity, as shown in Figure 11.6. Interestingly, in vitro activity assays of glycolytic enzymes show similar trends [21].

The data analysis with the model also lets us make predictions regarding alterations in protein amounts due to preconditioning. No direct data are available for comparisons. However, as we discussed earlier in the chapter, one may translate alterations in gene expression into amounts of protein, at least as a crude substitute [23, 24]. For the trehalose system, the comparison of the quantitative changes

TABLE 11.1: MODEL PARAMETERS (RATE CONSTANTS AND KINETIC ORDERS) AND PROTEIN CHANGES OBTAINED FROM TIME-SERIES DATA BY MEANS OF OPTIMIZATION*

Flux	*Model step*	*Rate constant γ*	*Kinetic orders for substrates*†	*Kinetic orders for regulators*‡	*Fold change* (2^{P_i})	Q_i
1	HXT	2.87×10^{-5}	0.526 (30°C) 0.472 (39°C)	−0.002 (G6P)	0.7	1.57
2	HXK	1.90×10^{-4}	0.510	−0.209 (Tre6P)	9.2	
3	PGM^F	5.66×10^{-6}	0.400		20.7	1.48
4	PGM^R	3.13×10^{-5}	0.471		17.3	
5	UGP^F	3.58×10^{-5}	0.767		16.2	
6	UGP^R	1.31×10^{-5}	0.159		26.0	
7	GSY	9.43×10^{-7}	0.459	0.000 (G6P)	0.9	
8	GPH	6.94×10^{-8}	0.311	−0.002 (G6P) −0.001 (UDPG)	61.8	
9	TPS1	1.19×10^{-6}	0.659 (G1P) 0.625 (UDPG)	0.000 (Glc)	21.5	2.48
10	TPS2	3.24×10^{-6}	0.361		14.2	2.35
11	NTH	1.99×10^{-7}	0.082		4.9	0.42
12	PFK	2.89×10^{-5}	0.693		1.0	
13	"FBA"§	6.13×10^{-5}	0.369		1.2	1.26
14	Leakage	5.54×10^{-6}	0.672		4.1	
15	ZWF	1.45×10^{-7}	0.693		1.0	

* From Fonseca L, Sánchez C, Santos H & Voit EO. Complex coordination of multi-scale cellular responses to environmental stress. *Mol. BioSyst.* 7 (2011) 731–741. With permission from The Royal Society of Chemistry.

† The temperatures in parentheses show the temperature associated with the kinetic order for glucose transport or the substrate associated with each kinetic order.

‡ Each abbreviation in parentheses shows the regulating metabolite.

§ "FBA" designates the collection of enzymatic steps between fructose 1,6-bisphosphate aldolase and the release of end products.

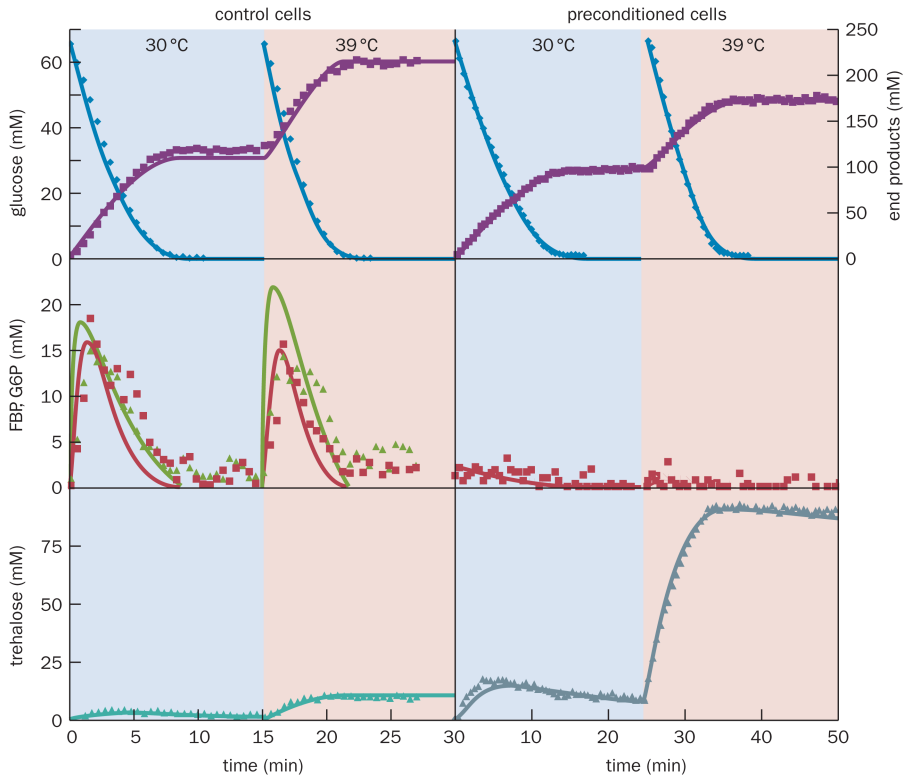


Figure 11.6 Experimental time courses (symbols) and model fits for key representatives of the trehalose cycle. The dark blue symbols and curves in the top row show glucose in the medium, while the burgundy symbols and curves show all end-products combined. Their concentration is much higher than that of glucose, because they are trioses (3-carbon sugar), while glucose is a hexose (6-carbon sugar). The profiles of FBP (red curves and symbols in the center row) are dramatically different for C and P. G6P (green curves and symbols in the center row) was not measured for the preconditioned cells, but the profile is expected to mirror that of FBP. Note the striking differences in trehalose production between C and P (bottom row). Also note the different time scales for C and P, which clearly indicate that the control cells take up glucose more efficiently. (Adapted from Fonseca L, Sanchez C, Santos H & Voit EO. *Mol. BioSyst.* 7 [2011] 731–741. With permission from The Royal Society of Chemistry.)

in protein levels with literature data is quite favorable (Table 11.2), even though they had been obtained with entirely different methods.

Table 11.2 exhibits one striking difference, namely in glycogen synthase. In the time series (bolus) experiments, the activity of this enzyme is slightly decreased,

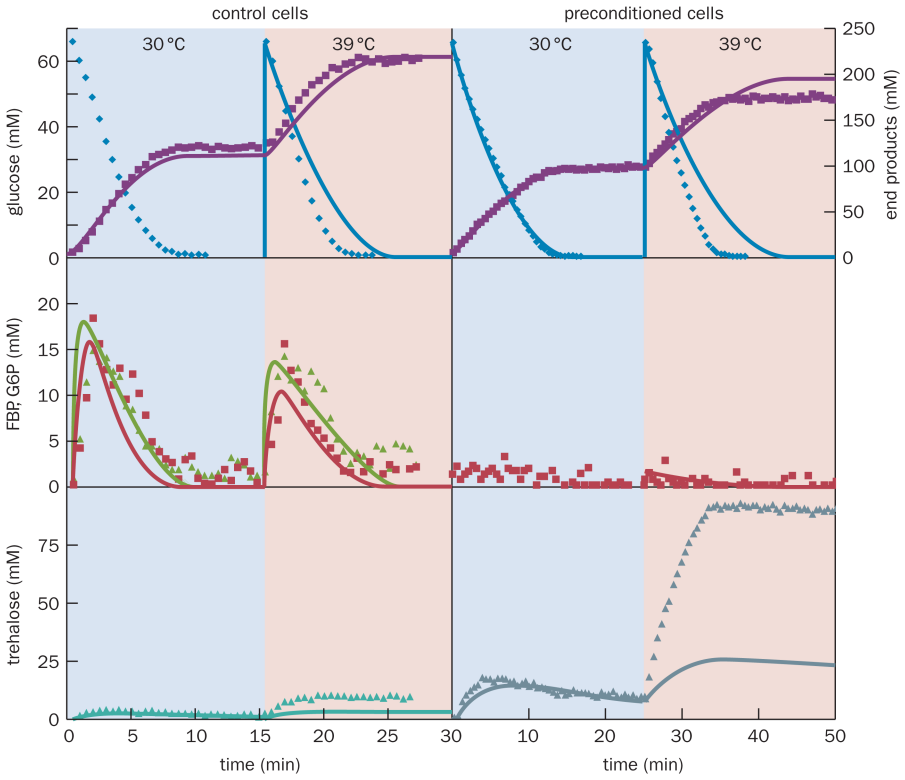


Figure 11.7 Testing the assumption that exclusively TPS1, TPS2, and NTH1/2 are affected by heat (35). Experimental time courses (symbols) and model fits for key representatives of the trehalose cycle, under the assumption that only the three enzymes TPS1, TPS2, and NTH1/2 are affected by heat. Symbols and colors are the same as in Figure 11.16. (Adapted from Fonseca L, Sanchez C, Santos H & Voit EO. *Mol. BioSyst.* 7 [2011] 731–741. With permission from The Royal Society of Chemistry.)

TABLE 11.2: CHANGES IN ENZYME AMOUNTS BETWEEN NORMALLY GROWN AND PRECONDITIONED CELLS, COMPARED WITH HEAT-INDUCED CHANGES IN MRNA LEVELS REPORTED IN THE LITERATURE

Enzyme	*Change in amount inferred from time data**		*Change in mRNA†*
	Forward reaction	**Reverse direction**	
Hexokinase	9	—	8
Phosphoglucosemutase	21	17	16
UDPG pyrophosphorylase	16	26	16
Glycogen synthase	0.9	—	16
Glycogen phosphorylase	—	62	50
T6P synthase	21	—	12
T6P phosphatase	14	—	18
Neutral trehalase	5	—	6
Phosphofructokinase	1	—	1

* Expressed as ratio between protein amount after preconditioning and protein amount in normally grown cells.
† Data from [2].

while it is strongly increased under constant glucose conditions. The reasons for this difference are unclear, but it could be that the constant glucose supply was more than the cells needed at the time, therefore allowing them to store some of the material as glycogen. Also, one must keep in mind that the constant glucose experiments were performed with growing cells under steady-state conditions, while the dynamic experiments used resting cells. Finally, the inferred change in FBP used for non-glycolytic pathways, which in the time series experiments increased about 4-fold, has no analogue in the mRNA study.

Summarizing the results, the model seems to capture the responses of the yeast cells under different conditions quite well. In particular, the model shows that heat-induced changes in enzyme activities are sufficient to mount a short-term

response, but that preconditioning makes such a response much stronger and more effective. Furthermore, the cellular adjustments between heat conditions and between normally grown and preconditioned cells are so far not validated but seem to be reasonable.

One may ask why there are such stark differences in expression levels (Table 11.2). At least partial answers can be obtained by mathematically analyzing alternative operating strategies, as discussed in Chapter 6, which the cells could apply to reach the observed levels of trehalose and glycogen [25].

There is clear evidence that the genes coding for trehalase (*NTH1/2* and *ATH1*) are immediately and strongly upregulated under heat stress. This up-regulation seems at first counterintuitive, because it suggests an increase in the degradation of a metabolite that is clearly needed. We have also seen that the activity of trehalase is directly reduced by heat. Thus, there are concurrent but apparently counteracting changes in response to heat stress that are occurring on different timescales. One way to reconcile the opposing changes is to have a closer look at the timing of these processes. The direct reducing effect of heat on the activity of the enzyme very quickly allows the necessary accumulation of trehalose. After about 20 minutes for transcription and translation, the increased gene expression results in a higher amount of enzyme, which apparently compensates for the heat-induced reduction in enzyme activity and keeps trehalose production and degradation in balance. More importantly, as soon as the heat stress ceases, the heat-induced reduction in activity stops, while the amount of enzyme is still increased. This combination of a large amount of enzyme with uninhibited active allows the cell to remove trehalose very quickly. This removal of unneeded trehalose is apparently crucial for a healthy reentry into normal physiological operation [26, 27] and is reflected in our multiscale model via Q_{10} coefficients (fast) and changes in protein levels (slow).

11.1.3 Case Study: SIR Modeling Revisited

This second case study revisits a series of models first discussed in Chapter 2. It is designed to demonstrate how one may bridge scales in a manner where a model at the cellular level is embedded directly within the equations of a population-level model. In the present case, we only consider rather small models at two levels, and the timescales differ just by about one order of magnitude, but the example will nevertheless give an impression of the challenges that larger models, spanning wider differences in scale, often face.

In Chapter 2, the concept of SIR and SIRS modeling was introduced as a means of describing the dynamics of an infectious disease spreading through a population. To recall, the population was stratified into susceptible (S), infected (I) and "recovered" (R) individuals; the latter typically referring to having gained immunity. Most processes were represented with linear terms and the key interaction between susceptible (S) and infected (I) individuals was very simplistically modeled as mass action kinetics representing a black-box mechanism of interaction, analogous to molecules colliding and forming complexes. In spite (or because?) of its simplicity, the model has been used thousands of times since its first use a century ago [28].

At the population level, the coarse description of infection, recovery, and death is sufficient for capturing the overall trends among subgroups of people. However, what would we need to do if we wanted to account for known details of the infection process that triggers a battle between pro-inflammatory and anti-inflammatory components of the host's immune system that clear the pathogen from the body? This task requires us to bridge scales from immune cells within an individual to the dynamics of an entire human population. This discrepancy causes a key challenge: the molecular and cellular events in viruses occur at a timescale between sub-seconds and hours, whereas the spread of the infection in a population easily spans weeks and months. If the virus sub-model requires a time resolution of 0.001 seconds, solving the system for a month would require $1{,}000 \times 60 \times 60 \times 24 \times 30 \approx 2.6$ billion steps. The number per se is no real problem for modern computers, but it is easy to see how larger models with greater time discrepancies can quickly get out of hand.

11.1.4 Accounting for the Host Response

To explore the impact of viral replication and clearance, we adapt the concepts proposed by Almocera *et al.* [29] and apply them to our SIRS system from Chapter 2 (Eqns. 2.6) by accounting for processes occurring in individuals of the infected (I) state (**Figure 11.8**). Specifically, we consider the introduction of viral particles into a susceptible host upon infection. The transition of a susceptible person into the infected state results in the reproduction and secretion of new virions. We formulate this aspect by describing the growth of the virus population over time with the logistic function, which we already saw in Chapter 5 and will discuss in more detail in Chapter 14. This function has the mathematical format $pV\left(1-\frac{V}{K_v}\right)$, where V is the size of the viral population, p is its growth rate, and K_v is the final size of the population, barring perturbations; K_v is often called the carrying capacity (see Chapter 14). For simplicity, we will write this function for our analysis as $p_1V - p_2V^2$.

The presence of virus triggers immune cell activation. Thus, the virus is sensed and subsequently removed by effector T cells, whose population we call E. We assume that this population is normally in a dynamic steady state and, if perturbed, grows back or decreases back to K_e according to the logistic function $qE\left(1-\frac{E}{K_e}\right)$, which is of the same type as the growth function for viruses. Like the logistic growth term for the virus population, we will reframe the growth of the immune cell population as $q_1E - q_2E^2$. When triggered by virus, the production of E is increased, and the effector cells remove virus at a higher rate (Figure 11.8). Eventually, the rate of virus removal outcompetes the replication rate with which new virus is added to the viral pool. This process mirrors the infection and recovery that is globally described at the population level. As a result, all viruses may be eliminated, or E and V assume a new steady state with $V > 0$, which corresponds to a chronic infection. Often this infection is "sub-threshold," meaning that it did not entirely disappear but is not severe either.

The clearance of virus depends on both V and E; we define the rate of this process as c_v. Furthermore, we must quantify the triggering of the effector cell population, which is driven by the presence of virus. We choose a Michaelis–Menten description as a function of V, that is, we define $c_G = r_G \cdot V/(K_G + V)$. Clearly, other options could be considered.

This competitive infection-recovery dynamics of V and E can now be incorporated into the SIRS model as its own set of ODEs, which we call the "VE module." The module zooms into the infection process and thereby refines the state of infected individuals (I) into a level of higher granularity [29]. Within compartment I, the virus is leading to an influx of effector immune cells (E) that are recruited to suppress further viral production. To connect the VE module to the original SIRS model, the infection rate rI is transformed from a static parameter into a dynamic expression $r_I(V)$ that is a function of the level of virus V within compartment I. For simplicity of notation, we omit the explicit dependence on V and retain the terminology of r_I. The rate can take on different forms. For example, Almocera *et al.* [29] explored the use of a simple linear relation, $r_I = \alpha \cdot V$, a Michaelis–Menten relation between viral load and infection rate, $r_I = r_W \cdot V/(K_W + V)$, and a sigmoidal relation between virus and infectivity in the form of a Hill function, $r_I = r_W \cdot V^2/(K_W{}^2 + V^2)$. We use the Michaelis–Menten format.

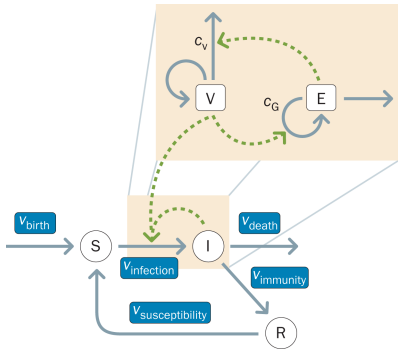


Figure 11.8 Implementation of the combined "two-scale" SIRS-VE model. The original SIRS system is expanded by zooming in on the infection process (yellow shading), which is now explicitly driven by virus particles and effector cells. The flux $v_{infection}$ is now regulated by r_I, which is a function of V. The virus level V is controlled by logistic self-replication growth and clearance by T cells (E), The T cell population is increased in the presence of virus. V and E are driven by processes associated with rates c_G (virus-induced T cell activation) and c_v (viral clearance per T cells). (Adapted from Almocera AES and Hernandez-Vargas EA, *Math. Biosci.* 309 [2019] 34–41. With permission from Elsevier.)

11.1.5 Design of a Two-Scale Model

Taking all these aspects together, the system including the SIRS and the VE modules is modeled by Eqns. (11.2) and (11.3):

$$\begin{aligned}
\dot{S} &= r_B - r_I S \cdot I + r_S R - r_{DS} S \\
\dot{I} &= r_I S \cdot I - r_R I - r_{DI} I \\
\dot{R} &= r_R I - r_S R \\
r_I &= \frac{r_W V}{V + K_W}
\end{aligned} \tag{11.2}$$

$$\begin{aligned}\dot{V} &= p_1 V - p_2 V^2 - c_v E \cdot V \\ \dot{E} &= q_1 E - q_2 E^2 + c_G \cdot E \\ c_G &= \frac{r_G V}{V + K_G}\end{aligned} \tag{11.3}$$

Recall that r_I and c_G are explicit functions of V, rather than ODEs.

We implement the two-scale model for SIRS and VE in three steps. The first just requires setting up the original SIRS model from Chapter 2 again. The second is the coding of the VE sub-module, and the third step consists of connecting the two parts by replacing rI in the SIRS model with the dynamic term $rI(V)$. Of course, we need to define numerical values for all parameters. For the SIRS module, we use the same values as in Chapter 2. The only difference is that we change the initial value to an entirely healthy population: $(S, I, R) = (1000, 0, 0)$. For the VE module and the connecting function $rI(V)$, we use the following values:

$$\begin{aligned}\dot{S} &= 3 + 0.01\,R - r_I(V)\,S\,I \\ \dot{I} &= r_I(V)\,S\,I - 0.005\,I - 0.02\,I \\ \dot{R} &= 0.05\,I - 0.01\,R \\ \dot{V} &= 100\,V - 1.67\cdot 10^{-4} V^2 - 2.4\cdot 10^{-5}\,E\,V \\ \dot{E} &= 0.4\,E - 4\cdot 10^{-7} E^2 + 6.5\frac{V}{V + 3\cdot 10^3} E \\ r_I(V) &= \begin{cases} 0.0005 & \textit{Original SIRS Model} \\ 0.024\dfrac{V}{V + 10^5} & \textit{Two-Scale Model} \end{cases}\end{aligned} \tag{11.4}$$

Note that the VE module in this set-up happens to be unaffected by the SIRS model. Thus, we can simulate it by itself. Assuming healthy conditions, we initiate the module with $V = 0$ and E at its healthy steady state, which is 1,000,000 (see Exercises).

It is now easy to run a simulation. For instance, suppose the first infection occurs at $t = 2$, which we can model by setting $I = 1$ and $V = 1{,}000$. Suppose further that a new strain of virus hits the population again at later times and with different severity (Table 11.3); generically, we call this type of one-time input a *bolus*.

As E and V are large populations, it is useful to define logarithms as $LE = \log(E)$ and $LV = \log(V + 1)$. We use $V + 1$ instead of V, because the difference is insignificant in important situations where V is large, but the setting avoids numerical problems if the virus is eliminated so that V becomes 0. The approximation itself is immaterial in our case, as we use LE and LV exclusively for plotting and not for computing the dynamics of the system.

Numerical integration yields a result as in **Figure 11.9**. We see that the effector cells successfully reduce the virus after each bout in a damped oscillatory manner. In some cases, a new virus emerges before the effector cells have completed the removal of the previous strain. We also notice that the population of effector cells ultimately reaches a new steady state, along with a non-zero titer of V. Note that the dynamics of the module in this case looks the same whether simulated within the larger multiscale SIR model or as an isolated two-variable ODE system (E, V). This equivalence is due to the unidirectional coupling of the virus/effector dynamics with the population system; in other words, the dynamics is passed from E and V to the next scale without any dynamic feedback from the levels of S, I, or R.

Integrating the two sub-models, the SIRS model and the VE module, into a single two-scale model results in complex behavior over time, which is hard to predict but can be easily demonstrated with simulations. Namely, we simply replace

TABLE 11.3: BOLUSES OF VIRUS INFECTION AT DIFFERENT TIME POINTS

t	2	5	6	20
$V(t)$	1,000	10^5	10^4	2×10^5

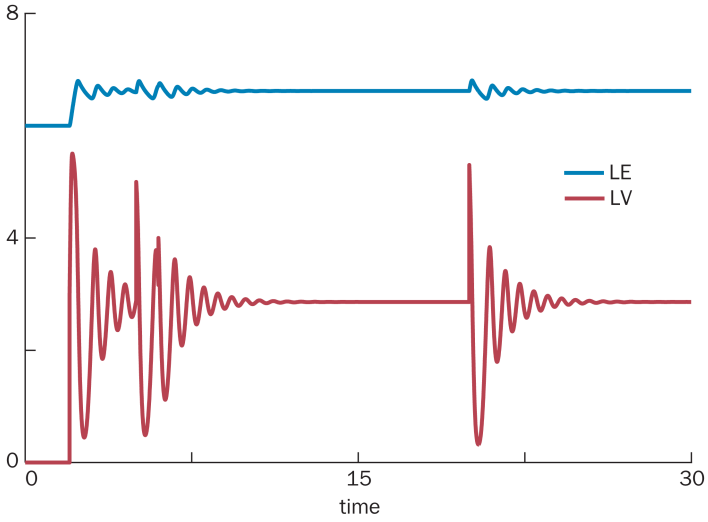


Figure 11.9 Trends in the VE module. The interactions between *V* and *E* can be tested without the rest of the model because the SIRS model in this specific implementation does not elicit feedback on the VE module. *V* and *E* are expressed in logarithms. The infections occur at the time points given in the text.

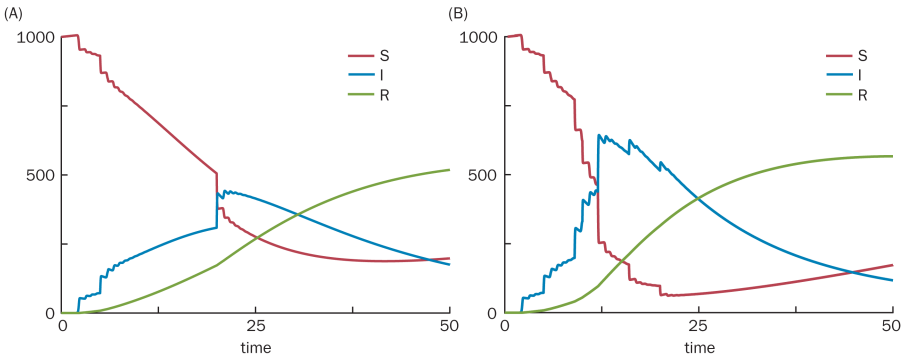


Figure 11.10 Trends in S, I, and R in the integrated two-scale model. (A) Results corresponding to the trends in *E* and *V* as shown in Figure 11.9 and generated by inputs from Table 11.3. (B) If the population is hit with more infections (Table 11.4), the trends in *S*, *I*, and *R* are strongly affected. In particular, more people become infected. The infections occur at the times given in the text.

TABLE 11.4: INCREASED NUMBER OF BOUTS OF VIRUS AT DIFFERENT TIME POINTS

t	2	5	6	9	10	12	16	20
V(*t*)	1,000	10^5	10^4	10^5	10^5	10^6	10^5	2×10^5

the constant infection rate r_I of the original SIRS model with $r_I(V)$, as indicated earlier, and we can immediately see the trends in *S*, *I*, and *R* that correspond to the virus dynamics in Figure 11.9. They are shown in **Figure 11.10A** for the same settings.

It turns out that these trends are strongly affected by the infection scenario. For instance, if we introduce additional bouts of infection (compare Tables 11.3 and 11.4), quite a few more individuals become infected, which makes sense (Figure 11.9B).

Overall, the simulations demonstrate in granular resolution how the within-host viral dynamics of repeated infections is passed to the population model. The damped oscillations in virus titer ripple through the SIRS system to generate erratic waves of newly infected individuals in population *I* and, concomitantly, in the number of susceptible individuals (Fig 11.10). It may seem that this system dynamics is stochastic, but it is in fact entirely deterministic. We had not seen this erratic behavior in the original SIR model. Note that the interaction between modules is unidirectional—the SIRS population dynamics does not impact the influx of virus in the system in the same manner.

Whether the system can capture a stable endemic level of state *I* level depends on the balance of within-host parameters and between-host interactions. As the size of the antiviral effector variable *E* shrinks, the virus oscillations increase in amplitude. In the original model [29], *E* corresponded to antibody responses for Ebola-infected individuals. Stability analysis (see Chapter 5) and simulations with

literature-derived parameter values indicate that addition of antiviral therapeutics that enhance the immune response at the onset of the endemic infection (as reflected by an increased value of c_ν) will suppress a short-lived outbreak and significantly diminish its peak.

11.2 AGENT-BASED MODELING

The complicated model output of the previous case study raises unexpected questions. Is it optimal to model the total viral load within the population in this manner? There is little doubt that it would be more accurate to reflect some infected individuals as having low a virus load or a rapid immune response while other patients remain sick and infect susceptible people for longer periods. So, representing all *persons* as having their own internal battle between effector cells and virus should be a more accurate description if we want to consider epidemiological dynamics. How might this be accomplished? One form of computational model that provides a natural platform for exploring the bridge between the internal dynamics in individuals and interactions within a population is an **agent-based model** (ABM).

ABMs are very flexible systems descriptions that operate on discrete timescales (Chapter 4). They are especially convenient for simulating complicated spatial behaviors that are often subject to randomness. One particularly intriguing aspect of ABMs is that emergent organizational features naturally arise from simple rules of interactions between agents. The agents in this approach may represent molecules, cells, organisms, subpopulations, or about any other entities. Contextual information regarding each agent's properties, its interactions with other agents, and its responses to the immediate environment allows the agent to make independent decisions with each discrete time step. Moreover, because the rules can be modified over the course of a simulation with new information, the agents can be made adaptable and permit a lot of variations in their properties.

To gain familiarity with the mechanics of ABMs, we return to our SIRS model and convert it into a model that accounts for space and random effects. We use for this illustration the specialized software NetLogo [30] (Box 11.1), a platform first developed by Uri Wilensky in 1998 and now widely used in education for developing agent-based models [31, 32]. Nonetheless, other specialized platforms have been proposed for developing complex multicellular models for biological applications, such as Morpheus, CompuCell3D, and PhysiCell [33–35].

BOX 11.1: NETLOGO

Many tools exist for describing spatial movement and agent-specific rules within a simulation environment, but no platform is as popular and long-standing as NetLogo [30]. Referencing the Logo programming language, which was derived from Lisp and popularized in the 1980s for introducing computer programming through graphing the movement of a turtle icon on screen, NetLogo still calls all agents *turtles*. NetLogo turtles can be defined by class types (known as *breeds*), placed at distinct coordinates in a 2D or 3D environment, and follow scripts for dictating movement with respect to their current state or through interactions with other turtles. Furthermore, environmental features can be explicitly defined through *patches*. Discrete, multi-agent interactions, such as predator-prey dynamics (see Chapter 14), can be explored through this platform.

11.2.1 Agent-Based Modeling in Netlogo

Each agent in our Netlogo simulation represents an individual, categorized as susceptible (S), infected (I), or recovered (R), as before. However, unlike the ODE model reflected in Figure 11.8, the ABM is a tool for discrete, stochastic simulations, where all agents move independently and act on a two-dimensional grid. An important consequence of the allowance for stochasticity is that the simulations result in different dynamic trends with every run. Stochasticity may be introduced in several ways. First, the placement of individuals on a simulation lattice may be initialized in a random fashion. Next, the movements of agents are typically randomized in the x and y directions, so that each agent reaches a new location on the grid with each timestep or stays where it is. In addition, a user-set fraction of the

population is randomly assigned as initially infected. Finally, each infected agent is given randomized initial values for viral load (V) and the number of effector cells (E), and these settings determine how long the agent remains sick. The population-based dynamics of the model consists of all agents collectively moving and changing state on the same grid, according to their rules for movements and to rules that determine what happens if two agents collide, as well as to subsequent rules for transition toward death or recovery.

For our SIRS example, a set of nested ODEs used for state I is implemented within each agent using the same equation (11.3) as in the continuous system model, with parameters either set uniformly or randomized across a range of values to reflect variability in immune rates across the population. Without interactions between S and I, no virus is spread. Upon collision between an S agent and an I agent, a transfer of V initiates the state transition from a susceptible to an infected individual, and the internal V–E dynamics commences. While the agent is fighting the infection, any future interaction with another S agent will transmit the virus further. Note that this mechanism substantially differs from the model in Eqn. (11.2) in which the term rI determines infection rates by the overall viral load in the system. By contrast, the spread of virus in the NetLogo model is defined by an interaction distance between an I individual and an S individual on the grid. This aspect of distance is analogous to the presumed minimal social distance of 6 feet to avoid infection from respiration (sneezing, yelling, and even singing) during the SARS-CoV-2 pandemic. The distance defined infectiousness for viral variants [36], with interactions (infection) possible over larger distances associated with higher contagiousness. In our Netlogo model, a value of 1 represents a single pixel unit apart, while a value of 4 is defined as a larger "zone of contamination" with a "radius" of 4 pixels on each side of an I individual to determine whether a transition to a susceptible agent occurs. Finally, the transition from state I to R for a given agent depends on the virus level dropping below a user-defined threshold value. Upon this I-to-R transition, a clock commences (e.g., 10 time steps) that defines the duration of immunity, i.e., how long the individual stays in the R state before returning to the susceptible S population. For this example, a combination of birth and immigration into the population, defined with a constant rate, results in a constant flow of new susceptible people (S). Death from the virus in the I agents occurs if V exceeds a user-defined threshold. Details can be found in Box 11.2 and in the description of the code in the Netlogo library.

BOX 11.2: NETLOGO CODE OF A MULTISCALE SIRS MODEL

```
breed [healthy healthys] ;; susceptible people
breed [sick sicks]       ;; infected people
breed [immune immunes]   ;; recovered people

turtles-own
   [remaining-immunity   ;; how many weeks of immunity the turtle has left
    sick-time            ;; how many weeks the turtle has been infectious
    V                    ;; virus load
    Ef                   ;; number of effector cells
    ]

globals
   [ %infected           ;; what % of the population is infectious
     %immune             ;; what % of the population is immune
     immunity-duration   ;; how many weeks immunity lasts
;;  note that this is a variable w/initial value = immune-duration
     dieprob             ;; sets death threshold for infected individual
     deltat              ;; timestep
     t ]                 ;; internal time

to update-display
   ask turtles [
     if shape != turtle-shape [ set shape turtle-shape set size 1.5 ]
     if breed = sick [ set color red ]
     if breed = immune [ set color grey ]
     if breed = healthy [ set color green]
```

```
 if count turtles > 0
   [ set %infected (count turtles with [breed = sick ]/ count turtles) * 100
     set %immune (count turtles with [ breed = immune ] / count turtles) * 100 ]
]
end

to setup
  clear-all
;;  setup-constants
  populate
;;  update-global-variables
  update-display
  reset-ticks
end

;; We create a population of turtles of which number-sick are infectious,
;; and distribute them randomly to populate the environment
  create-healthy (number-healthy)
    [ setxy random-xcor random-ycor
    set remaining-immunity 0]
    create-sick (number-infected)
    [ setxy random-xcor random-ycor
    set V random 1e3
    set Ef random 1e6
    set remaining-immunity immune-duration
    set dieprob 0.01]
   create-immune 1
    [ setxy random-xcor random-ycor
    set remaining-immunity immunity-duration]
end

to get-sick         ;; turtle procedure upon interaction
  set breed sick
  ask sick
  [set V random 5000
    set Ef random 1e6]
end

to become-immune    ;; turtle procedure upon decline in V
  set breed immune
  set remaining-immunity immune-duration
end

to go               ;; loop for all turtles
  create-healthy (birth-rate) ;; add to population with birth
  [ setxy random-xcor random-ycor
    set remaining-immunity 0]
  ask turtles [get-older move]
  ask turtles [
  if breed = sick [recover-or-die]
  ifelse breed = healthy [infect]
    [lose-immune]]
  update-display
  set t t + deltat
  tick
end

to get-older
  ask turtles-here  [
  if breed = immune      [
    if immune-level  [
      set label remaining-immunity]]
  if breed = immune [ set remaining-immunity remaining-immunity - 1 ]
    if breed = sick [prodV prodE]]
  if (breed = immune) or (breed = healthy) [
    if random-float 1 < 0.01 [die]] ;; 1% natural death for healthy and immune
  turtles
end
```

```
to move             ;; turtle procedure for random movement
  rt random 50
  lt random 50
  fd 1
end

to lose-immune      ;; turtle procedure for immune breed
  ask immune-here [
    if remaining-immunity < 1 [
      set breed healthy ]]
end

to infect           ;; turtle procedure for healthy breed
  ask healthy-here [
    if any? other turtles with [breed = sick] in-radius interaction-distance [
      get-sick]];
end

to recover-or-die   ;; turtle procedure for sick breed
  ask sick-here [
        if virus-level [
      set label V
  ]
  if effector-level [
      set label Ef
    ]
    if V < immune-threshold [become-immune] ] ;; defining I->R threshold
    if V > death-threshold [die] ;; defining death threshold
end

;; diff eqns approximated from Almocera Math Biosci 2019
;; first differential equation for virus in sick turtles
to prodV
 set V V + deltat * (100 * V - 1.6e-4 * V * V - 2.4e-5 * Ef * V)
end

;; second differential equation for effector cells in sick turtles
to prodE
  set Ef Ef + deltat * (0.4 * Ef - 4e-7 * Ef * Ef + (0.024 * V / ( 1e5 + V)) * Ef)
end
; Modified from Copyrighted 1998 Uri Wilensky SIR model.
```

Once Netlogo is installed, we can call up the code by opening the file within the simulation interface under File/Open. The interface appears, and we initialize agents and click. We can now start a simulation by clicking *go*. It will run until we hit *go* again.

For our first exploration of the model, we run a simulation over a time period of the 52 weeks of transmission. Looking at the output, it becomes immediately clear that high infectivity of the virus, associated with a larger infection distance of 4 results in a rapid rise of the sick population, with 90% of the population infected within a week (Fig 11.11, dashed lines). By comparison, setting a smaller interaction distance of 2 reduces the chance of infection because close encounters occur less frequently between susceptible and infected people, leading to a slower decline of the susceptible population S and a slower rise of the infected group I (Figure 11.11, solid lines). Note that if our simulation environment were extremely crowded, the sensitivity to this parameter would decrease due to more frequent encounters between individuals.

It is straightforward to examine the effects of the multiscale parameters that link the internal mechanism of V–E interactions to the outside world of population-level dynamics over 365 time steps, thus representing the impact of viral spread through a population over a year. As an example, we define different thresholds of virus titers needed to change an agent from immune to recovered. Our goal is to determine the effect of the two virus threshold parameters (transition from I to R and transition from I to death) on the trends in the three subpopulations S, I,

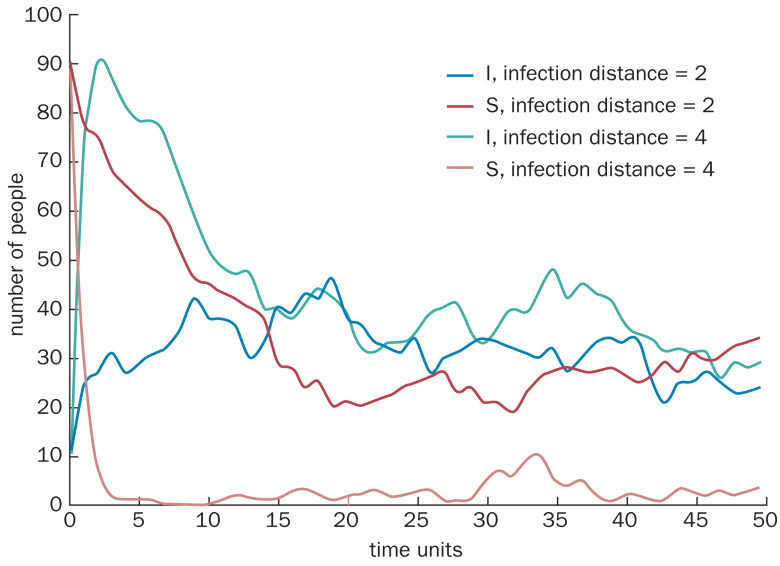


Figure 11.11 Trends in S and I during an infection. Doubling the radius of infectivity of the *I* agents from a distance of 2 pixels (solid lines) to 4 pixels (dotted lines) leads to dramatic increases in the magnitude and time to peak infection.

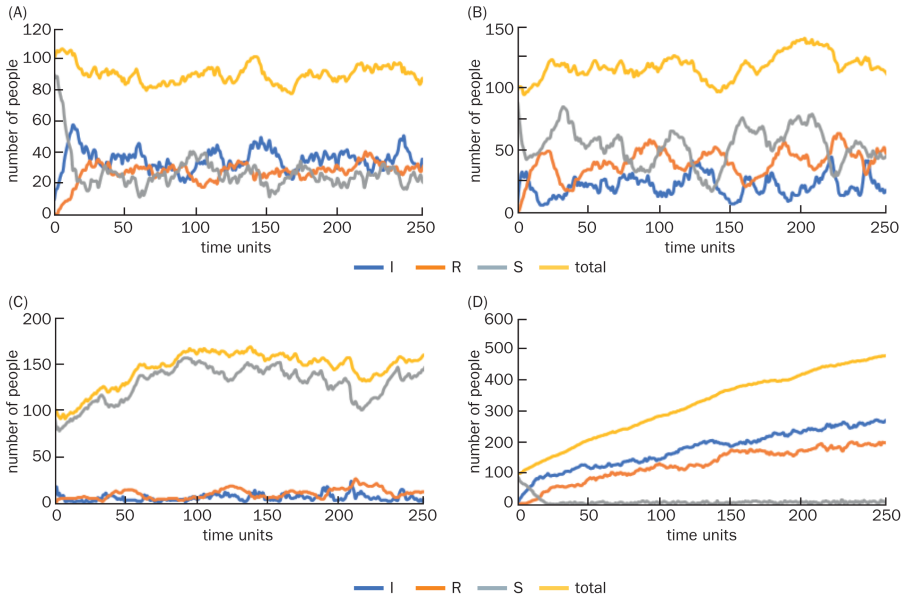


Figure 11.12 Examples of varying two parameters defining the viral thresholds for I-to-R transitions and I-to-death transitions within a given agent in the multiscale SIRS model. (A) Default settings of immune threshold $V \leq 200$, death threshold $V \geq 4{,}610$; (B) Immune threshold $V \leq 500$, death threshold $V \geq 4{,}610$; (C) Immune threshold $V \leq 500$, death threshold $V \geq 3{,}500$; (D) Immune threshold $V \leq 200$, death threshold $V \geq 6{,}000$.

and *R*. Sample output with various combinations of these two parameter values is shown in Figure 11.12. We quickly notice from inspection drastic differences in the total population levels in each plot (yellow lines). The net effect of death of infected individuals with a constant influx of new susceptible people is largely balanced at the default setting of $V > 4{,}610$, meaning that 4,610 viruses per individual are sufficient to cause an infected person to die (plots 11.12A and 11.12B). In this case, the fluctuating total population hovers around the starting value of 100 people. Changing this threshold parameter to a higher or lower value (plots C, D) leads to growth of the population. Why is that? One would think that a lower threshold of death, making it easier to remove infected individuals, would lead to a massive decline in the total population. It may at first be counterintuitive, but the model demonstrates quite clearly that a high death rate and thus rapid removal of infected agents prevents widespread contagion among the rest of the population (as is visible in the low *I* levels of plot 11.12C) because there are fewer chance encounters for viral propagation. Alternatively, we can examine the impact of transitioning more quickly to the recovered state by a higher immune threshold, which is visualized in the comparison of plots 11.12A and 11.12B. This quicker recovery reduces the actively infectious population, as evidenced by the lower *I* trendline and higher

R trendline in 11.12B compared to 11.12A. Secondarily, accelerating a transition to *R* results in a higher *S* population as well.

This case study exemplifies multiscale behavior resulting from passing one variable, *V*, that is spatially confined within an individual to the population-level interactions. With the chosen parameter values, the timescales of *V*–*E* interactions in this simple model are occurring on the order of days, which happens to correspond closely to the spread of infections in humans. At each timestep, the integration of two ODEs representing the virus and effector cells occurs within a host happens in a deterministic manner, while the simultaneous evaluation of movement and decision making (infect, recover, die) is stochastic. In this example, all events occur at similar timescales. In other settings, the difference in timescales could differ much more widely, possibly by several orders of magnitude. For instance, the two-week incubation time of the parental SARS-CoV-2 virus is very slow compared to the rapid replication of norovirus, which causes symptoms within hours of exposure [37]. If there is a large discrepancy in timescales between the within-individual *V*–*E* dynamics and the within-population SIRS dynamics, it is necessary to resort to true temporally multiscale models.

11.2.2 More Complicated Spatial Aspects in Agent-Based Models

The initial introduction to ABMs, revisiting the SIRS scenario, considered the group of individuals to be randomly interacting within a defined area—there were no cruise ships, dormitories, public transportation, or other "superspreader events" that could increase the likelihood of specific population groups coming into contact with an infected person. By contrast, a typical use of many ABMs is the exploration of the outcomes of particular agent behaviors in heterogeneous environments are often of interest.

With respect to more detailed spatial features, ABMs fall into several classes of spatial construction (**Figure 11.13**). The typical default is a **lattice-based model**,

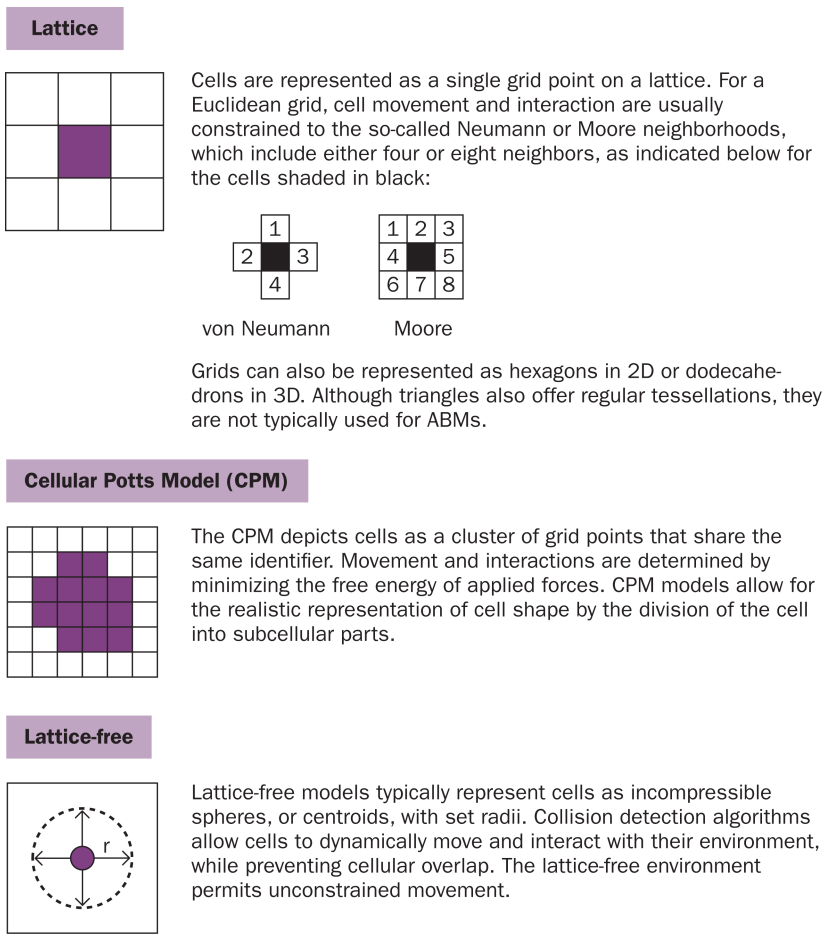


Figure 11.13 The three most common physical representations of cells in ABMs. (Adapted from Glen CM, Kemp ML & Voit EO. *PLoS Comput. Biol.* 15 [2019] e1006577.)

where the position and movement of each agent are implemented through individual points on a grid. In other words, the typically two-dimensional space is subdivided with regularly spaced gridlines that define cells of the size of a single agent; our Netlogo model is an example. Movements and engagement with other agents or with the environment are dictated by rules involving the agent, the nearest neighbors on the lattice, and sometimes the environment. This set-up is the workhorse of typical ABM modeling and is the least complex, but it lacks a refined description of how forces from the environment might alter spacing, deformation, and movement between agents, which can be important at the cellular and molecular levels. Nonetheless, just as a pixelated picture can communicate visual information effectively, a simple lattice-based representation is often sufficient for simulating spatial dynamics without high computational cost. Also similar to pixelated images, shrinking of the lattice spacing increases resolution and can generate greater spatial complexity. NetLogo is an example of lattice-based ABM software (Box 11.1) that can take advantage of the microenvironmental description of "patches" that coarsely approximate localized heterogeneity without the computational expense associated with explicit implementation of complex field equations [38]. For example, a model of resource competition between benign, sedentary cancer cells and invasive metastatic cancer cells used NetLogo to explore ecological costs associated with nutrient demands needed for motility [39].

A slightly more complicated, often-used subset of lattice-based ABMs for describing multicellular biological systems is a modeling variant called the **cellular Potts model** (Figure 11.13). This structure allows each agent to be represented as a region of lattice squares represented by a common cell index. Each agent may migrate and/or be deformed in some manner, with the technical details being dictated by minimization of an energy function that accounts for adhesion with neighbors and constraints on the overall area of the agent [33]. Sophisticated models of cell sorting [40, 41], tumor growth, and vasculature branching [42] have been achieved with this type of model.

Another form of ABM is a **lattice-free** physics-based approach, where agent displacement is constrained by springs that create tension between neighboring agents. The specifics are determined by the "thermal energy" reflected in jostling of the ball-spring forms. This format has proven to be effective and convenient for capturing features such as cell sorting by differential adhesion, chemotactic movement, and directed growth, thus making this modeling format particularly useful for describing developmental processes. Interactions that are explicitly defined on the spatial scale of cell-to-cell distances (on the order of 10μm) are manifested in outcomes that occur on tissue scales on the order of 1,000 μm. Furthermore, while subcellular resolution is typically not captured in these models, mechanisms occurring on faster timescales than cellular movement—such as soluble factor diffusion, signaling processes, and metabolism—can be captured by integrating differential equations and passing values of the updated species back to the slower timescale events of cell-based rules involving movement, division, and similar behaviors. Descriptions of tissue-scale patterning and cellular organization can result from phenotypic state changes that are functions of neighboring cell influences [43], morphogen levels [44], or juxtacrine mechanisms within stem cell niches [45, 46].

No matter how an ABM is implemented, challenges arise in fitting parameters and quantifying or predicting outcomes associated with multiscale, spatial modeling. A superb example of applying a form of sensitivity analysis (see Chapter 5) was proposed by Kirschner and Linderman, who developed multiscale models of tuberculosis based on bacterial interactions with host immune cells in an inflammatory microenvironment [47, 48]. Their models include receptor kinetics and intracellular signaling via ODE description, secretion, and uptake of soluble molecules that develop into gradients across the tissue and cellular actions (e.g., activation, killing, growth) performed by the bacterial cell agents and immune cell agents. Given this diversity of agents, the model parameters covered a wide range of processes, from protein synthesis rates to probability rates of cell movement. In order to evaluate the effects of parameter value assignment on the system, the authors performed a statistical technique called Latin Hypercube Sampling (see Chapter 6) to minimize the selection of parameter sets across a wide range of val-

ues. They also devised a quantitative metric, called a partial rank correlation coefficient, to evaluate each parameter's influence on the model output [48]. With targeted simulations of this type, it is possible to construct input/output correlations at different hierarchical levels of the model [49].

ABMs often incur substantial computational demands in terms of simulations supporting analyses that are standard in regular ODE modeling, such as stability or sensitivity analysis. In the tuberculosis example, for instance, the parameters were sampled across 250 numerical values, and each value was simulated four times to account for the stochasticity of the model. These types of sensitivity analyses demand many thousand simulations to yield critical insight into the relative importance of the properties of a biological system of interest.

ABMs are still relatively new, and new tools will continue to advance their use. Importantly, the great flexibility of ABMs means that the sky is the limit for all kinds of practical applications.

11.3 OUTLOOK FOR MULTISCALE MODELING

Multiscale modeling is and will be a grand challenge throughout the foreseeable future [6, 7]. In most real-world cases, it is not feasible to construct a single model simultaneously addressing the different organizational, spatial, and temporal scales, and the only option is a separation of time scales and the subsequent construction of sub-models, whose structure and complexity are directly related to the questions investigated at a given level. In opportune applications, the sub-models can be conceptually concatenated, thereby yielding a more or less complete chain of causes and effects that is able to explain macroscopic observations based on molecular mechanisms (for a detailed example, see Chapter 12). Such a chain of models requires the dissection of large-scale systems into functional modules. Within the human body, such systems of systems are abundant.

More likely than complete chains are sets of complementary models that focus on one or two levels and inform models at other levels, as has been proposed in the Physiome and Virtual Physiological Rat Projects [50–52]. The liver, for instance, consists of more or less homogeneous lobes, which themselves consist of hepatocytes and other cell types that produce over 1000 important enzymes and are essential for the production of proteins and other compounds needed for digestion and detoxification. Initially ignoring the requirements and means of metabolite transport, much of the function of the liver can be studied on the level of individual hepatocytes. However, moving to higher levels of functionality, one must address the overarching functions of the tissues and the whole organ, such as the regenerative potential of the liver with all its spatial and functional implications. The Virtual Liver Network and the KidneySim Project approach this multiscale topic with different types of computational models [53–55].

An alternative for organizing the model design could be a so-called template-and-anchor approach that separates a coarse, high-level template model from finer grained, lower-level anchor models [56, 57]. The anchor models themselves sometimes serve as templates for even more detailed models.

While any conceptual and computational dissection of large systems into smaller modules is of enormous help for modeling purposes, caution is necessary, because the dissection sometimes eliminates crucial interactions among components. A prime example is the brain, which consists of billions of neurons, whose individual function we understand quite well but which is governed by uncounted, highly regulated brain circuits and other physiological systems that are often ill characterized. We have a good grasp of the signal transduction processes occurring at synapses, where neurons communicate with each other, but the sheer number of neurons, their biochemical and physiological dynamics, and the complex distribution of synapses throughout the brain lead to emerging properties such as adaptation, learning, and memory that are still beyond our grasp (see Chapter 15). We also need to be recognizant of the fact that problems in one system easily spread to others. For instance, a full understanding of cancer without consideration of the immune system is without doubt incomplete. One thing is clear: micro- and macrophysiology will continue to present us with grand challenges but also with a glimpse of where we might want to target our research efforts in the years to come.

11.4 EXERCISES

11.1 It is straightforward to set up a metabolic model for the pathway diagram in Figures 11.4 and 11.5 formulated as a GMA system. With parameter values gleaned from the literature, it reads

$$
\begin{aligned}
&\text{Glucose:} && \dot{X}_1 = 30X_2^{-0.2} - 90X_1^{0.75}X_6^{-0.4} + 2.5X_7^{0.3} && X_1 = 0.03\\
&\text{G6P:} && \dot{X}_2 = 90X_1^{0.75}X_6^{-0.4} + 54X_2^{-0.6}X_3^{0.6} - 23X_2^{0.75} && \\
& && \quad -3X_2^{0.2} - 7.2X_2^{0.4}X_3^{-0.4} - 0.2X_1^{-0.3}X_2^{0.3}X_4^{0.3} && X_2 = 1\\
&\text{G1P:} && \dot{X}_3 = 7.2X_2^{0.4}X_3^{-0.4} + 0.9X_2^{-0.2}X_4^{-0.2}X_5^{0.25} - 54X_2^{-0.6}X_3^{0.6} && \\
& && \quad -11X_3^{0.5} - 12X_2^{-0.2}X_3^{0.8}X_4^{-0.2} && X_3 = 0.1\\
&\text{UDPG:} && \dot{X}_4 = 11X_3^{0.5} - 0.2X_1^{-0.3}X_2^{0.3}X_4^{0.3} - 3.5X_2^{0.2}X_4^{0.4} && X_4 = 0.66\\
&\text{Glycogen:} && \dot{X}_5 = 3.5X_2^{0.2}X_4^{0.4} + 12X_2^{-0.2}X_3^{0.8}X_4^{-0.2} - 0.9X_2^{-0.2}X_4^{-0.2}X_5^{0.25} && \\
& && \quad -4X_5^{0.25} && X_5 = 1.04\\
&\text{T6P:} && \dot{X}_6 = 0.2X_1^{-0.3}X_2^{0.3}X_4^{0.3} - 1.1X_6^{0.2} && X_6 = 0.02\\
&\text{Trehalose:} && \dot{X}_7 = 1.1X_6^{0.2} - 1.25X_7^{0.3} && X_7 = 0.05
\end{aligned}
\qquad (11.5)
$$

Here, the units are micromole for concentrations and micromole per minute for fluxes. Note that the model contains not only trehalose and T6P—in addition to the glycolytic metabolites—but also the pathway toward glycogen, which competes with the production of trehalose for the shared substrate UDPG and is also strongly affected by heat stress.

Implement the model in Eqn. (11.5) in a software program, such as MATLAB®, *Mathematica*®, Python, or PLAS. Check whether the system has one or more steady states and whether (any of) these steady states are stable. Revisit the meaning of eigenvalues and their real and imaginary parts in this context. Investigate the sensitivities of the system. Recall what exactly the magnitude and sign of a sensitivity or gain means and determine which parameters are particularly influential.

11.2 Perform perturbation experiments with the model in Eqn. (11.5) in order to develop a feel for how the model responds to slight alterations in numerical features. Investigate transient features of the model with simulations of temporary or persistent increases or decreases in some of the variables or parameters.

11.3 Expand your program to normalize all variables with respect to their steady-state values. In other words, scale all variables by their steady-state values so that all variables have a steady-state value of 1. What is a significant advantage of this normalization?

11.4 Simulate excess and shortage of glucose with the model in Eqn. (11.5) and report your findings.

11.5 Simulate mutations in TPS1 or TPS2 with the model in Eqn. (11.5) and report your findings. Search the literature for data and compare them with your modeling results.

11.6 Simulate mutations leading to decreased activities (80%, 20%) of all enzymes, one at a time and two at a time. Discuss to what degree single mutations are predictive of double mutations.

11.7 How would you simulate alterations in enzymatic steps that are catalyzed by two or more enzymes, such as hexokinase I, hexokinase II, and glucokinase? Search the literature for responses to deletions of one of several enzymes catalyzing the same reaction. Write a brief summary report.

11.8 It has been observed that TPS1 deletion mutants are unable to grow on glucose substrate. Search the literature for details. Discuss how this phenomenon can be demonstrated with the model or why such an analysis is not possible.

11.9 Temperature-induced changes in enzymes occur very rapidly, while the expression of genes and the synthesis of new proteins take some while. Discuss the implications of these time scale issues for models of the heat stress response.

11.10 Test with the model in Eqn. (11.1) whether changes in only one or two enzyme activities would result in appreciable increases in trehalose. Report how the metabolic profiles are affected by each change.

11.11 Explore with the model in Eqn. (11.1) different alterations in enzyme activities. Assemble a table showing which alterations lead to significant increases in trehalose. Add one sentence per simulation describing other effects on the system.

11.12 Predict what the influence of the size of the external glucose pool is on the dynamics of the model in Eqn. (11.1). Check your predictions with simulations.

11.13 Discuss which assumptions must be made explicitly and implicitly, when one substitutes heat-induced changes in enzyme activities with corresponding changes in gene expression.

11.14 Show with straightforward algebra that 1,000,000 is the nonzero steady-state value of E if $V = 0$.

11.15 Analytically compute the nontrivial steady state of V and E if $V \neq 0$; check the result in PLAS or some other software.

11.16 Linearize the VE module and compare the original and the linearization with respect to $\log(V+1)$ and $\log(E)$. Assess the stability of the non-zero steady state of the VE module and its approach to this state, as in Figure 5.9.

11.17 Assess with simulations how important the virus titer of a new bout is for the dynamics of the VE module.

11.18 Which parameter or parameters must be altered (1) to reduce the virus population permanently to below 100 or (2) to eliminate the virus altogether ($V \approx 1$)?

11.19 A key feature of the two-level SIRS-VE model is the difference in timescales. To explore this difference, multiply the entire equations for V and E with the same positive number, which alters the timescale of the VE module. At a minimum, test scaling values of 0.1, 0.5, 2, and 10. Summarize your findings in a paragraph.

11.20 The virus-dependent infection rate $rI(V)$ is the direct connection between the two levels of the two-level SIRS-VE model. The parameter, rW, defines the Michaelis–Menten rate of infection as a function of virus. Test the effect of different values of rW.

11.21 Removal of the virus is achieved by effector cells. Study the role of their growth rate on the dynamics of the two-scale system by altering cV.

11.22 Perform 10 NetLogo simulations of the multiscale SIR model under default settings and for 100 timesteps. Analyze the differences in the time trends of each variable and the total population among the different runs. Report your findings. Hint: exporting the resulting trend lines from the user interface (right click on graph, highlight export) allows superposition of the results from independent runs.

11.23 Using the NetLogo multiscale SIR model, explore how human behaviors alter the dynamics of viral propagation in the population. Specifically, create a scenario to investigate how readily the virus is spread if infected individuals are too sick to move.

11.24 Using the NetLogo multiscale SIR model, investigate how prophylactic administration of antivirals impacts an outbreak. Determine what term in the differential equation for V represents the self-replication rate parameter and reduce this value for all people.

11.25 Compare the steady-state output of the continuous multiscale model with the ABM output.

11.26 The model is set to assign a random value of V and E upon infection. What happens if each transmission leads to exactly the same amount of V and E? Assign V and E values that correspond to the initial conditions of the continuous multiscale model.

FURTHER READING

Ford Versypt AN. Multiscale modeling in disease. *Curr Opin Syst Biol.* 27 (2021) 100340.

Glen CM & Kemp ML, Voit EO. Agent-based modeling of morphogenetic systems: Advantages and challenges. *PLoS Comput Biol.* 15(3) (2019) e1006577.

Hohmann S & Mager WH, eds. *Yeast Stress Response*, pp. xiii + 389. Springer Verlag, Berlin, 2003.

Macal C & North M. Introductory tutorial: Agent-based modeling and simulation. In *Proceedings of the Winter Simulation Conference* (A Tolk, SY Diallo, IO Ryzhov, L Yilmaz, S Buckley, JA Miller, eds). IEEE Press, Savannah, GA, 2014.

Rodrigues-Pousada C, Nevitt T & Menezes, R. The yeast stress response. *FEBS J.* 222 (2005) 2639–2647.

Schnell S, Grima R & Maini P. Multiscale modeling in biology. *Amer Sci.* 95 (2007).

Shiflet AB & Shiflet GW, An introduction to agent-based modeling for undergraduates. *Procedia Comput Sci.* 29 (2014) 1392–1402.

Taymaz-Nikerel H, Cankorur A & Kirdar B. Genome-wide transcriptional response of *Saccharomyces cerevisiae* to stress-induced perturbations. *Front Bioeng Biotechnol.* 4 (2016), https://doi.org/10.3389/fbioe.2016.00017.

Walpole J, Papin JA & Peirce SM. Multiscale computational models of complex biological systems. *Annu Rev Biomed Eng.* 15 (2013) 37–54.

Weinan E. *Principles of Multiscale Modeling*. Cambridge University Press, Cambridge, UK, 2011.

Wilensky U & Rand W. *An Introduction to Agent-Based Modeling: Modeling Natural, Social, and Engineered Complex Systems with NetLogo*. MIT Press, Cambridge, MA, 2015.

REFERENCES

1. Kokotovic, P.V., et al., Singular perturbation and iterative separation of time scales. *Automatica*, 1980. **16**(1): p. 23–33.
2. Savageau, M.A., Allometric morphogenesis of complex systems: Derivation of the basic equations from first principles. *Proc Natl Acad Sci U S A*, 1979. **76**(12): p. 6023–6025.
3. Voit, E.O., et al., What's next for computational systems biology? *Front Syst Biol*, 2024. in press.
4. Ferreira, A.E., A.M. Ponces Freire, and E.O. Voit, A quantitative model of the generation of N(epsilon)-(carboxymethyl)lysine in the Maillard reaction between collagen and glucose. *Biochem J*, 2003. **376**(Part 1): p. 109–121.
5. Fletcher, A.G. and J.M. Osborne, Seven challenges in the multiscale modeling of multicellular tissues. *WIREs Mech Dis*, 2021. **14**(1): p. e1527.
6. Schnell, S., R. Grima, and P. Maini, Multiscale modeling in biology. *Am Sci*, 2007. **95**(2): p. 134.
7. Sloot, P.M.A. and A.G. Hoekstra, Multi-scale modelling in computational biomedicine. *Brief Bioinform*, 2009. **11**(1): p. 142–152.

8. Hohmann, S. and W.H. Mager, eds. *Yeast Stress Response*. 2003, Berlin: Springer Verlag. p. xiii+389.
9. Voit, E.O., Biochemical and genomic regulation of the trehalose cycle in yeast: Review of observations and canonical model analysis. *J Theor Biol*, 2003. **223**(1): p. 55–78.
10. Paul, M.J., et al., Trehalose metabolism and signaling. *Annu Rev Plant Biol*, 2008. **59**: p. 417–441.
11. Shima, J. and H. Takagi, Stress-tolerance of baker's-yeast (*Saccharomyces cerevisiae*) cells: stress-protective molecules and genes involved in stress tolerance. *Biotechnol Appl Biochem*, 2009. **53**(Pt 3): p. 155–164.
12. Hottiger, T., P. Schmutz, and A. Wiemken, Heat-induced accumulation and futile cycling of trehalose in *Saccharomyces cerevisiae*. *J Bacteriol*, 1987. **169**(12): p. 5518–5522.
13. Neves, M.J. and J. François, On the mechanism by which a heat shock induces trehalose accumulation in *Saccharomyces cerevisiae*. *Biochem J*, 1992. **288**(Pt 3): p. 859–864.
14. Hannun, Y.A. and L.M. Obeid, Principles of bioactive lipid signalling: Lessons from sphingolipids. *Nat Rev Mol Cell Biol*, 2008. **9**(2): p. 139–150.
15. Jenkins, G.M., et al., Involvement of yeast sphingolipids in the heat stress response of *Saccharomyces cerevisiae*. *J Biol Chem*, 1997. **272**(510): p. 32566–32572.
16. Cowart, L.A., et al., Revealing a signaling role of phytosphingosine-1-phosphate in yeast. *Mol Syst Biol*, 2010. **6**(349).
17. Gasch, A.P., et al., Genomic expression programs in the response of yeast cells to environmental changes. *Mol Biol Cell*, 2000. **11**(12): p. 4241–4257.
18. Vilaprinyo, E., R. Alves, and A. Sorribas, Minimization of biosynthetic costs in adaptive gene expression responses of yeast to environmental changes. *PLoS Comp Biol*, 2010. **6**(2).
19. Palotai, R., M.S. Szalay, and P. Csermely, Chaperones as integrators of cellular networks: changes of cellular integrity in stress and diseases. *IUBMB Life*, 2008. **60**(1): p. 10–18.
20. Hahn, J.S., D.W. Neef, and D.J. Thiele, A stress regulatory network for co-ordinated activation of proteasome expression mediated by yeast heat shock transcription factor. *Mol Microbiol*, 2006. **60**(1): p. 240–251.
21. Fonseca, L., et al., Complex coordination of multi-scale cellular responses to environmental stress. *Mol BioSyst*, 2011. **7**: p. 731–741, https://doi.org/10.1039/C0MB00102C.
22. Mundim, K.C., et al., Temperature coefficient (Q_{10}) and its applications in biological systems: Beyond the Arrhenius theory. *Ecol Model*, 2020. **431**.
23. Buccitelli, C. and M. Selbach, mRNAs, proteins and the emerging principles of gene expression control. *Nat Rev Genet*, 2020. **21**(10): p. 630–644.
24. Koussounadis, A., et al., Dynamic modulation of phosphoprotein expression in ovarian cancer xenograft models. *BMC Cancer*, 2016. **16**: p. 205.
25. Lee, Y., P.-W. Chen, and E.O. Voit, Analysis of operating principles with S-system models. *Math Biosci*, 2011. **231**: p. 49–60.
26. François, J. and J.L. Parrou, Reserve carbohydrates metabolism in the yeast *Saccharomyces cerevisiae*. *FEMS Microbiol Rev*, 2001. **25**(1): p. 125–145.
27. Nwaka, S. and H. Holzer, Molecular biology of trehalose and the trehalases in the yeast *Saccharomyces cerevisiae*. *Prog Nucleic Acid Res Mol Biol*, 1998. **58**: p. 197–237.
28. Kernack, W. and A.G. McKendrick, Contribution to the mathematical theory of epidemics. *Proc Royal Soc London Ser A* 1927. **115**: p. 700–721.
29. Almocera, A.E.S. and E.A. Hernandez-Vargas, Coupling multiscale within-host dynamics and between-host transmission with recovery (SIR) dynamics. *Math Biosci*, 2019. **309**: p. 34–41.
30. Netlogo, https://ccl.northwestern.edu/netlogo/, accessed 2024.
31. Wilensky, U., *NetLogo*, http://ccl.northwestern.edu/netlogo/, 1999.
32. Bodine, E.N., et al., Agent-Based Modeling and Simulation in Mathematics and Biology Education. *Bull Math Biol*, 2020. **82**(8): p. 101.
33. Swat, M.H., et al., Multi-scale modeling of tissues using CompuCell3D. *Methods Cell Biol*, 2012. **110**: p. 325–366.
34. Ghaffarizadeh, A., et al., PhysiCell: An open source physics-based cell simulator for 3-D multicellular systems. *PLoS Comput Biol*, 2018. **14**(2): p. e1005991.
35. Starruss, J., et al., Morpheus: A user-friendly modeling environment for multiscale and multicellular systems biology. *Bioinformatics*, 2014. **30**(9): p. 1331–1332.
36. Stadnytskyi, V., P. Anfinrud, and A. Bax, Breathing, speaking, coughing or sneezing: What drives transmission of SARS-CoV-2? *J Intern Med*, 2021. **290**(5): p. 1010–1027.
37. Robilotti, E., S. Deresinski, and B.A. Pinsky, Norovirus. *Clin Microbiol Rev*, 2015. **28**(1): p. 134–164.
38. Durrett, R. and S. Levin, The importance of being discrete (and spatial). *Theor Popul Biol*, 1994. **46**: p. 363–394.
39. Aktipis, C.A., C.C. Maley, and J.W. Pepper, Dispersal evolution in neoplasms: The role of disregulated metabolism in the evolution of cell motility. *Cancer Prev Res (Phila)*, 2012. **5**(2): p. 266–275.
40. Glen, C.M., M.L. Kemp, and E.O. Voit, Agent-based modeling of morphogenetic systems: Advantages and challenges. *PLoS Comput Biol*, 2019. **15**(3): p. e1006577.
41. Zhang, Y., et al., Computer simulations of cell sorting due to differential adhesion. *PLoS One*, 2011. **6**(10): p. e24999.
42. Shirinifard, A., et al., 3D multi-cell simulation of tumor growth and angiogenesis. *PLoS One*, 2009. **4**(10): p. e7190.
43. White, D.E., et al., Spatial pattern dynamics of 3D stem cell loss of pluripotency via rules-based computational modeling. *PLoS Comput Biol*, 2013. **9**(3): p. e1002952.
44. White, D.E., et al., Quantitative multivariate analysis of dynamic multicellular morphogenic trajectories. *Integr Biol (Camb)*, 2015. **7**(7): p. 825–833.
45. Du, H., Q. Nie, and W.R. Holmes, The interplay between Wnt mediated expansion and negative regulation of growth promotes robust intestinal crypt structure and homeostasis. *PLoS Comput Biol*, 2015. **11**(8): p. e1004285.
46. Glen, C.M., T.C. McDevitt, and M.L. Kemp, Dynamic intercellular transport modulates the spatial patterning of differentiation during early neural commitment. *Nat Commun*, 2018. **9**(1): p. 4111.
47. Marino, S., et al., Macrophage polarization drives granuloma outcome during Mycobacterium tuberculosis infection. *Infect Immun*, 2015. **83**(1): p. 324–338.
48. Fallahi-Sichani, M., et al., Multiscale computational modeling reveals a critical role for TNF-alpha receptor 1 dynamics in tuberculosis granuloma formation. *J Immunol*, 2011. **186**(6): p. 3472–3483.
49. Renardy, M., et al., Global sensitivity analysis of biological multi-scale models. *Curr Opin Biomed Eng*, 2019. **11**: p. 109–116.
50. Physiome, *The Physiome Project*, http://www.physiome.org/Links/websites.html, 2011.
51. Project, T.V.P.R., https://virtualrat.org/.
52. Hunter, P.J. and T.K. Borg, Integration from proteins to organs: The physiome project. *Nat Rev Mol Cell Biol*, 2003. **4**: p. 237–243.
53. Hoehme, S., et al., Prediction and validation of cell alignment along microvessels as order principle to restore tissue

architecture in liver regeneration. *Proc Natl Acad Sci U S A*, 2010. **107**(23): p. 10371–10376.
54. Virtual_Liver, *The Virtual Liver Network*, https://www.h-its.org/projects/virtual-liver/, 2011.
55. KidneySim, *KidneySim Project*, http://physiome.ibisc.fr/~srt/kidneysim/index.php, 2011.
56. Full, R.J. and D. E.Koditschek, Templates and anchors: Neuromechanical hypotheses of legged locomotion on land. *J Exp Biol*, 1999. **202**(23): p. 3325–3332.
57. Kumbale, C.M., Q. Zhang, and E.O. Voit, Hepatic cholesterol biosynthesis and dioxin-induced dysregulation: A multiscale computational approach. *Food Chem. Toxicol*, 2023. **181**: p. 114086.

12 Physiological Modeling: The Heart as an Example

When you have read this chapter, you should be able to:

- Understand the basics of the anatomy and physiology of the heart
- Discuss the electrochemical processes during contraction and relaxation
- Formulate black-box models of heartbeats
- Analyze the role of perturbations and pacemakers in oscillators
- Explain the biology and mathematical formulation of action potentials
- Understand the principles and challenges of multiscale modeling

The heart is an incredible machine. A little bigger than a fist and weighing just about half a pound, it pumps blood without ceasing, beating roughly 100,000 times every day, between two and three billion times in a normal lifetime. It keeps on working without us thinking about it, but beware if it skips even a few beats in a row! Many societies have given the heart exalted roles, associating it with life, love, and stress. Just think: what would Valentine's Day be without that heart shape? The fact that it is really quite different from the heart's true anatomy (**Figure 12.1**) is easily forgotten on February 14th.

Virtually unlimited sources of general and specific information are readily available about all aspects of the heart, from its cells and tissues to its anatomy, from its normal electrical and mechanical functioning to the plethora of possible problems and pathologies. The easiest access to matters of the heart is probably the large body of textbooks, as well as an enormous variety of websites (for example, [1–5]), which we will not specifically cite in the following.

The role of the heart is to move blood through the body and to supply the cells with most of what they need. The blood carries oxygen and nutrients and transports unwanted chemicals away. The heart accomplishes this feat by a regular pumping action that moves blood through a network of about 60,000 miles of arteries, veins, and capillaries. Every minute, it pumps about 5 liters of blood, for a total of over 7000 liters per day. If the heart stops, we are in trouble. Coronary heart disease is responsible for about half of all deaths in Western countries, and about 700,000 people die from heart attacks every year in the United States alone.

Clearly, the heart is a very complex system of systems that spans multiple scales, both in space and time. In a way, it could be another case study for the previous Chapter 11, but it is so important and spread out over scales that it deserves its own chapter. We begin by discussing several aspects of the function of the heart

DOI: 10.1201/9781003402770-12

with mathematical models in the form of vignettes. Conceptually, these can be concatenated into an overall systems model, as we will discuss at the end of the chapter. Some of the detailed models are very simple, others are more complicated, and some of the simulation models that have been developed for academic and commercial purposes are extremely complex, and we will only mention them. To some degree, the usefulness of these models correlates with their complexity, and the most sophisticated simulation models have become accurate enough to allow realistic studies of the normal heart and some of its diseases. Nonetheless, the simpler models are of interest as well, because they give us insights into the heart's macroscopic oscillatory patterns and its basic electrical and chemical properties, without overwhelming us with too much detail.

The strategy of the chapter is as follows. It begins with a general background section that is rather superficial and by no means does justice to the amazing features of the heart; a more detailed, very accessible introduction can be found in [4]. Next, we introduce some simple black-box models that describe and analyze heartbeat-like oscillations. Looking a little deeper into the physiology of the heart, we find that its function is directly associated with oscillations in the dynamics of calcium. Thus, the next step in our modeling efforts is to describe calcium oscillations with a minimalistic ordinary differential equation model and to explore how far such a model can lead us in understanding heart function. Indeed, we will see that quite simplistic models are sufficient to describe stable calcium oscillations. However, these models are of course very limited and, for instance, do not account for the **electrochemical** processes that we know to exist in the heart. We therefore dig deeper, trying to understand how chemistry and the laws of **electrophysiology** govern the contractions in individual heart cells and how the activities of all heart cells are coordinated in a fashion that results in proper heart pumping. We stop short of the most sophisticated simulation models of the heart, because they are naturally very complicated. Nonetheless, we do briefly discuss their underlying principles. Thus, we will start at the macroscopic level, dive into metabolism, electrophysiology, chemistry, and genetics, and ultimately return to the effects of molecular changes on the functioning of the whole heart.

This gradual procedure of starting at a high level with a rather coarse model and then studying selected aspects with a finer resolution and with an increasing level of sophistication mimics the way we learn many things when growing up. Early on, we distinguish static items form things that move, we learn to differentiate between living and engineered machines that move, we become able to distinguish cars from trucks and cheap cars from expensive cars, and in some cases we learn to discern the year a car was made even if the differences from one year to the next are very subtle. In modeling, a similar manner of learning has been proposed in biomechanics, where one might begin with a macroscopic, high-level assessment of locomotion, and subsequently embed increasingly finer-grained models into these first, coarse models [6]. In this biomechanical context, the high-level models are called templates. They help the modeler organize anchors, which are more complex models representing anatomical and physiological features in a detailed fashion and can be incorporated into the template models in a natural manner (see also [7]).

Numerous models of the heart have been developed over the past century that could serve as templates or anchors. In fact, many investigators have devoted their careers to heart modeling. Some giants in the field include James Bassingthwaighte, Peter Hunter, Denis Noble, Charles Peskin, and Raimond Winslow. Moreover, the electrochemical processes in heart muscle cells show quite a lot of resemblance to nerve cells, and descriptions of the dynamics of beating heart cells have been inspired directly by experimental and computational research on nerve cells, which began more than half a century ago with the pioneering work of Alan Lloyd Hodgkin and Andrew Huxley and was extended and further analyzed by leaders in the field of neuroscience, including Richard FitzHugh, Jin-Ichi Nagumo, and John Rinzel (see, for example, [8]). Indeed, 50 years ago, Noble [9, 10] showed with the first computer model of its kind that action potentials in the heart can be explained with equations similar to those proposed by Hodgkin and Huxley.

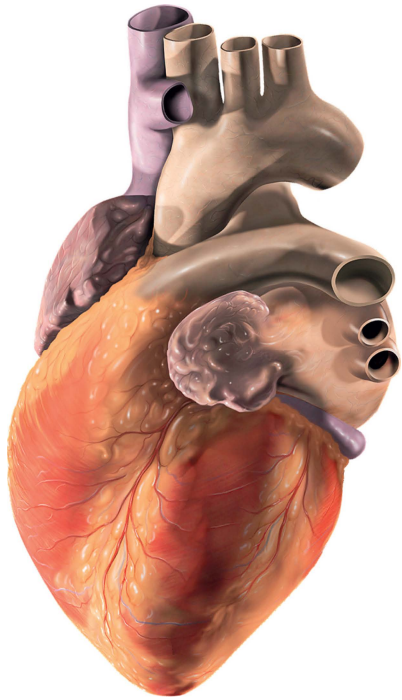


Figure 12.1 Model of the human heart. The heart is a complicated muscle that encloses four chambers. With each beat, the heart dynamically changes its shape and thereby pumps blood to the lungs and throughout the body. (Courtesy of Patrick J. Lynch under the Creative Commons Attribution 2.5 Generic license.)

12.1 HIERARCHY OF SCALES AND MODELING APPROACHES

It is quite obvious that the function of the heart spans many scales in size, organization, and process speed. Beginning with the most evident scale of a whole organ, the heart is a peristaltic pump that moves blood. This pump is composed of tissues, which need to work in a tightly coordinated fashion. The tissues consist of several types of cells, which have their specific functions within the concerted effort of making the heart beat. Within and between the cells, there is a lot of controlled movement of molecules and especially **ions**, which are directly associated with the electrical activity of the heart. Inside the cells, we find metabolites and proteins, some of which are found in many cells, while others are specific. Finally, there are genes, some of which are only expressed in heart cells and can cause problems if they are mutated.

Ideally, complementary models of all these aspects would seamlessly span the entire spectrum from electrical and chemical features to intracellular mechanical and metabolic events, from cells to cell aggregates and tissues, and from the different tissues to a complete picture of the normal physiology and pathology of the heart. However, such a complete multiscale story has not been written, although we will discuss some efforts in this direction at the end of the chapter. For the bulk of this chapter, we will select some aspects, while unfortunately having to minimize—or even omit—others. For instance, we will not discuss in any length very important issues of tissue mechanics and blood flow, three-dimensional models of the anatomy and physiology of the heart, four-dimensional changes in the heart over short and long time horizons, or mechanisms of adaptation and remodeling after infarction. Good reviews of pertinent research questions, modeling needs, and accomplishments at different hierarchical levels of organization have been presented within the framework of the Physiome Project, an international effort to understand normal and abnormal functioning of the heart through computational modeling and to translate the insights gained from modeling into advances in clinical applications (see, for example, [11–16]). Like other large-scale efforts, such as the Virtual Liver Network [17], the Lung Physiome [18], and the Blue Brain Project [19], the target of model development for these complex, multiscale systems is not necessarily a single all-encompassing supermodel that would account for every detail but rather a set of complementary models of different dimensionality and sophistication that explain certain aspects. The reason for not targeting a single model is that the relevant time and size scales in the heart and other organs are so vastly different that it seems impossible, at least with current methods, to merge them in a meaningful fashion [10]. Nonetheless, the various models should be able to talk to each other, which the Physiome Project facilitates by using cell- and tissue-specific mark-up languages (XMLs), such as CellML, SBML, and FieldML [14, 20, 21].

12.1.1 Basics of Heart Anatomy

Before we discuss different modeling approaches, we need at least a coarse description of our subject. The human heart is a muscle that contains four chambers, namely the left and right **atria** and the left and right **ventricles**

The left and right sides of a human heart become strictly separated during embryonic development so that there is normally no direct blood flow between them. With each heartbeat, blood flows through the chambers in a well-controlled, cyclic manner that consists of two phases called **diastole** and **systole**, which are Greek terms for dilation and contraction, respectively. Because the process is cyclic, it does not really matter where we start. So, let us begin with the phase of early diastole. At this point, the heart is relaxed and begins to receive blood in two states. The first is deoxygenated, that is, relatively low in oxygen content, because it returns from all parts throughout the body, where oxygen had diffused from the bloodstream into the various tissues. The second state of blood is oxygenated. It comes from the lungs, where hemoglobin molecules in the red blood cells have been loaded with oxygen.

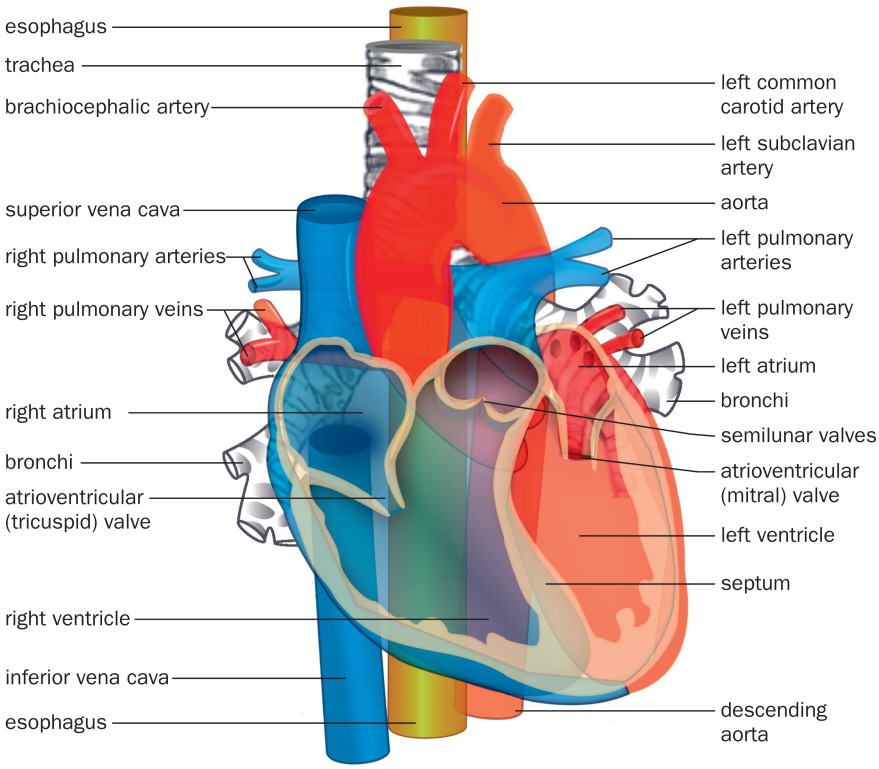


Figure 12.2 Diagram of the parts of the human heart. In healthy humans, the left chambers (atrium and ventricle) are completely separated from the right chambers. During the cardiac cycle, blood runs through the four chambers in a well-coordinated fashion. (Courtesy of ZooFari under Creative Commons Attribution-Share Alike 3.0 Unported license.)

The right atrium exclusively receives the deoxygenated blood through two branches of the vena cava, which collect blood from the various tissues of the body. Meanwhile, the left atrium exclusively receives oxygenated blood through the pulmonary vein that comes from the lungs. The atria fill quickly, and because the atrioventricular (AV) valves are open (see Figure 12.2), so do the ventricles. These are in a relaxed state where their wall tissue is thin and the plane of the AV valves shifts back up toward the base of the heart. A following contraction by the atria pushes additional blood into the ventricles. During the second phase, the ventricles begin to contract and their walls thicken. The AV valves close and the semilunar outflow valves open. Deoxygenated blood is sent through the pulmonary artery on its way to the lungs, while oxygenated blood enters the aorta, the largest artery in the body, from where it is distributed throughout the body, first through increasingly smaller arteries and finally through tiny capillaries that are found in all parts of the body and release oxygen to adjacent cells. The capillaries turn into small and then larger veins, which ultimately return the deoxygenated blood to the right atrium. The total volume of the heart does not change significantly during the heartbeat.

12.1.2 Modeling Targets at the Organ Level

At the macroscopic level of an organ, the heart acts like a peristaltic pump. Or, perhaps it is better to say that the heart is a combination of two pumps that drive different portions of the circulation of blood. Unlike most engineered pumps, the heart is highly sensitive to its mechanical surroundings and constantly adjusts influx and efflux in response to the current requirements of the body and various stresses. These adjustments occur through autonomic controls that do not require conscious input. In fact, very few people are capable of conscientiously affecting their heartbeat, although mental stresses such as worry or fear can easily alter it.

The adjustment mechanism of the heart to different pumping demands is characterized by the **Frank–Starling law** [22]. This law is derived from a nonlinear partial differential equation (PDE), developed in the nineteenth century and called the Young–Laplace equation, which generically relates pressure differences in fluids to the shape of the wall retaining the fluid. Broadly, it states that increased

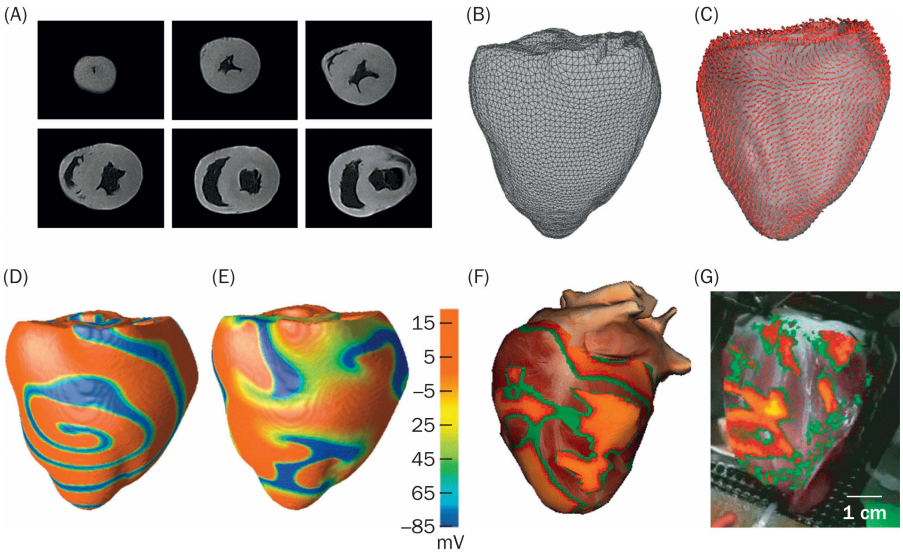


Figure 12.3 Images of ventricular tachycardia and fibrillation, which are potentially life-threatening arrhythmias that originate in one of the ventricles of the heart. (A) Transverse sections from a sophisticated MRI scan of a rabbit heart, proceeding from apex to base. (B) High-resolution reconstruction of the rabbit ventricular structure. (C) Orientation of fibers on the rabbit ventricular structure. (D) Simulation of ventricular tachycardia, in the form of a single spiral wave, in the rabbit ventricular structure. (E) Simulation of ventricular fibrillation, in the form of multiple spiral waves, in the rabbit ventricular structure. (F) Simulation of ventricular fibrillation in a reconstructed canine heart. (G) Experimental ventricular fibrillation in a canine wedge preparation recorded through optical mapping. Images of arrhythmias in motion can be found at the interactive website http://thevirtualheart.org. (Courtesy of Elizabeth M. Cherry, Alfonso Bueno-Orovio and Flavio H. Fenton, Cornell University. With permission from Elsevier.)

pressure requires increased wall thickness to maintain a stable wall tension. Applying these relationships to the heart, the Frank–Starling law formalizes how increasing the venous blood input to the ventricle stretches the ventricular wall, enhances contractility, and elevates the diastolic pressure and volume of the ventricle, which in turn leads to an increased stroke volume [23]. Thus, the heart uses an adaptive response mechanism that adjusts each ventricular output to its inflow and ensures the balance between the outputs of the two ventricles over time.

The balancing of venous input and **cardiac** output occurs without external regulation. The heart accomplishes this control task by adjusting the force of contraction proportionally to changes in the lengths of its muscle fibers. This adjustment is related to a decreased spacing of **filaments** within the **cardiomyocytes** (heart muscle cells), the formation of cross-bridges between filaments during each contraction of the heart muscle, and an increased sensitivity of the filaments to calcium (see later in this chapter and [24]).

The physical relationship between pressure and wall thickness expressed in the Frank–Starling law has been used to explain the gradual thickening of arteries and of the left ventricle wall in response to persistently high blood pressure. The unfortunate consequence of this trend is that a thicker left ventricle is stiffer than normal, so that filling it with blood requires elevated pressure. Over time, this altered pressure can lead to a decline in performance during diastole and eventually to diastolic heart failure.

Beyond the important yet somewhat generic insights gained from the Frank–Starling law, modern methods of **fluid mechanics**, engineering, computing, and modeling permit detailed and very sophisticated analyses of the spatial organization and three-dimensional architecture of the heart and their impact on proper functioning. These analyses are crucial for an understanding of healthy and perturbed blood flow in the atria and ventricles, as well as in the **coronaries**—the arteries and veins that provide the heart muscle with oxygen and nutrients and remove breakdown products. These vessels are rather narrow and prone to blockage, which can lead to atherosclerosis and heart attacks. Modeling blood flow as well as the mechanical deformation of the heart during the cardiac cycle requires systems of PDEs and **finite element approaches**, which subdivide a three-dimensional body into very small cubes and quantify changes within and between these cubes over time. Finite element models faithfully reflect the anatomy of the heart, including the orientation of fibers within the layers of the heart wall. They are fundamental to predicting whole-organ function based on features of the heart tissue that vary from person to person, for instance when the wall thickens owing to disease [16, 25]. Researchers at the École Polytechnique Fédérale de Lausanne (EPFL) have developed a very sophisticated three-dimensional supercomputer model of coronary blood flow that is able to predict heart attacks [26].

In addition to mechanical features and issues of blood flow, comprehensive organ-level models need to address the electrical activation of heart contractions, which initiate at the **sino-atrial node** and spread to the **Purkinje fibers** and the entire **myocardium** (heart muscle), as discussed in detail later. Fascinating pictures and movies of electrical activity waves can be found in [15, 27] and on many websites; an example is shown in **Figure 12.3**. Finally, all cells in the heart need energy, which requires efficient transport of metabolites and oxygen between the coronaries and the myocardium that needs to be taken into account in realistic organ models.

12.1.3 Modeling Targets at the Tissue Level

The pumping action of the heart is accomplished with contractile tissues that contain specialized cells, called cardiac cells, myocardial cells, or cardiomyocytes. These cells are in many ways similar to skeletal muscle cells, but, in contrast to the latter, the coordinated pumping action of the heart does not require our conscious input. Also, the heart muscle is very resistant to fatigue, because the cardiomyocytes contain many mitochondria, which are responsible for respiration and energy supply, and a large number of myoglobins, which are proteins that are able to store oxygen. Furthermore, the heart muscle is richly equipped with coronary blood vessels, which provide oxygen and nutrients to its cells. If the heart does not receive sufficient amounts of oxygen, an infarct results, leading to decreased cardiac function, tissue damage, and possibly death.

Models of the heart at the tissue level account for structural details and for properties of **electrical conductance**. Prominent structural aspects include the mechanics of the deformation of the heart tissue during contractions [12]. Because the heart contains roughly 10^{10} myocytes, a key concept of tissue-level modeling is the approximation of the discrete aggregate of these large numbers of cells and their interactions with a collective representation. This so-called **continuum approximation** permits the application of approaches gleaned from the physics of elastic materials, which deals with forces, stresses, strains, deformations, and distortions and helps explain shape changes during the cardiac cycle as well as the spread of electrical signals.

The electrical signals affecting the heart tissue and its billions of cardiomyocytes are coordinated by a primary pacemaker region in the wall of the right atrium, called the **sinoatrial (SA) node**. We will discuss this node several times in the following sections, so it suffices to state here that the SA node cells beat on their own, and that their oscillation in electric membrane charge is transduced to the **excitable** myocytes of the atria and ventricles. This traveling signal, originating at the SA node, ultimately makes the heart contract in a well-controlled manner. The SA node consists of a group of modified cardiac myocytes that spontaneously contract on the order of 100 times per minute.

The electrical signals generated in the SA node cause the atria to contract. They also spread along dedicated conduction channels in the walls of the atria and quickly reach the **atrioventricular (AV) node**, which constitutes the electrical connection between the atria and ventricles. Signals from the SA node reach the AV node with a delay of about 0.12 seconds. This delay may sound trivial, but it is very important because it ensures the correct timing between the ejection of blood from the atria and the contraction of the ventricles. The AV node contains a control system that slows down conduction if the node is stimulated too frequently. This mechanism, together with a **refractory period** where further triggering is not possible (see later), prevents contractions of the ventricles that are too rapid and could be hazardous. Signals from SA and AV nodes subsequently run through the conductive **bundle of His**, named after the Swiss cardiologist Wilhelm His, and to the **Purkinje fibers**, which branch out throughout the walls of the ventricles. The bundle and Purkinje fibers are specialized to conduct electrical signals rapidly to the myocardium. AV node and Purkinje cells can autonomously fire, but because their rate is slow, the electrical signals are normally dominated by the SA node.

The beating rate of the SA node is constantly modified by impulses from the **autonomic nervous system (ANS)**, which generally acts as a control system that maintains homeostasis in the body so that the ultimate heart rate under resting

conditions is only about 70 beats per minute but may go up to 200 beats during strenuous exercise. ANS activities occur without conscious control, sensation, or thought. However, the heart rate can be affected by noradrenaline (also known as norepinephrine)—the most common neurotransmitter of this system—and the hormone adrenaline (also known as epinephrine). We all know that sudden excitement or stress can lead to a burst in adrenaline (and noradrenaline) and sudden increases in heartbeat that get us ready for "fight or flight." Alas, phenomena like the ANS-controlled pacemaker activity in the SA node are usually more complex than they seem at first. As an analogy, consider the act of focusing on near or distant objects, which requires the combined voluntary actions of the extra-ocular muscles to move the eyes and of smooth muscles, controlled by the ANS, to change the shape of the lens. Clearly, the focusing task combines voluntary and involuntary actions. Similarly, the heartbeat is not entirely controlled by the ANS and, for instance, mobilization of the cardiovascular system is driven in advance of exercise to ready the musculature for motor activity.

As in the case of whole-organ studies, the spatial and mechanical properties of the heart tissue are of great importance at the tissue level, and computational (finite element) analyses are crucial for understanding the design principles underlying a successful electrical propagation from the pacemaker cells to the surrounding, excitable heart muscle [28]. For instance, it is important that the electrical coupling between pacemaker cells and excitable cells in the ventricles suppresses electrical interactions at inappropriate times. This suppression depends on a variety of factors, such as the intercellular coupling among the heart's pacemaker cells and differences in their membrane properties, as well as the communication among the surrounding cells, which depends on the number and spatial orientation of gap junctions connecting them. Thus, analyzing the spatial organization of the heart at the tissue level is important for a deep understanding of the correct propagation of electrical signals throughout the heart, the fine-tuned timing of contractions of the chambers, the pathological responses of the heart close to areas of infarction, and even for correct interpretations of ECG readings (see later).

12.1.4 Modeling Targets at the Cell Level

The cells of the heart have intriguing specific features, but of course they also contain generic components that one would have to include in a comprehensive model of the heart. Not surprisingly, heart cells express different genes than other cells, their metabolism requires more energy, they contain proteins some of which are uncommon in other cells, and some of their signaling systems are germane to the function of the heart. Like skeletal muscle cells, most cells of the heart muscle are contractile, and the mechanics of the responsible filaments constitutes an interesting target for modeling. The electrical activity of these cells is tightly associated with the flux of ions, so that ion transport and electrochemical models demand special attention (see later). Finally, specifically adapted genomic, proteomic, and metabolic models are required for understanding the particular features of the heart. Some interesting modeling work in this context is presented in [29, 30].

Of crucial importance is that the heart contains a number of different cell types, two of which are of particular relevance here, namely cells of the SA node and excitable cardiomyocytes. The cells in the SA node are modified cardiomyocytes and constitute only about 1% of all heart cells, but they are of immense importance. SA node cells exhibit a number of physiological differences in comparison with ventricular cells. For instance, while they do contain contractile filaments, they do not contract. The most important distinguishing feature is the ability of SA node cells to beat (**depolarize**) spontaneously (discussed in detail later), which has led to the terminology of SA node cells as **pacemaker cells**. This electrical depolarization, which is caused by ion fluxes across the cell membrane, makes the membrane more positive and results in an **action potential**, which ultimately spreads throughout the heart muscle, as we will discuss in more detail later. The spontaneous depolarization in SA node cells occurs at a rate of about 100 times per minute, which, however, is constantly modified by the ANS. Thus, SA and AV node cells are clearly different from most other cardiomyocytes, but there are also significant differences

within the latter group, for instance, with respect to the characteristics of their **resting potentials** and the shapes of their action potentials [31].

Interestingly, the transition between cells in the SA node and the peripheral atrial cells is smooth, with one cell type gradually changing into the other within a region surrounding the SA node. While this spatial transition might seem to be a subtle detail, it has turned out to be very important. Using computational three-dimensional tissue reconstructions, recent research has shown how cell types and their communication channels are distributed in these transition regions and how this spatial heterogeneity is crucial for proper signal propagation. In fact, even within the SA node, one finds differences in morphology and action potentials between central and peripheral regions [28].

The spreading of the electrical impulse from cell to cell occurs through gap junctions, which are specialized sets of half-channels connecting the cytoplasm of two neighboring cells and permitting the free flow of ions and some other molecules (see Chapter 8). Gap junctions are of utmost importance to the heart, because they allow the efficient passing of ion-based electrical signals from one cell to another and thereby enable the coordinated depolarization and contraction of the heart muscle. In addition to the spreading of electrical charge through gap junctions, the electrical signal is transduced through the bundle of His and Purkinje fibers, which consist of specialized cell types that conduct five to ten times as fast as signal transduction through gap junctions.

The receivers of the electrical impulses from the SA and AV nodes and the Purkinje fibers are the regular, excitable cardiomyocytes. These elongated cells slightly differ in structure and electrical properties between the atria and the ventricles, but their important common property is that they are contractile. This ability to contract is due to large numbers of protein filaments—or **myofibrils**—which are a few micrometers long and similar to those found in skeletal muscle cells. The contraction itself is a complex process, which is ultimately accomplished by the sliding motion of two myofibrils, **actin** and **myosin**. In simplified terms, the myosin filaments contain numerous hinged heads, which form cross-bridges with the thinner actin filaments (**Figure 12.4**). By changing the angle of the hinges, the actin and myosin filaments slide against each other in a ratcheting manner, for which the energy is provided by hydrolysis of adenosine triphosphate (ATP). As a result of the sliding action, the entire cell contracts and becomes shorter and thicker. A review of some of the pertinent modeling activities is given in [32].

In reality, the myofibrils are components of a complicated multi-protein complex that includes the important proteins titin and troponin. The details of the sliding motion are quite involved, and it suffices here to state that they are coupled to oscillations in the cytosolic calcium concentration (see later and, for example, [4, 33]). When the calcium concentration increases, the cell contracts. When the concentration decreases, the filaments slide in opposite directions, and the cell relaxes. This tight correspondence is called **excitation–contraction (EC) coupling** [13].

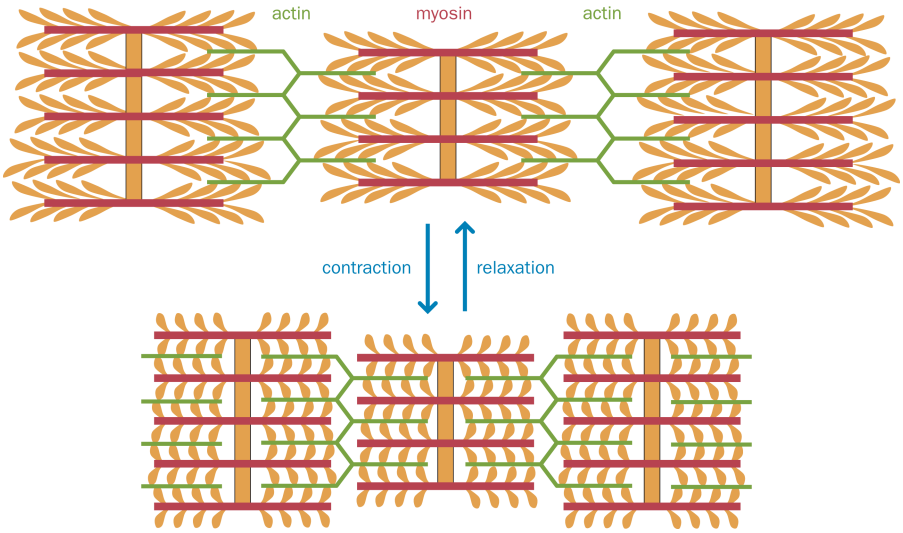


Figure 12.4 Simplified representation of the sliding action of myofilaments, which causes a cardiomyocyte to contract. The myosin filaments (brown) contain hinged heads (tan) that form cross-bridges with the thinner actin filaments (green). A change in the angle of the hinges leads to sliding and contraction. In reality, this process is much more complicated.

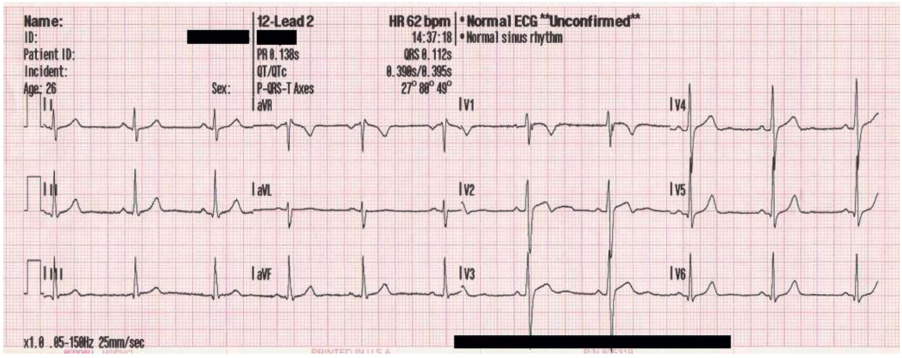


Figure 12.5 A 12-lead ECG of a healthy young man. Electrical currents generated during the cardiac cycle spread throughout the body and can be measured by an array of electrodes placed on the skin. See the text for further explanation.

Because the filament movement requires a lot of energy, cardiomyocytes contain many mitochondria.

Nobel Laureate Andrew Huxley proposed the first sliding filament model. While this model is not as prominent as his model of nerve action potentials, it has served as a basis for an entirely different class of models than the electrical models of nerve and heart cells.

Importantly, the contraction process in cardiomyocytes is initiated by electrical impulses and the flow of ions, as we will discuss later in greater detail. This electrical excitation in a cardiomyocyte is followed by a refractory period during which the cell normally does not respond to further electrical excitations. After this period, the cells are ready to fire again.

The membranes of cardiomyocytes change in their electrical charge once during every heartbeat. As a consequence, an electrical charge spreads throughout the heart and to the surrounding tissues and ultimately causes very subtle electrical changes on the skin. These changes can be measured over a period of time with electrodes attached to the skin and are interpreted with methods of electrocardiography. Because the electrical impulses move as waves throughout the heart and are modified by the tissues of the body, the shape of the resulting **electrocardiogram** (**ECG**; or **EKG** from the German analog) depends on the three-dimensional architecture of the heart and the exact locations of the electrodes on the body. In order to assess heart function as comprehensively as feasible, it is therefore common practice to place several sets of electrodes at defined locations on the chest and limbs. The ECG displays the voltages between pairs of electrodes, and their spatial as well as temporal interpretation gives clues regarding the overall health status of the heart, as well as abnormal rhythms that may be due to imbalances in electrolytes or to damage to the heart tissue in specific locations; some details will be discussed later. The ECG can also identify—and to some degree locate—tissue damage in the case of a myocardial infarction (MI; heart attack). Since the ECG only measures electrical activity, it does not truly measure the heart's ability to pump sufficient quantities of blood. The amounts of blood flowing through the atria and ventricles can be measured with an ultrasound-based echocardiogram or with methods of nuclear medicine. Figure 12.5 shows a 12-lead ECG of a healthy young man.

Thus, we started with organ-level models, pointed to modeling needs at the tissue level, discussed features of heart cells, and have now returned to their effects as they are manifest at the whole-body level.

12.2 SIMPLE MODELS OF OSCILLATIONS

The information we have discussed so far is already sufficient for some very simple mathematical heart models. Let us start by focusing on the most obvious—the oscillatory pattern. This oscillation can be observed in many forms, for instance in the ECG but also in the increasing and decreasing volume of blood in a ventricle, cyclic changes in electrical voltage, or rhythmic changes in the deformation of the ventricular wall. All these derivative oscillations ultimately originate from the autonomous depolarization and repolarization of the cells in the SA node, which result in a relatively regular oscillation in voltage (Figure 12.6). Here the term "autonomous" means that the cells exhibit sustained oscillations without needing repeated triggering events.

An intriguing aspect of these oscillations is that small perturbations are easily tolerated. In other words, if something temporarily disturbs or modulates the rhythm or the amplitude of the oscillation, the healthy heartbeat soon returns to the original rhythm. Of course, this tolerance is absolutely necessary for the heart to operate in a healthy way. Just imagine going to a horror movie, where a gruesome zombie attack makes your heart race. For a moment, you may enjoy the excitement, but you would certainly like your heart to go back to normal after a while and without conscious prodding.

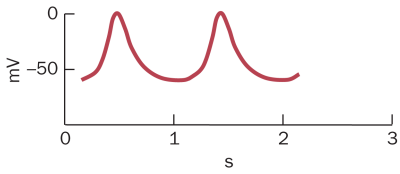


Figure 12.6 Oscillatory change in the voltage of an SA node cell. Different cell types and cells in different locations of the heart exhibit distinctly different shapes (see Figure 12.12), and these are again very different from the various patterns in an ECG (see Figure 12.5) [1, 29].

12.2.1 Black-Box Models of Oscillations

As we discussed in Chapter 5, tolerance to perturbations is a hallmark feature of stable limit-cycle oscillations. We mentioned different formats, including the van der Pol oscillation [34, 35], which was proposed many decades ago as a model for heartbeats, and we return here to the very flexible limit cycles based on S-systems [36]. Specifically, we consider the two examples in **Figure 12.7**, which correspond to the model equations

$$\begin{aligned}
\text{System A:}\quad & \dot{X}_1 = 2.5\cdot(X_2^{-6} - X_1^3 X_2^{-3.2}), && X_1 = 0.9,\\
& \dot{X}_2 = 1.001\cdot 2.5\cdot(1 - X_1^{-5} X_2^{-3}), && X_2 = 1.2;\\
\text{System B:}\quad & \dot{X}_1 = 1.005\cdot(X_2^{-8} - X_1^7 X_2^5), && X_1 = 0.25,\\
& \dot{X}_2 = X_1^6 X_2^5 - X_1^{-1} X_2^{-4}, && X_2 = 2.5.
\end{aligned} \tag{12.1}$$

Systems A and B lead to oscillations that are vaguely reminiscent of the dynamics in the SA node (see Figure 12.6) and of ECG traces (see Figure 12.5), and we can easily confirm that both oscillations are indeed stable, namely, by briefly perturbing them and watching them return to their original oscillations.

If stable oscillations are repeatedly "poked" with the "wrong" frequency, bad things can happen. Mathematically speaking, a stable limit cycle that is exposed to an oscillating stimulus pattern may become chaotic: although everything in the system is entirely deterministic, the oscillatory behavior is no longer predictable, unless one actually solves the equations (Chapter 5). The analogy with heart physiology is evident: repeated spurious electrical signals can affect the system, and the

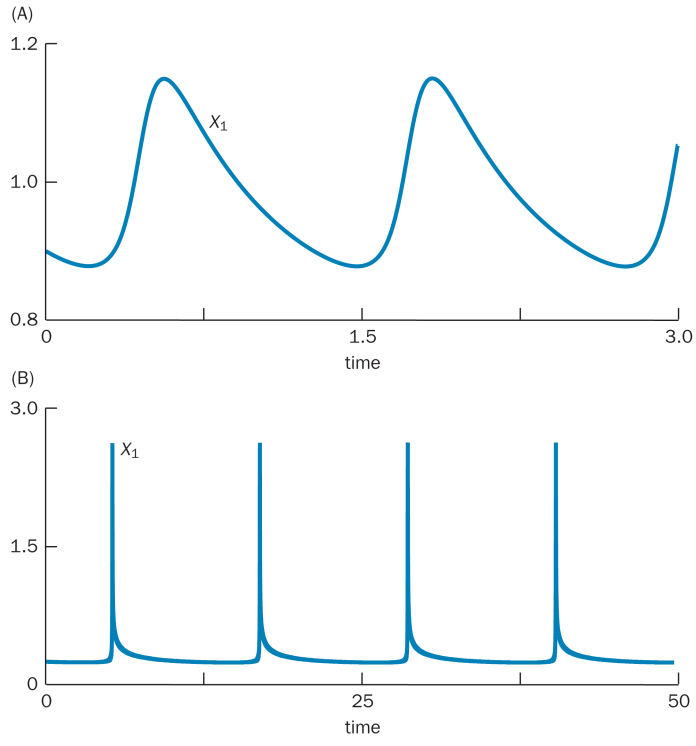


Figure 12.7 Stable S-system oscillations. (A) Variable X_1 of System A in (12.6), which resembles the oscillations in the SA node (see Figure 12.6). (B) Variable X_1 of System B in (12.6), which is similar to a simplified ECG trace (see Figure 12.5).

result may be an unpredictable beating pattern. Specific examples are blockages in the AV node or the fast conducting bundles, which may lead to repeated additional electrical signals that can interfere with the usual electrical patterns in the heart and cause arrhythmia and even death. At the same time, one must be careful with the interpretation of irregularities: while chaotic oscillations may be a sign of disease, a fetal heartbeat that is too regular is often a sign of trouble [37–39]. Indeed, the loss of variability in oscillations has been associated in recent times with different pathologies, frailty, and aging [40, 41].

As a numerical example, suppose the stable oscillatory System B in (12.1) is subjected to a small regular oscillation. We model this idea by adding $0.01(s+1)$ to both equations, where s is the value of a sine function, which we discussed in the form of ODEs (Eqns. 5.3 and 5.51 in Chapter 5), so that $s+1$ oscillates between 0 and 2. In addition, we introduce an extra parameter a, which permits us to change the frequency of the sine oscillation. Thus, our heartbeat together with spurious electrical signals is given by the following four-variable model:

$$\begin{aligned}
\dot{X}_1 &= 1.005\left(X_2^{-8} - X_1^7 X_2^5\right) + 0.01(s+1), \quad X_1 = 0.25, \\
\dot{X}_2 &= X_1^6 X_2^5 - X_1^{-1} X_2^{-4} + 0.01(s+1), \quad X_2 = 2.5, \\
\dot{s} &= a\,c, \qquad\qquad s(0) = 0, \\
\dot{c} &= -a\,s, \qquad\qquad c(0) = 1.
\end{aligned} \tag{12.2}$$

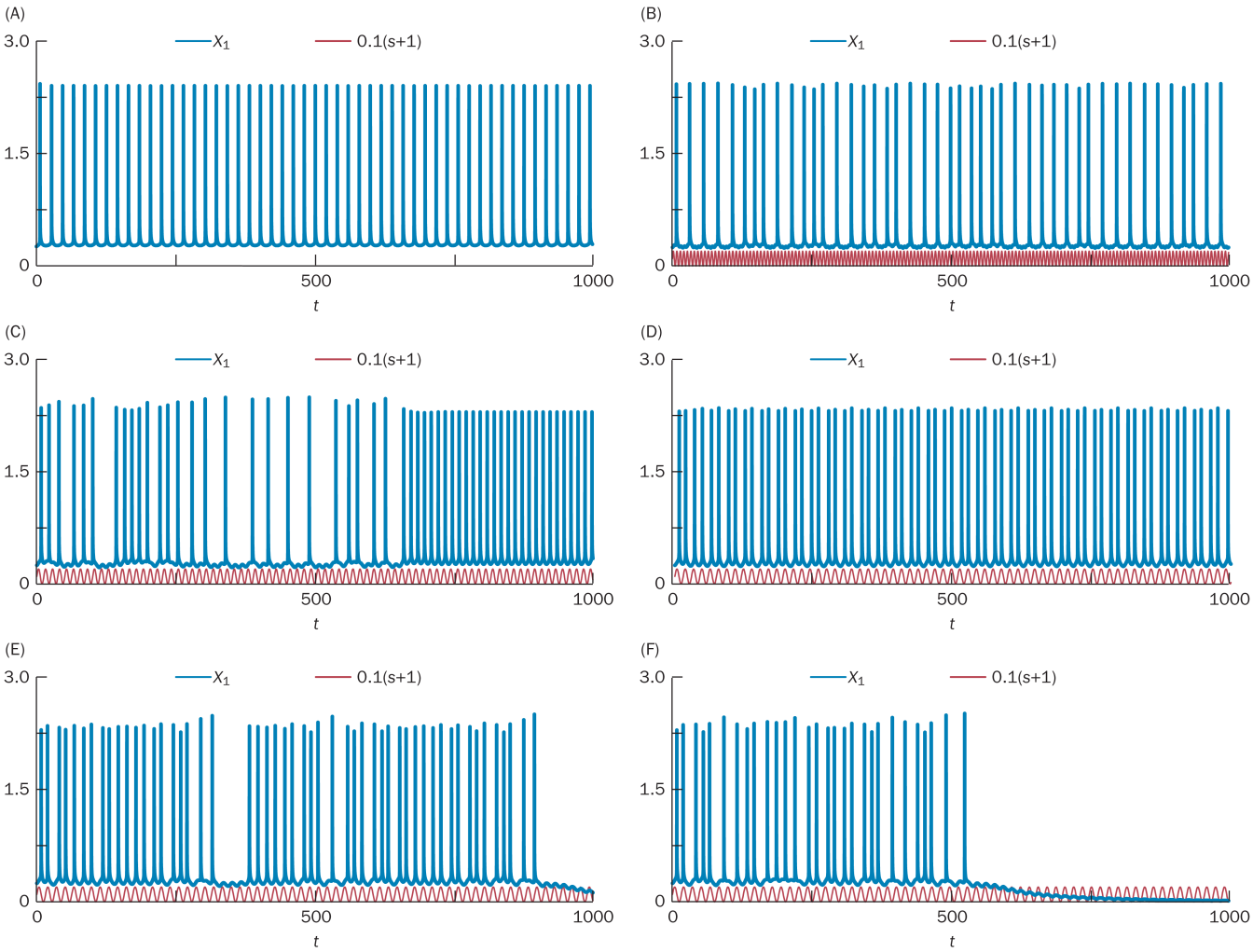


Figure 12.8 Simulation of a perturbed heartbeat, ultimately leading to fibrillation and death. If a stable oscillator is regularly "poked," for instance with a sine function, the results may be negligible or detrimental, depending on how the frequencies of the two oscillators relate to each other. The frequency of the sine oscillator is determined by a. See the text for details.

If $a = 0$, the frequency of the sine system is the same as before. Larger values mean faster oscillations, smaller values slower oscillations. If $a = 0$, the sine system does not oscillate at all (The reader should confirm this with mathematical arguments and with simulations!), which permits a good comparison between systems with and without spurious stimuli. The system is in this case almost the same as in (12.1), except that both equations have an extra term $+0.01$, which is apparently negligible. The oscillations of this system (without spurious oscillations) are shown in **Figure 12.8(A)** for a long time horizon.

In the subsequent panels (Figure 12.8B–F), the frequency parameter is varied; the sine oscillation itself is visualized for clarity with a 10-fold magnification of its amplitude. If $a = 10$, the sine oscillation is so fast that it does not change the appearance of the heart oscillation (not shown but very similar to Figure 12.8A). In Figure 12.8(B), with $a = 1$, the heartbeat is not quite even. The next panel (Figure 12.8C), with $a = 0.5$, spells trouble, with quite arrhythmic oscillations, which seem to recover after about 700 time units. However, the heart rate is much increased. Further lowering the parameter a seems to remedy the situation, but if we look closely, the heart goes into double beats. For $a = 0.40$, the heart starts skipping beats, and even very slight further decreases in a cause the heartbeat to stop. The smaller a is, the faster this happens (Figures 12.8D–F).

We can see that even simple models are capable of capturing the essence of the heartbeat and point to some very serious problems. Because these models are so simple, they are well suited for rigorous mathematical analysis. Indeed, many researchers have studied limit cycles in this and other contexts and explored their vulnerability with respect to input oscillations of an unfavorable frequency (see, for example, [42, 43]).

One may also use these simple models to explore whether a highly irregular heartbeat can be made regular by a pacemaker and how robust such a correction might be. Pacemakers are implanted electronic devices that send out electrical signals with the goal of correcting troublesome oscillations. If adjusted correctly, a modern pacemaker can ensure that the heart beats with the right speed at the right time.

To assess the function of a very simplistic pacemaker, we model the irregular heartbeat with the chaotic Lotka–Volterra (LV) oscillator (see Chapter 14 for details). To resemble the timing of a heartbeat, we multiply all equations by 50, which results in a rather erratic heartbeat with about 60 to 70 beats per minute. We model the overly simplistic pacemaker with a sine oscillator, represented as a pair of ODEs ($\dot{s} = qc$, $\dot{c} = -qs$; see previous mention and Chapter 5) and add to each equation in the LV-system a term of the form $p \cdot (s + 1)$. A typical result, with $q = 6$, is shown in **Figure 12.9**. Initially we set $p = 0$, which corresponds to the unaffected erratic heartbeat. At $t = 8$, we reset $p = 0.05$. One can see that it takes several beats but that the pacemaker indeed tames the chaos. In reality, pacemakers are much more sophisticated and respond to different demands of the body that require the heart to beat faster or more slowly. Exercise 12.5 explores this topic in more detail.

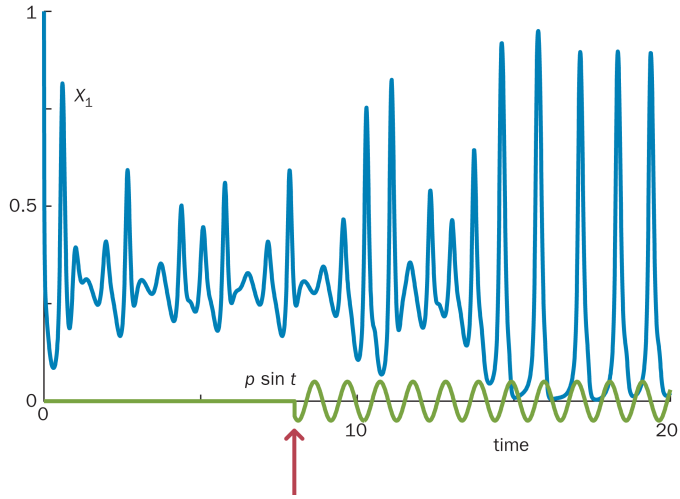


Figure 12.9 Effectiveness of a simple pacemaker. An irregular heartbeat, here modeled as a chaotic Lotka–Volterra (LV) oscillator, becomes regular through the effect of an appropriate sine function. Only the first variable of the LV system is shown, along with a scaled sine function, which oscillates between –0.05 and 0.05, starting at $t = 8$. For better visibility, the sine function is not shifted in this graph, although it is shifted by 1 in the model.

12.2.2 Summary of Black-Box Oscillation Models

Rather simple oscillation models can represent some of the phenomena that have been observed in well-functioning and diseased hearts, such as arrhythmias, beat skipping, the potentially detrimental effects of spurious electrical signals, and the basic function of pacemakers. So, we have to ask ourselves whether these simple models are sufficient. As is so often the case, the answer depends on the purpose of the modeling effort. For instance, if we want to demonstrate arrhythmias, stable black-box oscillation models might be most instructive, because nothing distracts from the simplicity of their mathematical structure. Or if we plan to use the heartbeat as an input module for a larger model and we are not specifically interested in mechanistic details, a small ordinary differential equation model might be a good choice. As an example, suppose we want to study the metabolic responses of cells to altered oxygen supply. In this case, the mechanisms that fundamentally lead to oscillations in blood oxygen may not be particularly relevant. In fact, since larger models always require more effort in terms of parameter estimation and diagnostics, investing more time on the oscillator may not be worth the effort. Also, as mentioned earlier, a lot can be learned mathematically about small limit cycle oscillators, and many books have used simple models like those shown earlier for analyzing different types of oscillations, questions of structural stability, and bifurcation points in nerves, hearts, and other oscillating systems, which would not be possible in larger models [8, 42, 43].

While small models have strong advantages, they obviously have limitations as well. For instance, if we want to dissect a disease pattern into possible molecular, physiological, or electrochemical causes, the small models simply cannot provide adequate answers. It is not even possible to ask the right questions, because these models do not contain the relevant components, such as a membrane, concentrations of ions, or gated channels, which are very important for proper functioning, as we will see later in this chapter. In order to explain the processes that are ultimately responsible for a regular heartbeat and for the purposes of studying and treating heart disease, we need to dig deeper, down to the real molecular and physical drivers, formulate them quantitatively, and then use mathematical models and computation to integrate results from molecular and electrochemical levels to gain insight into macroscopic function or failure. We begin with an exploration of calcium dynamics, which is crucial for proper function, develop a model for this particular aspect, and then work toward more comprehensive models that include chemical and electrical processes. To venture toward such more sophisticated models, we need to understand more about the biology of the heart: what makes the heart tick?

12.2.3 From a Black Box to Meaningful Models

We have seen that it is surprisingly easy to construct oscillations and even limit cycle oscillations as black-box models. The question is now whether we can construct a simple yet biologically meaningful model that exhibits sustained and robust oscillations. One approach, which we will pursue here, is to study the movement of ions during the cardiac cycle.

The chain of contraction and relaxation events in heart cells, including SA node cells, is closely associated with the periodic movement of calcium ions between three locations, namely the extracellular fluid, the cytosol, and an intracellular compartment, called the **sarcoplasmic reticulum (SR)**. The cycle begins when the membrane of a cell depolarizes, which is associated with an influx of sodium and a relatively small amount of calcium from the extracellular space into the cytosol. The calcium binds to proteins in the membrane of the SR, where it triggers a mechanism called **calcium-induced calcium release (CICR)**, which results in the flow of large amounts of calcium from the SR into the cytosol [44, 45]. In regular cardiomyocytes, this calcium binds to troponin and leads to the sliding action of the myfibrils actin and myosin, as discussed before, and this sliding action causes contraction of the myocyte. Calcium now disengages from the myofibrils. Most of it is pumped back into the SR, and some leaves the cell through specialized calcium exit carriers that are located on the cell's membrane and simultaneously

export calcium and import sodium. The cell relaxes and is ready for a new contraction.

In regular cardiomyocytes, the initial depolarization is triggered by electrical signals coming from the SA node or from neighboring cells, as discussed earlier. In contrast, the membrane in an SA node cell depolarizes spontaneously, starting from a minimum charge of about –60 mV (see Figure 12.6). While the initial trigger is different, the release of calcium from the SR also happens in the SA node [45]. Thus, both SA node cells and ventricular cardiomyocytes depend on the movement of calcium among three compartments.

The question then becomes whether a mathematical model of the three-compartment calcium dynamics can in principle lead to sustained, stable oscillations, as observed in cardiomyocytes. Because contractions in heart and skeletal muscle, as well as calcium bursting patterns in organs such as the pancreas, are of great clinical importance, it is not surprising that there is plenty of literature on models of calcium oscillations. For a simple proof of principle, we only look at one minimalistic model that captures the calcium dynamics nicely and quite easily and serves our purposes of providing a biologically motivated oscillator. This model was proposed by Somogyi and Stucki [46, 47], and many more complex models followed later; a review is given in [48].

The Somogyi–Stucki model describes a dynamical system that consists of the three physical spaces mentioned earlier and calcium movements among them (**Figure 12.10**). These spaces are the cytosol, the SR inside the cell, and the extracellular space. The model assumes that the extracellular space contains such a high concentration of calcium, $[Ca^{2+}]_{ext}$, in comparison with the cytosol that it can be considered an independent, constant variable. The calcium concentrations in the SR and the cytosol are denoted by $x = [Ca^{2+}]_{SR}$ and $y = [Ca^{2+}]_{cyt}$, respectively, pump rates by p, simple transport rates by r, and channel transport by ct, with subscripts indicating source and target. According to Somogyi and Stucki, the following assumptions are made in the construction of the model:

1. The extracellular space serves as an inexhaustible source for an influx of $Ca^{2+} = [Ca^{2+}]_{ext}$ at a constant rate r_{EC}.
2. Ca^{2+} is pumped from the cytosol into the extracellular space at a constant rate p_{CE}.
3. Ca^{2+} is pumped from the cytosol into the SR at a constant rate p_{CS}.
4. Ca^{2+} leaks from the SR into the cytosol proportionally to the concentration difference between the SR and the cytosol and at a rate r_{SE}.
5. Ca^{2+} is actively pumped from the SR into the cytosol. The process depends on the concentration difference between the SR and the cytosol, and its rate is a sigmoidal function of the concentration in the SR.

The equations for this small dynamic system are easily set up according to the guidelines discussed throughout earlier chapters. Because we are interested in time-dependent processes, we use differential equations, and if we ignore spatial details, for reasons of simplicity, these equations are ordinary differential

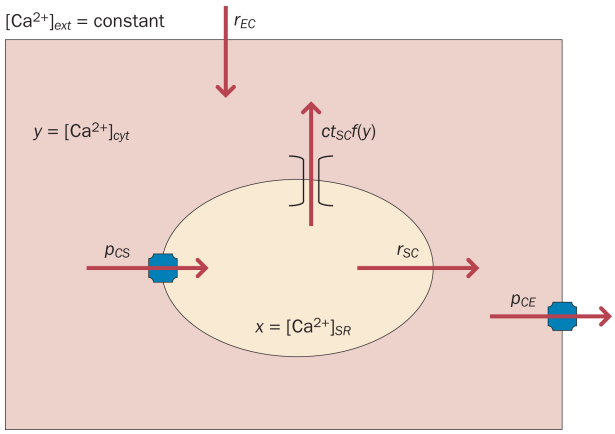


Figure 12.10 Simplified diagram of calcium flow associated with a myocyte. The illustration shows the processes and symbols used in the Somogyi–Stucki model, which leads to stable calcium oscillations. The red area represents the cytosol of the cell, while the pale area represents the sarcoplasmic reticulum. The notation is explained in the text.

equations (ODEs). Thus, we formulate ODEs for the two dependent variables, x and y and include on the right-hand sides all relevant variables, functions, and rates. To implement this strategy, we make the simplifying assumption that all processes can be adequately represented as linear functions. This assumption is typical for transport processes and for processes whose turnover rates are proportional to their substrate concentrations. The only exception is $f(y)$, which describes the channel transport. For this function, Somogyi and Stucki use a Hill function with Hill parameter h, which saturates at the maximal rate $(ct)_{SC}$. Thus, the nonlinear channel transport process is modeled as

$$f(y) = (ct)_{SC} \frac{y^h}{K^h + y^h}. \tag{12.3}$$

Taken together, x is affected by transport into and out of the SR, and y is affected by transport into and out of the cytosol, and the mathematical formulation is

$$\begin{aligned} \dot{x} &= p_{CS} y - r_{SC}(x-y) - (ct)_{SC} \frac{y^h}{K^h + y^h}(x-y), \\ \dot{y} &= r_{EC} + r_{SC}(x-y) + (ct)_{SC} \frac{y^h}{K^h + y^h}(x-y) - p_{CE} y - p_{CS} y. \end{aligned} \tag{12.4}$$

As in so many cases, it is not easy to pull suitable parameter values out of a hat (see Chapter 6). Since we are not really interested in parameter estimation here, we simply take values similar to those used by Somogy and Stucki [46, 47], namely

$$\begin{aligned} p_{CS} &= 4.5, & r_{SC} &= 0.05, & (ct)_{SC} &= 3, \\ h &= 4, & K &= 0.1, \\ r_{EC} &= 0.075, & p_{CE} &= 1.5. \end{aligned}$$

With these values, the system becomes

$$\begin{aligned} \dot{x} &= 4.5y - 0.05(x-y) - 3\frac{y^4}{0.1^4 + y^4}(x-y), \\ \dot{y} &= 0.075 + 0.05(x-y) + 3\frac{y^4}{0.1^4 + y^4}(x-y) - 6y. \end{aligned} \tag{12.5}$$

The initial values turn out to be rather unimportant and we set $x_0 = 2$, $y_0 = 0.1$. The result of a simulation with these conditions is shown in Figure 12.11. We see that this rather simple model of the calcium dynamics in a cell is indeed able to oscillate. In fact, perturbations demonstrate that it oscillates all by itself in a stable manner. Our proof of principle is complete.

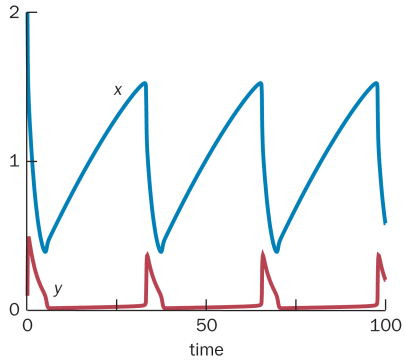


Figure 12.11 Oscillations generated by the Somogyi–Stucki system (12.4). Even without external cues, the system oscillates in a stable manner. The variables x and y represent calcium concentrations in the SR and the cytosol, respectively.

12.3 ELECTROCHEMISTRY IN CARDIOMYOCYTES

The oscillations in the heart are ultimately due to ion movements and electrical charges, but we have not really discussed how these two are related. The key to understanding these relationships lies in transport processes at the cell membrane and the membrane of the SR. A superb description of these cross-membrane processes can be found in [49]; much of the following is excerpted from this source.

The transport processes across the membranes are tied to two distinctly different yet intimately related gradients, namely a concentration gradient and an electrical (charge) gradient. While the cell contains many types of molecules, the concentrations of interest here are those of ions. To refresh our memory, ions are atoms or molecules carrying electrical charges that result from a difference between their numbers of electrons and protons. In the case of an atom, the dense nucleus contains protons, which are positively charged and neutrons, which are

electrically neutral. The nucleus is surrounded by a cloud of negatively charged electrons. In a negatively charged ion, also called an anion, the cloud contains more electrons than there are protons in the nucleus. The most important examples here include ionized atoms such as chloride ion and some larger molecules such as negatively charged proteins. Positively charged ions, called cations, have one or several more protons in the nucleus than electrons in the shells of their electron clouds. The most important examples for membrane activities are sodium, potassium, and calcium ions.

In the case of a cell that is surrounded by a watery medium and separated from the medium by a porous membrane, ions can be transported in and out through the membrane either by means of specific transport mechanisms or, to some degree, through leakage. As we know from many real-life situations, differences within gradients tend to equalize if left alone. Expressed in the terminology of physics, the total energy of the system approaches the minimum that is achievable in the given situation. Hot coffee in a cool room gets cold and eventually reaches the same temperature as its surroundings. Sugar poured into a glass of water dissolves through random molecular movements and its concentration eventually becomes homogeneous. This natural tendency to equalize concentration differences also applies to concentrations inside and outside a cell. However, in the process of equalizing the concentrations of ions, it is possible that the charges become unbalanced. Thus, the total energy of the system is composed of two components: the chemical energy associated with molecular movement within the concentration gradient and the electrical energy associated with charges. Nature tends to minimize the sum of these two components, and, given enough time, the sum becomes zero. We will discuss this aspect later more rigorously.

The gatekeeper of ion movements between the cytosol and the extracellular space is the cell membrane. It consists of a lipid bilayer that creates a substantial barrier for polar molecules and ions in particular, because of strong attractive electrostatic forces between ions and water molecules. This impediment to free ion flow is crucially important, because it allows an excitable cell, such as a cardiomyocyte, to create and retain concentrations and electric charges in its cytosol that differ drastically from the conditions in the surrounding extracellular fluid. The cell manages the creation of concentration gradients across membranes—and thus the storage of potential energy and the possibility of electrical signal transduction—through transmembrane **channel proteins** and **carrier proteins** that are often specialized to transport particular ions or water-soluble organic molecules into or out of the cell.

Channel proteins form hydrophilic, water-filled pores that cross the lipid bilayer and allow select ions to travel through the membrane. This travel is very fast, with up to 100 million ions being able to pass through a channel within 1 second ([49], p. 667). With such speed, channel transport can be 100,000 times faster than transport that is facilitated by carrier proteins, a property that is of great import in electric signaling. The most characteristic feature of channels is that transport must always occur down the concentration gradient. In the case of ions, this passive transport results in a change in membrane potential, which in turn may influence further ion transport. Because most cells exhibit a negative charge toward the inside of the membrane, the entry of positively charged ions is favored. A fascinating feature of channel proteins is their fine-tuned specificity, which is a consequence of their molecular structure. A pertinent example is the potassium leak channel, which conducts K^+ ions 10,000 times better than it does Na^+. This difference is intriguing, because a sodium atom is slightly smaller than a potassium atom. Nature solved this problem with a potassium leak channel protein that creates a specially charged ion selectivity filter, which almost exclusively prevents sodium from traversing the membrane [49].

In contrast to simple aqueous pores, carrier proteins and ion channels are controllable by the cell. Specifically, they may be opened and closed through electrical or mechanical means, as well as through the binding of ligands. Most important among the channels are **voltage-gated** and receptor-gated channels that are, respectively, controlled by voltage and by binding of molecules such as the neurotransmitter acetylcholine.

Carrier proteins, also called **pumps** or permeases, are transporters that bind a specific ion and, while undergoing a transformational change, actively facilitate its transport through the membrane. Carrier proteins are specific with respect to the molecules they transport and require energy, which may be supplied, for instance, in the form of ATP. Their dynamics resembles that of enzyme-catalyzed reactions and shows a concentration dependence resembling that of a Michaelis–Menten process (see Chapters 5 and 9), with a characteristic binding constant K_M and saturation at a maximal rate V_{max} for high ion concentrations. Some carrier proteins also permit passive transport in either direction, as long as it occurs down the concentration gradient, but their main role is active transport against the electrochemical gradient.

Carrier proteins come in several variations. In the first case, uphill transport (against the concentration gradient) is coupled to downhill transport of another molecule. If both the ion and the other molecule are transported in the same direction, the transport protein is called a symporter. If they are transported in opposite directions, we talk about an antiporter. For example, moving an ion down the gradient releases free energy that can be used to move a larger molecule, such as a sugar, up the gradient. The second type of a carrier protein is a uniporter. In this case, only one ion is transported and the process is coupled to the hydrolysis of ATP. It is also possible that an antiporter is combined with ATP hydrolysis. As a further possibility, which, however, is not of relevance here, bacteria may use light as an energy source for ion transport against the gradient.

The most important transport process in our context is the **Na^+–K^+ pump**, which hydrolyzes ATP. Because Na^+ is pumped out of the cell, against a steep electrochemical gradient, while K^+ is pumped in, the transporter is an antiporter, and, since it uses ATP, it is sometimes called a Na^+–K^+ ATPase. The pump can also run in reverse, where it generates ATP from ADP. The Na^+–K^+ pump functions in such a manner that during each cycle three Na^+ ions are pumped out, while only two K^+ ions are pumped in. As a consequence, the pump changes the charge balance and is therefore called electrogenic, that is, it alters the membrane potential. However, the electrogenic change is small in comparison with other processes and contributes little to the electrical state of the cell's membrane.

A second crucial antiporter protein for the transport of ions during the cardiac cycle is the sodium–calcium exchanger (**Na^+–Ca^{2+}exchanger**), which removes calcium from the cell, in particular after an action potential. It allows sodium to flow across the plasma membrane, down its electrochemical gradient and thereby transports calcium ions in the opposite direction. The removal of calcium is very fast, with several thousand calcium ions crossing the membrane per second. This speed is necessary for effective signal transduction, but it is also expensive: the removal of each calcium ion in this process requires the import of three sodium ions. The transport through the Na^+–Ca^{2+} exchanger is electrogenic, and strong depolarization of the membrane can actually reverse the direction of ion exchange, which is possible under the influence of some toxins and drugs. Alterations in the activity of the Na^+–Ca^{2+} exchanger can have serious consequences and lead to ventricular arrhythmias [49].

An additional pump that removes calcium is the plasma membrane Ca^{2+} ATPase (also known as the PMCA), which has a higher affinity for calcium than the Na^+–Ca^{2+} exchanger, but a much lower transport capacity. Because of its higher affinity, the Ca^{2+} ATPase is effective when the calcium concentration is low, and thus it complements the Na^+–Ca^{2+} exchanger. The Ca^{2+} ATPase is well understood in terms of its structure and function ([50], pp. 660–661).

Normally, the Na^+–K^+ exchanger sets up a sodium gradient that drives the Na^+–Ca^{2+} exchanger to extrude calcium. The foxglove plant (*Digitalis purpurea*) produces the toxic substance digitalis, which is used as a medication to control arrhythmias of the heart. Digitalis functions by poisoning the Na^+–K^+ exchanger and thereby decreases the Na^+ gradient, which in turns reduces Ca^{2+} extrusion. The accumulated intracellular Ca^{2+} promotes contraction [51].

The channels are the mediators for transfer of ions into and out of cardiomyocytes, and because ions are electrically charged, their transport can lead to changes in the electrical state of the cell and, specifically, its membrane. How is it possible that these transport processes lead to autonomous oscillations?

12.3.1 Biophysical Description of Electrochemical Processes at the Membrane of Cardiomyocytes

Ion transport and changes in the cell's electrical state constitute one of the rather rare cases in biological systems analysis where processes are so close to processes in physics that the corresponding laws of physics can be used directly for explanations or at least as an inspiration. Here we borrow heavily from electricity theory. As discussed before, nature tends to minimize energy, which in the case of excitable cells consists of two components, namely the chemical energy associated with molecular movement within the concentration gradient and the electrical energy associated with charges. Given enough time without outside influences, the sum of the two sources of energy approaches zero.

Formulating this fundamental phenomenon in the terminology of physics leads to the **Nernst equation**, which quantifies electrochemical gradients. One begins with the free-energy change per mole ion, ΔG_{conc}, which is associated with the concentration gradient across the membrane. Nernst found that this quantity can be computed as $-RT\ln(C_o/C_i)$, where R is the gas constant, T is the absolute temperature in Kelvin, and C_o and C_i are the concentrations of ions outside and inside the cell, respectively. Moving an ion across the membrane into the cell causes a charge-related free-energy change of size ΔG_{charge}. This change depends on the charge z of the ion and the voltage V between the inside and outside of the cell. Specifically, the change in energy ΔG_{charge} is given as zFV per mole ion, where F is Faraday's constant. A resting cell approaches a state where the total energy is minimal. This state is therefore characterized by

$$-RT\ln\left(\frac{C_o}{C_i}\right)+zFV=0. \tag{12.6}$$

It is typically more useful to turn this equation around in order to compute the equilibrium potential or voltage at rest as

$$V=\frac{RT}{zF}\ln\left(\frac{C_o}{C_i}\right). \tag{12.7}$$

Furthermore, the logarithm with base 10 ($\log_{10}$) is often preferred over the natural logarithm (ln), and if we are considering a human cell, the temperature is more or less constant at 37°C, which corresponds to 310.15 K. Thus, with R = 1.987 cal K^{-1} mol^{-1}, $F = 2.3\times10^4$ cal V^{-1} mol^{-1} and ln 10 = 2.3026, the equilibrium potential for a singly charged ion is

$$V\approx 61.7\log_{10}\left(\frac{C_o}{C_i}\right)[\text{mV}]. \tag{12.8}$$

For the example of a typical cardiomyocyte, the intra- and extracellular potassium concentrations, $[K]_i$ and $[K]_o$, are about 140–150 mM and 4–5 mM, respectively, so that the equilibrium potential for potassium, V_{K}, is roughly between –90 and –100 mV.

The same type of computation holds for other ions, and the laws of physics allow us to compute the total potential by an appropriate summation of concentrations, if we account for the fact that different ions have different permeabilities through the membrane. Specifically, the membrane potential V_M is the equilibrium potential at which the voltage gradients and the concentration gradients of all ions are in balance, so that there is no flux of ions in either direction. This membrane potential is formulated as the **Goldman–Hodgkin–Katz equation**, which for a combined potential including K^+, Na^+, and Cl^- reads

$$V_M=\frac{RT}{zF}\ln\left(\frac{P_{\text{K}}[\text{K}^+]_o+P_{\text{Na}}[\text{Na}^+]_o+P_{\text{Cl}}[\text{Cl}^-]_i}{P_{\text{K}}[\text{K}^+]_i+P_{\text{Na}}[\text{Na}^+]_i+P_{\text{Cl}}[\text{Cl}^-]_o}\right). \tag{12.9}$$

Note that the subscripts for chloride are exchanged—since chloride is an anion—and that the different quantities P quantify the permeability for each ion and depend on the properties of the membrane [8].

Although they are very important, ions like hydrogen and magnesium are often ignored in this computation, because their concentrations are orders of magnitude smaller than those of K^+, Na^+, and Cl^- ([49], p. 652). The membranes of most excitable cells are much more permeable to K^+ than to Na^+ and Cl^-, which results in a resting membrane potential that is closer to the equilibrium potential for K^+ than to that for Na^+ and Cl^- [52].

12.3.2 Resting Potentials and Action Potentials

When an excitable cardiomyocyte is relaxed, its membrane is negatively charged at about –90 mV. This state is called the **resting potential** and describes the electrical potential across the cell membrane, when the outside is considered 0 mV. Electrical stimulation of sufficient strength—or a change in ion balance—changes this state by causing ion channels to open. When positively charged Na^+ and Ca^{2+} ions enter the cell, the membrane begins to depolarize, that is, its charge becomes more positive. Intriguingly, only relatively few Na^+ ions are needed to trigger depolarization. After a brief period of time, K^+ ions begin to leave the cell, the cell repolarizes (its membrane potential becomes more negative), and the cell returns toward its negative charge. Na^+–Ca^{2+} and Na^+–K^+ exchangers reestablish the resting potential. One cycle of up-and-down changes in voltage constitutes an action potential. While some aspects of action potentials are the same for excitable cardiomyocytes and for cells of the SA node, there are also significant differences. We discuss the two cell types separately.

An intriguing feature of SA node cells is that they do not have a true resting potential (**Figure 12.12A**). Instead, they spontaneously depolarize and therefore can generate repeated action potentials, a phenomenon called **automaticity**. In comparison with other cardiomyoctyes, which cycle through five phases during each action potential (see later mention), SA node cells only have three phases (4, 0, and 3). Phase 4 is characterized by the most negative membrane potential, about –60 mV. This potential is less negative than in ventricular cardiomyocytes and sufficiently high for the activation of slow inward-pointing, mixed sodium–potassium currents, called **funny currents**. These funny currents initiate the depolarization of the membrane and thereby control the spontaneous activity of the SA node [54]. Once the potential reaches about –50 mV, transient T-type calcium channels open and Ca^{2+} rushes into the cell, thereby depolarizing the cell even further. At –40 mV, additional L-type calcium channels open and increase the Ca^{2+} influx further. At the same time, potassium ceases to move out of the cell. During the following phase 0, more Ca^{2+} enters the cell through L-channels, while the T-channels begin to close. Overall, the depolarization takes longer than in other cardiac cells. In phase 3, the SA node cells **repolarize**. Potassium channels open, leading to an increased outward flow of K^+. L-channels close, and the depolarizing inward Ca^{2+} currents subside. The action potentials in atrioventricular (AV) node cells are similar to those in the SA node.

Of note is that the controlling funny currents can also be activated by cyclic nucleotides such as cyclic adenosine monophosphate (cAMP) and by **G-protein-coupled receptor** signaling (see Chapter 10). This metabolic signaling option provides a mechanism for the autonomic nervous system to stimulate the beating of the heart [53].

In contrast to SA and AV node cells, ventricular and atrial cardiomyocytes, as well as Purkinje cells, have a true resting potential, which is associated with diastole and characterized by a selective permeability to different ions. Specifically, the membrane is permeable to K^+ ions but essentially impermeable to other ions. As a consequence, the resting potential of about –90 mV is close to the equilibrium potential with respect to K^+ Cells in this resting state generate an action potential only when externally stimulated. Under normal conditions, this stimulus ultimately originates at the SA node and spreads throughout the heart muscle, as we discussed before. Once triggered by adjacent cells, the action potential in these cells develops very quickly, and its overall appearance is distinctly different from that of SA node cells (Figure 12.12B). Starting again with phase 4, the cells have a

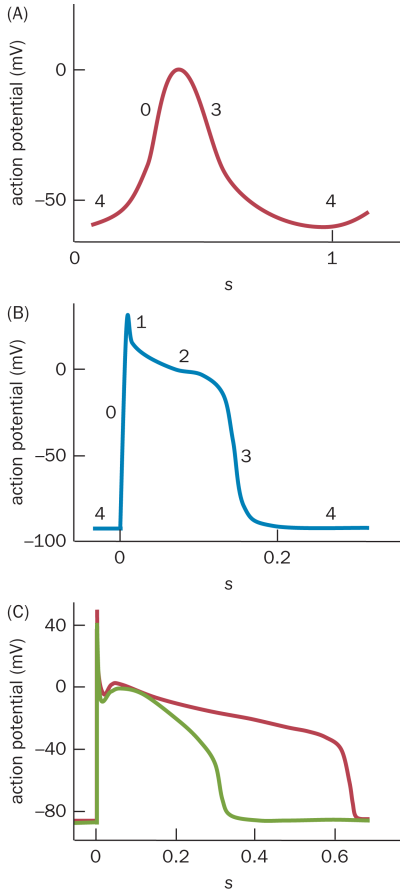


Figure 12.12 Action potentials in heart cells [4, 29, 53]. (A) Action potential of a spontaneously depolarizing SA node cell. (B) Action potential of a typical ventricular cardiomyocyte. Phases 1 and 2 of the latter are missing in the action potential in the SA node. (C) Canine cardiac action potential (green) in a healthy animal and in an animal with congestive heart failure (red).

resting membrane potential of about –90 mV. When the cell receives an action potential from a neighboring cell, it begins to depolarize (phase 0). As soon as the charge exceeds a threshold of about –70 mV, the process is followed by further, rapid depolarization that is mediated by an increase in fast inward Na^+ currents. During this phase, the cell begins to contract. At the same time, the K^+ channels close. Subsequently, the fast Na^+ channels are inactivated and outward K^+ channels begin to open, leading to a brief hyperpolarization, before the cell begins to repolarize (phase 1). At this time, contraction is still in progress. As we discussed before, the sodium influx changes the membrane such that calcium begins to flow in as well, thereby triggering an amplification through the calcium-induced calcium release mechanism and resulting in the flow of large amounts of calcium from the sarcoplasmic reticulum into the cytosol [44], which we also mentioned before and which we will discuss in detail later. Due to the influx of calcium, the initial repolarization is delayed, which can be seen in a plateau of the action potential (phase 2). The contraction ends, and the cell begins to relax. The full repolarization occurs during phase 3, where the potassium efflux is high, the Ca^{2+} channels are inactivated, and the membrane potential becomes more negative. To restore the resting potential, Na^+–K^+ and Na^+–Ca^{2+} exchangers, as well as the plasma membrane Ca^{2+} ATPase, become active. To retain a long-term balance, the amount of Ca^{2+} leaving the cell equals the amount of trigger Ca^{2+} that entered at the start of the cycle. Between phase 0 and most of phase 3, the so-called effective refractory period, the cell cannot be excited again. This mechanism normally prevents spurious signals from spreading throughout the heart muscle and assures sufficient time for filling the heart with blood and ejecting it in a controlled manner.

It should be noted that the actual shapes of action potentials vary among different regions of the heart, which is important for the correct propagation of electric activity and normal cardiac function throughout the heart muscle [30]. Some heart diseases are associated with a change in the pattern of a normal action potential (Figure 12.12C).

The different phases of the action potential can be directly related to features of an ECG (**Figure 12.13**). The PR interval between the onset of the P wave and Q has a length of 120–200 ms and measures the time between the onset of atrial and ventricular depolarization. Within this interval, the P wave reflects the depolarization spreading from the SA node throughout the atria, which lasts for 80–100 ms. During the brief isoelectric period afterwards, the impulse travels to the AV node. The QRS complex is the phase of rapid ventricular depolarization, which lasts for 80–100 ms. The ST segment represents the isoelectric period during which the entire ventricle is depolarized. It corresponds more or less to the plateau phase in the ventricular action potential. The T wave represents ventricular repolarization. This process takes longer than the depolarization during the P wave. The QT

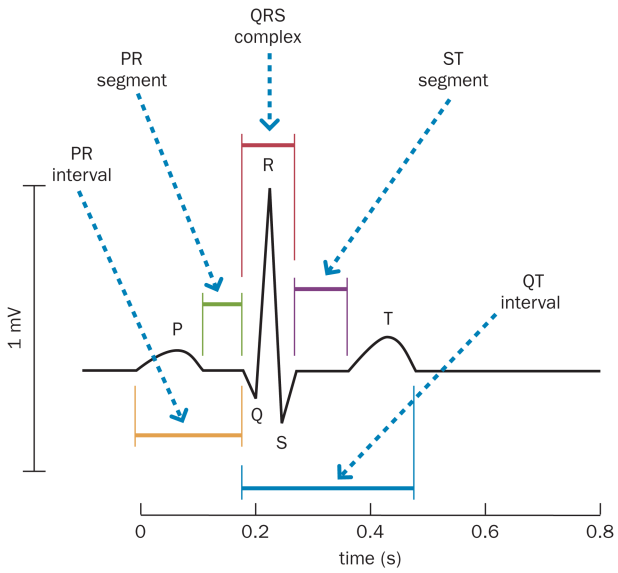


Figure 12.13 Diagram of an ECG trace, showing depolarization and repolarization of the atria and ventricles in the heart. Standardization of the recording speed allows measurements of the heart rate from the intervals between different waves. See the text for further explanation.

interval corresponds roughly to the average ventricular action potential and ranges between 200 and 400 ms, but is shorter at high heart rates. Even seemingly minute abnormalities in any of these constituents of the ECG trace are often signs of various pathologies.

12.3.3 Models of Action Potentials

The first scientists to study action potentials in excitable cells quantitatively were Alan Hodgkin and Andrew Huxley. They did their legendary work on axons of squid neurons in the early 1950s [55] and were awarded the Nobel Prize for their efforts in 1963. A key to their success was the recognition of a direct analogy between the functioning of a nerve cell and a simple electrical circuit. A few years later, Noble [9, 10, 55] and others [56–58] demonstrated that the Hodgkin–Huxley model for nerve cells could be adapted to represent cardiac action potentials, which have a much longer repolarization phase than neurons.

Hodgkin and Huxley exploited the analogy with electrical circuits to apply well-established laws from the theory of electricity to action potentials. They interpreted the lipid bilayer of the membrane as a capacitor, ion channels as conductors, electrochemical gradients driving the flow of ions as batteries, and ion pumps as current sources. To refresh our memory, let us revisit the basic terms of electricity.

- The most fundamental, subatomic property that describes relative amounts of electrons and protons in atoms and molecules is the electrical charge q. Of particular importance here are ions, which have either an excess or a shortage of electrons compared with neutral atoms. The electrical charge of a macroscopic entity is equal to the sum of the charges of all its particles. Thus, in excitable cells, the electrical charge is the net positive or negative charge carried by the ions in the system. A charge creates an electrical field in the surrounding space, which exerts a force on other electrically charged objects.
- An electrical conductor is a material that contains movable electrical charges. In electrical circuits, conductors typically consist of metals such as copper, while the conductor in our case consists of watery solutions containing ions and the channels that allow ions to cross the lipid bilayer.
- Electric current is the actual movement or flow of electric charge. It is typically denoted by I and, because it describes the change in charge at a given point, it is given as

$$I = \frac{dq}{dt}. \tag{12.10}$$

- The electrical potential is the potential energy per unit of charge in an electric field. The difference between the electrical potentials at two points is called voltage (denoted by V) or voltage drop and quantifies the ability to move electrical charge though a resistance. Sometimes this potential difference is called the electromotive force, or EMF.
- Capacitance, denoted by C, is a measure of the amount of stored electrical charge in relation to an electric potential. It is related to charge and voltage as

$$C = \frac{q}{V}. \tag{12.11}$$

- Electrical conductance and resistance are properties of materials and are inverse to each other. Resistance R describes how much the flow of charged particles, such as ions, is impeded, while conductance g measures how easy it is for charge to flow through some material. It is the inverse of resistance:

$$g = \frac{1}{R}. \tag{12.12}$$

Famous physicists such as Maxwell, Faraday, Ohm, Coulomb, and Kirchhoff discovered laws relating the various quantities in electrical systems. Here are three of particular relevance.

- Ohm's law states that the current through a conductor is directly proportional to the voltage across the two points and inversely proportional to the resistance between them. Thus, for time-dependent V and I, we have

$$V = IR = \frac{I}{g}. \tag{12.13}$$

- Kirchhoff's two most important laws address the conservation of charge and energy at any point in an electrical circuit (with the exception of a capacitor). Thus, if there are k currents flowing toward or away from a point, their sum is zero. In mathematical terms,

$$\sum_k I_k = 0. \tag{12.14}$$

- Similarly, the sum of all k voltages around any closed electrical circuit is zero, if the positive and negative signs are assigned correctly. Thus,

$$\sum_k V_k = 0. \tag{12.15}$$

Because an electric current describes the flow of electric charge, the total current equals the sum of all currents in the system. In other words, the currents in a system with k resistances are additive in a sense that

$$I = \sum_k I_k = \sum_k \left(\frac{V}{R_k} \right) = V \sum_k g_k. \tag{12.16}$$

These relationships look rather static, but they also apply in a system whose electrical state changes over time. Of particular importance here, dynamic changes in voltages and currents across the membrane of a cardiac cell can lead to the generation of a time-dependent action potential. To describe this action potential mathematically, we acknowledge explicitly that voltage and currents are time-dependent by defining $V(t)$ as the difference between internal and external voltages: $V(t) = V_i(t) - V_e(t)$. $V(t)$ is a time-dependent variable, which we may differentiate with respect to time. Using (12.10), we obtain

$$\frac{dV}{dt} = \frac{1}{C}\frac{dq}{dt} = \frac{I(t)}{C}. \tag{12.17}$$

According to the laws of electricity, the sum of the capacitive current $I(t)$ and the ionic current $I_{\text{ion}}(t)$ at the membrane must be zero. Substituting this conservation law into (12.17) and rearranging yields

$$C\frac{dV}{dt} + I_{\text{ion}}(t) = 0. \tag{12.18}$$

This is our fundamental equation. However, in order to account for the various ion fluxes, which sometimes run in opposite directions, it is now necessary to split the voltage and ionic current into partial voltages and currents that are associated with Na^+, K^+, and a pool L, which accounts for all other ions, such as calcium, chloride, and magnesium. The symbol L was chosen to suggest leakage. Thus, we define V_{Na}, V_{K}, and V_L, representing contributions to the resting potential, along with the corresponding currents $I_{\text{Na}} = g_{\text{Na}}(V - V_{\text{Na}})$, $I_{\text{K}} = g_{\text{K}}(V - V_{\text{K}})$, and $I_L = g_L(V - V_L)$. Recalling the relationship (12.13) between current, conductance, and voltage, we can rewrite (12.18) as

$$C\frac{dV}{dt} = -g_{\text{eff}}(V - V_{\text{eq}}). \tag{12.19}$$

Here, the effective conductance $g_{\text{eff}} = g_{\text{Na}} + g_{\text{K}} + g_L$ is composed of the conductances related to Na^+, K^+, and the collective pool L for other ions. The resting (or equilibrium) potential is composed of individual contributions of conductances and voltages as

$$V_{\text{eq}} = \frac{g_{\text{Na}}V_{\text{Na}} + g_{\text{K}}V_{\text{K}} + g_{\text{L}}V_{\text{L}}}{g_{\text{eff}}}. \tag{12.20}$$

These terms are substituted into (12.19), and one may furthermore account for the membrane current, which is 0 at equilibrium or is equal to I_{app} applied to the system in typical experiments. It is customary in this field not to express relationships in absolute potentials or voltages but instead to consider a relative potential v that describes the deviation from the resting potential:

$$v = V - V_{\text{eq}}. \tag{12.21}$$

With this convention, we obtain an ODE describing dynamic changes in voltage as

$$C\frac{dv}{dt} = -g_{\text{Na}}(v - v_{\text{Na}}) - g_{\text{K}}(v - v_{\text{K}}) - g_{\text{L}}(v - v_{\text{L}}) + I_{\text{app}}. \tag{12.22}$$

Note that the capacitance C and the different conductances g are properties of the material and are therefore constant over time in most electrical circuits. In the case of the heart, the capacitance is related to the thickness of the membrane, which is not entirely constant in a cardiac cell, but C is usually considered constant, which is probably a reasonable approximation over the short period of time of an action potential. Importantly, the conductances correspond to channels in membranes, and some of these are ion-gated or voltage-gated and are therefore functions of the electrochemical milieu of the cell. In other words, both the voltage and the conductances are functions of time and far from constant.

Hodgkin and Huxley formulated the dependences of conductances on voltage essentially as black-box models that were inspired by experimental data but do not have a direct interpretation in terms of biological mechanisms. In a stroke of genius, they defined

$$\begin{aligned} g_{\text{K}} &= \bar{g}_{\text{K}} n^4, \\ g_{\text{Na}} &= \bar{g}_{\text{Na}} m^3 h, \end{aligned} \tag{12.23}$$

and considered g_{L} as constant, because it was seen as negligible in comparison with g_{Na} and g_{K}. The terms $\bar{g}_{\text{K}}$ and $\bar{g}_{\text{Na}}$ were defined as constant conductivity parameters, while the quantities n, m, and h, which may be interpreted as characteristics of the potassium and sodium gates, were given as the solutions of differential equations that also depend on voltage:

$$\begin{aligned} \frac{dn}{dt} &= \alpha_n(v)(1-n) - \beta_n(v)n, \\ \frac{dm}{dt} &= \alpha_m(v)(1-m) - \beta_m(v)m, \\ \frac{dh}{dt} &= \alpha_h(v)(1-h) - \beta_h(v)h. \end{aligned} \tag{12.24}$$

The terms n^4 and m^3h are proportional to the fractions of open ion channels or the probability that these channels are open. If these quantities are 1 or 0, all corresponding channels are open or closed, respectively. The power of n was chosen to reflect four subunits of the potassium channel, which at the time was a postulate. The dependences on voltage in the Hodgkin–Huxley model are typically formulated as

$$\begin{aligned} \alpha_n(v) &= 0.01(10-v)(e^{(10-v)/10} - 1)^{-1}, & \beta_n(v) &= 0.125e^{-v/80}, \\ \alpha_m(v) &= 0.1(25-v)(e^{(25-v)/10} - 1)^{-1}, & \beta_m(v) &= 4.0e^{-v/18}, \\ \alpha_h(v) &= 0.07e^{-v/20}, & \beta_h(v) &= (e^{(30-v)/10} + 1)^{-1}, \end{aligned} \tag{12.25}$$

with initial values specified as $v(0) = 10.5$, $n(0) = 0.33$, $m(0) = 0.05$, $h(0) = 0.6$, and $C = 1$, $v_{Na} = 115$, $v_K = -12$, $v_L = 10.6$, $\bar{g}_{Na} = 120$, $\bar{g}_K = 36$, $g_L = 3$ (see, for example, [8, 59]). The apparently arbitrary form of the equations for n, m, and h comes from arguments describing how these quantities decay over time in a first-order fashion.

With these definitions and settings, the dynamics at the membrane is described as a system of four ODEs. The first represents the action potential itself, (12.22), while the remaining three are equations describing time-dependent changes in conductances and are defined as in (12.24) and the associated functional dependences (12.25). Putting the four equations into a numerical solver and specifying the initial values, we quickly obtain the fruit of our labor (**Figure 12.14**): an action potential in a nerve cell that shows rapid depolarization and repolarization. The numerical solution also shows the dynamics of the auxiliary variables n, m, and h, even though they do not really have a direct meaning, except insofar as they represent features of the different ion channels (**Figure 12.15**).

The appearance of this action potential reflects that of nerve cells. It is much faster than in cardiomyocytes, and the pronounced plateau during repolarization in phase 2 (see Figure 12.12B) is entirely missing. Furthermore, the nerve model exhibits a slight undershoot before returning to the resting potential. Nonetheless, variations in Hodgkin and Huxley's equations, such as adaptations in the representation of the potassium current, were shown to lead to longer action potentials and repolarization plateaus as we find them in cardiomyocytes [9, 10, 56–58].

Before we continue, let us summarize the key points of this section. The electrochemical changes during contraction and relaxation in a nerve cell—and by extension in a cardiac cell—can be modeled in analogy to an electrical circuit, and Hodgkin and Huxley showed that the physical laws describing such a circuit can be applied to this physiological system. The most significant difference between a standard electrical circuit and a cell is that some of the channels in the cell membrane are gated, that is, controlled by voltage or ions, whereas conductance in a typical electrical circuit is constant. As a consequence, there is a logical cycle of influences where ion concentrations affect voltage and voltage affects the movement of ions and thereby the chemical and electrical balances in the system. Hodgkin and Huxley modeled the voltage dependence of the channels based on mathematical assumptions regarding the functional dependences and by fitting experimental data. The amazing outcome is that their model works very well under a number of conditions.

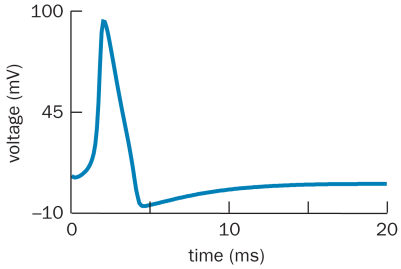


Figure 12.14 Simulated action potential in a nerve cell. The action potential was computed from the Hodgkin–Huxley model (12.22)–(12.25). The shape is different than in cardiomyocytes, but an adjusted Hodgkin–Huxley model can generate cardiac action potentials as in Figure 12.13(B).

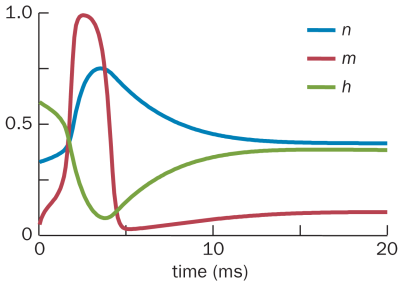


Figure 12.15 Temporal responses of gating variables in the Hodgkin–Huxley model (12.24). Close inspection shows that the variables do not return to their starting values after the action potential.

12.3.4 Repeated Heartbeats

We now have an action potential model at our disposal, (12.22)–(12.24), but it only fires once. If we simply run the simulation longer, nothing happens. The system has approached a steady state and stays there. How does it lead to pacemaker activity and a heartbeat? The vast majority of the heart cells receive electrical signals from the SA node, through the AV node and the Purkinje fibers, which are mediated through gap junctions. Thus, their triggers are changes in the action potential of their membrane. The pacemaker cells, in turn, autonomously fire in a process of depolarization that involves influxes and effluxes of ions, and we have seen before that depolarization begins with funny currents but really proceeds when Ca^{2+} rushes into the cytosol.

Let us recall that n, m, and h in the Hodgkin–Huxley model describe voltage-dependent gates and that their conductance is affected by changes in voltage and thus by changes in ions such as calcium. If we look more closely at the dynamics of n in the action potential model, we see that its final value is about 0.415, while its initial value is about 0.33 (see Figure 12.14). What happens if we artificially reset the value of n to the initial value after the action potential has fired? This is a simple simulation experiment and, *voilà*, the action potential fires again. In **Figure 12.16**, we repeatedly reset n at times $t = 20$, 40, and 60. We can see that the artificially induced potentials are a little lower than the first potential, but that the shape of the subsequent beats is identical. Would we receive the same peaks in all cases if we not only reset n, but also m and/or h?

Because a four-variable system is difficult to study mathematically, Fitzhugh and Nagumo developed a simpler variant of the Hodgkin–Huxley model, in which

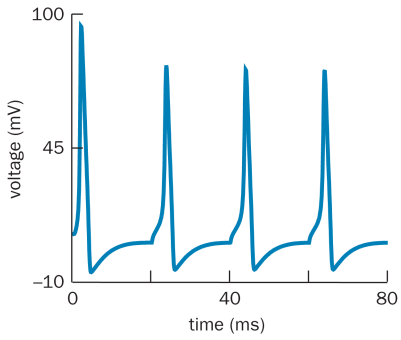


Figure 12.16 Creating repeated heartbeats. Artificially resetting the value of n in (12.24) to the initial values at times 20, 40, and 60 results in a beating pattern of repeated action potentials.

they separated the variables into fast and slow variables [60, 61]. The result had only two variables (see below), and it turns out that their model is a variant of the van der Pol oscillator that we discussed in the beginning of the chapter and in Chapter 5! We have come full circle and in the process have given some biological justification to van der Pol's expectation [34] that his oscillator might describe heart rhythms and maybe even serve as a pacemaker. Many researchers have performed very elegant analyses of the mathematical structure of the Fitzhugh–Nagumo equations and their bifurcation structure; the interested reader is referred to [8, 59].

To generate the repeated action potential, we cheated a little bit by manually resetting n at times 20, 40, and 60. Obviously, the real SA node does not need our input for every beat. So, we should ask whether it is possible to replace this artificial repeated triggering with a more natural pacemaker. A promising candidate could be our calcium oscillation model, because we can easily argue that the calcium oscillations are associated with depolarization and repolarization. The strategy is therefore simple: we set up a simulation program that contains the Hodgkin–Huxley model (12.22)–(12.25) and also the calcium oscillation model (12.5) and then couple the two systems.

Let us simulate a situation where the calcium concentration in the cytosol, y, affects the conductance of h. The easiest—but not the only—way of achieving the coupling is simply to add y to the equation for h. Revisiting the output of the calcium model, we see that the value of y is mostly low, with a value close to 0, but spikes regularly up to a value of about 0.4. These low values make sense, because the calcium level in the cytosol is about 10,000 times lower than calcium levels in the extracellular space. The simulation indeed shows the desired effect: without artificial intervention, the action potential spikes and depolarizes in a repeated manner (Figure 12.17), as we see it in an ECG. Of course, our success is no proof that the stitched-together model is an appropriate model of a beating heart, but it demonstrates how complex problems might be approached in a modular fashion, at least as a proof of principle and a means for investigating the internal logic of a system.

As mentioned above, Fitzhugh and Nagumo [60–62] introduced a 2-variable approximation to the Hodgkin-Huxley system with repeated oscillations that retains key properties but simplifies mathematical analysis. The system consists of two ODEs of the form

$$\dot{V} = V(a - V)(V - 1) - W$$
$$\dot{W} = bV - cW,$$

where V and W represent the membrane voltage and the activation of the current out of the cell. The parameters b and c have positive values; a may be positive or negative. Under the right conditions, the system can switch from a stable steady state to a limit cycle oscillation (Exercise 12.17).

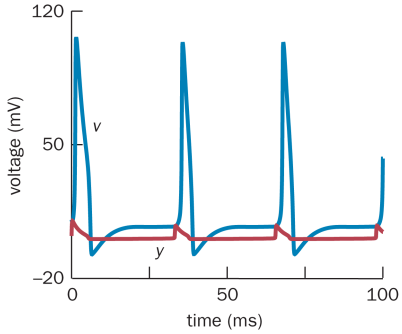


Figure 12.17 Creating self-sustained heartbeats. The combined Hodgkin–Huxley/Somogy–Stucki model generates repeated action potentials without external cues. Note that y is shown with 20-fold magnification.

12.4 ISSUES OF A FAILING HEART

Although the heart is incredibly resistant to fatigue and faithfully keeps on pumping, age and disease eventually take a toll. Uncounted things in the complex systems that work within the heart with unparalleled precision for most of our lives can go wrong, and the result is often heart failure, which presently affects about 5 million individuals in the United States, of whom about 20% die within 1 year of diagnosis. Indeed, heart failure is the leading cause of death in the Western world.

It is very clearly beyond the scope of this chapter to elaborate on all or even the most prevalent physiological changes in the heart that might result in heart failure. Clearly, this is a true multiscale issue, and the challenges discussed in Chapter 11 should be recalled. To address this issue, one could think of a sophisticated agent-based model or a finite-element model as it is used in the Physiome Project [15] but exclusively to explain one specific type of heart failure—the following sections discuss just one thread that connects the intracellular level with failure at the organ level. In line with our emphasis throughout the chapter, we focus on action potentials and the dynamics of calcium.

A prominent class of heart problems is related to abnormalities in the pacemaker system, which can lead to different arrhythmias. For instance, in sick sinus syndrome, which is often related to senescence of the SA node, the heart rate is too slow, which can lead to dizziness, sluggishness, and shortness of breath. The slow heartbeat (bradycardia; *bradys* is Greek for slow) may alternate with fast atrial arrhythmia (tachyarrhythmia; *tachys* is Greek for fast), leading to the so-called **brady–tachy syndrome**. A delay in conduction between the SA node and the AV node can lead to a first-degree heart block, which often does not lead to noticeable symptoms, because the AV node can serve as a sort of back-up pacemaker. By contrast, a block below the AV node, namely in the bundle of His and the Purkinje system, may be fatal if untreated. An implanted pacemaker may alleviate such a block and prevent cardiac arrest.

A fast heartbeat sometimes originates in one of the ventricles, leading to a life-threatening arrhythmia called **ventricular tachycardia**. A very common cause of this condition is scarring of the heart muscle due to an earlier heart attack. The scar does not conduct electrical signals so that the signal propagates on both sides around the damaged tissue. Earlier, we mentioned that three-dimensional modeling with finite element methods has become sophisticated enough to mimic this disease on the computer, and the result of such an analysis was shown in Figure 12.3.

A seemingly minute but indeed significant change in the proper functioning of the heart during aging is a slowing down of the calcium fluxes between the sarcoplasmic reticulum (SR) and the cytosol [63]. If the calcium dynamics in individual myocytes becomes less efficient, the consequences are felt in the contractions of the heart: it does not fill with blood and empty as quickly as it used to. The slower calcium dynamics often goes hand in hand with an age-related stiffening of the arteries, and together they can lead to shortness of breath, especially during strenuous activities. These systemic connections can be explained by the mechanical features of the heart: perfusion is reduced, and this reduction leads to an accumulation of fluid beneath the skin—called edema—and to the entire spectrum of problems associated with congestive heart failure.

Another change in cardiomyocyte activity that is associated with slower calcium dynamics is a lengthening of the action potential during repolarization (see Figure 12.12C). The longer action potential keeps the ion gates in the cell membrane open for a slightly longer period of time, thereby allowing more calcium to enter and exit the cytosol between beats and giving the weakened SR more time to release or take up calcium. The downside is a higher risk of unbalanced calcium flow and a slower response to increased demands on heart activity. Age also affects the contractile elements in the myocytes, and in particular the heavy chains of myosin, which are responsible for forming cross-bridges with actin (see Figure 12.4). Some of these changes can be detected in the expression of genes, which is different in young and old individuals. Finally, as with nearly all cells, cardiomyocytes are susceptible to unstable oxygen molecules, called free radicals, that create oxidative stress within the cells and may impair energy metabolism and other functionalities by damaging proteins, DNA, membranes, and other complex structures in the cell. In cardiomyocytes, free radicals are doubly problematic, because of the high energy demands of these cells and because free radicals can damage the calcium pumps in the membrane of the SR, which connects this aging mechanism again to reduced calcium dynamics. During aging, many cardiomyocytes die, and even in healthy septuagenarians, the number of cardiac cells may be decreased by 30% .

To the detriment of many elderly people, some of the effects of aging on the heart form a vicious cycle that gradually diminishes efficiency. When the calcium dynamics slows down and the arteries stiffen, the heart has to work harder to compensate. As in other muscles, the harder work leads to enlargement of the cardiomyocytes, which in turn leads to a thickening of the heart walls, as described by the Frank–Starling law (see earlier in the chapter). This normal thickening is exacerbated by diseases such as coronary heart disease and high blood pressure. Together, the metabolic and anatomical changes make the heart more susceptible to cardiovascular conditions such as left ventricular hypertrophy and atrial fibrillation. The latter is a common cardiac arrhythmia that involves the two atria: the heart muscle quivers instead of contracting in a coordinated fashion. Strong fibrillations may lead to heart palpitations and congestive heart failure.

12.4.1 Modeling Heart Function and Failure Based on Molecular Events

The models that we have developed in this chapter are snapshots of heart function at different levels. Considering these snapshots collectively will give an impression of how difficult it is to connect heart function and failure to specific molecular events through comprehensive, true multiscale models, as they are envisioned in the Physiome [15], the KidneySim [64], and Virtual Physiological Rat [65] Projects. Nonetheless, the following will discuss in broad strokes how our models can inform models at higher levels in order to connect intracellular calcium-associated events with macroscopic heart problems at the organ level. This line of reasoning with models was proposed by the late Raimond Winslow and his collaborators over a span of many years. Most of the technical details will have to be skipped here as they involve a lot of mathematics, but the interested reader may start diving into this field with reviews such as [66–68].

We begin our journey at the molecular scale and specifically at the cell membrane of the cardiomyocyte, the sarcolemma. The sarcolemma contains invaginations into the cytosol that effectively increase its surface area. The invaginations, called T-tubules, contain calcium channels that functionally connect the sarcolemma and the membrane of the tubular network of the SR (**Figure 12.18**). This functional connection occurs in spatially restricted microdomains of about 10 nm width, called **dyads**. Each cardiomyocyte contains roughly 12,000 such dyads. From the lumen of the T-tubules, the dyads can receive calcium through specific **L-type Ca^{2+}** channels (LCCs) that reach through the sarcolemma. They can also receive calcium from the SR through ryanodine receptor (RyR) proteins (**Figure 12.19**). The role of this structural set-up is to facilitate the calcium-induced calcium release (CICR) that we discussed before. In response to membrane depolarization, the LCCs open and thereby produce a small flux of Ca^{2+} ions from the T-tubule into the dyad. The Ca^{2+} ions cause the RyRs on the opposite site of the dyad to open, and the result is a strong calcium flow from the SR into the cytoplasm that ultimately leads to cardiomyocyte contraction. This LCC-RyR excitation-contraction mechanism is very fast and occurs in a range of micro- and milliseconds [13]. With a length in the nanometer range and a volume of only about 10^{-19} liter, the dyads may be very small, but they are "at the heart of the heart." To allow relaxation of the filament complex, intracellular calcium is returned to the SR by SERCA (SR Ca^{2+} ATPase) pumps and also pumped out of the cell through sodium–calcium exchangers or plasma membrane Ca^{2+} ATPase. The

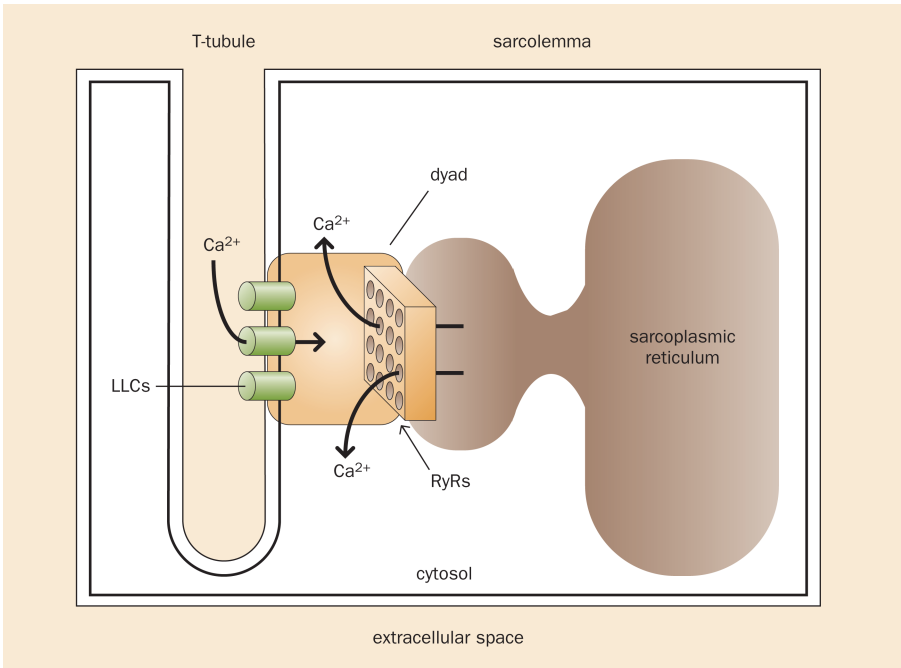


Figure 12.18 Simplified illustration of a dyad and its role in the cardiac cycle [68]. During the action potential, calcium from the extracellular space enters the dyad through L-type Ca^{2+} channels (LCCs) in the T-tubules. The influx activates arrays of ryanodine receptors (RyRs), which are located on the opposite side of the dyad on the sarcoplasmic reticulum (SR). The opening of the RyRs leads to a calcium flux from the SR into the dyad, from where it diffuses into the cytosol, increasing its calcium concentration greatly. The cytosolic calcium binds to myofilaments, which contract and shorten the cell. (Data from Tanskanen AJ, Greenstein JL, Chen A, et al. *Biophys. J.* 92 [2007] 3379–3396.) [69].

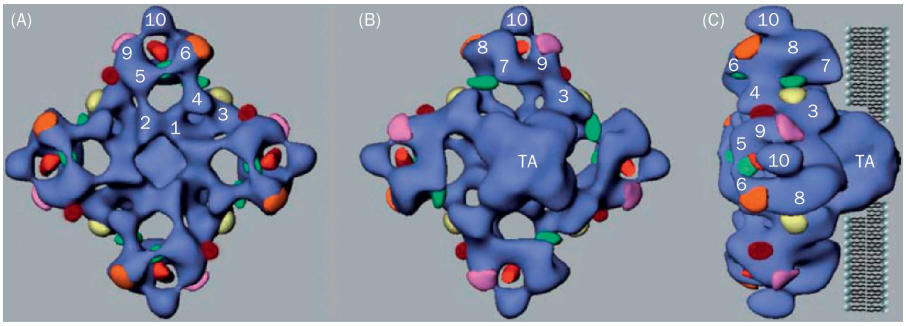


Figure 12.19 Three-dimensional structure of a ryanodine receptor. (A) View from the cytoplasm to the lumen of the sarcoplasmic reticulum (SR). (B) View in the opposite direction. (C) Side view. The numbers identify different globular domains and TA denotes the transmembrane assembly. (From Zissimopoulos S & Lai FA. Ryanodine receptor structure, function and pathophysiology. In Calcium: A Matter of Life or Death [J Krebs & M Michalak, eds], pp 287–342. Elsevier, 2007. With permission from Elsevier.)

reduction in calcium concentration causes the bonds between the myosin and actin filaments to loosen.

Experiments have shown that not every RyR is associated with an LCC but that there is a stochastic distribution of one LCC among about five RyRs (**Figure 12.20**). A careful computational analysis [68] showed that the particular shape of the RyR protein restricts the movement of Ca^{2+} ions and funnels them to their target binding sites. In contrast to a random process such as Brownian motion, this guiding action enormously increases the efficiency with which an external excitation by calcium ultimately leads to myocyte contraction.

The number of calcium molecules needed for mediating the signal from LCC to RyR is small. Even at the peak of the calcium flux, only about 10–100 free Ca^{2+} ions are found in each dyad. This low number has two consequences for any model representation: the process is stochastic and the noise level is high—one may find an average number of 50 ions in a dyad, but the number in any given dyad could also be 20 or 120. A situation leading to such a random occupancy is typically represented as a stochastic process [25, 70]. We briefly discussed stochastic processes in Chapters 2, 4, and 5, comparing and contrasting them with deterministic dynamic processes. The key feature of stochastic processes is that random numbers affect their progression. The simplest stochastic processes are probably flipping a coin and rolling a die. Typical models for stochastic processes are Poisson processes and Markov models.

The defining feature of the dyad is the channeling of ions, and the location of all Ca^{2+} ions in the vicinity of dyads is therefore of great importance. As we discussed in Chapters 2, 4, and 5, the consideration of spatial aspects in a dynamical process typically calls for a representation in the form of a partial differential equation

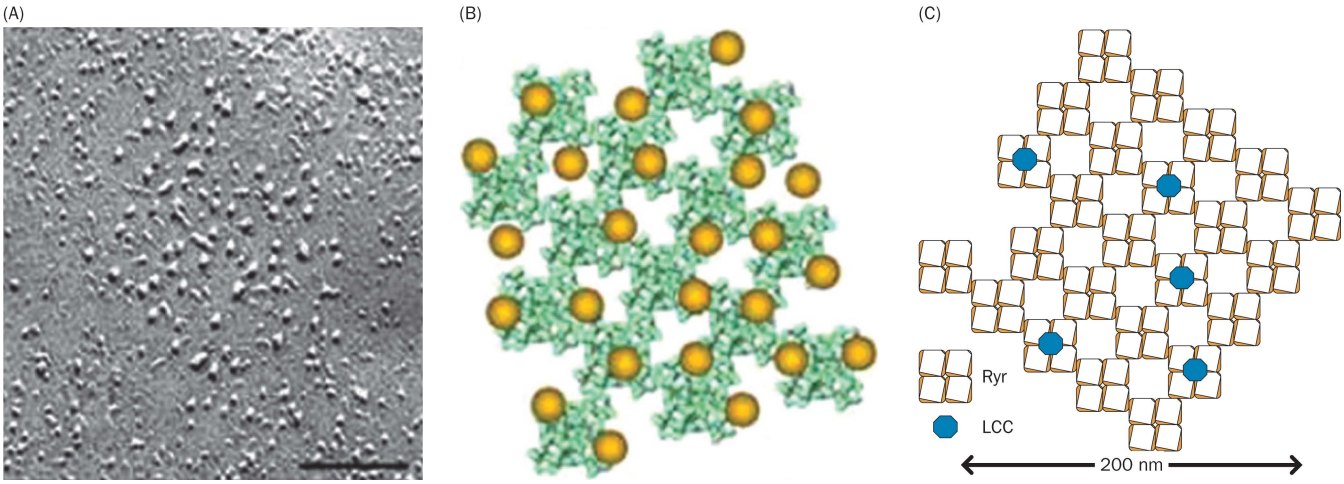


Figure 12.20 Spatial distribution of RyRs on the surface of the sarcoplasmic reticulum and of fewer LCCs on the adjacent membrane of the T-tubule, which is part of the sarcolemma. (A) Electron micrograph of Ca^{2+} release units in myocytes of a chicken heart. (B) Organization of cardiac RyR units into ordered arrays. (C) Schematic representation of RyR and LCC units on opposing sides of a dyad. ((B) from Yin C-C, D'Cruz LG & Lai FA. *Trends Cell Biol* 18 [2008] 149–156. With permission from Elsevier. (C) adapted from Tanskanen AJ, Greenstein JL, Chen A, et al. *Biophys. J.* 92 [2007] 3379–3396 and from Winslow RL, Tanskanen AJ, Chen M & Greenstein JL. *Ann. NY Acad. Sci.* 1080 [2006] 362–375. With permission from Elsevier. With permission from John Wiley & Sons.)

(PDE). Thus, to determine whether a Ca^{2+} ion will enter a dyad, a detailed model should quantify the probability $P(x, t)$ that a Ca^{2+} ion is close enough to and properly aligned with a dyad position x at time t. From a mathematical point of view, the combination of free motion and stochastic binding events would suggest as the default model a representation of the movement of individual calcium molecules in the form of a stochastic PDE, which is very complicated. Indeed, such detailed focus on each and every dyad—or maybe even every Ca^{2+} ion—would be extremely involved, because of the roughly 12,000 dyads per cell and the billions of Ca^{2+} ions flowing between the extracellular space, the cytosol, and the SR. As a consequence, simulations would take weeks to complete, even on a high-performance computer, and computation time would become prohibitive even for modeling one single cardiomyocyte, let alone the entire heart.

It may be infeasible to model processes with such resolution and detail, but discussion of the intricate molecular details and biologically observed mechanisms is important because it indicates how a stochastic model design would proceed in principle and also because it allows us to assess where one might make valid simplifying assumptions without compromising realism. To the rescue in our specific case comes the very fact that makes a direct simulation so complicated, namely the large number of dyads and the fact that they are more or less randomly distributed along the T-tubules. A reasonable first simplification in such a situation is to ignore spatial considerations, with the argument that one may study a local area of the cell membrane, a single T-tubule, or even a single dyad and subsequently scale-up the results statistically to the thousands of dyads in each cell. The important advantage is that this simplification step circumvents the need for PDEs to describe the detailed mechanisms in favor of a model consisting of ordinary differential equations that use average numbers of ions in dyads instead of a specific number in each dyad. Before we make this step, we must convince ourselves that we do not lose too much resolution. A strict mathematical justification would require setting up a detailed model and comparing it with the simplifying approximation. Indeed, Winslow's group performed a careful analysis demonstrating that a simplified model based on a single dyad retained important features of the CICR mechanism.

We cannot present the details of this analysis here, but we can ask ourselves which aspects are really important to us. In the end, we need to know how many calcium ions enter the collection of dyads. Does the large population of dyads buffer the stochastic fluctuations among individual dyads and make their collective behavior predictable? The affirmative answer comes from a fundamental law of statistics, known as the **central limit theorem**. This theorem states that the sum of many random events of similar character follows a normal distribution and that the standard deviation of this distribution becomes smaller if more and more events are considered. In other words, the more items that are part of the system, the more the population of items will behave predictably like their average. The theorem is very difficult to prove, but Exercise 12.18 suggests some easy computational experiments that might convince you that the theorem is correct.

Unfortunately, even the simplified model of one dyad is still too complicated as a module in a multiscale model of a whole cardiomyocyte. A second opportunity for simplification arises from a technique called **separation of timescales**, which we mentioned in Chapter 11. To recall, this generic technique is useful if a system involves different types of processes, some of which are very fast and others comparatively slow. The argument is that on the timescale of the fast processes, the slow processes are essentially constant and their dynamics can simply be ignored. On the timescale of the slow processes, it is often reasonable to expect that much faster processes assume their steady-state level essentially instantaneously and are therefore almost always constant. The result of this separation is a set of two or more simpler models, and one only analyzes the one whose timescale is of particular interest in a given situation. In the case of calcium transport associated with the dyads, the processes at LCCs and RyRs are comparable in speed but much faster than the release of large quantities of calcium from the SR. A possible strategy is therefore to focus only on the LCC and RyR as a coupled system that can be represented as a low-dimensional system of ordinary differential equations [67, 71].

Experimental work has shown that opening and closing of each LCC depend on the local calcium concentration. Specifically, the data suggest that Ca^{2+} binding

induces the channel to switch from an active mode (Mode Normal) to an inactive mode (Mode Ca) where transitions to the open state are very rare. A first model describing the opening and closing dynamics assumed that four subunits could independently close each LCC [72, 73]. Thus, five active states ($C_0, \ldots, C_4$) were defined, with the index representing the number of permissive subunits. Similarly, five inactive states were defined. As a result, any given LCC could be in one of 12 states, namely ten closed states (five active and five inactive) and two open states (active or inactive). In line with experimental results, the transitions among active states and among inactive states were made voltage-dependent, while transitions between closed and open states were made voltage-independent. The transitions from active to inactive states were formulated to depend on the Ca^{2+} concentration, with an increase in calcium causing the LCC to shift to a gating mode permitting only infrequent opening.

Rigorous analysis of this model justified a simplification from five to two pairs of active and inactive states. It also showed that the transition probability from an inactive closed state to an open state was negligibly small. Thus, the result was a simplified LCC model consisting of five states: two closed active states C_1 and C_2; one open state O, which is accessible from 0.005; and two closed (Ca^{2+} inactivated) states I_1 and I_2 accessible from C_1 and C_2, respectively (**Figure 12.21A**).

The rates of transitions between the two active states and between the two inactivated states are fast (rates denoted by f in Figure 12.21A). By contrast, inactivation by Ca^{2+} and activation are slower (rates denoted by s). This difference in rates suggested a separation of timescales and thus a possible further simplification to only three states (Figure 12.21B), using the argument that the fast transitions become less relevant on the timescale of the slow transitions.

It is straightforward to set up mass action equations for the two systems. Implementing both in the same program allows us to judge the loss in resolution due to model reduction. The two models, inspired by Hinch and coworkers [67, 71], have the following forms:

Five-state model

$$\begin{aligned}
\dot{C}_1 &= 50C_2 + 0.5I_1 - 40C_1 - C_1, & C_1 &= 0.60,\\
\dot{C}_2 &= 40C_1 + I_2 - 50C_2 - 2C_2 + 0.5O - 2C_2, & C_2 &= 0.49,\\
\dot{I}_1 &= 40I_2 + C_1 - 60I_1 - 0.5I_1, & I_1 &= 0.79,\\
\dot{I}_2 &= 60I_1 + 2C_2 - 40I_2 - I_2, & I_2 &= 1.18,\\
\dot{O} &= 2C_2 - 0.5O + perturb, & O &= 1.95;
\end{aligned} \tag{12.26}$$

Three-state model

$$\begin{aligned}
\dot{C} &= 1.1I - 2C + 1.1OO - 2C, & C &= 1.09,\\
\dot{I} &= 2C - 1.1I, & I &= 1.97,\\
\dot{OO} &= 2C - 1.1OO + perturb, & OO &= 1.95:
\end{aligned} \tag{12.27}$$

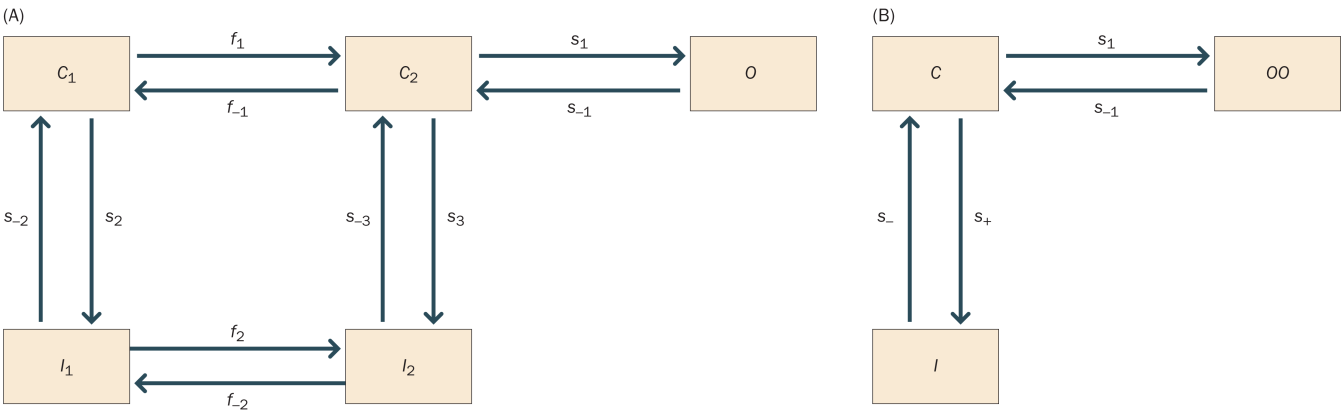


Figure 12.21 Original and simplified models of calcium dynamics in dyads. Computational validation experiments are needed to assess to what degree the simpler three-state model in (B) reflects the responses of the more complex five-state model in (A). (Adapted from Hinch R, Greenstein JL & Winslow RL. *Prog. Biophys. Mol. Biol.* 90 [2006] 136–150. With permission from Elsevier.)

The two systems have parameters that lead to similar steady states for $C_1 + C_2$ and C, $I_1 + I_2$ and I, as well as O and OO. The simulation begins close to these states. At time $t = 2$, we perturb the open state (O, OO) by setting the parameter *perturb* = 1. At $t = 4$, we send the signal *perturb* = –1, and at $t = 6$, we return to normal by setting *perturb* = 0. The result is shown in **Figure 12.22(A)**. We see that the two systems respond quite similarly.

For the second part of the simulation, we multiply all fast (f) parameters by 0.05—which makes them similar in magnitude to the slow (s) parameters—or by 0.005, which makes them even slower. Figure 12.22(B) and (C) demonstrate that the systems return to similar—although not exactly the steady states as before—but that the transient dynamics diverge a little more than before (Figure 12.22A). This one set of simulations suggests that the three-state model is doing quite well. For due diligence, one should execute many more and different types of perturbations with actual parameter values and compare the transients of the five- and three-state systems. If the differences in results can be deemed negligible for all relevant scenarios, the simpler three-state model is a sufficient representation of the larger, five-state system.

The more realistic model of Hinch and colleagues [67] is substantially more complicated than this illustration, because the voltage and calcium dependences of the transitions are explicitly taken into account, parameters are measurable quantities, obtainable from specific experiments, and the channel current is represented with the Goldman–Hodgkin–Katz equation (12.9) that we discussed before. Even in this more complex implementation, validation studies attested that the reduction to the three-state model was of acceptable accuracy.

Both experiments and mechanistic models indicate that RyR inactivation is the primary mechanism of termination of calcium release from the SR. This inactivation process has characteristics similar to the LCC activation process, and model construction begins again with five variables for different states of the RyR, which can be reduced to three states, with similar arguments of timescale separation as for the LCC model.

The next step toward a more comprehensive model is the merging of the LCC and RyR models. A minimal representation for each such calcium release unit (CaRU) consists of one LCC, a closely apposed RyR, and the dyadic space. One LCC opening may activate four to six RyRs (see Figure 12.20). The Ca^{2+} flux from the dyadic space to the cytoplasm is assumed to be governed by simple diffusion. Based on this assumption, it is possible to establish an approximate relationship between the concentration of calcium in the dyadic space and that in the cytoplasm. This relationship, which also depends on the currents through the LCC and RyR, is very important because it greatly reduces the number of differential equations in the model: assuming that the relationship is valid, it is no longer necessary to compute the calcium concentration for each dyad. The CaRU model can have nine different states, resulting from combinations of the three LCC states and the three RyR states, but it can again be simplified to three states under the assumption that a separation of timescales is legitimate.

Hinch and collaborators showed that all the different pieces (component models) can be merged into a whole-cell Ca^{2+} regulation model that describes the

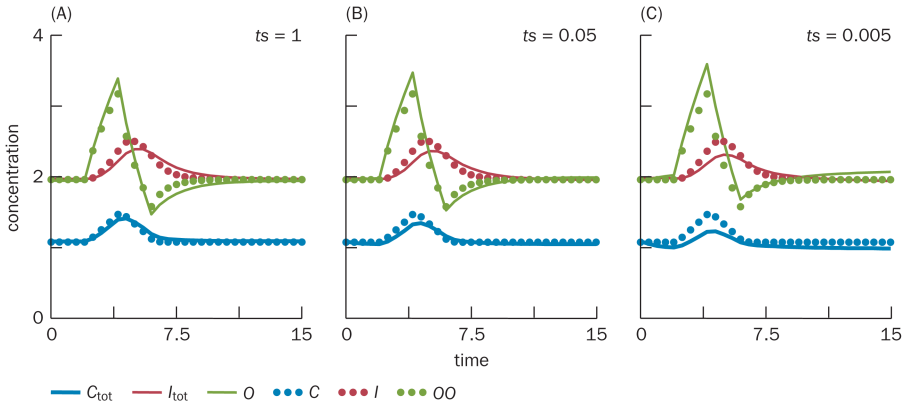


Figure 12.22 Comparison of responses of the five-state and three-state models (12.26) and (12.27). The accuracy of the approximation generated with the simpler model depends on the parameter settings and, in particular, on the magnitude of the timescale separation *ts*. See the text for details.

electrical, chemical, and transport processes in a cardiomyocyte very well [67]. This model contains the CaRU system and the different pumps and ion channels that we discussed before, as well as mechanisms of energy balance. Needless to say, this model is rather complicated. Because of the strategy of simplifying details, followed by validating each simplification, the model is not 100% realistic, but it seems sufficiently accurate as well as amenable to solution and analysis in a reasonable amount of computer time. Indeed, this model is accurate enough to allow the representation of some heart diseases. Of particular importance is sudden cardiac death, which may have a wide variety of root causes.

To summarize, the collection of models sketched above permits a plausible chain of causes and effects, leading from changes in LLCs and RyRs to a prolongation of the action potential (see Figure 12.12C), reduced Ca^{2+} flux, increased Ca^{2+} relaxation time, and subsequent arrhythmia (cf. [68]). Interestingly, some of these molecular events can also be traced back to the expression of ion channels and of genes that code for transporters and Ca^{2+} handling proteins [30], thus extending the concatenation of molecular events and the corresponding models. More generally, changes in any of the parameters in this line of models can become the basis for personalizing models and studying variability in disease outcomes in a population (see Chapter 13).

The models thus connect genetic, molecular, physiological, and clinical phenomena in a step-wise, causative fashion. The same models can furthermore be extended into a whole-heart model, which demonstrates how molecular aberrations at the cellular level may lead to macroscopic changes, such as arrhythmias, ventricular tachycardia, and fibrillation. For such a whole-heart model, one must account for the shape of the heart and the complex orientation of fibers within the heart muscle, as we discussed early in the chapter. At this point, it is no longer possible to avoid or simplify the partial differential equations or finite element representations that permit a detailed description of how electric impulses move along the surface of the heart (see Figure 12.3). Nonetheless, due to the simplicity of these validated sub-models, in particular for LCC and RyR, it is actually feasible to construct detailed whole-heart models that capture the entire chain of causes and effects, from genes to channels and transporters to the calcium dynamics within individual cardiomyocytes, to alterations in the electrical conductance in the heart tissue, and ultimately to arrhythmias and myocardial infarction.

12.5 EXERCISES

12.1 As a nerdy Valentine's gift, produce two- and three-dimensional hearts, using one of the many available heart curves [74–76], such as

(1) $(x^2+y^2-1)^3-x^2y^3=0$;

(2) $x=\sin t\cos t\ln|t|,\quad y=\sqrt{|t|}\cos t,\quad -1\le t\le 1$;

(3) $(x^2+\frac{9}{4}y^2+z^2-1)^3-x^2z^3 -\frac{9}{80}y^2z^3=0,\qquad -3\le x,y,z\le 3.$

Customize your hearts by changing or adding parameters of the models.

12.2 Analyze the van der Pol oscillator in Chapter 5 (Eqn. 5.56).

12.2.1 Explore the role of the parameter k in the van der Pol oscillator. Report your findings.

12.2.2 What happens if both equations in (5.59) are multiplied by the same positive number? What happens if the multiplier is negative? What happens if the two equations are multiplied by different numbers?

12.2.3 Study the effects of larger perturbations on v. Study the effects of perturbations on w; interpret what such perturbations mean.

12.2.4 Study simultaneous perturbations in both v and w.

12.2.5 Study the effects of regularly iterated perturbations (given to the system at regular intervals by an outside pacemaker).

12.3 Analyze the S-system oscillator in (12.1).

12.3.1 Initiate the stable oscillator System B in (12.1) with different initial values. Begin with $X_1 = 0.05$, $X_2 = 2.5$, and $X_1 = 0.75$, $X_2 = 2.5$. Try other combinations. Report and interpret your findings.

12.3.2 Modulate the oscillator by slightly changing some of the exponents or by multiplying the entire systems or some terms by constant (positive or negative) factors. Report your findings.

12.3.3 Multiply both equations in the oscillator System A in (12.1) by the same X_1 or X_2,

raised to some positive or negative power, such as X_2^{-4}. Visualize the results with time courses and phase-plane plots in which X_2 is plotted against X_1. Describe in a report how the shape of the oscillation changes.

12.4 Impulses affecting the heartbeat

12.4.1 In the series of examples of spurious sine function impulses, we changed the frequency. As a consequence, the oscillations became chaotic and eventually the entire heartbeat stopped. Explore what happens if the frequency is retained (for instance, $a = 10$, $a = 1$, $a = 0.5$, or $a = 0.0001$) but the amplitude of the oscillation is altered.

12.4.2 Explain why it is not clinically sufficient that a pacemaker generates a regular impulse pattern of, say, 70 beats per minute. What else is needed?

12.5 Explore further to what degree a highly irregular heartbeat can be made regular by a simplistic pacemaker (see discussion of Figure 12.19). Multiply all equations of the chaotic Lotka–Volterra (LV) oscillator in Chapter 14 by 50, which should result in a rather irregular heartbeat with about 60 to 70 beats per minute. Formulate a sine oscillator (pacemaker) as a pair of ODEs ($\dot{s} = qc$, $\dot{c} = -qs$). Add to each equation in the LV-system a term of the form $p(s + 1)$. Confirm with $p = 0$ the chaotic arrhythmia. Perform three series of experiments, with the goal of regularizing the arrhythmia. In the first series, fix $q = 6$, increase p from 0 in small steps, and study the resulting heartbeat. In the second series, set p at some small value and vary q. In the third series, vary both. Finally, select a combination of p and q that seems to work well, and then alter the demand of the body for more or fewer heartbeats by multiplying all LV equations with the same value from an interval of [0.5, 1.5].

12.6 Describe what the setting of $[Ca^{2+}]_{ext}$ as an independent variable in Section 12.7 entails. What is the importance of its numerical value? What assumptions have to be made and justified to assign the status of an independent rather than a dependent variable? What would have to be changed if $[Ca^{2+}]_{ext}$ were considered a dependent variable? Explain biologically and mathematically.

12.7 Adding the two equations in (12.4) together yields the much simpler equation $\dot{x} + \dot{y} = r_{EC} - p_{CE}y$. Explain the biological and mathematical meaning of this equation. What would it mean (biologically and mathematically) if $r_{EC} = p_{CE} = 0$; $r_{EC} - p_{CE} = 0$; $r_{EC} - p_{CE} \neq 0$?

12.8 Confirm numerically that the oscillation in the system (12.5) is stable. Is (12.5) a closed system? Explain.

12.9 Mine the literature or the Internet for three-dimensional molecular structures of channel proteins in cardiomyocytes.

12.10 The Na^+ concentration is usually much higher outside the cell (about 150mM) than inside the cell (between 5 and 20 mM). Furthermore, while there is a lot of calcium in cells, much is bound and the free calcium concentrations are much lower than those of Na^+, with values of $[Ca^{2+}]_i = 10^{-4}$ mM and $[Ca^{2+}]_o$ between 1 and 3 mm. Compute the equilibrium potentials for Na^+ and $Ca^{2+}$and discuss their contributions to the membrane potential.

12.11 The form of the equations for n, m, and h in the Hodgkin–Huxley model derives from the following argument. Let x be one of the three quantities; then we can write its differential equation in simplified form as

$$\frac{dx}{dt} = \alpha_x(1-x) - \beta_x x$$

Show that this equation has the solution $x(t) = x_\infty + (x_0 - x_\infty)e^{-t/\tau}$, where $x(0) = x_0$, $x_\infty = \alpha x / (\alpha x + \beta x)$, and $\tau_x = 1/\alpha x + \beta x)$. Interpret x_∞.

12.12 Study the effects of initial conditions (of v, n, m, and h) on the shape of the action potential. Use as settings for other parameters: $v_{Na} = 120$ mV, $v_K = -12$ mV, $v_L = 10.6$ mV, $C = 1$, and $I_{app} = 1$.

12.13 Compare the results of the Hodgkin–Huxley model in the text with two other representations for the gating functions n, m, and h. Use in both cases the parameters $v(0) = 12$, $n = 0.33$, $m = 0.05$, $h = 0.6$, $C = 1$, $v_{Na} = 115$, $v_K = -12$, and $v_L = 10.6$. Study the voltage and gating functions and discuss repeated beats.

(A) Explore the following functions (adapted from J. Peterson, Clemson University) with $\bar{g}_{Na} = 120$, $\bar{g}_K = 3.6$, and $g_L = 0.3$:

$$\begin{aligned}
\alpha_n(v) &= \frac{0.01(v+50)}{1-e^{-(v+50)/10}},\\
\beta_n(v) &= 0.125e^{-(v+60)/80},\\
\alpha_m(v) &= \frac{0.1(v+35)}{1-e^{-(v+35)/10}},\\
\beta_m(v) &= 4.0e^{-(v+60)/18},\\
\alpha_h(v) &= 0.07e^{-(v+60)/20},\\
\beta_h(v) &= \frac{1}{1+e^{-(v+30)/10}}.
\end{aligned}$$

(B) Use the following functions with $\bar{g}_{Na} = 120$, $\bar{g}_K = 36$, and $g_L = 3$:

$$\begin{aligned}
\alpha_n(v) &= \frac{0.01(50-v)}{1-e^{(v-50)/10}},\\
\beta_n(v) &= 0.125e^{(v-60)/80},\\
\alpha_m(v) &= \frac{0.1(35-v)}{1-e^{(v-35)/10}},\\
\beta_m(v) &= 4.0e^{(v-60)/18},\\
\alpha_h(v) &= 0.07e^{(v-60)/20},\\
\beta_h(v) &= \frac{1}{1+e^{(v-30)/10}}.
\end{aligned}$$

12.14 Perform simulations assessing the effects of resetting v, n, m, and h, and combinations thereof, in the Hodgkin–Huxley model. Report and interpret your results.

12.15 Add to the equation for n in (12.24) an artificial leakage term of the type $-pn$, where p is a small number. Test with simulations the effects of different values of p on the dynamics of voltage v. If the system starts to beat, test whether the pattern is a limit cycle. A strict mathematical test is difficult for these types of systems, leading Fitzhugh and others to study a reduced system in which n and h were assumed to be constant. This assumption allowed them to study the bifurcation and limit cycle behavior in a two-variable system that was much easier to handle. If you have sufficient mathematical background, read the elegant analysis in the original papers [60, 61] or in textbooks in the field [8, 59] and summarize the highlights of the analysis. See also Exercise 12.17.

12.16 Test with simulations whether the combined Hodgkin–Huxley/Somogy–Stucki model exhibits a stable limit cycle.

12.17 FitzHugh and Nagumo [58, 60, 61] introduced a two-variable ODE system for repeated action potentials, namely

$$\dot{V} = V(a - V)(V - 1) - W$$

$$\dot{W} = bV - cW\ ,$$

where V and W represent the membrane voltage and the activation of the current out of the cell. The parameters b and c have positive values; a may be positive or negative. Under the right conditions, the system can switch from a stable steady state to a limit cycle oscillation. Analyze at least three combinations of a and c, while $b = 0.01$, namely:

$$\text{Case 1}: (a,c) = (0.1, 0.02)$$

$$\text{Case 2}: (a,c) = (-0.1, 0.02)$$

$$\text{Case 3}: (a,c) = (0.1, 0.1)$$

For each case:

1. Draw the nullclines of the system in a plot of W against V.
2. Determine all steady states of the system.
3. Determine the stability of each steady state.
4. Use the methods of Chapter 5 to determine in which manner the system approaches each steady state (as shown in Figure 5.8).
5. Implement the system in PLAS or some other software. Run simulations of V and W and plot time courses, as well as the corresponding 2-d phase plots.
6. Create vector field plots and superimpose the corresponding nullclines, which are lines where either $\dot{V}$ or $\dot{W}$ equals 0.
7. The system is supposed to allow stable steady states as well as limit cycles. For $b = 0.01$ and $c = 0.02$, explore the transitions of the system for different values of a.

12.18 In the context of calcium dynamics in the dyad, we evoked the central limit theorem. Explore this theorem in four steps:

12.18.1 Roll one die 50 times and report the outcome on a graph showing on the horizontal axis the number of the roll (1, . . . , 50) and on the vertical axis the result (the number rolled, between 1 and 6). Explain to which aspect of the CICR system this experiment corresponds. Create a histogram of the results.

12.18.2 Repeat the experiment with two, three, and four dice simultaneously and report on the vertical axis the sum of numbers rolled, divided by the number of dice. Compare and interpret the results of these experiments and the previous experiment with one die. Create a histogram of the results.

12.18.3 Write a computer program to do a series of analogous experiments, where you assume that each of 12,000 dyads contains a random number of Ca^{2+} ions that is between 10 and 100. Let the program graph the sum of ions, divided by the number of dyads, for each step in the experiment. Interpret the results mathematically and biologically. Create a histogram of the results.

12.19 Analyze the accuracy (or loss of accuracy) due to the separation of timescales with a simple linear pathway model with feedback that is affected by noise. The pathway has the structure shown in **Figure 12.23** and the describing model is given by

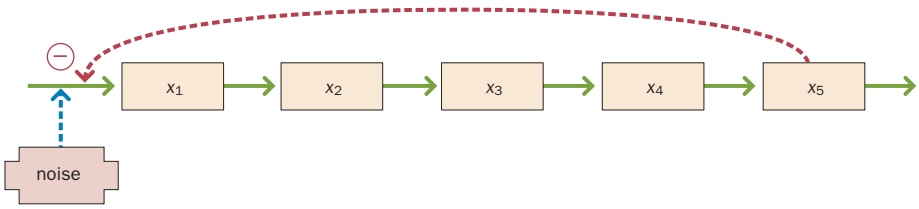


Figure 12.23 Simple pathway with feedback and exposure to noise. The effect of the noise depends on the timescales of the pathway and the input noise.

$$\begin{aligned}
\dot{x}_1 &= p x_5^{-0.3} - x_1^{0.5}, \quad x_1 = 2, \quad p = 1,\\
\dot{x}_2 &= x_1^{0.5} - x_2^{0.75}, \quad x_2 = 1,\\
\dot{x}_3 &= x_2^{0.75} - x_3^{0.4}, \quad x_3 = 1,\\
\dot{x}_4 &= x_3^{0.4} - x_4^{0.5}, \quad x_4 = 1,\\
\dot{x}_5 &= x_4^{0.5} - x_5^{0.4}, \quad x_5 = 1.
\end{aligned}$$

The noise is modeled as a sine wave with small amplitude, namely

$$\begin{aligned}
\dot{s} &= (1-c)\cdot(ts), \quad s = 1, \quad ts = 1,\\
\dot{c} &= (s-1)\cdot(ts), \quad c = 0.8.
\end{aligned}$$

The parameter *ts* represents the timescale, i.e., the frequency. To familiarize yourself with the two components, implement them separately in a differential equation solver and explore the effects of changing the initial value of x_1, the value of the parameter *p*, and the initial values of *s* and *c*. Once you have a feel for these components, connect the two models by putting them together in your solver, replacing *p* with *s* in the first equation and setting *ts* = 0. What do these settings mean? Now the actual series of experiments begins. For each experiment, set *ts* to a different value between 0 and 1000. Report and interpret your findings.

12.20 Explore different LCC models.

12.20.1 Introduce more and different perturbations to one or more variables of the systems in (12.26) and (12.27). Explain the results.

12.20.2 In the perturbations in (12.26) and (12.27), we set the parameter *perturb* first to 1, then to –1, and then to 0. At the end, the systems are back at their original steady states. Explain why the system does not return to its original steady state if, for instance, the second perturbation is –2.

12.21 Search the literature and the Internet for genes whose altered expression is associated with different diseases or temporary problems with the heart. Similarly search for proteins. Pinpoint as closely as possible where and how alterations in these genes or proteins affect heart function and where they would enter any of the models we discussed.

12.22 Mine the literature and the Internet for finite element methods in heart modeling. Select one study using these methods and summarize it in a two-page report.

12.23 Write a two-page report about spiral waves in cardiovascular disease.

FURTHER READING

Alberts B, Johnson A, Lewis J, et al. *Molecular Biology of the Cell*, 5th ed. Garland Science, New York, NY, 2008.

Bassingthwaighte J, Hunter P & Noble D. The Cardiac Physiome: Perspectives for the future. *Exp Physiol.* 94 (2009) 597–605.

Crampin EJ, Halstead M, Hunter P, et al. Computational physiology and the Physiome Project. *Exp Physiol.* 89 (2004) 1–26.

Keener J & Sneyd J. *Mathematical Physiology*. Springer, New York, NY, 1998.

Klabunde RE. *Cardiovascular Physiology Concepts*, 2nd ed. Lippincott Williams & Wilkins, Philadelphia, PA, 2011.

Nerbonne JM & Kass RS. Molecular physiology of cardiac repolarization. *Physiol Rev.* 85 (2005) 1205–1253.

Noble D. From the Hodgkin–Huxley axon to the virtual heart. *J Physiol.* 580 (2007) 15–22.

Winslow RL, Tanskanen A, Chen M & Greenstein JL. Multiscale modeling of calcium signaling in the cardiac dyad. *Ann NY Acad Sci.* 1080 (2006) 362–375.

REFERENCES

1. National_Institute_of_Aging, https://www.nia.nih.gov/health/heart-health-and-aging, Heart Health and Aging.
2. Clinic, C., *Heart*, https://my.clevelandclinic.org/health/body/21704-heart.
3. Cherry, E.M. and F. Fenton, *Virtual Heart*, https://thevirtualheart.org/whoweare.html.
4. Klabunde, R.E., *Cardiovascular Physiology Concepts*. 2nd ed. 2005, Philadelphia: Lippincott Williams & Wilkins.
5. Heart, *Human Heart*, http://science.nationalgeographic.com/staticfiles/NGS/Shared/StaticFiles/Science/Images/Content/heart-display-1037731-sw.jpg, 2011.
6. Full, R.J. and D.E. Koditschek, Templates and anchors: Neuromechanical hypotheses of legged locomotion on land. *J Exp Biol*, 1999. **202**(23): p. 3325–3332.
7. Kumbale, C.M., Q. Zhang, and E.O. Voit, Hepatic cholesterol biosynthesis and dioxin-induced dysregulation: A multiscale computational approach. *Food Chem Toxicol*, 2023. **181**: p. 114086.
8. Keener, J. and J. Sneyd, *Mathematical Physiology*. 2nd ed. 1998, New York, NY: Springer Verlag.
9. Noble, D., Cardiac action and pacemaker potentials based on the Hodgkin-Huxley equations. *Nature*, 1960. **4749**(188): p. 495–497.
10. Noble, D., A modification of the Hodgkin-Huxley equations applicable to Purkinje fibre action and pace-maker potentials. *J Physiol*, 1962. **160**: p. 317–352.
11. Bassingthwaighte, J., P. Hunter, and D. Noble, The Cardiac Physiome: Perspectives for the future. *Exp Physiol*, 2009. **94**: p. 597–605.
12. Crampin, E.J., et al., Computational physiology and the Physiome Project. *Exp Physiol*, 2004. **89**(1): p. 1–26.
13. Greenstein, J.L. and R.L. Winslow, Integrative systems models of cardiac excitation-contraction coupling. *Circ Res*, 2011. **108**(1): p. 70–84.
14. Hunter, P.J. and T.K. Borg, Integration from proteins to organs: The Physiome Project. *Nat Rev Molec & Cell Biol*, 2003. **4**: p. 237–243.
15. Physiome, *The Physiome Project*, http://www.physiome.org/Links/websites.html, 2011.

16. Waters, S.L., et al., Theoretical models for coronary vascular biomechanics: Progress & challenges. *Prog Biophys Mol Biol*, 2011. **104**(1–3): p. 49–76.
17. Virtual_Liver, *The Virtual Liver Network*, https://www.h-its.org/projects/virtual-liver/, 2011.
18. Tawhai, M.H., E.A. Hoffman, and C.L. Lin, The lung physiome: Merging imaging-based measures with predictive computational models. *Wiley Interdiscip Rev Syst Biol Med*, 2009. **1**(1): p. 61–72.
19. Blue-Brain-Project, http://bluebrain.epfl.ch/. 2010.
20. CellML, *CellML*, http://www.cellml.org, 2011.
21. SBML, *Systems Biology Markup Language*, https://synonym.caltech.edu/, 2011.
22. Allen, D. and J. Kentish, The cellular basis of the length-tension relation in cardiac muscle. *J Mol Cell Cardiol*, 1985. **9**: p. 821–840.
23. Shiels, H.A. and E. White, The Frank-Starling mechanism in vertebrate cardiac myocytes. *J Exp Biol*, 2008. **211**(Pt 13): p. 2005–2013.
24. Smith, L., et al., Cooperative cross-bridge activation of thin filaments contributes to the Frank-Starling mechanism in cardiac muscle. *Biophys J*, 2009. **96**(6): p. 3692–3702.
25. Smith, N.P., et al., Mathematical modeling of the heart: Cell to organ. *Chaos Solutions Fractals*, 2002. **13**: p. 1613–1621.
26. EPFL, *3D Model of Blood Flow by Supercomputer Predicts Heart Attacks*, https://actu.epfl.ch/news/3d-model-of-blood-flow-by-supercomputer-predicts-h/, 2011.
27. Cherry, E.M. and F.H. Fenton, Visualization of spiral and scroll waves in simulated and experimental cardiac tissue. *New J Phys*, 2008. **10**: p. 125016, https://doi.org/10.1088/1367-2630/10/12/125016.
28. Joyner, R.W., R. Wilders, and M.B. Wagner, Propagation of pacemaker activity. *Med Biol Eng Comput,* 2007. **45**(2): p. 177–187.
29. DiFrancesco, D. and D. Noble, A model of cardiac electrical activity incorporating ionic pumps and concentration changes. *Phil Trans R Soc Lond B Biol Sci*, 1985. **10**: p. 353–398.
30. Nerbonne, J.M. and R.S. Kass, Molecular physiology of cardiac repolarization. *Physiol Rev*, 2005. **85**(4): p. 1205–1253.
31. Nattel, S., New ideas about atrial fibrillation 50 years on. *Nature*, 2002. **415**: p. 219–226.
32. Fink, M., et al., Cardiac cell modelling: Observations from the heart of the cardiac physiome project. *Prog Biophys Mol Biol*, 2011. **104**(1–3): p. 2–21.
33. Nyhan, D. and T.J.J. Blanck, Cardiac physiology, in *Foundations of Anesthesia*. 1993, St. Louis, MO: Mosby Ltd., p. 473–484.
34. van der Pol, B. and J. van der Mark, Frequency demultiplication. *Nature*, 1927. **120**: p. 363–364.
35. van der Pol, B. and J. van der Mark, The heart beat considered as a relaxation oscillation. *Phil Mag (7th series)*, 1928. **6**(Suppl.): p. 763–775.
36. Yin, W. and E.O. Voit, Construction and customization of stable oscillation models in biology. *J Biol Systems*, 2008. **16**(4): p. 463–478.
37. Beard, R.W., et al., The significance of the changes in the continuous fetal heart rate in the first stage of labour. *J Obstet Gynaecol Br Commonw*, 1971. **78**(10): p. 865–881.
38. Ferrario, M., et al., Comparison of entropy-based regularity estimators: Application to the fetal heart rate signal for the identification of fetal distress. *IEEE Trans Biomed Eng*, 2006. **53**(1): p. 119–125.
39. Williams, K.P. and F. Galerneau, Intrapartum fetal heart rate patterns in the prediction of neonatal acidemia. *Am J Obstet Gynecol*, 2003. **188**(3): p. 820–823.
40. Lipsitz, L.A., Physiological complexity, aging, and the path to frailty. *Sci Aging Knowledge Environ*, 2004. **2004**(16): p. pe16.
41. Lipsitz, L.A. and A.L. Goldberger, Loss of 'complexity' and aging. Potential applications of fractals and chaos theory to senescence. *JAMA*, 1992. **267**(13): p. 1806–1809.
42. Glass, L. and M.C. Mackey, *From Clock to Chaos. The Rhythms of Life*. 1988, Princeton, NJ: Princeton University Press.
43. Thompson, J.M.T. and H.B. Stewart, *Nonlinear Dynamics and Chaos*. 2002, New York: Wiley and Sons.
44. Endo, M., Calcium release from the sarcoplasmic reticulum. *Physiol Rev*, 1977. **57**(1): p. 71–108.
45. Vinogradova, T.M., et al., Sarcoplasmic reticulum Ca^{2+} pumping kinetics regulates timing of local Ca^{2+} releases and spontaneous beating rate of rabbit sinoatrial node pacemaker cells. *Circ Res*, 2010. **107**(6): p. 767–775.
46. Somogyi, R. and J. Stucki, Hormone-induced calcium oscillations in liver cells can be explained by a simple one pool model. *J Biol Chem*, 1991. **266**(17): p. 11068–11077.
47. Stucki, J. and R. Somogyi, A dialogue on Ca^{2+} oscillations: An attempt to understand the essentials of mechanisms leading to hormone-induced intracellular Ca^{2+} oscillations in various kinds of cell on a theoretical level. *Biochim Biophys Acta*, 1994. **1183**: p. 453–472.
48. Bertram, R., A. Sherman, and L.S. Satin, Electrical bursting, calcium oscillations, and synchronization of pancreatic islets. *Adv Exp Med Biol*, 2010. **654**: p. 261–279.
49. Alberts, B., et al., *Molecular Biology of the Cell*. 2002, New York, NY: Garlands Science.
50. Sipido, K.R., et al., Na/Ca exchange and cardiac ventricular arrhythmias. *Ann NY Acad Sci,* 2007. **1099**: p. 339–348.
51. Ojetti, V., et al., Calcium channel blockers, beta-blockers and digitalis poisoning: Management in the emergency room. *Eur Rev Med Pharmacol Sci,* 2005. **9**(4): p. 241–246.
52. Sherwood, L., *Human Physiology: From Cells to Systems*. 7th ed. 2010, Totnes, UK: Brooks/Cole.
53. Lakatta, E.G., V.A. Maltsev, and T.M. Vinogradova, A coupled SYSTEM of lintracellular Ca 2+ clocks and surface membrane voltage clocks controls the timekeeping mechanism of the heart's pacemaker. *Circ Res*, 2010. **106**: p. 659–673.
54. DiFrancesco, D., The role of the funny current in pacemaker activity. *Circ Res,* 2010. **106**: p. 434–446.
55. Noble, D., From the Hodgkin–Huxley axon to the virtual heart. *J Physiol*, 2007. **580**(1): p. 15–22.
56. Winslow, R.L., et al., Mechanisms of altered excitation-contraction coupling in canine tachycardia-induced heart failure, II: Model studies. *Circ Res*, 1999. **84**: p. 571–586.
57. Brady, A.J. and J.W. Woodbury, The sodium-potassium hypothesis as the basis of electrical activity in frog ventricle. *J Physiol*, 1960. **154**: p. 385–407.
58. Fitzhugh, R., Thresholds and plateaus in the Hodgkin-Huxley nerve equations. *J Gen Physiol*, 1960. **43**: p. 867–896.
59. Izhikevich, E.M., *Dynamical Systems in Neuroscience*. 2007, Cambridge, MA: MIT Press.
60. Fitzhugh, R., Impulses and physiological states in theoretical models of nerve membrane. *Biophys J*, 1961. **1**(6): p. 445–466.
61. Nagumo, J.A. and S. Yoshizawa, An active pulse transmission line simulating nerve axon. *Proc Inst Radio Eng*, 1962. **50**: p. 2061–2070.
62. Edelstein-Keshet, L., *Mathematical Models in Biology*. 2005, New York: Random House.
63. Mubagwa, K., et al., Calcium uptake by the sarcoplasmic reticulum, high energy content and histological changes in ischemic cardiomyopathy. *Cardiovasc Res*, 1998. **37**: p. 515–523.
64. KidneySim, *KidneySim Project*, http://physiome.ibisc.fr/~srt/kidneysim/index.php, 2011.
65. Project, T.V.P.R., https://virtualrat.org/.
66. Cortassa, S., et al., An integrated model of cardiac mitochondrial energy metabolism and calcium dynamics. *Biophys J*, 2003. **84**: p. 2734–2755.

67. Hinch, R., J.L. Greenstein, and R.L. Winslow, Multi-scale models of local control of calcium induced calcium release. *Progr Biophys Molec Biol*, 2006. **90**: p. 136–150.
68. Winslow, R.L., et al., Multiscale modeling of calcium signaling in the cardiac dyad. *Ann NY Acad Sci*, 2006. **1080**: p. 362–375.
69. Tanskanen, A.J., et al., Protein geometry and placement in the cardiac dyad influence macroscopic properties of calcium-induced calcium release. *Biophys J*, 2007. **92**(10): p. 3379–3396.
70. Xing, J., H. Wang, and G. Oster, From continuum Fokker-Planck models to discrete kinetic models. *Biophys J*, 2005. **89**: p. 1551–1563.
71. Hinch, R., et al., A simplified local control model of calcium-Induced calcium release in cardiac ventricular myocytes. *Biophys J*, 2004. **87**(6): p. 3723–3736.
72. Imredy, J.P. and D.T. Yue, Mechanism of Ca^{++}-sensitive inactivation of L-type Ca^{++} channels. *Neuron*, 1994. **12**: p. 1301–1318.
73. Jafri, M.S., J.J. Rice, and R.L. Winslow, Cardiac Ca^{++} dynamics: The roles of ryanodine receptor adaptation and sarcoplasmic reticulum load. *Biophys J*, 1998. **74**: p. 1149–1168.
74. Beutel, E., *Algebraische Kurven*. 1909–11, Leipzig: G.J. Göschen.
75. Taubin, G., An accurate algorithm for rasterizing algebraic curves, in *Second ACM/IEEE Symposium on Solid Modeling and Applications Proceedings*. 1993, New York, NY: Association for Computing Machinery (ACM), p. 221–230.
76. Curve, H., *Mathematische Basteleien*, http://www.mathematische-basteleien.de/heart.htm.

Taylor & Francis
Taylor & Francis Group
http://taylorandfrancis.com

13 Systems Biology in Medicine and Drug Development

When you have read this chapter, you should be able to:

- Identify molecular causes of human variation
- Understand disease progression as a dynamic transition between stable attractors or as a trajectory along a multivariate continuum
- Describe the concepts of transforming an average disease model into a personalized disease model
- Understand the different phases and challenges of the drug development process
- Construct and analyze antibody–target binding models
- Construct and analyze pharmacokinetic models
- Create and analyze differential equation models of disease models
- Realize the enormous potential of virtual clinical trials in the future

13.1 ARE YOU UNIQUE?

13.1.1 Biological Variability and Disease

If you and I and everybody else were exactly the same, life would be rather boring, but medicine and drug development would be much easier. In a simplistic analogy, if one knows how to exchange a windshield wiper motor in one 2005 Ford Mustang, one knows how to do it in another 2005 Ford Mustang. Not so with complex diseases. Two people may present with exactly the same symptoms but react in a distinctly different manner to the same dose of the same drug. As a consequence and in contrast to car repair, it is not sufficient to test a drug regimen on only one person—instead it is necessary to perform large, complex, and expensive clinical trials that often require many years of investigation. Then again, in spite of the clear differences, we humans cannot be all that different from each other. After all, medicine has become quite successful, and most prescription drugs actually work as advertised.

So, how different are we really? Let us start with our genes. As we discussed in Chapter 6, each of us has roughly three billion nucleotide pairs. For most of them,

DOI: 10.1201/9781003402770-13

there are no differences between two people and even among the entire human population. If indeed there is a genetic difference, say in base 5,000,000, between any two people on Earth, then this base is the location of a **SNP** (pronounced "snip" and meaning **single nucleotide polymorphism**, which roughly translates into "many different forms in a given base of the DNA"). According to the best current estimates, SNPs occur within the entire human population about every 100–300 bases, which implies that you and I differ genetically by far, far less than 1% [1]. This does not sound like much, but SNPs account for about 90% of all genetic variation, with other changes occurring in deletions, multiplications of DNA, translocations of DNA from one chromosome to another, and other rather rare alterations. Just considering SNPs, is there a sufficient repertoire of differences to make 6 or 7 billion people unique? A quick and very rough computation leaves no doubt that we will not run out of unique people any time soon. If every hundredth nucleotide is changeable and if we simplistically assume that the probability of a single nucleotide (A, T, C, or G) is the same at each SNP, then there are 30 million locations for SNPs, with a choice of four nucleotides at each location. Try to compute $4^{30,000,000}$. It is a truly enormous number that dwarfs the number of all humans who have ever lived. It even dwarfs by far the number of all atoms in the observable universe, which astronomers believe to be about 10^{80}, give or take a few zeros. Adding to this genetic interpersonal diversity, SNPs are not the only determinants of our individuality. While genes are generally considered the blueprint of an organism, we know that even organisms with exactly the same genetic material are not identical. First, there are identical twins, who may look indistinguishable but may have rather different traits and personalities. Also, we have learned from the cloning of animals such as Dolly the Sheep and CopyCat the Cat that the "offspring" differs from the mother, for instance, in the coloration pattern.

If genes are not telling the whole story, what else is there? One relatively new field of biology focuses on **epigenetics**, which studies differences in the activation and deactivation patterns of genes among otherwise similar cells or organisms (see Chapter 6). These differences are often the consequence of modifications at the gene level, which, however, are not encoded in the DNA sequence. Instead, the DNA is methylated or otherwise modulated with residues at certain locations, which tends to affect the rate of gene transcription. Another possibility is a change in histone proteins around which the DNA is wrapped and which may again affect the rate of gene activation or deactivation. Other variations in an organism are, of course, related to the environment, including exposure to harmful chemicals, diet, and all our lifestyle choices.

All in all, we are different, and no two people are the same. This simple fact makes medicine and the development and application of prescription drugs challenging. Fortunately, the situation is not entirely hopeless. First, not every point mutation or SNP has a major effect. Much of our DNA does not code for proteins, and mutations in coding or noncoding sections may not always be critical. As a result of a mutation, an amino acid may be different in the protein for which the gene codes, but the incorrect amino acid may be located outside the active site of the protein, where no great harm is done. It is also possible that the protein is slightly impaired but that the many control mechanisms in the body offer effective compensation. Maybe most importantly, if we concentrate on one particular disease, very many features and processes elsewhere in the body become irrelevant. Because these facts come to our rescue, we are usually successful treating a child's ear infection, irrespective of her size, weight, race, or hair color. Nonetheless, many diseases are so severe, and prescription drugs may have such worrisome potential side effects, that a new focus in the medical field is on personalized or "precision" medicine that seeks to develop treatments customized to an individual patient.

13.1.2 Modeling Variability and Disease

About 20 years ago, Weston and Hood [2] foresaw that "systems biology . . . will catalyze fundamental changes in the future of health care [. . . and] have a major role in creating a predictive, preventative, and personalized approach to medicine." Similarly, Kitano [3] predicted that "although the road ahead is long and winding, it leads to a future where biology and medicine are transformed into pre-

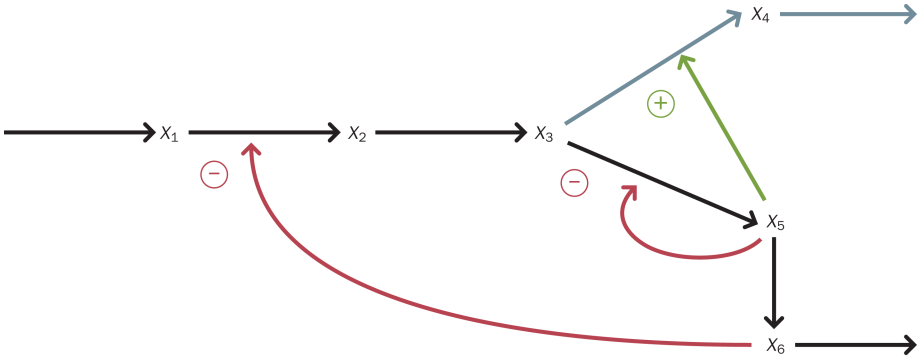


Figure 13.1 Example pathway for illustrating variability and disease. The generic pathway consists of five metabolites, two feedback signals (red), and one activation signal (green). The fluxes to and from X_4 (gray) are very small.

cision engineering." Since then, wider recognition of the importance of the complexity of biological systems, individual variability, and the complicated transitions between health and disease has spawned the new field of systems medicine, whose state of the art was recently reported in the several-thousand-page *Encyclopedia of Systems Medicine* [4].

Individuality in medicine may manifest at the level of static molecular networks (e.g., [5–7]) but it is more likely also to involve regulation and other dynamic features [8–11]. The role of individuality can be readily illustrated with an example from metabolism, where, in a typical case, an inter-personal difference will be manifest in the activity of some enzyme or modulator. It is easy to set up small dynamic models to get a feel for what such a difference may involve (see Chapters 2 and 5). As an example, let us study a generic branched pathway with some regulatory signals (**Figure 13.1**). The mathematical form is not really important for this type of illustration, and we translate the diagram into a mixed **Michaelis–Menten/power-law** model:

$$
\begin{aligned}
\dot{X}_1 &= Input - \frac{V_{\max 1} X_1}{K_{M1}(1 + X_6/K_{I6}) + X_1}, \\
\dot{X}_2 &= \frac{V_{\max 1} X_1}{K_{M1}(1 + X_6/K_{I6}) + X_1} - \frac{V_{\max 2} X_2}{K_{M2} + X_2}, \\
\dot{X}_3 &= \frac{V_{\max 2} X_2}{K_{M2} + X_2} - \beta_{31} X_3^{h_{331}} X_5^{h_{351}} - \beta_{32} X_3^{h_{332}} X_5^{h_{352}}, \\
\dot{X}_4 &= \beta_{31} X_3^{h_{331}} X_5^{h_{351}} - \beta_4 X_4^{h_{44}}, \\
\dot{X}_5 &= \beta_{32} X_3^{h_{332}} X_5^{h_{352}} - \frac{V_{\max 5} X_5}{K_{M5} + X_5}, \\
\dot{X}_6 &= \frac{V_{\max 5} X_5}{K_{M5} + X_5} - \beta_6 X_6^{h_{66}}.
\end{aligned}
\tag{13.1}
$$

Parameter values for the model were chosen almost arbitrarily and are given in **Table 13.1**. We begin our analysis by computing the steady state, which, with three significant digits, is $(X_1, \ldots, X_6) = (2.57, 4.00, 3.23, 3.70, 3.30, 3.13)$. For a typical simulation, we initiate the system at this state and double the input during the time period [8, 12]. A plot of the results is shown in **Figure 13.2**. As is common for this type of pathway structure, the variables rise and fall in order of their position and then return to the original steady state. Notably, X_4 responds very slowly, because the flux going through this pool is very small in comparison with the others.

TABLE 13.1: PARAMETER VALUES FOR THE PATHWAY IN FIGURE 13.1 AND (13.1)

Input	V_{max1}	K_{M1}	K_{I6}	V_{max2}	K_{M2}	β_{31}	h_{331}	h_{351}
2	4	1	2	5	6	0.005	0.8	0.2
β_{32}	h_{332}	h_{352}	β_4	h_{44}	V_{max5}	K_{M5}	β_6	h_{66}
2	0.4	−0.4	0.005	0.9	8	10	1	0.6

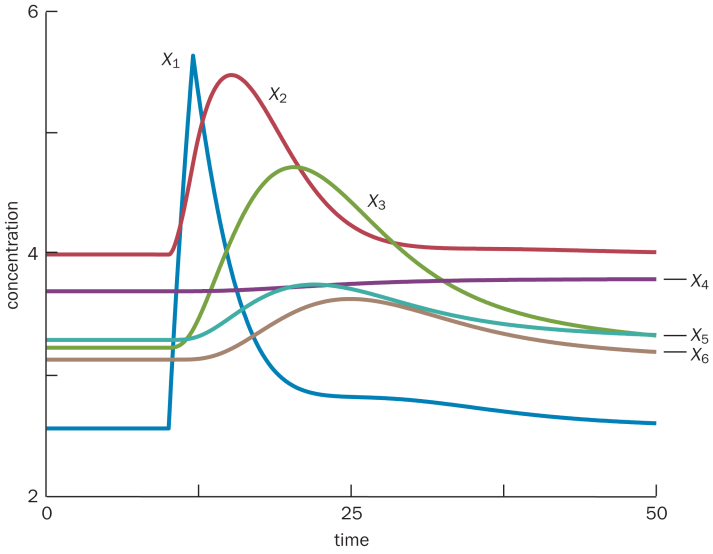


Figure 13.2 Typical scenario simulation of the pathway in Figure 13.1. The simulation used the equations in (13.1) and the parameter values in Table 13.1. From time $t = 10$ to time $t = 12$, the input to the system is doubled from 2 to 4.

It is difficult to predict intuitively how such a system responds to perturbations in parameter values, but it is easy to check computationally. For instance, we may perform a series of simulations in which we raise one parameter at a time by and study the effect on the steady state. Interestingly, we find that most perturbations are very well buffered. Usually, only one or two of the steady-state values deviate by more than 10% from their original values, and deviations of between 20% and 30% only occur in the variables directly affected by the alteration. As a specific example, if V_{max5} is raised by 20% , then X_3 and X_5 decrease by about 21%, but all other variables are essentially unchanged. Exercises 13.1–13.4 explore this example further. In reality, a deviation of 20% often goes unnoticed and does not lead to disease or even symptoms.

While quite simplistic, this cursory analysis implies that regulated systems have a strong buffering capacity. Indeed, a Japanese group made the same observation, when they exposed *Escherichia coli* cells to various genetic and environmental perturbations [13]. While the gene activities and enzymes close to the perturbation were often strongly affected, the remaining metabolic profile was by and large unaffected, owing to a compensatory rerouting of fluxes in the cells. In this light, the reader should revisit the discussions regarding robustness of **small-world networks** (see Chapter 3).

The analysis of perturbations and their ultimate effects provides several insights. First, most steady-state values are not much affected by changes in any of the parameters. Second, an enzyme alteration at the beginning or center of the pathway may affect immediately associated metabolites, but the end products are often only mildly influenced. A direct implication is that, in many disease-related cases, changes in the intermediate metabolites may not be of relevance. Third, the dynamic response to a change in input or enzyme activity may consist of a transient metabolic deviation, which, however, may not be of long-term pertinence with respect to disease. What is not evident from the example is that larger and more complex systems are usually less prone to local perturbations, such as a single altered enzyme activity, than smaller, less-regulated systems. The reason is often—although not always—an enhanced buffering capacity of larger systems.

The ultimate effect of alterations in specific parameters toward a disease are impossible to predict intuitively. Theoretically, given an adequate dynamic model, one could perform sensitivity analysis to assess how sensitive the health-and-disease system is to a specific parameter (Chapter 5), but such an analysis seldom accounts for synergistic and compensatory effects among the parameters of the system. Nevertheless, it is possible to employ Monte-Carlo simulations (Chapter 2) that account for simultaneous changes in several parameter values, which enables them to characterize the variability of outcomes within a population. **Figure 13.3** illustrates the concept. In this case, the "average person," which actually does not exist in reality but in the model results from using values for all parameters that are averaged over the population, displays damped oscillations in response to some stimulus (top right). Replacing these average parameter values with "data" from different individuals

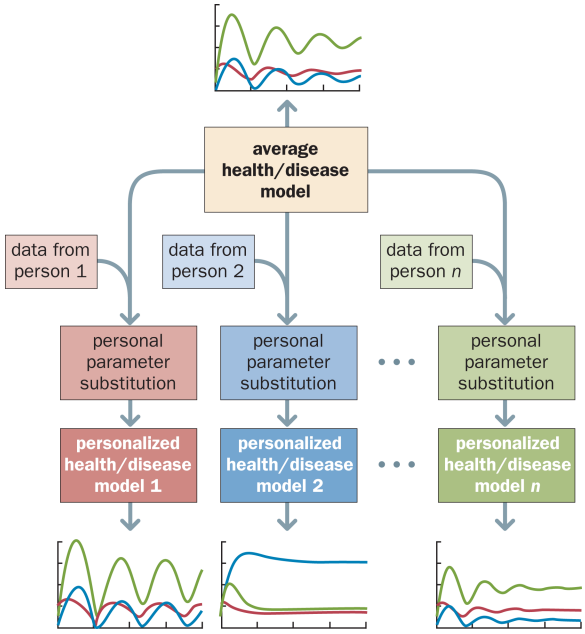


Figure 13.3 Outcome variability within a population. The implementation of the model with averaged parameters yields damped oscillations in response to some stimulus (top right). In variation to this result, personalized parameter values may lead to differently shaped oscillations or reveal responses where the oscillations do not exist. Adapted from [19].

reveals that the oscillations may have different shapes or disappear altogether. The necessary individualized data may be obtained with traditional methods but in the future will very likely be collected with modern methods of multi-omics analysis, combined with single-cell techniques and sophisticated tools of data science, machine learning, and artificial intelligence [14–18]. Section 13.9 pursues these concepts of personalization toward the construction of virtual clinical trials.

13.2 PERSONALIZED MEDICINE AND PREDICTIVE HEALTH

Even if it seems that biomedical systems can compensate for altered components and processes quite well, we know of course that there are many diseases that cause long-term pain and suffering. In a relatively small number of cases, these diseases are caused by a single variation at the genome level. Perhaps the best example is sickle cell anemia, which is caused by a point mutation in the hemoglobin gene *HBB*. While there are many variants, the most common is very well understood: it consists of the substitution of the hydrophilic glutamic acid with the hydrophobic amino acid valine at the sixth amino acid position of the *HBB* polypeptide chain. So, we know exactly what is wrong, but we cannot do much about it.

In contrast to single gene diseases like sickle cell anemia and cystic fibrosis, most disease pathologies result from a combination of genetic predisposition (as manifest in several gene alterations) and epigenetics, as well as environmental and possibly other factors, which nowadays is collectively called the *exposome* [20]. These combinations make it difficult to answer questions such as: how can it be that some nonsmokers develop lung cancer while some long-term chain-smokers do not? Why do people respond differently to the same treatment? These types of questions are at the core of two new and related approaches to medicine: **personalized** or **precision medicine** and **predictive health.**

The key goal of personalized medicine is the prevention and treatment of disease in a fashion that is custom-tailored to the individual. For instance, instead of determining the dose of a prescription drug simply based on body weight and using average information obtained from large populations to assess the efficacy of the drug per kilogram body weight, the dose is computed based on the personal characteristics of the patient. These include obvious information such as body weight, sex, and age, but also gene markers, relevant enzyme profiles, metabolic rate, and a host of other **biomarkers** that could be of relevance to the disease and its treatment. Ideally, such a personalized, precise treatment in the future will maximize efficacy and minimize undesired side effects.

Predictive health is a conceptual continuation of personalized medicine. Its key goal is to maintain health rather than to treat disease. Thus, predictive health aims to predict disease in an individual before it manifests itself. The main targets are chronic diseases such as diabetes, cancer, and neurodegeneration. These diseases often begin with a long, silent phase, where a person does not even perceive symptoms. However, this early phase constitutes a measurable deviation from the norm, and although the person is apparently healthy, her or his biochemistry, physiology, gene expression, or protein and metabolite profiles already contain early warning signs indicating that the person is on an undesirable trajectory toward a chronic disease. If we had a reliable and comprehensive catalog of such warning signs and if healthy people received early screening and treatment, while they were still apparently healthy, chronic diseases could potentially be prevented or managed from their early stages on. There is plenty of indication that such a treatment option would be much cheaper than treatments of chronic disease that are typical in today's world, and efforts are underway to promote predictive health [21].

One intriguing challenge for any type of personalized medicine and predictive health is that it is not at all easy to define what exactly health or disease means. We have seen previously that there can be a lot of variability among healthy individuals, but we also know that the transition step from health to disease does not have to be large. A straightforward distinction between health and disease may be a different set of important parameters, such as blood pressure, cholesterol, or some genetic predisposition (see previous discussion and [8]). An extension of this view is that such changes collectively may disturb the normal state of **homeostasis** and lead to a less desirable, stable attractor (see Chapter 5 and [11, 22, 23]).

The transition from health to disease can occur gradually or abruptly [24], sometimes in the manner of a stochastic multi-variate system with hysteresis [11], corresponding to the situation of simple hysteresis we encountered in Chapter 10 (see Box 10.1 and [25, 26]). Some authors have proposed that disease may be the trade-off to the physiological robustness of the human body or the loss of complexity in the dynamics of a physiological subsystem [27, 28].

13.2.1 Data Needs and Biomarkers

Two components are required to realize ideas of personalized medicine and predictive health: first, lots of data; second, computational methods that are capable of integrating and analyzing these data. The data must cover many physiological aspects of individuals when they are healthy and follow them throughout their healthy life and possibly into disease. The data from a single person will not shed sufficient light on complex disease development processes, but a growing collection of many such personal trajectories, some staying within the healthy range and some diverting toward disease, will eventually reveal an increasingly comprehensive picture of how disease develops and may suggest which early and very early signs (biomarkers) should be considered in regular screening examinations of healthy individuals. In more general terms, these data and their analysis will permit new and sharper definitions of what it means to be healthy, premorbid (with early signs of disease but without too severe symptoms), or diseased [8].

The tracing back of a disease to different disease predictors, late biomarkers, and ultimately early biomarkers (**Figure 13.4**) may seem to lie far in the future; however, in some instances, it has already become standard. A good example is an elevated

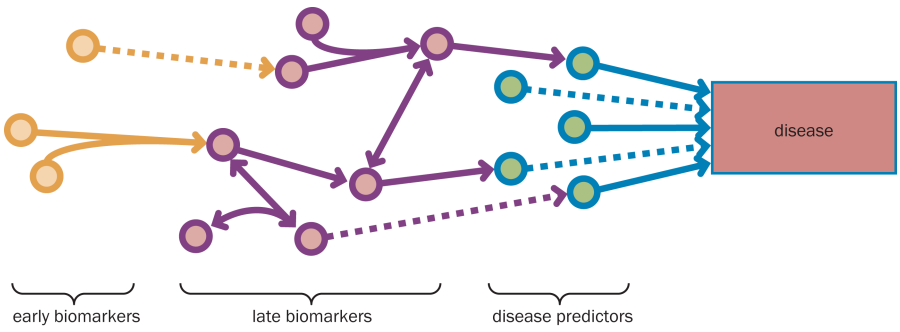


Figure 13.4 Hierarchical biomarker networks. Disease is usually preceded by disease predictors. These are most likely preceded by late and early biomarkers.

blood sugar level, which is not even measured anymore but estimated from the percentage of red blood cells with sugar attached to hemoglobin. In itself, elevated blood sugar does not constitute a disease but is often the precursor of type 2 diabetes. Another example is amniocentesis, a means of diagnosing a fetus within the womb by testing a small amount of amniotic fluid for genetic abnormalities and infections. Even though a baby might appear healthy in an ultrasound examination, the amniocentesis might foreshadow diseases later in life, based on genetic markers.

Biomarkers can also be indicators of whether a particular drug treatment might be effective, before the drug is actually administered [29]. In the new field of **pharmacogenomics**, analyses of expressed genes, especially those from the class encoding cytochrome P450 enzymes, are used to predict the sensitivity of individuals to specific drugs. A beautiful example is a prognostic genome analysis in children with acute lymphoblastic leukemia (ALL). It had been known that many children with ALL responded very well to standard chemotherapy, whereas others developed severe side effects that greatly compromised the efficacy and applicability of the drugs. Scientists at St. Jude Children's Research Hospital in Memphis, Tennessee, discovered that microarray and in vitro leukemia cell killing assays revealed between 20 and 45 differentially expressed genes that were associated with single-drug resistance or cross-resistance [30], which led to a highly accurate predictive test for young ALL patients, distinguishing responders from nonresponders to drug treatment [31]. For other examples of predictive biomarker studies based on personalized-omics and multi-omics analysis see [16, 17].

As in the example of ALL, the first step toward rational predictions of disease and treatment outcome will be a statistical association analysis, where certain patterns in the data are predominantly observed in cases of disease or drug resistance, while other patterns are more often associated with health and a successful drug treatment. As an extension of common statistical association analyses, it is possible to train an **artificial neural network (ANN)** or some other machine learning technique to consider many markers and to classify individuals as healthy or at-risk. Because ANNs can be large, they have the potential of identifying at-risk patients earlier and more accurately than standard association methods of statistics. An example is the computer-assisted ProstAsure test for prostate cancer [32].

The identification of associations between biomarkers and disease is invaluable but not the ultimate goal of predictive health. Building upon association data, the next step is the construction of causal models, which not only associate certain biomarkers with disease but also demonstrate that a particular disease is actually caused by a variation in these biomarkers. Once this causality has been established, the biomarker is no longer only a diagnostic or prognostic symptom but becomes a true drug target. The following sections indicate how modeling and systems biology of the future might contribute to our understanding of personalized health, disease, and treatment.

13.2.2 Personalizing Mathematical Models

Let us begin by painting a picture, in very broad strokes, of how today's medicine works. **Figure 13.5** shows a simplified diagram. The left side depicts, in reddish colors, the typical paths of current knowledge discovery. The two main sources of input are epidemiological studies, which study large populations in order to associate diseases with risk factors and modern biology, which characterizes biochemical and physiological mechanisms that might lead to disease if they are altered and quantifies these mechanisms with pertinent parameters. The results of these two approaches ultimately enter clinical trials, which lead to prescriptions of treatments that are, by the nature of the process, based on population averages. Process descriptions and parameters have also been the basis for the design of disease models that, due to the underlying data, again address average individuals. These models can be diagnosed and analyzed as we discussed in Chapter 5 and may lead to better characterizations of a disease, as well as to a better understanding of averaged treatment options.

Systems biology enters the field with two components. Experimental systems biology offers new and very powerful means of identifying molecular and physiological mechanisms and their parameters. It will also be the foundation of information gathering in personalized precision medicine and predictive health, as

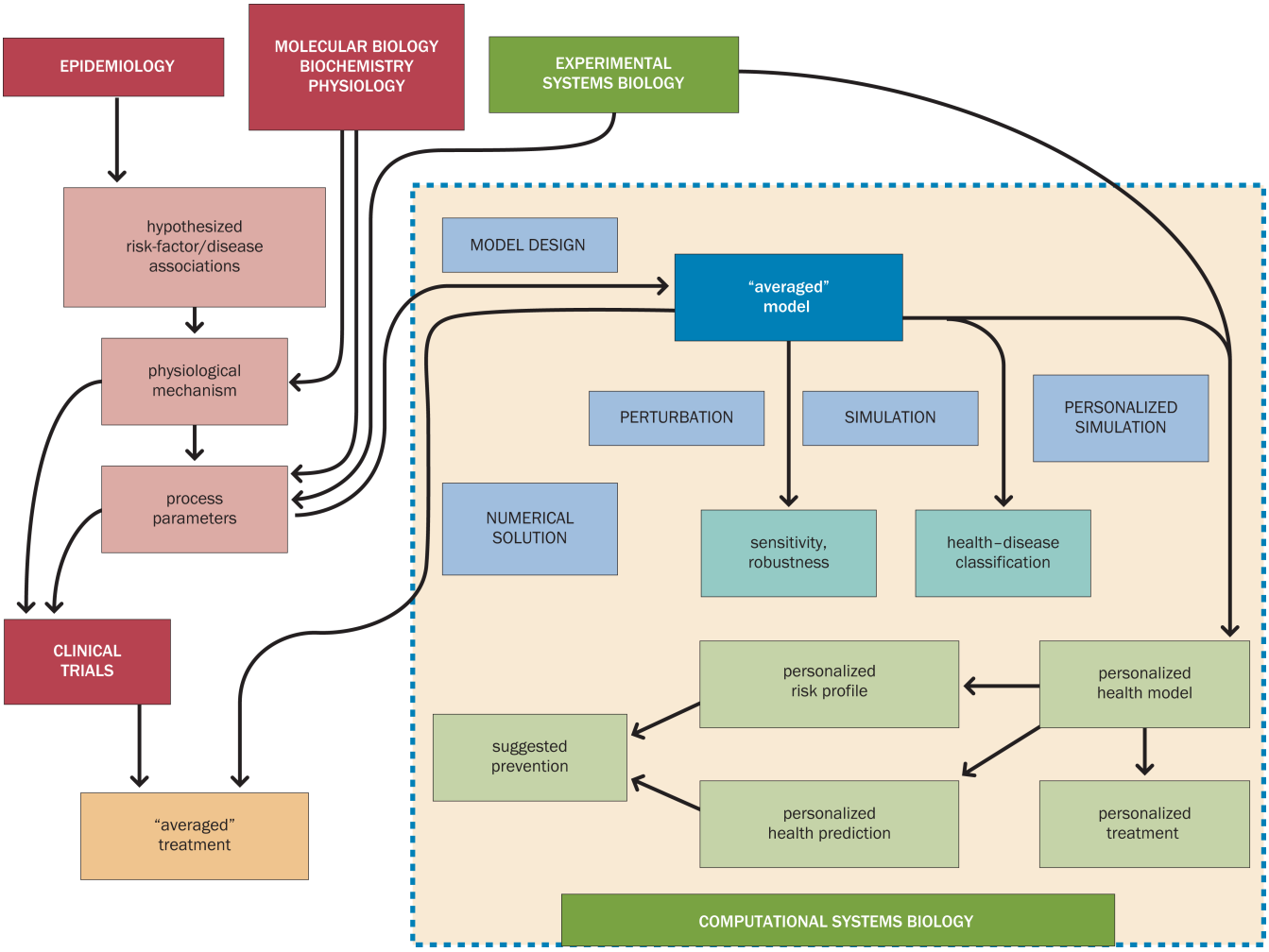


Figure 13.5 Traditional and personalized medicine. The simplified diagram shows in reddish colors the current paradigm of medicine and in blue and green colors an emerging paradigm involving systems biology. (Adapted from Voit EO & Brigham KL. *Open Pathol. J.* 2 [2008] 68–70.)

discussed before. Computational systems biology will use averaged disease models, personalize them based on individualized data, yield personal risk profiles, and suggest possible prevention strategies.

The computational aspect of personalized medicine and predictive health requires the personalization of models [8, 9, 11, 33]. Conceptually, this step is fairly simple, and some companies have achieved successes in simulating virtual patients. In generic terms, one begins with a disease model that had been constructed and parameterized with the best available data and information currently available. This model reflects population averages, rather than one single individual. Suppose now that many biomarkers had been measured for a specific person and that they do not wildly fluctuate over time. A comprehensive model should directly or indirectly permit these biomarkers to be entered into the model. For instance, an altered enzyme activity might be translated into the K_M value of some enzymatic step. A deactivated gene could, at least in first approximation, be translated into the decreased activity of a protein, which would in turn correspond to a decreased transport rate, altered catalytic conversion, a muffled signal transduction process, or some other effect on the model. Because the biomarkers were measured in a specific person, their values are most likely different from the average over an entire population. Thus, the original average model becomes gradually customized by changing each average biomarker value to the measured, personal biomarker value, and every person can be seen as a unique point in a huge, dynamically changing parameter space.

In the simple example at the beginning of the chapter, we modified parameters by increasing them by 20%. In some sense, this exercise corresponded to an

abstract personalization. Here, we simultaneously change many parameters, namely all those for which personalized biomarker measurements are available. Presumably, this personal biomarker profile is not complete, which requires us to retain average values for the remaining parameters, in the hope that the individual is more or less normal in aspects that were not specifically measured. If possible, missing measurements can later be substituted with personal values, or one might perform a simulation study, for instance, using **Monte Carlo simulation** (see Chapter 2), to determine how influential the average parameters really are. If the system is not sensitive to these parameters, it might not even be necessary to obtain personalized values.

The personalization of a model opens new avenues for various prevention and treatment options. First, one will probably simulate the model from the current "initial" state forward. Such a simulation shows whether the individual converges to some healthy steady state or whether the individual biomarker profile leads to the slow but unrelenting accumulation of molecules such as cholesterol in the plasma and liver (see Exercises 13.2 and 13.3), reflecting a progression along the spectrum of hyperlipidemia. In addition to the steady state, the model also indicates the speed of change and shows whether any transients during the scenario simulation could be dangerous. This baseline simulation is expanded into scenario simulations, where one attempts to intercept undesired trajectories, such as the accumulation of harmful substances. It is expected that a reliable model will identify successful strategies. It may also help screen for interventions that at first seem a good idea but in fact do not really work, for one unexpected reason or another.

The collection of many personalized health and disease trajectories will begin to shape a deeper understanding of causes, symptoms, and progressions of disease processes. It will show that some abnormal biomarkers alone are not dangerous but that they may forecast disease if they are combined with other altered biomarkers. Ultimately, a comprehensive collection of trajectories will help us determine effective early biomarkers that precede late biomarkers and precursors of disease, as shown in Figure 13.5.

Looking at the complexity of molecular and physiological systems, one might come to the conclusion that personalized medicine is a dream without much of a chance of realization. Three arguments counter this pessimistic view.

First, physicians have actually been pursuing the same strategy since the beginnings of medicine: they gained their knowledge from population averages and learned to customize it toward a patient. Maybe the biomarkers were initially simpler, consisting of blood pressure, temperature, and past personal trends such as weight loss. However, in more recent times, physicians have been using many more and specific biomarkers, and the only true difference from a computerized system of personalized medicine is that doctors perform the integration of data in their heads, aided by their knowledge and experience. Looking into the future, this integration will become increasingly more difficult, because the number of measurable biomarkers will continue to climb rapidly, eventually overwhelming the most astute human mind. It seems that the only remedy against this overload will be computer-aided data integration. One should add that computer predictions will eventually become so complex that the physician is no longer able to explain how a suggestion from a computer was obtained.

The second argument is the success of computational expert systems in select areas of medicine, such as the identification and severity of poisonings from symptoms [34] and emergency medicine [35]. While these systems are based on methods of artificial intelligence, which are quite different from systems modeling with differential equations, their successes suggest that computer-assisted diagnostics and treatment are not out of the question in medicine. In the much simpler case of car mechanics, computerized diagnostics has become an accepted standard.

Third, while most of our computational models of molecular systems are presently too coarse and insufficiently comprehensive to facilitate personalized medicine, specific numerical, model-based predictions are already possible in other medical fields. For instance, before cosmetic surgeries are performed, they are often simulated first in a virtual environment that permits personalization. Similarly, a group at Emory University and the Georgia Institute of Technology have recently developed an image-based software system for blood flow reconstruction surgery on the

heart that is custom-tailored to a specific patient [36]. The computational platform allows surgeons to perform, in a simulation, alternative surgical scenarios, and the system computes the postoperative hemodynamic flow properties of each resulting reconstruction. If the dream of personalizing models becomes reality, enormous costs may be saved during the development of new prescription drugs. This aspect is discussed in Section 13.9.

It is clear that the complexity of physiological systems in health and disease will render it very challenging to design and implement precise models in specific individuals. Nonetheless, in most cases, it is not necessary to target the entire human physiology at once. Instead, it will be feasible to move gradually from what we have already accomplished in medicine to increasingly detailed systems of disease processes.

The starting point is a consideration of what health and disease really mean. Precise definitions are difficult, but it is sufficient here to begin with the pragmatic definition of a normal range for a biomarker. We all know that blood pressure of 220 over 120 is probably not so good in the long run and that the body core temperature should remain within a rather tight range for us to function normally. Generically, one may define a normal range for a biomarker as shown in **Figure 13.6(A)**, which acknowledges that the boundary between normal and abnormal is fuzzy. The situation becomes more complicated when we consider two biomarkers simultaneously (Figure 13.6B). In many cases, combinations of extreme values of both biomarkers are no longer healthy so that the green domain is not a rectangle but an ellipse or some other shape with truncated corners (**Figure 13.7A**). The consequence is that the normal domain of a biomarker is no longer absolute. In Figure 13.7B, the orange and blue "patients" have exactly the same value for biomarker 1, but blue is healthy, while orange is not. The purple

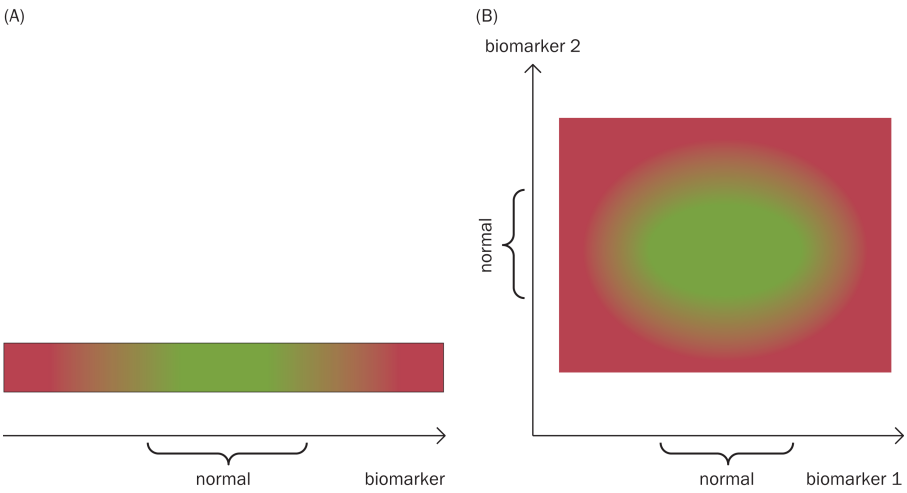


Figure 13.6 Health is related to biomarkers within their normal ranges. (A) For a single biomarker, a normal range is relatively easy to define. (B) For several biomarkers, the healthy domain may be different from a simple combination of normal ranges.

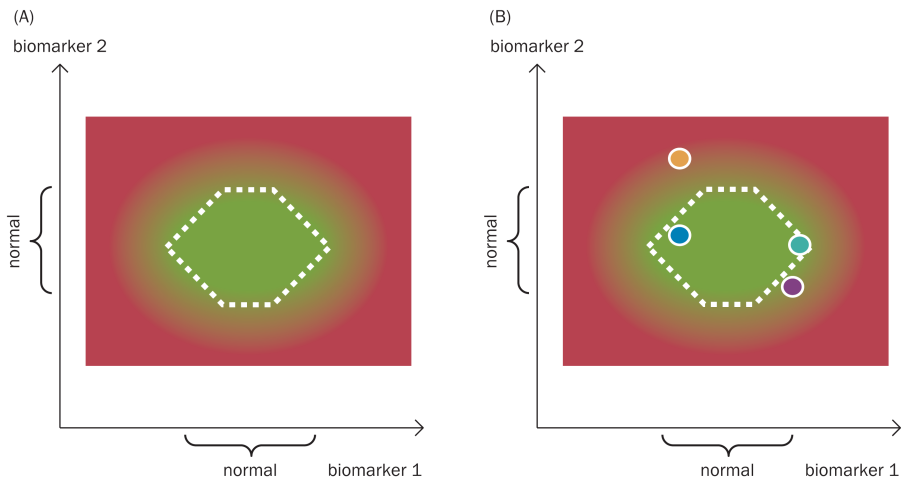


Figure 13.7 Delineation of health and disease. For simplicity, the healthy domain in (A) is defined by the dotted hexagon. (B) shows cases where the consideration of a single biomarker leads to "false positives" or "false negatives." The issue is much more pronounced when many more biomarkers are involved.

patient has (borderline) healthy biomarkers 1 and 2, but the combination is no longer in the healthy domain. The turquoise patient, by contrast, shows an unhealthy reading for biomarker 1 but is still considered healthy. Expressed differently, the outcome of only a single biomarker test can lead to **false-positive** or **false-negative** outcomes. In the first case, the test falsely suggests disease, whereas in the second case, the test falsely indicates health. While simplified, this discussion indicates that individual biomarkers are seldom absolute predictors of health or disease and that their values must be seen within the context of a larger health system where biomarkers are interdependent on one another.

Today's physicians know the normal ranges for many biomarkers, as well as some of the more important combinations of a few biomarkers. With this knowledge, it is possible to develop models that take account of information from many sources and integrate it in a meaningful manner. Over time, such models will improve in quality and reliability and eventually will be able to take hundreds of biomarkers and classify whether their combinations are healthy, premorbid, or indicative of disease. Furthermore, model-based integration of repeated measurements of biomarker combinations over the lifetime of an individual will show whether this individual is on a healthy trajectory or whether he or she is moving dangerously close to the realm of disease. Analyzing and interpreting thousands of such trajectories together will create a rational basis for predictive health [11].

13.3 THE DRUG DEVELOPMENT PROCESS

Between the dawn and dusk of the twentieth century, the average human life expectancy in the United States went up from about 45 to 77 years [37, 38]. Much of this enormous extension of life can be credited to vast improvements in medicine and the availability of drugs and biomedical devices, along with better hygiene, plenty of food, and a deeper understanding of physiology, molecular biology, and microbiology. While the maximal lifespan for humans is unknown, current estimates lie between 120 and 130 years, and as more and more individuals are becoming centenarians, the emphasis of extending life is shifting toward good health at an advanced age. Realizing this emphasis will only be possible with continued developments and improvements in medicine and drug development.

Unfortunately, the rate at which new drugs are introduced has been slowing down considerably in recent years. It is not the case that the pharmaceutical industry has run out of drug targets. In fact, high-throughput biology has been suggesting new drug targets galore. Rather, the main reason seems to be that "Big Pharma" is leery of the length and cost of the drug development process, which may take 10–20 years and require an investment of between one and two billion dollars [39, 40]. Patents protect new drugs for only about 15–20 years, depending on the country, before cheaper generic drugs become competitors. In areas such as HIV/AIDS, the investment in a drug must be recouped much faster, because new formulations and cocktails may begin to challenge a drug after just a few years. Adding to this situation is the enormous attrition rate of new chemical entities (**NCEs**), that is, of active ingredients, during the drug development process [39]. Out of 100 NCEs, only one to five make it into preclinical trials, and only 10% of these "survivors" will ultimately be approved [41]. According to other estimates, only one hundredth of one percent will ultimately make it to the market [42]. Obviously, these numbers depend on many aspects of the disease and the NCE but indicate that the a priori probability of success of an NCE is low.

Reasons for the low success rate are plentiful and include the fact that the pharmaceutical industry has begun to attack very complex diseases—often with more resources for research and development than in the past—and that the standards set forth by the Food and Drug Administration (FDA) in the United States and similar health agencies elsewhere are becoming more and more stringent. The reasons for dismissal of an NCE are manifold as well, but two of them stand out. First, 30–60% of the dismissed drugs are abandoned owing to issues of efficacy and bioavailability. In other words, the human body does not accept the drug in sufficient quantities to permit a therapeutic effect. Another 30% have to be eliminated because of toxicity and serious side effects, while 7% are not com-

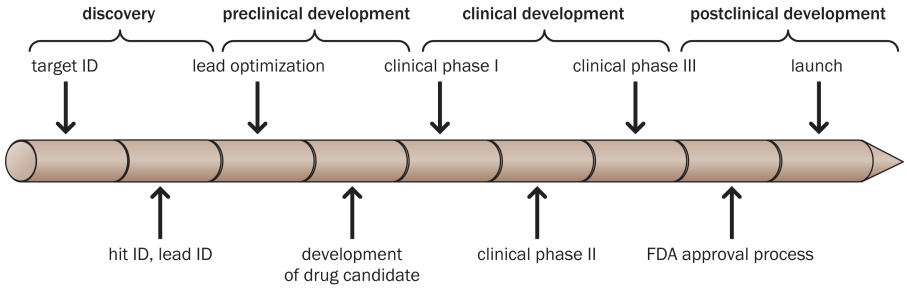


Figure 13.8 The drug development pipeline. The pipeline consists of distinct phases, which any drug has to survive before it can be used for treatment. The process is very costly and time-consuming. Without compromising safety, costs and time can only be saved if drug candidates with undesirable properties are screened out early.

mercially viable [37, 42]. Thus, the landscape of drug discovery, development, and marketing is treacherous and requires significant resources.

So, what exactly are the problems, and are there ways of solving them? Certainly, nobody intends to cut corners or to simplify the testing procedures. Instead, the challenge to be solved may be posed as (A) increasing the overall success rate of the drug development pipeline and (B) accelerating the elimination of an NCE from the pipeline. The former is rather obvious. The latter is important because the later phases of the pipeline, and in particular the phases of clinical trials, may cost hundreds of millions of dollars [40]. Thus, if a problematic new NCE can be eliminated early, substantial amounts of resources are saved. The drug pipeline is depicted in **Figure 13.8**. It consists of four major phases that are readily subdivided into further segments. The outcome of testing at each segment of each phase may suggest abandoning further consideration of the NCE.

The first phase is concerned with early-stage research and discovery, target identification, the identification of small-molecule **hits**, and validation. At the beginning, a specific biological target is identified as being associated with a particular disease. The target could be a receptor, a signaling molecule, an enzyme, or some other protein that plays a crucial role in the disease. The hit one is hoping for is a molecule that interacts with the target in such a fashion that the disease process is interrupted. In the case of a receptor, a hit molecule may be an artificial ligand that attaches to the receptor and blocks it from accepting the disease-associated ligand. In the case of an enzyme, the hit might be an inhibitor. It is not surprising that many possible hits must be investigated. Often, these hits come from extensive combinatorial molecular libraries that a pharmaceutical company has assembled. In most cases, only one or a few of the hit molecules will continue in the drug development pipeline as **leads** for further testing. Hit and lead identifications are nowadays executed in a high-throughput fashion that utilizes many of the methods of modern molecular biology and experimental systems biology [43]. In addition to the initial identification, the target, hit, and lead must be independently validated. This step provides proof of concept that the target may be worth pursuing with the identified leads. For instance, if the target is a receptor, then binding affinities, length of receptor occupancy, and accessibility of the receptor to the lead compound must be investigated.

Validated hits with favorable characteristics that show potential for an improved medication become leads in the next phase of preclinical drug development. The first component of this phase is lead optimization. The lead is subjected to further tests of medicinal chemistry and, in particular, to an optimization process that investigates small molecular variations, which may affect receptor binding or the strength of inhibition with respect to a target enzyme. An important guideline is the exploration of quantitative structure–activity relationships **(QSARs)**, which attempt to assign specific features to chemical structures that might be used as side groups of the lead molecule [44, 45]. In the case of proteins, a similar approach is the creation of peptoids, which are artificial peptide-like compounds that can be designed to work as drugs with desirable properties [46]. Such drugs could be intriguing because of the preeminent role of specific peptide sequences in protein–protein interactions, molecular recognition, and signaling.

The second component of lead optimization is the assessment of dose, efficacy, and acute toxicology in animals. Compounds that are found to be effective in animals and do not lead to acute toxicity are moved forward in the drug development pipeline as drug candidates. The subsequent investigation of drug candidates

constitutes the second step of preclinical drug development. Here, toxicity over longer time horizons is assessed, along with **pharmacokinetic** tests that reveal how long the compound resides within some tissue and the bloodstream. The half-lives in different organs may differ drastically, because, for instance, metabolism in fat is typically slow, while it is the role of the liver to detoxify the body as quickly as possible. In the context of pharmacokinetics, one sometimes speaks of investigating ADME, which refers to absorption, distribution, metabolism, and excretion of a drug (see Section 13.7 for details). Drug candidates surviving the muster of preclinical assessment are moved forward to clinical testing [47].

The clinical phase may easily consume half of all costs of the entire drug development process and half of the time [40]. It is commonly subdivided into three subsequent types of clinical trials. The first, phase I, is primarily concerned with safety issues. The compound, which will already have been found to be safe in animal studies, is administered to a few dozen healthy human volunteers over a time span of 1 to 2 years. If toxicity or other undesired side effects are detected, the testing is stopped, and the drug is not permitted to move forward in its current form. Throughout the clinical phase, studies also investigate different formulations and dosing regimens. In phase II, both efficacy and safety are tested on a relatively small cohort of 100 – 300 patients. This phase again lasts for 1 to 2 years. Finally, in phase III, the drug candidate is used in a randomized clinical trial sometimes involving thousands of patients. The foci of this 2- to 3-year-long study are again efficacy and safety. Because volunteers, patients, and hospitals are involved in clinical trials, this phase of the drug development pipeline is very costly. Roughly 20% of drug candidates survive the three phases of clinical trials.

Once all preclinical and clinical testing has yielded satisfactory results, the drug candidate must be submitted for approval by the Food and Drug Administration (FDA) in the United States and/or corresponding health agencies in other countries. Drug applications to the FDA require vast amounts of documentation about the positive and negative results of all former tests and trials, and approval typically takes 1 to 2 years. The final, post-approval phase of the drug pipeline contains the launch. It consists primarily of the sharing of information with physicians, hospitals, and trade magazines, as well as direct advertising and marketing. This phase can cost $100 million or more [48].

13.4 THE ROLE OF SYSTEMS BIOLOGY IN DRUG DEVELOPMENT

Results of deliberations by think tanks have suggested that academia and the pharmaceutical industry should form partnerships, in which academicians would primarily focus on drug discovery and possibly some of the preclinical studies, while the pharmaceutical industry would take the lead in the expensive clinical trials. However the roles of all potential contributors to the process are split up, it is more important here to ask how systems biology might be able to help improve drug discovery and development. Several roles of systems biology are already emerging, and the industry has been embracing them to various degrees for quite a while. Indeed, the potential of systems biology in drug discovery has been recognized for some time (see, for example, [3, 49, 50]).

The drug development process and its regulation will continue throughout the foreseeable future, but computational modeling within this process received an encouraging boost in December 2022, when the US Federal Drug Administration (FDA) Modernization Act 2.0 (S.52002) was enacted into law. This law permits the use of certain alternatives to animal testing, including cell-based assays, organs chips, and computer modeling [51].

Looking at the drug development pipeline shown in Figure 13.8 from a systems point of view, two distinct types of improvements would obviously contribute to desirable solutions. First, it would be desirable to increase the number of promising hits. And second, it would be highly desirable to find out as early as possible whether particular hits or leads would later be doomed. This activity would not reduce overall attrition, but it would screen out hits and leads before their testing became very expensive, which happens particularly in the clinical development phase.

Regarding increased numbers of hits, there is no doubt that methods of experimental systems biology and of high-throughput data acquisition and analysis are very promising tools for the separate or combined identification of targets, hits, and leads, as well as for global toxicity studies. All major pharmaceutical industries are already heavily engaged in parallelized "omics," robotic screening, bioinformatics, and machine learning, and some smaller companies have sprung up that concentrate on genomic analyses that are made available to larger companies; examples include Selventa [52] and Aitia [53]. Proteomic studies are also being pursued, especially within the realm of signaling systems, where sophisticated methods are beginning to permit crisp assessments of the phosphorylation states of specific signaling proteins [54, 55].

In addition to searching for NCEs, the pharmaceutical industry has begun to invest heavily in the exploration of **biologics** [54, 55]. In contrast to synthetically produced, small-molecule NCEs, these biologics are derived from naturally occurring biological materials. They often have a favorable safety profile, because they are less likely to elicit an immune response, and they break down into natural molecules such as amino acids. In fact, it is sometimes even therapeutically beneficial to supply an unchanged biomolecule to a patient who shows a corresponding deficiency. For example, insulin is given to patients with type 1 diabetes and factor VIII is prescribed for hemophilia A. In other cases, natural compounds are used as templates that are chemically modified, for instance, in order to improve the binding to a receptor. Because it is possible to attach a wide variety of residues to the natural base compounds, the number of potentially efficacious biologics is enormous. As a consequence, the biologics sector of the pharmaceutical industry has in recent years been growing at a swift annual rate and now is a multi-billion-dollar market. Monoclonal antibodies constitute the best-known category of biologics. Other examples include engineered proteases—for instance, from the blood clotting cascade—and aptamers [56]. Aptamers are artificial oligonucleotides that do not code for a peptide or interact with DNA but are designed to bind and thereby deactivate a target, such as the active domain of a disease-linked protein. Modern methods of experimental systems biology make it relatively easy and cheap to produce aptamers in vast variations. Overall, the role of experimental systems biology in drug discovery seems quite evident, and we will not discuss it further.

13.4.1 Computational Target and Lead Identification

Different branches of computational systems biology can be helpful for target and lead identification (see, for example, [57]). For instance, methods of molecular modeling and combinatorial chemistry permit the computational assessment of molecules and active groups and are becoming increasingly reliable predictors of the efficacy and toxicity of permutated drugs and their side groups, as well as of biologics [58–60]. These predictions are, in some sense, based on a new generation of sophisticated QSARs that have been in use for many years. Systems biology is also used increasingly for receptor binding and dosing studies with newly created molecules. Not to be overlooked, computational methods of intelligent data and **text mining** are rapidly moving to the forefront. With thousands of biomedical journals being published on a regular basis, automated alerts and sophisticated searches have become a necessity. In addition to simple key words or phrases, modern search engines are becoming quickly better in semantic searches that may even include images.

Many aspects of systems biology look promising for reducing the cost of drug discovery through effective early screening and hit elimination. The key to moving the elimination of NCEs and biologics closer to the front of the drug pipeline is a solid understanding of the mechanisms of action of a hit or lead with respect to a particular target. Because disease processes are complex, such enhanced understanding greatly benefits from computational models. Beautiful examples of very comprehensive disease models, for instance for diabetes and rheumatoid arthritis, have been developed by Entelos, now part of Rosa [61] and utilized by larger pharmaceutical companies. Companies of this type construct these models with methods discussed in earlier chapters of this book, by mining the literature exhaustively, curating the results very carefully, and converting all pertinent information into processes and customized parameters for simulations of virtual patients. These examples are real-world extensions of the sandbox analysis that we used in the beginning of the chapter (see (13.1)) for illustrative purposes.

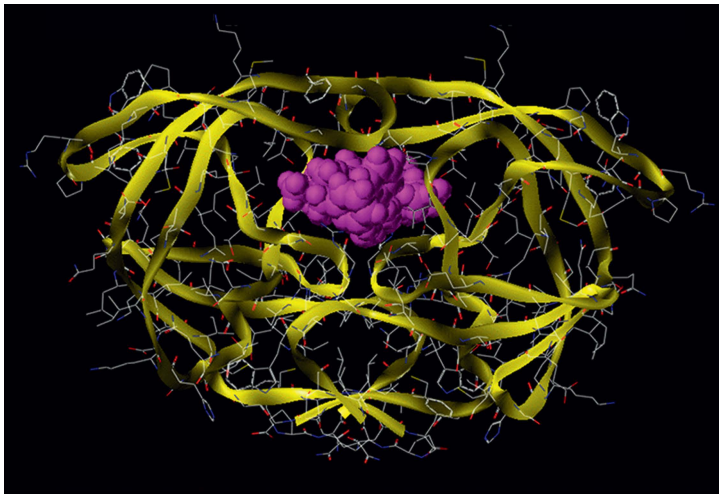


Figure 13.9 Many drugs work by binding to proteins. Here, the FDA-approved drug indinavir docks into the cavity of an HIV protease in a lock-and-key mechanism (Protein Data Bank: PDB 10DW and 1HSG). Courtesy of Juan Cui and Ying Xu, University of Georgia.

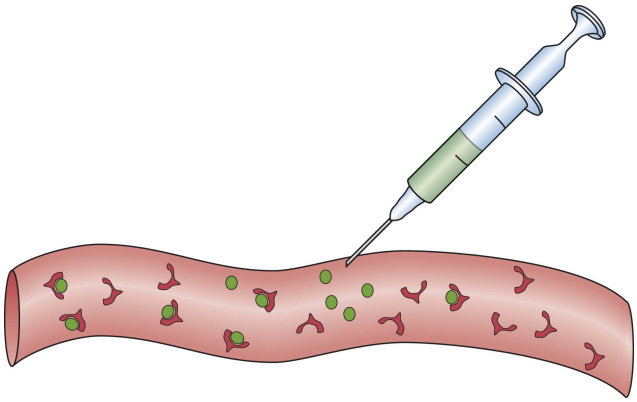


Figure 13.10 Treatment of disease with biologics. Monoclonal antibodies (green circles) are injected into the bloodstream, where they bind reversibly to their target molecules.

While comprehensive models show clear potential, even simple models can be of great value. For instance, it is not always necessary to model a signal transduction pathway in great detail, and it may suffice instead to construct a black-box model that is used as a surrogate for a comprehensive systems model [62]. In the following, we discuss three examples. The first pertains to receptor dynamics, which plays a major role for very many drugs. The second example discusses compartment modeling and pharmacokinetics, which have already been mentioned in Chapter 4. Finally, we will show how dynamic pathway models may help weed out NCEs and contribute to an understanding of safety and toxicity.

13.4.2 Receptor Dynamics

Essentially all prescription drugs need to be taken up by cells, and very many of them are associated with membrane-bound receptors (Figure 13.9). It is therefore of great interest to study binding characteristics, and even relatively small mathematical models can yield valuable insights. As an example, let us consider two scenarios involving therapeutic antibodies. In the first, very much simplified, case, adapted from a study by the pharmaceutical industry [63], a monoclonal antibody M is injected intravenously and, in the bloodstream, binds to a target T, which could be a signaling peptide or a hormone. On encountering one another, M and T form a complex C, which prevents T from exerting a negative effect on the system (Figure 13.10). The goal of the treatment is to reduce T to a low level, for example, 20%. Ignoring all spatial aspects, it is easy to set up a diagram and a dynamic model of the system (Figure 13.11; see also Exercises 13.15–13.17). It shows that the target is synthesized and cleared by the body. The antibody is also cleared, although at a different rate. The complex in this simple example is assumed to be relatively stable and, owing to its size, not cleared by the body within the time horizon of

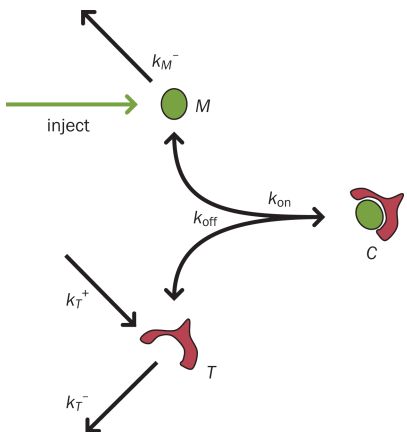


Figure 13.11 Diagram of a monoclonal antibody M binding to a target T. The binding yields a complex C that renders the target dysfunctional. Rate constants are defined in (13.2) and the text.

interest [62]. The default assumption for setting up a model of the system is mass action kinetics, which leads to the following representation:

$$\begin{aligned}\dot{T} &= k_T^+ - k_T^- T - k_{\text{on}} TM + k_{\text{off}} C,\\ \dot{M} &= k_{\text{off}} C - k_{\text{on}} TM - k_M^- M,\\ \dot{C} &= k_{\text{on}} TM - k_{\text{off}} C.\end{aligned} \tag{13.2}$$

Note that the injection of antibody is not modeled explicitly. Instead, at time t_0, the value of M is set equal to a value corresponding to the antibody concentration in the bloodstream. We simulate the system with initial values $(T(t_0), M(t_0), C(t_0)) = (T_0, M_0, C_0)$ = (different values, 100, 0.001) and parameter values $k_T^+ = T_0 k_T^-$, $k_T^- = \ln(2)/10$, $k_M^- = \ln(2)/(21 \cdot 24)$, $k_{on} = 0.036$, $k_{off} = 0.00072$, given for a time-scale of hours, as suggested in [63, 64]. Let us check the rationale for these values. The synthesis rate of T, k_T^+, is set such that the system without any antibodies or complexes is in a steady state. This assumption is often (although not always) true. Setting all terms containing M or C to zero and equating the right-hand side of the first equation in (13.2) to zero results in the given definition of the synthesis rate. The degradation rate of T, k_T^-, corresponds to a half-life of 10 hours. The clearance rate of M corresponds to a half-life of 3 weeks. The rates of complex formation and dissociation are typical for these types of complexes.

It is now easy to run simulations over a typical period of, say, 40 days, with different values for $T(t_0)$. Some results are shown in **Figure 13.12**. They demonstrate that the concentration of the target drops immediately and then recovers after some while. The higher the initial target concentration, the faster T recovers. The figure also shows the disappearance of free antibody, which is due to binding to T as well as to clearance.

The second example is similar in spirit, but it is now assumed that the system consists of a receptor R, a ligand L that naturally binds to the receptor—thereby contributing to the disease process—and a therapeutic antibody A. An important question for the pharmaceutical industry designing a biologic is whether the antibody should be developed to bind to the receptor or to the ligand. In either case, the association of ligand and receptor would be impeded, so does it really matter? A sketch of the situation (**Figure 13.13**) accounts for both scenarios; in reality, binding occurs only to the ligand or only to the receptor. The structural diagram, again accounting for both situations, is given in **Figure 13.14**, and the equations describing the diagram are as follows:

$$\begin{aligned}\dot{R} &= k_R^+ - k_R^- R - k_1^+ RL + k_{1b}^- C_1 - k_2^+ RA + k_{2b}^- C_2,\\ \dot{L} &= k_L^+ - k_L^- L - k_1^+ RL + k_{1b}^- C_1 - k_3^+ LA + k_{3b}^- C_3,\\ \dot{A} &= -k_A^- A - k_2^+ RA + k_{2b}^- C_2 - k_3^+ LA + k_{3b}^- C_3,\\ \dot{C}_1 &= k_1^+ RL - k_{1b}^- C_1,\\ \dot{C}_2 &= k_2^+ RA - k_{2b}^- C_2,\\ \dot{C}_3 &= k_1^+ LA - k_{3b}^- C_3.\end{aligned} \tag{13.3}$$

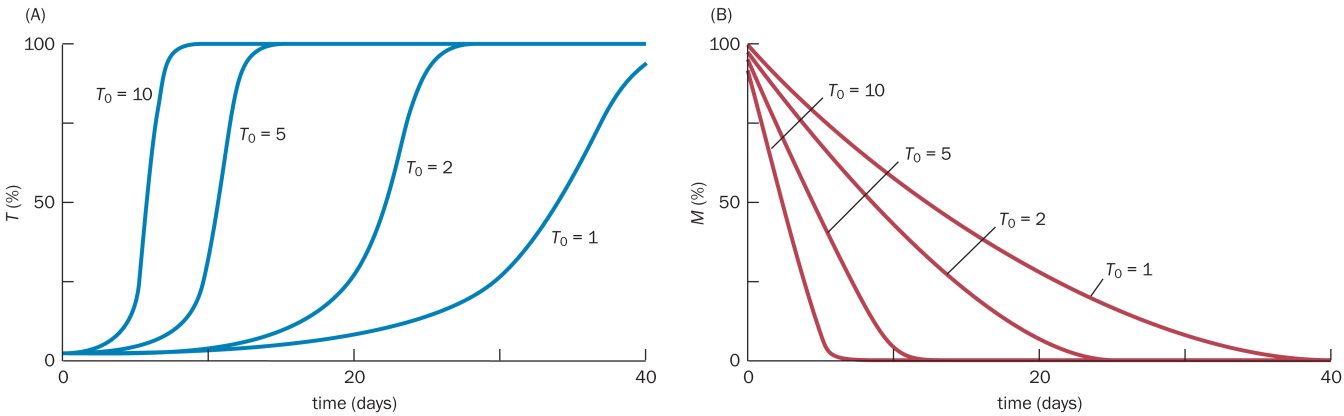


Figure 13.12 Simulation results with the model (13.2) of antibody binding. For a fixed initial monoclonal antibody concentration M_0, the rebounding dynamics of the free target T, as a percentage of the initial target concentration T_0 (A) and the free antibody concentration M, as a percentage of M_0 (B), depend strongly on T_0. Note that time is given in days, rather than hours.

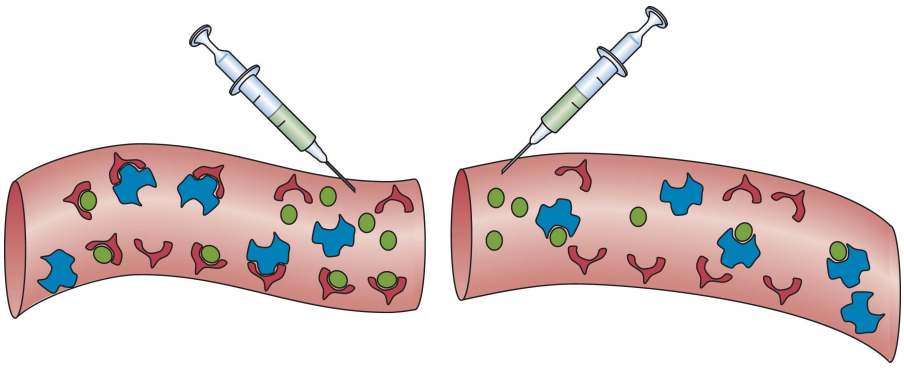


Figure 13.13 Alternative options for antibody treatment. The monoclonal antibody (green circles), which is injected into the bloodstream, binds reversibly either to the receptor (left) or to the ligand (right) associated with the target disease.

For the case of receptor binding by the antibody, the blue boxes are zero, whereas the pink boxes are zero for ligand binding. Parameter values are given in Table 13.2; they are inspired by actual measurements [63, 64]. The initial values R_0, L_0, and A_0 are at first set to 4, 100, and 100, respectively, but are varied in different simulations.

As an illustration, we explore the system for the case where the antibody binds to the ligand. Thus, $R_0 = 4$, $A_0 = 100$, and L_0 is varied for different scenarios. Representative results (Figure 13.15) suggest the following. For low numbers of ligands, the antibody reduces their concentration effectively, and the ligand concentration remains below 30% of the initial value for about 2 days. By contrast, if the initial ligand concentration is at the order of 10, the antibodies reduce it initially only to about 40%, and within 2 days the concentration has reached 30%. For even higher concentrations, for example, $L_0 = 100$, the ligand concentration is almost back to 100% within 2 days. In addition, there is a strong accumulation of ligand–antibody complex, which might cause undesirable side effects of the treatment. These results suggest that designing an antibody to a ligand can be effective if the ligand concentration is relatively low. If the concentration is high, the antibody is no longer efficacious. Exercise 13.18 explores whether an antibody against the receptor is to be preferred in this case.

In addition to changing R_0 or L_0, we can also use the model to explore different doses of antibody. For instance, one might surmise that a higher dose might counteract the decreasing efficacy of the treatment for higher ligand concentrations. To test this hypothesis, we suppose $L_0 = 10$, and quadruple the dose of antibodies

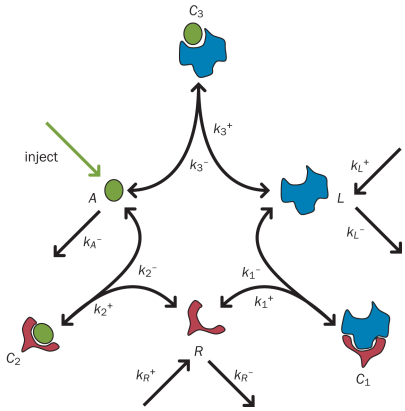


Figure 13.14 Diagram of a monoclonal antibody A binding to a target receptor R or to a target ligand L. Three types of complexes arise: between receptor and ligand (C_1) and either between receptor and antibody (C_2) or between ligand and antibody (C_3). Rate constants are defined in (13.3) and the text.

TABLE 13.2: PARAMETER VALUES FOR THE SYSTEM (13.3)

k_R^+	k_R^-	k_L^+	k_L^-	k_A^+	k_1^+	k_1^-	k_2^+	k_2^-	k_3^+	k_3^-	C_i
$R_0 k_R^-$	0.0693	$L_0\, k_L -$	0.1155	0.0014	4	1	0.036	0.036	0.0018	0.036	0.001

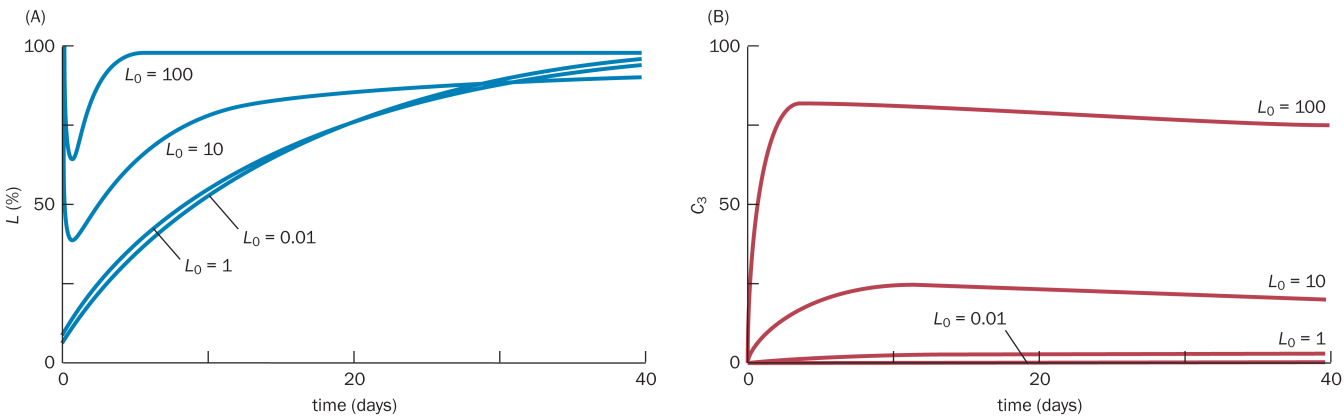


Figure 13.15 Simulation results with a model of different types of antibody binding (13.3). For fixed initial monoclonal antibody and receptor concentrations (A_0 and R_0), the dynamics of free ligand rebounding (L; in percent of the initial the target concentration L_0; left) and the concentration of complex C_3 (right) critically depend on L_0.

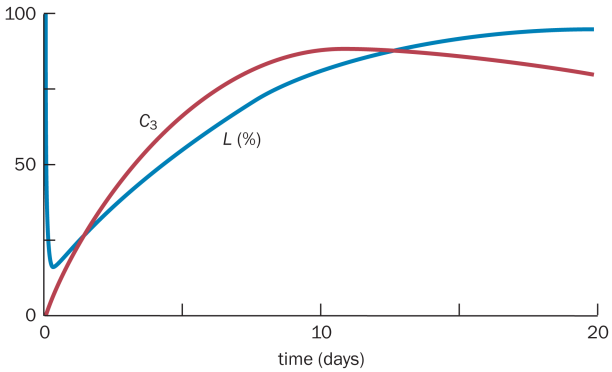


Figure 13.16 Simulation result with the model (13.3) for increased antibody loads. If the initial antibody concentration A_0 is increased to 400, the free ligand rebounds more slowly (compare with the curve $L_0 = 10$ in Figure 13.14), but the concentration of the complex C_3 is much higher.

to 400. In comparison with the lower dose, the ligand concentration is better controlled, but the concentration of the complex is again high (Figure 13.16). An alternative to a higher dose is repeated injection of antibodies. Exercise 13.19 asks you to explore variations on this theme.

13.4.3 Pharmacokinetic Modeling

Computational systems biology has been playing a significant role in drug development for several decades by supporting pharmacokinetic and pharmacodynamic assessments of the levels of therapeutic agents in different tissues and organs of the body (see, for example, [65]). Two classes of approaches use either relatively simple **compartment** models [66] or a particularly powerful expansion of these toward **physiologically based pharmacokinetic (PBPK) models** [12, 67–69].

The base concepts for pharmacokinetic analyses and simulations are relatively simple, at least conceptually. The body is represented as a set of homogeneous compartments, and a prescription drug that was administered orally or through an injection migrates among these compartments. Some compartments may excrete the drug (lungs through inhalation, kidneys through urine, the gastrointestinal (GI) tract through feces), and the liver, in particular, may break it down through enzymatic activity. The key question is how long the drug resides in these different compartments and in what concentrations because this information is important for developing an effective dosing regimen. The drug in question could be a small-molecule drug, a nutrient or metabolite, a recombinant protein or antibody, or even a toxic substance.

To obtain an impression of the principles of this type of approach, let's consider a simple **compartment model** that consists of just two compartments: the bloodstream and the liver (Figure 13.17). In this simplified system, the bloodstream

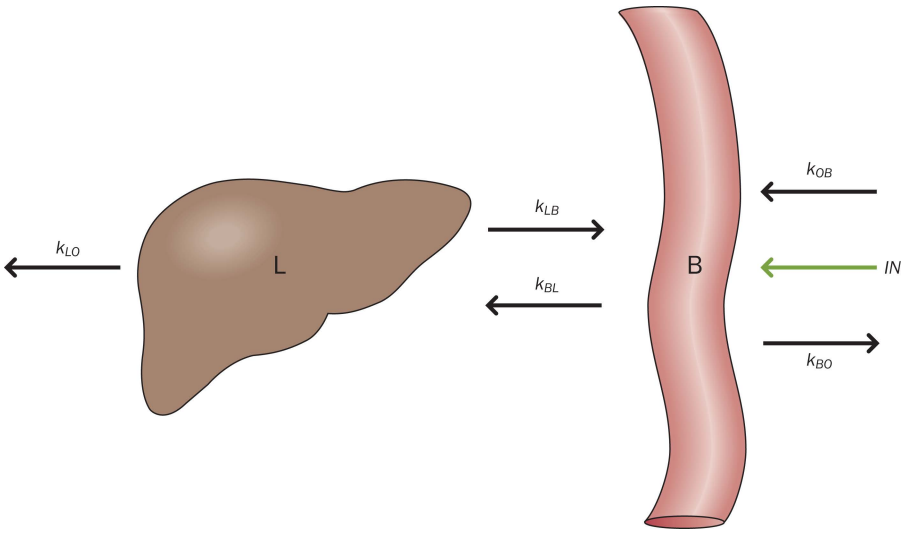


Figure 13.17 A two-compartment model for the exchange of a metabolite between blood (B) and liver (L). Several processes govern this exchange. They can be formulated as a mathematical model such as (13.4) in a straightforward manner (see Chapter 2). The parameters are explained in the text.

receives a drug through injection or from elsewhere in the body. Examples of the latter might be the endogenous production of insulin or the baseline level of a metabolite produced by the patient. The input rate is k_{OB}. Some of the drug moves to other organs (with rate k_{BO}) and some is exchanged with the liver, with rates k_{BL} and k_{LB}, respectively. The liver delivers some of the drug to the intestines or gallbladder with rate k_{LO}.

Calling the concentration of the drug in the liver L and that in the blood B, a linear model with possible additional injection input of "dose" IN reads

$$\begin{aligned} \dot{B} &= k_{OB} + k_{LB}L - (k_{BO} + k_{BL})B + IN, \\ \dot{L} &= k_{BL}B - (k_{LO} + k_{LB})L. \end{aligned} \tag{13.4}$$

Mathematically, k_{OB} and IN could be merged, but we keep them separate to distinguish normal physiology, where B is continually augmented with material from the rest of the body and additional input from an intravenous injection.

Let us suppose for a moment that no metabolite is transported from the bloodstream into the liver ($k_{BL} = 0$). The equation for L then has the form $\dot{L} = -kL$, with rate $k = k_{LO} = k_{LB}$, which immediately signals that L is losing material in an exponential fashion (see Chapters 2 and 4). As a second situation, imagine no influx through k_{OB} but a bolus injection IN at some time point. In other words, after the brief injection of IN, there is no further input and the only active processes are the exchanges between the two compartments and efflux from both. After a short initial phase of equilibration between L and B, the dynamics now consists merely of material shuttling between or leaving the pools, and these losses occur in an exponential fashion, which corresponds to straight lines in a logarithmic plot (**Figure 13.18A**). This solution has the explicit form

$$\begin{aligned} B(t) &= c_{11}e^{\lambda_1 t} + c_{12}e^{\lambda_2 t}, \\ L(t) &= c_{21}e^{\lambda_1 t} + c_{22}e^{\lambda_2 t}, \end{aligned} \tag{13.5}$$

where λ_1 and λ_2 are the eigenvalues of the system matrix $\mathbf{A}$ and $c_{11}, \ldots, c_{22}$ are constants that depend on the numerical specifications of the model (see Chapter 5). Because the eigenvalues might be real or imaginary numbers, the functions in (13.5) could in principle decrease toward zero or oscillate.

The most interesting situation occurs if none of the rate constants are zero and the system is in a steady state but then receives an injection IN. For instance, if $k_{OB} = 1.2$, $k_{LB} = 0.5$, $k_{BO} = 0.9$, $k_{BL} = 0.8$, $k_{LO} = 0.3$, and $IN = 0$, the system happens to

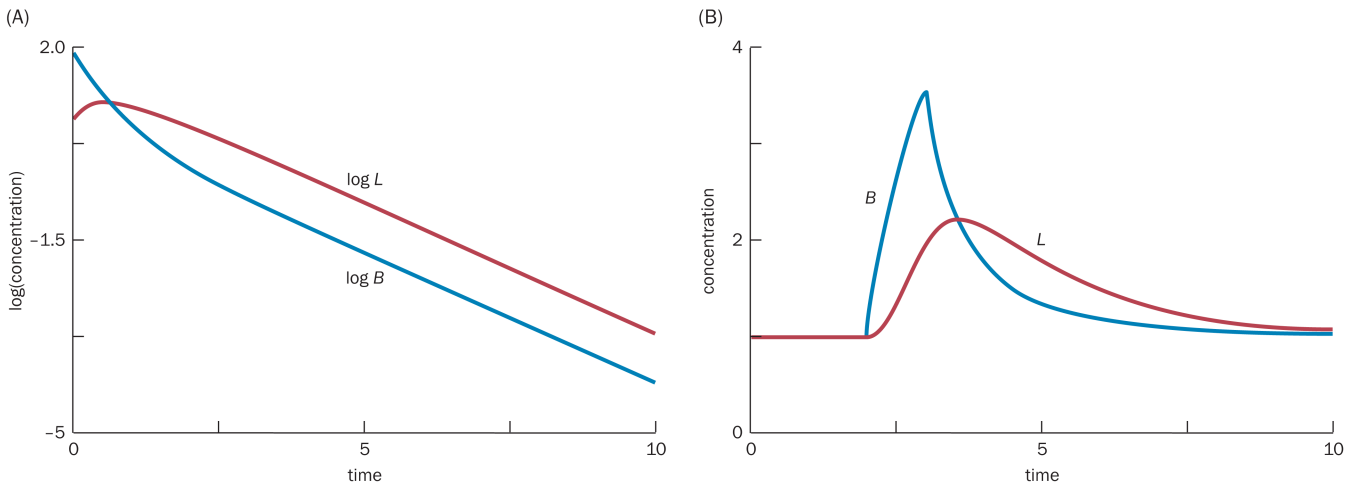


Figure 13.18 Responses of the two-compartment model (13.4) for the exchange of a metabolite between blood (B, with concentration B) and liver (L, with concentration L). (A) After a one-time input, the material equilibrates and afterwards is lost in an exponential fashion (note the logarithmic vertical axis). (B) The system operates at a steady state with constant influx, effluxes, and exchanges between B and L. Between times 3 and 4, B receives an injection with additional material, which moves between compartments and is ultimately cleared from the system.

have the steady state $B = L = 1$ (in some unspecified unit). Suppose that we inject 5 units of *IN* within the time interval $3 = t = 4$. Since *B* is the primary recipient, it increases, but not in a linear fashion, because some of the new material immediately starts moving into *L*, which increases subsequently. After some while, *IN* has been cleared from the system, which returns to its steady state (Figure 13.18B). Many scenarios can easily be tested in this fashion (see Exercises 13.20–13.23).

If we realistically assume that the liver enzymatically degrades the drug, the equation of *L* is augmented with additional degradation terms, maybe in the form of **Michaelis–Menten, Hill, or power-law functions**, which make the equation nonlinear (see Exercise 13.24). The nonlinearities make the model more realistic but preclude some strictly mathematical analyses; however, they are not a hindrance for all kinds of simulations. As so often, the true challenge of this expansion lies in an accurate determination of suitable functions and parameter values.

More sophisticated and realistic than simple compartment models are PBPK models that consist of multiple compartments representing organs or tissues, which are functionally connected through the circulatory system (**Figure 13.19**). The compartments are quantitatively represented with measurable characteristics such as volume and perfusion rate. A drug enters the body by mouth, inhalation, or injection and is transported to and among the various tissues via the bloodstream. Organs such as the liver metabolize some of the drug, or one could consider tumor-targeting as a "sink" tissue that removes circulating drug [70]. The rest cycles through the body and is eventually excreted through feces and urine or, in some cases, by exhalation or sweating, which causes the concentration eventually to decrease to insignificantly low levels. Typical questions are: how long will it take before a drug reaches a target tissue? How long will it remain in this tissue in therapeutically relevant doses? What is the best route and dosing regimen for drug administration? These questions are not easy to answer intuitively, because some tissues are very strongly perfused (for example, the brain and liver), while others are not (for example, fat). Also, many prescription drugs are lipophilic, so that they have a much higher affinity to fat than, for instance, muscle. Furthermore, the drug usually cycles through the system several times, and its residence time in the various organs depends critically on its lipid content and other physicochemical properties. PBPK models can not only be used to study the dynamics of the original drug but also the accumulation of toxic substances, including breakdown products of the drug [68, 69, 71].

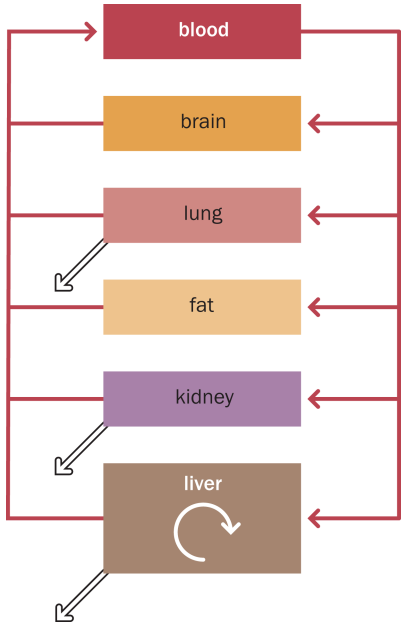


Figure 13.19 Diagram of a generic physiologically based pharmacokinetic (PBPK) model. Blood circulates among organs and tissues, carrying a drug of interest. The liver metabolizes some of the drug, and the lung, kidney, and liver "lose" drug by exhalation or excretion.

Probably the most important feature of PBPK models is that they can be utilized as a mechanistic tool for scaling drug scenarios from animal experiments with mice, rats, or dogs to humans. The same models are set up for the different species but with parameter values that are adjusted, for instance, to the volumes of organs, tested for one or two species, and then used to make predictions about the drug dynamics in humans. A typical case study is presented in Box 13.1. It is clear that the success rate of the clinical drug development phase rises substantially if one can rely on PBPK models that were set up for animals as reliable predictors of efficacy, safety, and toxicity in humans.

One somewhat tricky technical issue with PBPK models is that the typical measurement unit in biochemistry, physiology, and pharmaceutical science is a concentration but that the various organs and tissues have very different volumes so that the same amount of a drug moving, say, from the bloodstream to the lung tissue all of the sudden corresponds to a different concentration. PBPK models deal with this issue by multiplying concentrations with volumes, thereby computing the drug dynamics in absolute amounts. Outside this feature, most PBPK models use standard mass-action kinetics and Michaelis–Menten functions, but they could of course also be formulated with other rate functions, such as power-laws.

As an important example, the dynamics of a drug in the blood plasma is typically written as

$$V_P \dot{P} = IN + \frac{q_L}{r_L} L + \frac{q_K}{r_K} K + \frac{q_M}{r_M} M - (q_L + q_K + q_M) P \qquad (13.6)$$

The left-hand side shows the derivative of the concentration P of the drug in plasma, multiplied by the volume of plasma, V_P and thus of the actual amount of drug. On the right-hand side we have amounts of drug entering (positive signs) and leaving (negative signs) the bloodstream. Beyond the input (IN), typically an intravenous injection, we see terms with two new types of parameters: flow rates between compartments, traditionally denoted with q and a subscript identifying the compartment, and so-called partition coefficients, which are denoted with subscripted r parameters and represent the fact that only portions of a tissue actively take up or release the drug. The quantities with capital letters are concentrations of the drug in liver (L), kidney (K) and muscle (M). Depending on the application, other compartments, such as brain, lung, and fatty tissue could be considered.

The equations for other organs are constructed similarly but are usually simpler. For instance, the drug concentration (or amount) in muscle is formulated as

$$V_M \dot{M} = q_M \left(P - \frac{M}{r_M} \right) \tag{13.7}$$

Similarly, the typical equation for the kidneys is written as

$$V_K \dot{K} = q_K \left(P - \frac{K}{r_K} \right) - k_K \frac{K}{r_K} \tag{13.8}$$

It contains an extra term for the excretion of the drug through urine. The case study in Box 1 provides further details and a few case-specific twists.

In principle, PBPK models may account for arbitrarily many compartments, and there is no real limit to their complexity. Typically, all transport steps are modeled as first-order processes and metabolism within organs is not considered, with the possible exception of the liver, so that standard PBPK models consist of systems of linear differential equations, for which there are many methods of analysis (see Chapter 5). Of course, this is an approximation.

All in all, PBPK models have been very successful in the pharmaceutical industry because they permit predictions of dynamically changing drug distributions among organs and tissues based on animal experiments and on individual human-specific parameters like organ volumes, partition coefficients, and flow rates that can be estimated for humans without exposing them to the new drug.

BOX 13.1: PHARMACOKINETICS OF METHOTRAXATE

Methotrexate (MTX) is a powerful prescription drug that inhibits folate metabolism, which is crucial for the production and repair of DNA and therefore the growth of cell populations. MTX is used to treat various cancers and autoimmune diseases and induces medical abortions. It is on the World Health Organization's List of Essential Medicines, which contains the most important medications needed in a basic health system [72]. Being a very powerful medication, it is not surprising that MTX can have serious, even life-threatening side effects in the liver, kidneys, lungs, and other tissues [73]. Any treatment with MTX therefore requires a well-balanced dosing regimen.

Bischoff and colleagues [74, 75] studied the pharmacokinetics of MTX with a PBPK model that was originally implemented for mice but then also tested in rats and dogs. Model predictions were subsequently compared with plasma concentrations in humans, where they reflected the actual observations quite well. Bischoff's papers are quite old but illustrate the principles of pharmacokinetic modeling very well. The following analysis is directly adapted from this work [75].

The specific model diagram is shown in **Figure 1**. It is fundamentally the same as Figure 13.17, but the authors found it necessary to account for multi-step processes of bile secretion (B_1–B_3) and transport through the gut lumen (G_1–G_4).

The model is set up as explained in the text. In fact, the equations for plasma, muscle, and kidney are identical with Eqns. (13.6)–(13.8). The equation for liver is very similar but additionally contains terms for input from the gastro-intestinal (GI) tract and also for biliary secretion, which is modeled as the Michaelis–Menten process B_0. Thus, it has the form

$$V_L \dot{L} = (q_L - q_G)\left(P - \frac{L}{r_L} \right) + q_G \left(\frac{G}{r_G} - \frac{L}{r_L} \right) - B_0 \text{ , where}$$

$$B_0 = \frac{k_L \left(\frac{L}{r_L} \right)}{K_L + \frac{L}{r_L}} . \tag{1}$$

The movement through the bile duct is modeled with the equations

$$\dot{B}_i = \tau^{-1} \left(B_{i-1} - B_i \right) \text{ for } i = 1,2,3 . \tag{2}$$

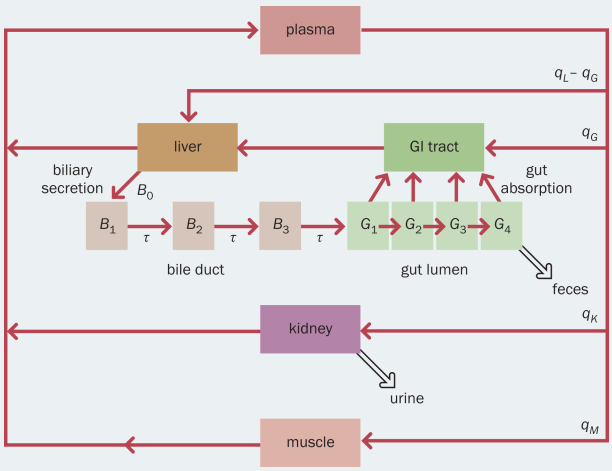


Figure 1 Diagram from the movement of methotrexate among different compartments within the body. Note that secretion from the liver is subdivided into three compartments of the bile duct and that the gut lumen is subdivided into four sections. (Adapted from Bischoff KB, Dedrick RL, Zaharko DS & Longstreth JA. *J. Pharmacol. Sci.* 60 [1971] 1128–1133. With permission from Elsevier.)

Here, τ is the holding time in each section of the bile duct, and B_0 is the Michaelis–Menten term in Eqn. (1). No scaling factor is needed, because only bile duct sections are involved. The dynamics in the gut tissue is represented as

$$V_G \dot{G} = q_G \left(P - \frac{G}{r_G} \right) + \frac{1}{4} \sum_{i=1}^{4} \left(\frac{k_G G_i}{K_G + G_i} + b G_i \right). \qquad (3)$$

The gut tissue receives material from the four compartments $G_1, \ldots, G_4$ of the gut lumen (see Figure 1). These transport processes are represented with a Michaelis–Menten process that accounts for saturation as well as a non-saturable effect. The concentrations of MTX in these compartments are modeled individually and for the entire gut lumen as

$$\frac{V_{GL}}{4} \dot{G}_1 = B_3 - k_f V_{GL} G_1 - \frac{1}{4} \left(\frac{k_G G_1}{K_G + G_1} + b G_1 \right)$$

$$\frac{V_{GL}}{4} \dot{G}_i = k_f V_{GL} \left(G_{i-1} - G_i \right) - \frac{1}{4} \left(\frac{k_G G_i}{K_G + G_i} + b G_i \right) \qquad i = 2,3,4$$

$$\dot{GL} = \frac{1}{4} \sum_{i=1}^{4} \dot{G}_i \qquad (4)$$

Entering the equations into a solver like PLAS or some other software and using the parameter values in Table 1, it is now possible to test the model against actual measurements. As a first example, **Figure 2** shows the results of a simulation where 3 mg MTX per gram body weight were injected intravenously into a mouse. The injection may be modeled with an input (*IN*) to the equation for plasma or, more simply, by setting the initial value of *P* as 66 (3 mg times bodyweight divided by the volume of plasma) and all other initial values close to zero (e.g., 0.001). Note that results of PBPK analyses are traditionally displayed on a logarithmic scale of the y-axis. Similar to the simple two-compartment model (Figure 13.18), one observes that all concentration profiles, after the initial phase, become more or less linear, which corresponds to exponential decay in all tissues. Exercises 13.25 and 13.26 address a simulation of 300 mg MTX per gram body weight and a variant modeling strategy where the injection is represented as a bolus.

As a second example, let's consider an injection of 6mg MTX per gram bodyweight into a rat. The results are shown in **Figure 3**. The overall appearance is similar to the results of the mouse experiment, even though the concentration values are of course different. Exercise 13.27 analyzes the MTX distribution among tissues in humans.

TABLE 1: MODEL PARAMETERS FOR METHOTREXATE CASE STUDY FOR THREE SPECIES (ADAPTED FROM BISCHOFF ET AL. [1971] [75]).

Parameter	*Model Component*	*Mouse*	*Rat*	*Human*
Body weight [g]		22	200	70,000
Volume [ml]				
Plasma	V_P	1	9.0	3,000
Muscle	V_M	10.0	100	35,000
Kidney	V_K	0.34	1.9	280
Liver	V_L	1.3	8.3	1,350
Gut Tissue	V_G	1.5	11.0	2,100
Gut Lumen	V_{GL}	1.5	11.0	2,100
Plasma Flow Rate [ml/min]				
Muscle	q_M	0.5	3.0	420
Kidney	q_K	0.8	5.0	700
Liver	q_L	1.1	6.5	800
Gut Tissue	q_G	0.9	5.3	700
Partition Coefficient				
Muscle	r_M	0.18	0.11	0.15
Kidney	r_K	3.0	3.3	3.0
Liver	r_L	7.0	3.3	3.0
Gut Tissue	r_G	1.2	1.0	1.0
Kidney Clearance [ml/min]	k_K	0.2	1.1	190
Bile Secretion Parameters				
Clearance [ml/min]	k_L/K_L	0.5	3.0	200
Residence Time [min]	τ	2.0	2.0	10
Gut Lumen Parameters				
Transit Time [min]		100	100	1,000
Reciprocal Transit Time [min-1]	k_f	0.01	0.01	0.001
Maximal Velocity	k_G	0.2	20	1,900
Michaelis Constant	K_G	6.0	200	200
Rate Constant, Absorption [ml/min]	b	0.001	0.0	0.0

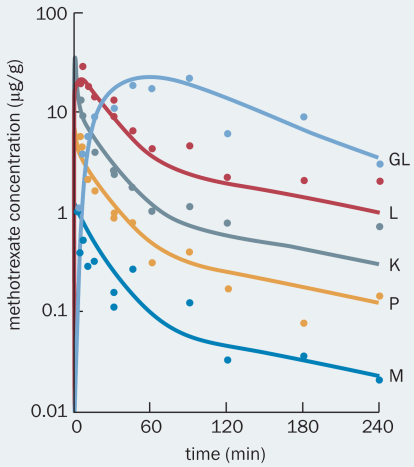


Figure 2 Methotrexate distribution among mouse tissues over time. Comparison of model predictions (lines) and experimental data (dots) for intravenous injection of 3 mg methotrexate per gram of body weight in a mouse. GL, gut lumen; K, kidney; L, liver; M, muscle; P, plasma. (Adapted from Bischoff KB, Dedrick RL, Zaharko DS & Longstreth JA. *J. Pharmacol. Sci.* 60 [1971] 1128–1133. With permission from Elsevier.)

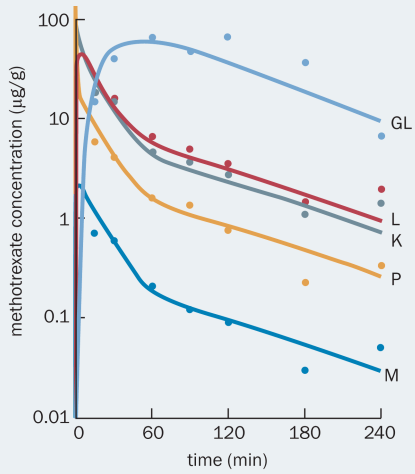


Figure 3 Methotrexate distribution among rat tissues over time. Comparison of model predictions (lines) and experimental data (dots) for intravenous injection of 6 mg methotrexate per gram of body weight in a rat. GL, gut lumen; K, kidney; L, liver; M, muscle; P, plasma. (Adapted from Bischoff KB, Dedrick RL, Zaharko DS & Longstreth JA. *J. Pharmacol. Sci.* 60 [1971] 1128–1133. With permission from Elsevier.)

13.4.4 Pathway Screening with Dynamical Models

Most dynamic pathway models do not account explicitly for different spatial locations such as organs. Instead, they attempt to capture quantitatively how a drug alters the movement of metabolites through a diseased pathway system and hopefully leads to a more desirable steady-state metabolic profile. The specific goal is typically an increase or decrease in a specific target metabolite. In addition to representing the primary pathway metabolites, dynamic models permit analyses of adverse effects within the pathway, such as the accumulation of toxic substances or the depletion of vital metabolites [71]. Sufficiently accurate pathway models are thus very versatile and can play a role in efficacy, safety, toxicity, and dosing assessments [76]. Furthermore, similar to pharmacokinetic models, they have the potential to aid extrapolations from animal to human test results. Sophisticated models in the future will be capable of capturing large portions of human physiology and metabolism in sufficient detail. They will become reliable enough to screen out seemingly promising but ultimately ineffective leads early in the drug development pipeline and thereby save valuable resources. Pathway models for drug discovery are reviewed in [77, 78].

As an illustration of the role of pathway models in target screening, consider a very simplified model of purine metabolism (**Figure 13.20**). This biosynthetic pathway is responsible for generating sufficient amounts of purines, which are components of DNA and RNA and can also be converted into a number of other crucial metabolites. Here, we assume that the cells are not growing so that hardly any new DNA is needed and that the RNA pool is in equilibrium, where as much is produced as degraded.

For our simplistic illustration, we focus on a disorder called hyperuricemia, which is an elevation of the uric acid concentration in the blood and can be the result of various perturbations within the purine pathway. A common disease associated with hyperuricemia is gout. The model corresponding to the pathway in Figure 13.20 was inspired by earlier modeling efforts [79–82], and, although it is much simpler than the original model, it allows us to illustrate some salient points. The mathematical representation in power-law format is as follows:

$$
\begin{aligned}
\dot{P} &= v_{In} - v_{PG} - v_{PI} - v_{PHI} - v_{PH}, \\
\dot{I} &= v_{PI} + v_{PHI} + v_{GI} - v_{IG} - v_{IH}, \\
\dot{G} &= v_{PG} + v_{IG} - v_{GI} - v_{GX}, \\
\dot{H} &= v_{PH} + v_{IH} - v_{PHI} - v_{HX} - v_{Hout}, \\
\dot{X} &= v_{GX} + v_{HX} - v_{XU}, \\
\dot{U} &= v_{XU} - v_{Uout},
\end{aligned}
\tag{13.9}
$$

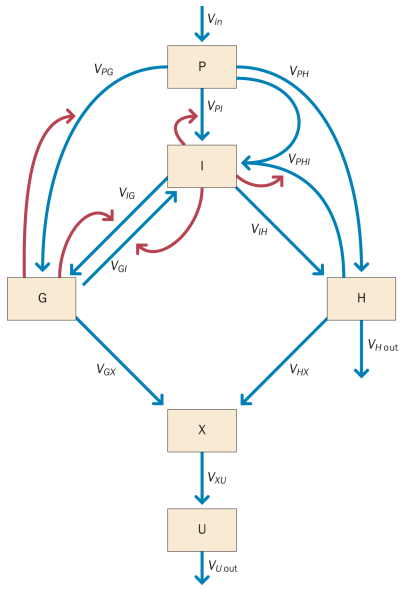


Figure 13.20 Simplified representation of purine metabolism. The diagram and the corresponding model (13.9), (13.10) consist of metabolites (boxes), enzymatic conversions and transport steps (blue arrows), and inhibitory signals (red arrows). P, phosphoribosylpyrophosphate; I, inosine monophosphate; G, guanylates; H, hypoxanthine and inosine; X, xanthine; U, uric acid.

where the fluxes, in self-explanatory notation, take the form

$$\begin{aligned}
v_{\mathrm{In}} &= 5, \\
v_{PG} &= 320P^{1.2}G^{-1.2}, & v_{GX} &= 0.01G^{0.5}, \\
v_{PI} &= 0.5P^{2}I^{-0.6}, & v_{IH} &= 0.1I^{0.8}, \\
v_{PH} &= 0.3P^{0.5}, & v_{HX} &= 0.75H^{0.6}, \\
v_{PHI} &= 1.2PI^{-0.5}H^{0.5}, & v_{Hout} &= 0.004H, \\
v_{IG} &= 2I^{0.2}G^{-0.2}, & v_{XU} &= 1.4X^{0.5}, \\
v_{GI} &= 12G^{0.7}I^{-1.2}, & v_{U\mathrm{out}} &= 0.031U.
\end{aligned} \tag{13.10}$$

With these functions and numerical settings, the steady state is approximately $(P, I, G, H, X, U) = (5, 100, 400, 10, 5, 100)$ (cf. [80]).

We discuss two perturbations within this pathway that lead to hyperuricemia. The first is hyperactivity of the enzyme phosphoribosylpyrophosphate synthase (PRPPS), which catalyzes the input step v_{In}. The second perturbation is due to a severe deficiency in the enzyme hypoxanthine–guanine phosphosribosyltransferase (HGPRT), which in the simplified model affects v_{PG} and v_{PHI}. In both cases, the concentration of uric acid (U) is significantly increased, among other problems. It is easy to simulate these conditions with the model in (13.9) and (13.10).

To represent PRPPS hyperactivity, we simply multiply v_{In} by an appropriate factor, such as 2. In response, the steady-state profile changes to about $(P, I, G, H, X, U) = (13, 260, 1327, 23, 14, 170)$. All metabolites are elevated, including U, which is not surprising, because more material is flowing into the system. For the simulation of HGPRT deficiency, we multiply v_{PG} and v_{PHI} by a number between 0 and 1, where 0 would correspond to total absence of enzyme activity and 1 is the normal case. If the enzyme has only two-thirds of its normal activity, the situation is actually not so bad: the steady-state profile in this case is about $(P, I, G, H, X, U) = (6, 103, 370, 11, 6, 109)$, which is different from the normal profile but not by all that much. The situation deteriorates critically if the activity drops much further, leading to a disease that is known as Lesch–Nyhan syndrome and is accompanied by severely compromised brain function. If we reduce the activity in the model to 1%, the metabolite profile becomes $(P, I, G, H, X, U) = (11, 97, 168, 21, 12, 157)$, where the guanylate levels are low and U is high.

The pertinent question here is whether it is possible to interfere effectively with any of the enzymes in order to compensate the changes in the metabolite profile that are caused by a disorder like PRPPS hyperactivity. The most straightforward strategy would be to inhibit PRPPS, but that is currently not possible. One active ingredient on the market is allopurinol, which is structurally similar to hypoxanthine and inhibits the enzyme xanthine oxidase, which oxidizes both hypoxanthine to xanthine and xanthine into uric acid. Multiplying v_{HX} and v_{XU} by 0.33, reflecting 30% remaining enzyme activity, does indeed control the uric acid level. The resulting profile is $(P, I, G, H, X, U) = (10, 259, 1124, 66, 54, 109)$. However, as is also observed in clinical studies, hypoxanthine and xanthine accumulate, along with the other metabolites in the system.

It is rather evident that an effective remedy should remove disease-associated excess material from the system. Uricase is an enzyme that catalyzes the oxidation of uric acid to 5-hydroxyisourate and has been formulated as a drug for hyperuricemia [83, 84]. To test its efficacy, we multiply $v_{U\mathrm{out}}$ by a factor representing the efficacy of the drug. If the activity of $v_{U\mathrm{out}}$ is increased by 50%, the uric acid concentration indeed comes within 10% of the normal value, and the metabolite profile is $(P, I, G, H, X, U) = (13, 252, 1215, 22, 13, 110)$. Alas, as in the previous scenario, P, I, and G accumulate quite significantly.

The enzyme information system BRENDA [85] indicates that the key initial enzyme of purine biosynthesis, amidophosphoribosyltransferase (EC 2.4.2.14), has a variety of inhibitors. Would it be useful to design a drug based on one of these? We can easily explore a hypothetical inhibitor affecting the corresponding process v_{PI}. For example, reducing the activity to 5% leads to the metabolite profile $(P, I, G, H, X, U) = (22, 235, 1475, 11, 7, 118)$. As in the previous case, the uric acid is more or

less under control. Also as before, P, I, and G are greatly increased. In contrast to the allopurinol treatment, hypxanthine and xanthine are now close to normal.

Are there other options? The diagram of the pathway contains two exit routes: one through U and one through H. Increasing $v_{H\text{out}}$ by 100 does indeed control U, along with X and H, but P, G, and I still accumulate. Could it potentially be effective to reroute excess P so that intermediates do not accumulate? Indeed, if we increase v_{PH} by a factor of 8 in addition to increasing $v_{H\text{out}}$ by 100, essentially all excess material is funneled out of the system and the resulting profile is almost normal, namely, $(P, I, G, H, X, U) = (5, 110, 426, 11, 6, 111)$. The model obviously cannot tell whether such a combined strategy can be developed as a useful pharmaceutical option or if it leads to complications elsewhere in the body. However, it demonstrates that modeling can be a powerful tool for screening hypotheses (see Exercises 13.28 and 13.29).

Dynamical models in pharmaceutical research are not limited to metabolism and transport between organs, the two systems chosen here because of their relative simplicity. Chemical, hormonal, electrical, and mechanical signaling, as well as aspects of higher-level systems physiology, will play increasingly important roles as diseases of the brain and of the cardiovascular, endocrine, and immune systems are becoming sufficiently well understood for mechanical investigations at the molecular level. An example of such efforts is the Physiome Project [86], which has as its central theme the creation of integrated computational models of complete organ systems and, eventually, organisms.

13.4.5 Virtual Clinical Trials

The personalization of dynamic disease models (see Section 13.4) has the potential to become very powerful tools of biomedical and pharmaceutical research that could save enormous costs within the drug development process. The key concepts consist of the following steps [87–91]:

1. Develop, test, and validate the disease model.
2. Identify all parameters that are likely to vary within the general population or a target population.
3. For each parameter identify the expected range and, possibly, the distribution of the value of the parameter throughout this range.
4. Run a Monte-Carlo simulation of maybe 5,000 iterations (see Chapter 2 and 6), thereby creating a population of 5,000 virtual patients. Even for a moderate number of parameters, every person will likely be different. Among these patients, a certain percentage is healthy, and the rest suffers from the disease, possibly with different severity. The simulation approach accounts for the role of influential and less influential parameters as well as compensation among parameters, as discussed in Section 13.4.
5. Apply the candidate drug to all patients by altering the parameters that are affected by the drug.
6. Study how safe and effective the candidate drug is in terms of curing the disease, the number of patients helped, and the number of patients suffering side effects. Investigate steady states as well as the dynamics in response to drug administration.
7. Apply statistics in the same manner as for actual healthy and diseased subpopulations and interpret the results in terms of drug safety and efficacy.
8. Dismiss candidate drugs that are, according to the model analysis, not safe or efficacious.

Considering that clinical trials are expensive, especially if a new drug is to be compared to an existing drug rather than a placebo, virtual clinical trials could save a lot of money and time. A report in 2018, analyzing 59 new therapeutic agents approved by the FDA from 2015 to 2016, found that a single FDA-mandated clinical trial may cost between $2 million and $350 million, with 50% of trials costing between $12 million and $33 million [40]. The highest cost is associated with failure of the drug in clinical trials [39], and if a virtual trial could reliably eliminate a

drug from further consideration early on, much of these costs could be avoided. In the immediate future, virtual clinical trials will not replace actual trials, but mathematical modeling will increasingly inform preclinical applications and drug development [92–94].

13.4.6 Emerging Roles of Systems Biology in Drug Development

The examples shown in this chapter indicate that the role of computational systems biology in drug development has already begun to shift from "potential" to "actively contributing." In many cases, the contributions are still modest, but they do suggest the growing contribution of systems biology to this field. In addition to molecular and QSAR modeling, systems biology should be expected to play an increasingly important role in exploring the following [65]:

- The feasibility of an envisioned disease-disrupting drug treatment
- The choice of a target (ligand, receptor, complex; upstream or downstream)
- The efficacy of a therapeutic molecule
- The balance between safety and efficacy
- The toxicity of NCEs and biologics
- The consequences of properties of a therapeutic molecule (half-life, affinity, attainable concentration, etc.)
- The best therapeutic modality (antibodies, peptide mimics, engineered enzymes, RNA interference [RNAi], etc.)
- The route of administration
- The optimal dosing regimen
- The need to account for diffusion and transport (within tumors, through membranes, across the blood-brain barrier, etc.)
- The effects of clearance
- The extrapolation of test results between animals and from animals to humans

Considering the fact that the young field of systems biology is already involved in many areas of drug discovery, it is fair to expect that systems biology ad systems medicine will play an increasingly important role in the mainstream of pharmaceutical research and development in the future. At the same time, the traditional focus of health care—or rather sick care—will shift more and more from treating to preventing disease, and the application of prescription drugs will become increasingly personalized. Systems biology and systems medicine will provide powerful tools supporting these shifts.

13.5 EXERCISES

13.1 Using the example in (13.1), perform the same type of analysis by decreasing one enzyme activity at a time. Are the results symmetric in a sense that they mirror the results of increasing activities?

13.2 In the example in (13.1), the flux through X_4 is very small. Suppose that this flux represents clearance of an undesired metabolite by the kidneys and that a patient with kidney disease can eliminate X_4 at only 10% capacity or not at all. Implement these two scenarios in the model and report the results.

13.3 Using the example in (13.1), study the effects of a decreased efflux from X_6. Compare the results with those of decreasing the clearance of X_4.

13.4 Using the example in (13.1), determine the most influential combination of two parameters with respect to X_6. Develop a strategy for testing the effects of three or four simultaneous perturbations (between 5% and 50%) in enzyme activities in the model. Look in the modeling chapters for alternatives to exhaustive analyses, which become infeasible owing to combinatorial explosion.

13.5 Define health and disease without expressing one in terms of the other.

13.6 In a model of a metabolic health and disease system, are biomarkers to be represented by parameters, independent or dependent variables, two of the three, or all three? Explain your answer and provide examples where applicable.

13.7 Search the literature for examples of biomarkers that are not genomic or related to enzymes. Discuss how they would enter mathematical models of some health and disease system under investigation.

13.8 Discuss whether false-positive or false-negative interpretations of biomarker readings, such as those corresponding to the pink and green points in Figure 13.7, become more of a problem or less of a problem in a larger system of biomarkers.

13.9 Search the literature for reports on drug treatments whose efficacy can be predicted from genome analyses.

13.10 Search the literature for nongenomic biomarkers that are predictive of disease or the success of some specific medication.

13.11 Discuss the role of cytochrome P450 enzymes as possible biomarkers for disease, disease resistance, or the efficacy of drugs. Formulate your findings as a report.

13.12 Using the simple model (13.1) at the beginning of the chapter, define health and disease states and analyze parameter settings within the disease domain. Is it possible to simulate situations where the analysis of a single biomarker suggests a healthy state, whereas the consideration of several biomarkers suggests disease? Either show examples or prove that such a situation is not possible.

13.13 Again using the simple model (13.1), study the long-term transient effects of a decreased clearance of X_4. For instance, if a concentration of X_4 above 12 or 40 (roughly 3 or 10 times the normal level, respectively) is deleterious, how long does it take before the patient should be considered sick?

13.14 Search the literature for data on the success or attrition of NCEs and biologics in different disease areas (inflammation, cardiovascular, oncology, etc.). Identify at which stages of the drug development process the failure rate is the highest and the lowest.

13.15 Expand the simple antibody–target model (13.2) to include clearance of the complex C. Perform simulations and compare the results with corresponding results of the model without clearance of C. Explore different half-lives for the clearance process.

13.16 Expand the simple antibody–target model (13.2) to allow for multiple doses of antibody injection. Perform simulations with different dosing regimens. Determine an optimal dosing schedule that keeps the target under control, while minimizing the total dose and adhering to a patient-friendly schedule.

13.17 Chabot and Gomes [62] state that the half-life of a collection of different clearance processes can be computed as

$$\frac{1}{t_{1/2;\text{overall}}} = \frac{1}{t_{1/2;\text{proteolysis}}} + \frac{1}{t_{1/2;\text{kidney}}} + \frac{1}{t_{1/2;\text{endocytosis}}} + \frac{1}{t_{1/2;\text{other}}}.$$

Prove this statement mathematically or design an ordinary differential equation model to test it computationally. Compare the collective half-life with the half-lives of the individual clearance processes.

13.18 Use the receptor–ligand–antibody model (13.3) to study the dynamics of ligands if the antibody binds to the receptor. Compare your results with those in the text.

13.19 Use the receptor–ligand–antibody model (13.3) to allow for multiple doses of antibody injection. Perform simulations with different dosing regimens.

13.20 Simulate constant input IN in the compartment model (13.4).

13.21 Simulate three equal daily doses of a drug that enters the compartment system (13.4) as input IN. Can you set the doses such that the drug concentration in blood is constant? Or such that it at least stays within some predefined range?

13.22 What happens if the efflux parameters k_{BO} and k_{LO} in the compartment model (13.4) are zero? Make a prediction before you simulate.

13.23 Develop a compartment model like (13.4) that contains fatty tissue in addition to blood and liver. Set the parameters such that they reflect very low perfusion in fat. Perform representative simulations.

13.24 Develop a compartment model like (13.4) that accounts for degradation of the toxic substance in the liver. Compare the results for degradation processes in the form of a Michaelis–Menten rate law and a power-law function.

13.25 Simulate an injection of 300 mg MTX per gram body weight and compare the results with the data in Table 13.3.

13.26 Using the information from Box 13.1, explore the differences between modeling the drug injection either as a change in the initial value of P or as an input bolus IN, which is set at 66 for one time unit, 33 for two time units, or 22 for three time units.

13.27 Using the information from Box 13.1, simulate the dynamic MTX distribution among tissues in

TABLE 13.3: MEASURED CONCENTRATIONS (MG/G) OF MTX IN RESPONSE TO AN INJECTION OF 300 MG MTX PER GRAM BODY WEIGHT IN MOUSE (FROM [75])

Time After Injection [min]	*20*	*50*	*70*	*80*	*150*	*180*	*200*
Plasma	100	25.6	18.2	10	2.2	1.5	4.3
Gut Lumen	1292	1668	1978	2154	2346	426	256

TABLE 13.4: MEASURED CONCENTRATIONS (MG/G) OF MTX IN PLASMA FOLLOWING AN INJECTION OF 1 MG MTX PER GRAM BODY WEIGHT IN TWO HUMANS (FROM [75]) (THE DATA WERE OBTAINED IN TWO EXPERIMENTS)

Time After Injection [min]	*15*	*45*	*60*	*120*	*150*	*240*	*300*	*360*
Experiment 1	1.10	0.821	0.693	0.453	–	0.284	–	0.211
Experiment 2	1.91	1.31	1.06	–	0.560	–	0.23	–

humans, given an injection of 1 mg per kg body weight. Compare the results with the measurements in Table 13.4. Note that measurements for other tissues are not available.

13.28 Explore other possible treatments of hyperuricemia with the simplified purine model (13.9).

13.29 Using the simplified purine model (13.9), compare the efficacy of combined treatments (at least at two target sites) with inhibition or activation of individual steps. Is it possible that drugs are synergistic in a sense that their combination is stronger than the sum of the two individual treatments?

FURTHER READING

Auffray C, Chen Z & Hood L. Systems medicine: The future of medical genomics and healthcare. *Genome Med.* 1 (2009) 2.

Bonate PL. *Pharmacokinetic–Pharmacodynamic Modeling and Simulation,* 2nd ed. Springer, New York, NY, 2011.

Gonzalez-Angulo AM, Hennessy BT & Mills GB. Future of personalized medicine in oncology: A systems biology approach. *J Clin Oncol.* 28 (2010) 2777–2783.

Kell DB. Systems biology, metabolic modelling and metabolomics in drug discovery and development. *Drug Discov Today.* 11 (2006) 1085–1092.

Kitano H. Computational systems biology. *Nature.* 420 (2002) 206–210.

Kitano H, Oda K, Kimura T, et al. Metabolic syndrome and robustness tradeoffs. *Diabetes.* 53 (Suppl 3) (2004) S6–S15.

Lipsitz LA. Physiological complexity, aging, and the path to frailty. *Sci Aging Knowledge Environ.* 2004 (2004) pe16.

McDonald JF. Integrated cancer systems biology: Current progress and future promise. *Future Oncol.* 7 (2011) 599–601.

Reddy MB, Yang RS, Clewell HJ III & Andersen ME (eds). *Physiologically Based Pharmacokinetic Modeling: Science and Applications*. Wiley, Hoboken, NJ, 2005.

Vodovotz Y, Bartels J & An G, In silico trials and personalized therapy for sepsis and trauma. In *Complex Systems and Computational Biology Approaches to Acute Inflammation: A Framework for Model-based Precision Medicine* (Y Vodovotz & G An, eds), pp. 155–168. Springer Nature Switzerland AG, Cham, Switzerland, 2021.

Voit EO. A systems-theoretical framework for health and disease. Abstract modeling of inflammation and preconditioning. *Math Biosci.* 217 (2008) 11–18.

Voit EO. *The Inner Workings of Life. Vignettes in Systems Biology*, pp. xii + 210. Cambridge University Press, Cambridge, UK, 2016.

Voit EO & Brigham KL. The role of systems biology in predictive health and personalized medicine. *Open Pathol J.* 2 (2008) 68–70.

West GB & Bergman A. Toward a systems biology framework for understanding aging and health span. *J Gerontol A Biol Sci Med Sci*. 64 (2009) 205–208.

Weston AD & Hood L. Systems biology, proteomics, and the future of health care: Toward predictive, preventative, and personalized medicine. *J Proteome Res.* 3 (2004) 179–196.

REFERENCES

1. http://www.ornl.gov/sci/techresources/Human_Genome/faq/snps.shtml.
2. Weston, A.D. and L. Hood, Systems biology, proteomics, and the future of health care: toward predictive, preventative, and personalized medicine. *J Proteome Res*, 2004. **3**(2): p. 179–196.
3. Kitano, H., Computational systems biology. *Nature*, 2002. **420**(6912): p. 206–210.
4. Wolkenhauer, O. (editor-in-chief), *Systems Medicine: Integrative, Qualitative and Computational Approaches*. 2021, Amsterdam: Elsevier/Academic Press.
5. Koch, I. and F. Schreiber, *Modeling in Systems Biology-the Petri Net Approach*. 2011, New York, NY: Springer Verlag.
6. Brier, M.R., J.B. Thomas, A.M. Fagan, J. Hassenstab, D.M. Holtzman, T.L. Benzinger, J.C. Morris, and B.M. Ances, Functional connectivity and graph theory in preclinical Alzheimer's disease. *Neurobiol Aging*, 2014. **35**(4): p. 757–768.
7. Vecchio, F., F. Miraglia, C. Marra, D. Quaranta, M.G. Vita, P. Bramanti, and P.M. Rossini, Human brain networks in cognitive decline: A graph theoretical analysis of cortical connectivity from EEG data. *J Alzheimers Dis*, 2014. **41**(1): p. 113–127.
8. Voit, E.O. and K.L. Brigham, The role of systems biology in predictive health and personalized medicine. *The Open Path J*, 2008. **2**: p. 68–70.
9. Deisboeck, T.S., Personalizing medicine: A systems biology perspective. *Mol Syst Biol*, 2009. **5**: p. 249.
10. Nelson, B., Solving the equation for personalized cancer care. *Cancer Cytopathol*, 2017. **125**(1): p. 7–8.
11. Davis, J.D., C.M. Kumbale, Q. Zhang, and E.O. Voit, Dynamical systems approaches to personalized medicine. *Curr Opin Biotechnol*, 2019. **58**: p. 168–174.
12. Reddy, M.B., R.S. Yang, and M.E. Andersen, *Physiologically Based Pharmacokinetic Modeling: Science and Applications*. 2005, New York, NY: John Wiley and Sons, p. 420.
13. Ishii, N., K. Nakahigashi, T. Baba, M. Robert, T. Soga, A. Kanai, T. Hirasawa, M. Naba, K. Hirai, A. Hoque, P.Y. Ho, Y. Kakazu, K. Sugawara, S. Igarashi, S. Harada, T. Masuda, N. Sugiyama, T.

Togashi, M. Hasegawa, Y. Takai, K. Yugi, K. Arakawa, N. Iwata, Y. Toya, Y. Nakayama, T. Nishioka, K. Shimizu, H. Mori, and M. Tomita, Multiple high-throughput analyses monitor the response of *E. coli* to perturbations. *Science*, 2007. **316**(5824): p. 593–597.
14. Liu, J., Z. Fan, W. Zhao, and X. Zhou, Machine intelligence in single-cell data analysis: Advances and new challenges. *Front Genet*, 2021. **12**: p. 655536.
15. Ma, Q. and D. Xu, Deep learning shapes single-cell data analysis. *Nat Rev Mol Cell Biol*, 2022. **23**(5): p. 303–304.
16. Chen, R. and M. Snyder, Systems biology: Personalized medicine for the future? *Current Opinion in Pharmacology*, 2012. **12**(5): p. 623–628.
17. Nardini, C., J. Dent, and P. Tieri, Editorial: Multi-omic data integration. *Front Cell Dev Biol*, 2015. **3**: p. 46.
18. Colijn, C., N. Jones, I.G. Johnston, S. Yaliraki, and M. Barahona, Toward precision healthcare: Context and mathematical challenges. *Front Physiol*, 2017. **8**: p. 136.
19. Kumbale, C.M., J.D. Davis, and E.O. Voit, Models for personalized medicine. *Syst Med*, 2021. **3**: p. 306–317.
20. Miller, G.W., *The Exposome: A Primer*. 2014, Oxford, UK: Associated Press.
21. http://predictivehealth.emory.edu/.
22. Sterling, P., Principles of allostasis: Optimal design, predictive regulation, pathophysiology and rational therapeutics, in *Allostasis, Homeostasis, and the Costs of Adaptation*, J. Schulkin, Editor. 2004, Cambridge, UK: Cambridge University Press, p. 17–64.
23. Scheffer, M., S.R. Carpenter, T.M. Lenton, J. Bascompte, W. Brock, V. Dakos, J. van de Koppel, I.A. van de Leemput, A.A. Levin, E.H. van Nes, M. Pascul, and J. Vandermeer, Anticipating critical transitions. *Science*, 2012. **338**: p. 344–348.
24. Janga, S.C. and M.M.R. Edupuganti, Systems and network-based approaches for personalized medicine. *Curr Synth Syst Biol*, 2014. **2**(3).
25. Bhattacharya, S., R.B. Conolly, N.E. Kaminski, R.S. Thomas, M.E. Andersen, and Q. Zhang, A bistable switch underlying B-cell differentiation and its disruption by the environmental contaminant 2,3,7,8-tetrachlorodibenzo-p-dioxin. *Toxicol Sci*, 2010. **115**(1): p. 51–65.
26. Zhang, Q., S. Bhattacharya, D.E. Kline, R.B. Crawford, R.B. Conolly, R.S. Thomas, N.E. Kaminski, and M.E. Andersen, Stochastic modeling of B lymphocyte terminal differentiation and its suppression by dioxin. *BMC Syst Biol*, 2010. **4**: p. 40.
27. Kitano, H., et al., Metabolic syndrome and robustness tradeoffs. *Diabetes*, 2004. **53**: p. S6–S15.
28. Lipsitz, L.A., Physiological complexity, aging, and the path to frailty. *Sci Aging Knowledge Environ*, 2004: p. pe16.
29. van't Veer, L.J. and R. Bernards, Enabling personalized cancer medicine through analysis of gene-expression patterns. *Nature*, 2008. **452**(7187): p. 564–570.
30. Holleman, A., M.H. Cheok, M.L. den Boer, W. Yang, A.J. Veerman, K.M. Kazemier, D. Pei, C. Cheng, C.H. Pui, M.V. Relling, G.E. Janka-Schaub, R. Pieters, and W.E. Evans, Gene-expression patterns in drug-resistant acute lymphoblastic leukemia cells and response to treatment. *N Engl J Med*, 2004. **351**(6): p. 533–542.
31. Foubister, V., Genes predict childhood leukemia outcome. *Drug Discov Today*, 2005. **10**(12): p. 812.
32. Babaian, R.J., H.A. Fritsche, Z. Zhang, K.H. Zhang, K.R. Madyastha, and S.D. Barnhill, Evaluation of prostAsure index in the detection of prostate cancer: a preliminary report. *Urology*, 1998. **51**(1): p. 132–136.
33. Kumbale, C.M. and E.O. Voit, Toward personalized medicine for HIV/AIDS. *J AIDS HIV Treat*, 2021. **3**(2): p. 37–41.
34. Darmoni, S.J., P. Massari, J.-M. Droy, N. Mahe, T. Blanc, E. Moirot, and J. Leroy, SETH: An expert system for the management on acute drug poisoning in adults. *Comput Methods Programs Biomed*, 1994. **43**(3–4): p. 171–176.
35. Luciani, D., S. Cavuto, L. Antiga, M. Miniati, S. Monti, M. Pistolesi, and G. Bertolini, Bayes pulmonary embolism assisted diagnosis: A new expert system for clinical use. *Emerg Med J*, 2007. **24**(3): p. 157–164.
36. Sundareswaran, K.S., D. de Zelicourt, S. Sharma, K.R. Kanter, T.L. Spray, J. Rossignac, F. Sotiropoulos, M.A. Fogel, and A.P. Yoganathan, Correction of pulmonary arteriovenous malformation using image-based surgical planning. *JACC Cardiovasc Imaging*, 2009. **2**(8): p. 1024–1030.
37. Kola, I. and J. Landis, Can the pharmaceutical industry reduce attrition rates? *Nat Rev Drug Discov*, 2004. **3**: p. 711–715.
38. www.cia.gov/library/publications/the-world-factbook/fields/2102.html.
39. Herper, M., The truly staggering cost of inventing new drugs. *Forbes*, February 10, 2012.
40. Moore, T.J., H. Zhang, G. Anderson, and G.C. Alexander, Estimated costs of pivotal trials for novel therapeutic agents approved by the US Food and Drug Administration, 2015–2016. *JAMA Intern Med*, 2018. **178**(11): p. 1451–1457.
41. www.netsci.org/Courseware/Drugs/Intro/slide06.html.
42. pharmalicensing.com/features/images/Razvif3.gif.
43. Goodnow, R.A.J., Hit and lead identification: Integrated technology-based approaches. *Drug Discov Today Technol*, 2006. **3**(4): p. 367–375.
44. Andrade, C.H., K.F. Pasqualoto, E.I. Ferreira, and A.J. Hopfinger, 4D-QSAR: Perspectives in drug design. *Molecules*, 2010. **15**: p. 3281–3294.
45. Funatsu, K., T. Miyao, and M. Arakawa, Systematic generation of chemical structures for rational drug design based on QSAR models. *Curr Comput Aided Drug Des*, 2011. **7**: p. 1–9.
46. Zuckermann, R.N. and T. Kodadek, Peptoids as potential therapeutics. *Curr Opin Mol Ther*, 2009. **11**(3): p. 299–307.
47. Wishart, D.S., Improving early drug discovery through ADME modelling: An overview. *Drugs RD*, 2007. **8**: p. 349–362.
48. DiMasi, J.A., R.W. Hansen, and H.G. Grabowski, The price of innovation: New estimates of drug development costs. *J Health Econ,* 2003. **835**: p. 1–35.
49. Hood, L., Systems biology: Integrating technology, biology, and computation. *Mech Ageing Dev*, 2003. **124**(1): p. 9–16.
50. Voit, E.O., Metabolic modeling: A tool of drug discovery in the post-genomic era. *Drug Discovery Today*, 2002. **7**(11): p. 621–628.
51. Modernization_Act, *CONGRESS.GOV. H.R.2617—Consolidated Appropriations Act, 2023. Became Public Law No: 117–328*, December 29, 2022, https://www.congress.gov/bill/117th-congress/house-bill/2617/text, accessed 2023.
52. Selventa, https://www.flagshippioneering.com/companies/selventa, accessed 2023.
53. Aitia, https://www.aitiabio.com/, accessed 2023.
54. Huang, P.H., A. Mukasa, R. Bonavia, R.A. Flynn, Z.E. Brewer, W.K. Cavenee, F.B. Furnari, and F.M. White, Quantitative analysis of EGFRvIII cellular signaling networks reveals a combinatorial therapeutic strategy for glioblastoma. *Proc Natl Acad Sci USA*, 2007. **104**(31): p. 12867–12872.
55. Sachs, K., O. Perez, S. Pe'er, D.A. Lauffenburger, and G.P. Nolan, Causal protein-signaling networks derived from multiparameter single-cell data. *Science*, 2005. **308**: p. 523–529.
56. Nimjee, S.M., C.P. Rusconi, and B.A. Sullenger, Aptamers: An emerging class of therapeutics. *Ann Rev Med*, 2005. **56**: p. 555–583.
57. Arakaki, A.K., R. Mezencev, N. Bowen, J. Huang, J. McDonald, and J. Skolnick, Identification of metabolites with anticancer properties by computational metabolomics. *Molec Cancer*, 2008. **7**: p. 57.
58. Adelusi, T.I., A.-Q.K. Oyedele, I.D. Boyenle, A.T. Ogunlana, R.O. Adeyemi, C.D. Ukachi, M.O. Idris, O.T. Olaoba, I.O. Adedotun, O.E. Kolawole, Y. Xiaoxing, and M. Abdul-Hammed, Molecular modeling in drug discovery. *Inform Med Unlocked*, 2022. **29**.

59. Aminpour, M., C. Montemagno, and J.A. Tuszynski, An Overview of molecular modeling for drug discovery with specific illustrative examples of applications. *Molecules*, 2019. **24**(9).
60. Salo-Ahen, O.M.H., I. Alanko, R. Bhadane, A.M.J.J. Bonvin, R.V. Honorato, S. Hossain, A.H. Juffer, A. Kabedev, M. Lahtela-Kakkonen, A.S. Larsen, E. Lescrinier, P. Marimuthu, M.U. Mirza, G. Mustafa, A. Nunes-Alves, T. Pantsar, A. Saadabadi, K. Singaravelu, and M. Vanmeert, Molecular dynamics simulations in drug discovery and pharmaceutical development. *Processes*, 2021. **9**(1): p. 71.
61. Rosa, https://rosaandco.com/, accessed 2023.
62. Chabot, J.R. and B. Gomes, Modeling efficacy and safety of engineered biologics, in *Drug Efficacy, Safety, and Biologics Discovery: Emerging Technologies and Tools*, J. Xu and S. Ekins, Editors. 2009, Hoboken, NJ: Wiley Interscience.
63. Mayawala, K. and B. Gomes, Prediction of therapeutic index of antibody-based therapeutics: Mathematical modeling approaches, in *Predictive Toxicology in Drug Safety*, J. Xu and L. Urban, Editors. 2010, Cambridge, UK: Cambridge University Press.
64. Gomes, B., Industrial systems biology. *Presentation at Foundations of Systems Biology in Engineering, FOSBE'09*, Denver, CO, August 2009.
65. Bonate, P.L., *Pharmacokinetic-Pharmacodynamic Modeling and Simulation*. 2006, New York, NY: Springer.
66. Jacquez, J.A., *Compartmental Analysis in Biology and Medicine*. 3rd ed. 1996, Ann Arbor, MI: BioMedware.
67. Lipscomb, J.C., S. Haddad, T. Poet, and K. Krishnan, Physiologically-based pharmacokinetic (PBPK) models in toxicity testing and risk assessment, in *New Technologies for Toxicity Testing*, M. Balls, R.D. Combes, and N. Bhogal, Editors. 2012, New York: Landes Bioscience and Springer Science+Business Media.
68. Zhuang, X. and C. Lu, PBPK modeling and simulation in drug research and development. *Acta Pharmaceutica Sinica B*, 2016. **6**(5): p. 430–440.
69. Kuepfer, L., C. Niederalt, T. Wendl, J.F. Schlender, S. Willmann, J. Lippert, M. Block, T. Eissing, and D. Teutonico, Applied concepts in PBPK modeling: How to build a PBPK/PD model. *CPT Pharmacometrics Syst Pharmacol*, 2016. **5**(10): p. 516–531.
70. Stroh, M., J. Sagert, J.M. Burke, J.F. Apgar, L. Lin, B.L. Millard, and W.M. Kavanaugh, Quantitative systems pharmacology model of a masked, tumor-activated antibody. *CPT: Pharmacometrics Syst Pharmacol*, 2019. **8**(9): p. 676–684.
71. Perry, C., G. Davis, T.M. Conner, and T. Zhang, Utilization of physiologically based pharmacokinetic modeling in clinical pharmacology and therapeutics: An overview. *Curr Pharmacol Rep*, 2020. **6**(3): p. 71–84.
72. World Health Organization (WHO), *WHO Model List of Essential Medicines*, http://www.who.int/medicines/publications/essentialmedicines/en/.
73. Mayo, *Side Effects of Methotraxate*, https://www.mayoclinic.org/drugs-supplements/methotrexate-oral-route/side-effects/drg-20084837, accessed 2023.
74. Bischoff, K.B., R.L. Dedrick, and D.S. Zaharko, Preliminary model for methotrexate pharmacokinetics. *J Pharmac Sci*, 1970. **59**(2): p. 149–154.
75. Bischoff, K.B., R.L. Dedrick, D.S. Zaharko, and J.A. Longstreth, Methotrexate pharmacokinetics. *Pharmac Sci*, 1971. **60**(8): p. 1128–1133.
76. Kell, D.B., Systems biology, metabolic modelling and metabolomics in drug discovery and development. *Drug Discov Today*, 2006. **11**(23/24): p. 1085–1092.
77. Yuryev, A., Pathway analysis for drug discovery: Computational infrastructure and applications, in *Technologies for the Pharmaceutical Industry*, S. Elkins, Editor. 2008, Hoboken, NJ: John Wiley and Sons, p. 304.
78. Ballesta, A., Q. Zhou, X. Zhang, H. Lv, and J.M. Gallo, Multiscale design of cell-type–specific pharmacokinetic/pharmacodynamic models for personalized medicine: Application to temozolomide in brain tumors. *CPT Pharmacometrics Syst Pharmac*, 2014. **3**(4): p. 1–11.
79. Curto, R., E.O. Voit, and M. Cascante, Analysis of abnormalities in purine metabolism leading to gout and to neurological dysfunctions in man. *Biochem J*, 1998. **329** (Pt 3): p. 477–487.
80. Curto, R., E.O. Voit, A. Sorribas, and M. Cascante, Validation and steady-state analysis of a power-law model of purine metabolism in man. *Biochem J*, 1997. **324** (Pt 3): p. 761–775.
81. Curto, R., E.O. Voit, A. Sorribas, and M. Cascante, Mathematical models of purine metabolism in man. *Math Biosci*, 1998. **151**(1): p. 1–49.
82. Voit, E.O., *Computational Analysis of Biochemical Systems: A Practical Guide for Biochemists and Molecular Biologists*. 2000, Cambridge, UK: Cambridge University Press. xii + 530.
83. Yang, X., Y. Yuan, C.G. Zhan, and F. Liao, Uricases as therapeutic agents to treat refractory gout: Current states and future directions. *Drug Dev Res*, 2012. **73**(2): p. 66–72.
84. Pierzynowska, K., A. Deshpande, N. Mosiichuk, R. Terkeltaub, P. Szczurek, E. Salido, S. Pierzynowski, and D. Grujic, Oral treatment with an engineered uricase, ALLN-346, reduces hyperuricemia, and uricosuria in urate oxidase-deficient mice. *Front Med (Lausanne)*, 2020. **7**: p. 569215.
85. www.brenda-enzymes.org.
86. http://www.physiome.org.nz/.
87. Clermont, G., J. Bartels, R. Kumar, G. Constantine, Y. Vodovotz, and C. Chow, In silico design of clinical trials: A method coming of age. *Crit Care Med*, 2004. **32**(10): p. 2061–2070.
88. Craig, M., J.L. Gevertz, I. Kareva, and K.P. Wilkie, A practical guide for the generation of model-based virtual clinical trials. *Front Syst Biol*, 2023. **3**.
89. Pappalardo, F., G. Russo, F.M. Tshinanu, and M. Viceconti, In silico clinical trials: Concepts and early adoptions. *Brief Bioinform*, 2019. **20**(5): p. 1699–1708.
90. Stalidzans, E., M. Zanin, P. Tieri, F. Castiglione, A. Polster, S. Scheiner, J. Pahle, B. Stres, M. List, J. Baumbach, M. Lautizi, K.V. Steen, and H.H.H.W. Schmidt, Mechanistic modeling and multiscale applications for precision medicine: Theory and practice. *Network Syst Med*, 2020. **3**(1): p. 36–56.
91. Viceconti, M., A. Kenney, and E. Morley-Fletcher, *In silico* clinical trials: How computer simulation will transform the biomedical industry. *Int J Clin Trials*, 2016. **3**(2): p. 37–46.
92. Barbolosi, D., J. Ciccolini, B. Lacarelle, F. Barlesi, and N. Andre, Computational oncology—mathematical modelling of drug regimens for precision medicine. *Nat Rev Clin Oncol*, 2016. **13**(4): p. 242–254.
93. Hamis, S., G.G. Powathil, and M.A.J. Chaplain, Blackboard to bedside: A mathematical modeling bottom-up approach toward personalized cancer treatments. *JCO Clin Cancer Inform*, 2019. **3**: p. 1–11.
94. Kovatchev, B.P., M. Breton, C.D. Man, and C. Cobelli, *In silico* preclinical trials: A proof of concept in closed-loop control of type 1 diabetes. *J Diabetes Sci Technol*, 2009. **3**(1): p. 44–55.

Population Systems

14

When you have read this chapter, you should be able to:

- Identify and discuss standard approaches to studying populations
- Distinguish models for uniform and stratified populations
- Design models for age-structured populations
- Develop and analyze models for competing populations
- Understand how to incorporate population growth into larger systems models
- Recognize the enormous importance of microbiomes for health, disease, and sustainable environments
- Realize the challenges to understanding the dynamics of mixed communities

Trends in the sizes of populations have fascinated humans for a long time. Hunters and gatherers were not only interested in the size of their own population but had a strongly vested interest in the populations of animals and plants in their surroundings, upon which their existence depended [1]. Kings, emperors, and other politicians throughout the ages found population sizes important for forecasting food needs and for levying taxes. Apparently, the art of quantifying population sizes can be traced back at least to the Paleolithic period of 30,000 years ago, when early humans in Central Europe used tally sticks to keep track of changes in populations; **exponential growth** was recorded as early as 4,000 years ago by the Babylonians [1–3]. Throughout recorded history, population growth has been measured, documented, predicted, and analyzed.

This chapter takes a very broad view of populations. While we might immediately think of the human world population or the population growth in our hometown or country, populations may also consist of free-living bacteria, viruses, or tumor cells. One might even include unconventional populations, such as the alleles of a gene within a human or animal population [4] or a population of molecules that is converted into a population of different molecules by the action of an enzyme.

14.1 POPULATION GROWTH

The earliest rigorous mathematical descriptions of population sizes and their trends are often attributed to the British clergyman and economist Thomas Robert Malthus [5], who formulated the law of exponential growth and to the Belgian mathematician Pierre-François Verhulst [6], who proposed the sigmoidal **logistic growth** function that we often encounter in microbial populations (see Chapter 5),

DOI: 10.1201/9781003402770-14

which starts out exponentially and then approaches and saturates at some constant called the **carrying capacity**. An enormous number of other growth functions were subsequently proposed for diverse populations, ranging from humans, animals, and plants to bacteria, cells, and viruses, and the literature on growth functions is huge. Most of these are relatively simple nonlinear functions, some discrete, some continuous, while others can be represented as the solutions of differential equations [1].

One might find it odd that population sizes, which are clearly discrete integers, are often modeled with differential equations, which could predict something like 3,20.214 individuals. The simple rationale for this strategy is that differential equations are often easier to set up and analyze than the corresponding discrete models, and we can always round those non-integer numbers. The situation is in reality even more complicated, because processes like growth, which evolve over a long period of time, are almost always affected by **stochastic** events. Yet, differential equations that capture the average growth behavior are often more useful than fully stochastic models, because they are so much easier to analyze and understand. Furthermore, one might rationalize the non-integer outcomes of these models as representing population density values rather than absolute quantities, such that 320.214 represents individuals per acre or square mile. This interpretation is akin to considering biochemical reactions in terms of concentrations (e.g., millimolar) instead of moles.

Previous chapters have made it clear that biological systems, even at the subcellular level, are exceedingly complex. Since higher organisms consist of very many cells, one should therefore expect that the growth of organisms—and of populations of organisms—is several orders of magnitude more complex. In some sense, this would be true if we wanted to keep track of every cell—or even molecule—in a population of organisms. However, while molecular and intracellular details are of course crucial, the study of populations focuses on a different level, where cells, individuals, or organisms are the units of choice and processes occurring within these units are not considered. As an analogy, recall the simple growth curves for humans in Figure 2.21.

One rationale supporting this simplification is the fact that intracellular processes are much, much faster than the **growth dynamics** of individuals and organisms. In fact, they are so fast that they are essentially always in a **steady state** so that their fast dynamics is not of importance for the much slower dynamics of the growth and aging of individuals and organisms. Similarly, processes at a geological scale, such as plate tectonics or even climate change, are so slow in comparison to an organism's lifespan that they can usually be considered constant. As a result, there are often only a few processes that are occurring on just the right timescale to affect growth in a substantial manner [1]. Furthermore, population studies by their nature consider large numbers of individuals, which leads to an averaging of individual variability. Thus, in simplified growth models, all individuals or members of the population are considered essentially the same, and one studies their collective behavior.

The logistic growth function, which we discussed already in Chapter 5, is a typical example of these approximation and averaging effects. By describing millions of cells, aspects of individuality and diversity, which can actually be quite significant even in bacteria, are averaged away, and the populations follow simple, smooth trends (as, for example, in **Figure 14.1**) that are characterized by just a very few population **parameters** such as the **growth rate** and the **carrying capacity** (see later discussion). This approach might seem overly crude, but it is very powerful. Consider, as an analogy, an ideal gas: while it is impossible to keep track of every gas molecule and its interactions with other molecules or the wall of the container, physicists have developed surprisingly simple gas laws that relate volume, pressure, and temperature to each other in a very accurate manner.

14.1.1 Traditional Models of Population Growth

The exponential and logistic growth functions are not the only growth functions in use. In fact, very many **growth laws** have been proposed over the past 200 years,

either as explicit functions or as ordinary differential equations (ODEs). One example is the Richards function [7]

$$N_R(t) = K\{1 + Q\exp[-\alpha\nu(t - t_0)]\}^{-1/\nu}, \tag{14.1}$$

which is very flexible in shape and has been used widely in plant science, agriculture, and forestry. Clearly, N_R is a function of time, t. $N_0 = N_R(t_0) = N_R(0)$ is the population size at some time point declared as 0, K is the carrying capacity, α and ν are positive shape parameters, and $Q = (K/N_0)^{\nu - 1}$. The various parameters permit shifting and stretching the sigmoid curve in different directions.

Another growth function, which is often used in actuarial sciences, is the Gompertz growth law [8]:

$$N_G(t) = K\exp[-b\exp(-rt)], \tag{14.2}$$

where K and r are again the carrying capacity and the growth rate, respectively, and b is an additional positive shape parameter.

It has turned out that it is beneficial for many purposes to represent growth functions as ODEs. For instance, the logistic growth function $N_{(t)} = 0.2524/(e^{-2.128t} + 0.005125)$ (see Figure 14.1 and Chapter 5) may be written as

$$\dot{N} = rN - \frac{r}{K}N^2. \tag{14.3}$$

For the specific example in Figure 14.1, the corresponding parameter values are $r = 2.128$, $K = 49.25$, and $N_0 = 0.2511$.

Mathematically, this conversion is sometimes not too difficult (Exercise 14.2), and from a modeling point of view, the ODE format facilitates the incorporation of the growth process into more complicated dynamic systems [9]. Specifically, the ODE formulation treats growth like any other process in a system, and (14.3) could be interpreted as a small system that describes the balance between one augmenting term, rN and one diminishing term, $(r/K)\,N^2$. One advantage of this formulation is that the growth process may be modulated by other components of the system. For instance, suppose that the bacteria are growing in a human lung during pneumonia. The body's immune system kills bacteria, and a pneumonia model would include this killing in the negative degradation term of (14.3). By contrast, drug treatment with penicillin does not kill bacteria but prevents them from proliferating. Thus, this effect would be incorporated into the positive term of the population model (14.3) as an inhibitor. Another example with wide applicability involves ODEs that describe how different populations living in the same environment interact and affect each other's growth characteristics. The typical approach for assessing this dynamics is the formulation of logistic functions with added interaction terms. We will discuss this example later in this chapter.

In many cases, ODE representations of growth processes are also more intuitive than an explicit growth function, because they can be interpreted in comparison with exponential growth, for which we have a good intuitive feel. In (14.3), the first term on the right-hand side represents unabated exponential growth, whereas the second term is sometimes interpreted as a diminishing **crowding** effect, which is proportional to the square of the number of individuals in a population. As an alternative, one may formulate (14.3) as

$$\dot{N} = r\left(\frac{K - N}{K}\right)N. \tag{14.4}$$

Mathematically, (14.3) and (14.4) are exactly equivalent, but (14.4) suggests the interpretation of a growth rate $r\left(\frac{K-N}{K}\right)$ that depends on the population density. This growth rate is almost equal to r for very small populations, where N is much smaller than K and decreases toward 0 if the population reaches the carrying capacity K.

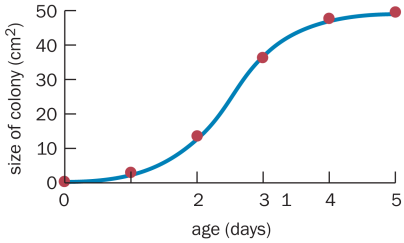


Figure 14.1 Growth of a bacterial colony. The colony, measured by its area on a Petri dish, was assessed over time (red dots). The logistic function $N_{(t)} = 0.2524 / (e^{-2.128t} + 0.005125)$ (blue line; see Chapter 5) models the data very well. (Data from [42].)

Similarly, the Richards function (14.1) may be written as

$$\dot{N}_R = \alpha\left[1-\left(\frac{N_R}{K}\right)^{\nu}\right]N_R, \tag{14.5}$$

which again can be interpreted as a variant of an exponential process with a population-density-dependent growth rate, which consists of α and the term in square brackets.

The Gompertz function (14.2) may also be written in ODE form with density-dependent rate:

$$\dot{N}_G = r\ln\left(\frac{K}{N_G}\right)N_G. \tag{14.6}$$

Alternately, the same Gompertz function may be formulated as a set of two ODEs:

$$\begin{aligned}\dot{N}_G &= RN_G,\\ \dot{R} &= -rR.\end{aligned} \tag{14.7}$$

While this representation may seem unnecessarily complicated, it is actually easier to interpret than (14.2) and (14.6). In the first equation, the new variable R is easily identified as the rate of an exponential growth process. The second equation shows that this growth rate is time-dependent; namely, it represents an exponential decay of the type R_0e^{-rt} (see Chapter 5). Thus, different representations may be mathematically equivalent but can have different biological interpretations. Furthermore, the embedding of functions into differential equations can be a tool for comparing and classifying growth laws [1, 9–11].

14.1.2 More Complex Growth Phenomena

Although many growth laws have been collected over time, there is still no guarantee that an observed dataset can be accurately modeled by any of them. As a case in point, the maximal sustainable size of the human population continues to be a matter of debate (**Box 14.1**). If we knew the growth function of the world population, we could easily determine the maximal size.

BOX 14.1: TRENDS IN THE GROWTH OF THE HUMAN POPULATION

The human world population has been steadily growing throughout recorded history, except for relatively minor "blips" caused by epidemics or wars. The big question on many people's minds is whether or when this growth will slow down, stop, or even be reversed. Census bureaus around the world try to answer this enormously important question, using estimates of current population sizes (**Figure 1**) and birth rates (**Figure 2**). Even with the best data available, the resulting predictions vary tremendously (**Figure 3**).

In some cases, the parameters of growth functions are relatively easy to assess. For instance, exponential growth is characterized by a growth rate parameter, which often is directly

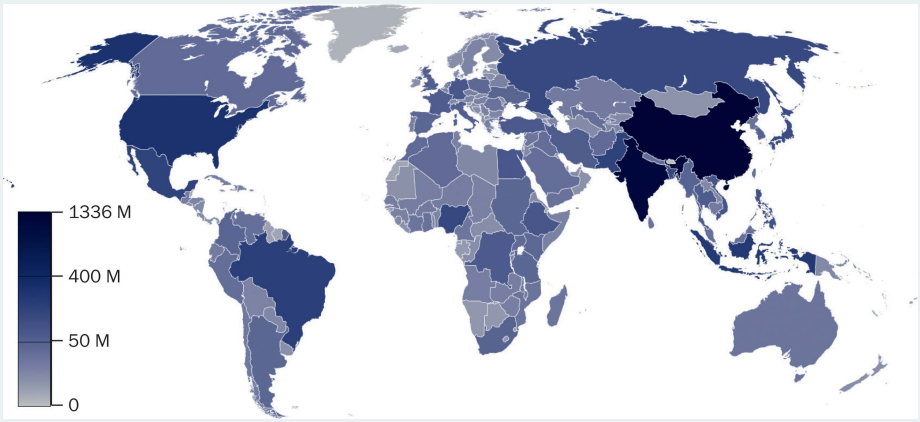


Figure 1 World population by countries and regions. The map is composed from data published by the United Nations (2019), the United States Census Bureau (2023), the US Central Intelligence Agency (2020), and Wikipedia (2023). Estimates of the total population range from 7.7 to 8.2 billion individuals. From https://learnertrip.com/demography/world-population-2023/

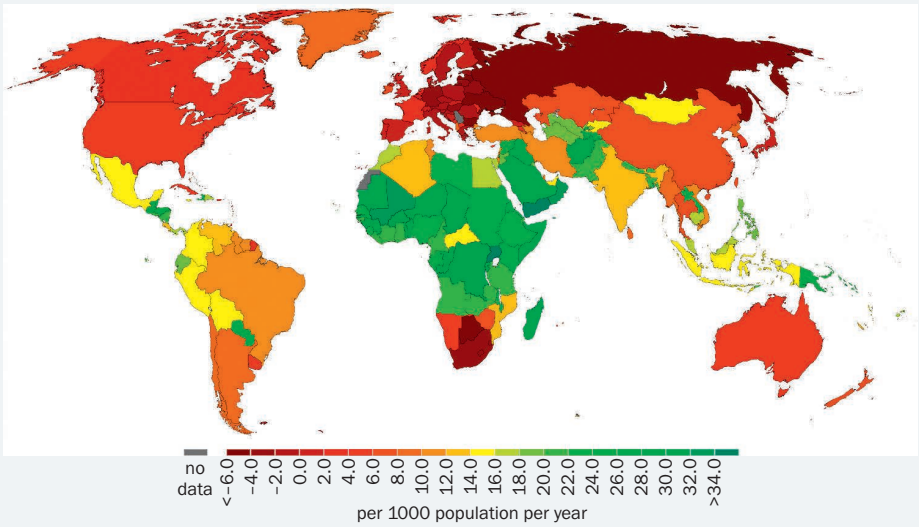


Figure 2 Global fertility rates as of 2020. Birth rate information, combined with actual population distributions, is used to predict future population trends.

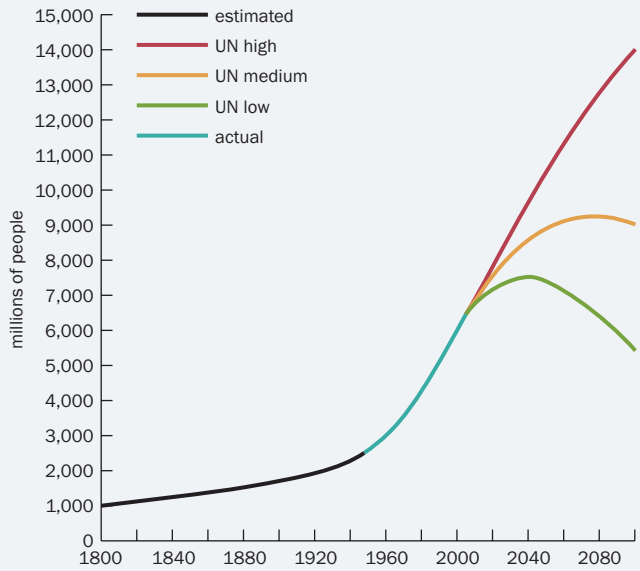


Figure 3 History and differing forecasts regarding the growth of the human world population. The predictions, based on UN 2004 projections and US Census Bureau historical estimates, differ drastically. (Loren Cobb, published under CC BY-SA 3.0 license.)

related to features such as the division time of cells (see [12, 13] and Chapters 2, 4 and 5). However, other population parameters are not directly related to parameters that can be measured in individuals or populations. For instance, the bacterial colony in Figure 14.1 approaches a final size of about 50 cm^2 on an agar plate, but this value cannot be inferred from studying individual bacteria or subpopulations, because it reflects genetic, physiological, and environmental factors in some unknown combination. Trees in a planted plot of land grow according to a widely acknowledged "3/2 rule" that relates their size to the planting density. This relationship of a **power-law function** with negative exponent 3/2 has been observed uncounted times, but there is no true rationale for the relationship or the specific numerical value [14, 15].

A different unsolved problem regarding population dynamics is the formulation and analysis of models for so-called **mixed communities**, including **microbiomes**. These communities sometimes consist of hundreds or thousands of different species of microbes that at once compete for space and nutrients and rely on each other for survival. Indeed, mixed communities are the rule rather than the exception because we can find them in "environments" from our mouth to soil, sewage pipes, lakes and oceans. We will discuss them later in this chapter.

Many typical population studies ignore spatial considerations, even though they may be very important. Intriguing examples are the growth of tumors, the incremental spread of introduced pests, such as the red fire ant *Solenopsis invicta* (see Exercise 14.34), and the spread of a disease throughout a country. A complicating aspect of human populations is that humans behave in complex and often unpredictable ways. For instance, imagine the task of predicting the trends in a population of people with diabetes under different proposed health-care schemes. A powerful approach to studying such cases, via computer simulation, is agent-based modeling, where individuals are modeled with heuristic rules that guide their decisions and interactions with others (see Chapter 11 and [16, 17]). A disadvantage of this method is the fact that explicit mathematical analyses are often difficult.

14.2 POPULATION DYNAMICS UNDER EXTERNAL PERTURBATIONS

The dynamics of a single unperturbed population is usually not very complicated, as we have seen in the previous section. For instance, it does not really matter all that much how small the initial population size is: in a realistic model, the population increases toward some final, stable size, which is easily determined through simulation (**Figure 14.2**). We can also often compute this final size with elementary mathematical methods. If the growth process is formulated as an explicit function of time, we consider the growth function for time going to infinity. For example, in the case of the Gompertz function (14.2), the exponential term $\exp(-rt)$ approaches zero, which causes the term $\exp[-b\exp(-rt)]$ to approach 1 so that $N_G(t)$ approaches K.

If the growth process is represented by a differential equation, we know from Chapter 5 how to proceed: the time derivative is set equal to zero and the resulting algebraic equation is solved. For instance, the steady states N_{ss} of the logistic function (14.3) are determined by solving the equation

$$0 = rN_{ss} - \frac{r}{K}N_{ss}^2. \tag{14.8}$$

The problem has two solutions. First, for $N_{ss} = 0$, the population is "empty" and remains that way. Second, if $N_{ss} = 0$, we are allowed to divide by r and by N_{ss}, which leads to $1 - N_{ss}/K = 0$ and yields $N_{ss} = K$ as the steady state, which corresponds to the carrying capacity.

Is it possible that the population ever exceeds this value? Sure. It could be that individuals immigrate from other areas, thus creating a higher value. It might also happen that the conditions in the previous years were very favorable, thereby permitting the population to grow beyond its typical carrying capacity. In contrast to growth from a small size, which is s-shaped, the decrease in population size looks like an exponential decay function, which approaches the same steady state, K, as before (**Figure 14.3**).

In nature, many organisms are subjected to predation, and bacteria are no exception. For instance, some amoeba, other protozoans, and worms eat bacteria. As Richter [18] pointed out in the context of whale hunting, the dynamics of a species depends critically on the type of predation. In many models, predation is assumed to be proportional to the current number of individuals, while it may occur with a constant rate in another typical case. It is easy to explore the consequences. If predation is proportional to population size, we formulate the dynamics as

$$\dot{N}_1 = rN_1 - \frac{r}{K}N_1^2 - pN_1, \tag{14.9}$$

where we have added the index 1 merely for easy discussion and comparisons. In the case of constant predation, the formulation is

$$\dot{N}_2 = rN_2 - \frac{r}{K}N_2^2 - P. \tag{14.10}$$

Let us suppose that the common parameters in the two models are the same as before and that both populations start with an initial size of 50. When we set the additional predation parameters more or less arbitrarily as $p = 0.775$ and $P = 25$, the two populations have very similar dynamics and roughly the same steady state (**Figure 14.4A**). Thus, one might think that the type of predation does not really matter. However, the two populations exhibit distinctly different responses to perturbations. For example, if the two populations are reduced to 15 at some point, maybe due to other predators, disease, or some environmental stress, the first population recovers quite quickly, while the second population never recovers and eventually dies out (Figure 14.4B). Stability analysis (Chapter 5) allows the

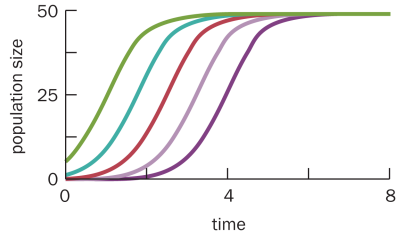


Figure 14.2 Shifted logistic functions. Different initial values shift the logistic function (14.3) in the horizontal direction, but eventually all curves reach the same final value, K. The red curve $N(0) = 0.2511$ corresponds to the process in Figure 14.1. Initial values for the curves, from left to right, are 20×0.2511; 5×0.2511; 0.2511; 0.2511/5; 0.2511/20.

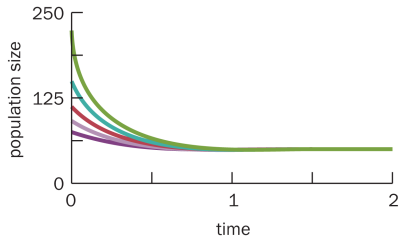


Figure 14.3 Logistic functions above their carrying capacity. If the logistic function is initialized above K, it decreases toward K in a monotone fashion that does not resemble the s-shaped growth from low initial values. Initial values, from bottom to top, are 75; 1.2×75; 1.5×75; 2×75; 3×75.

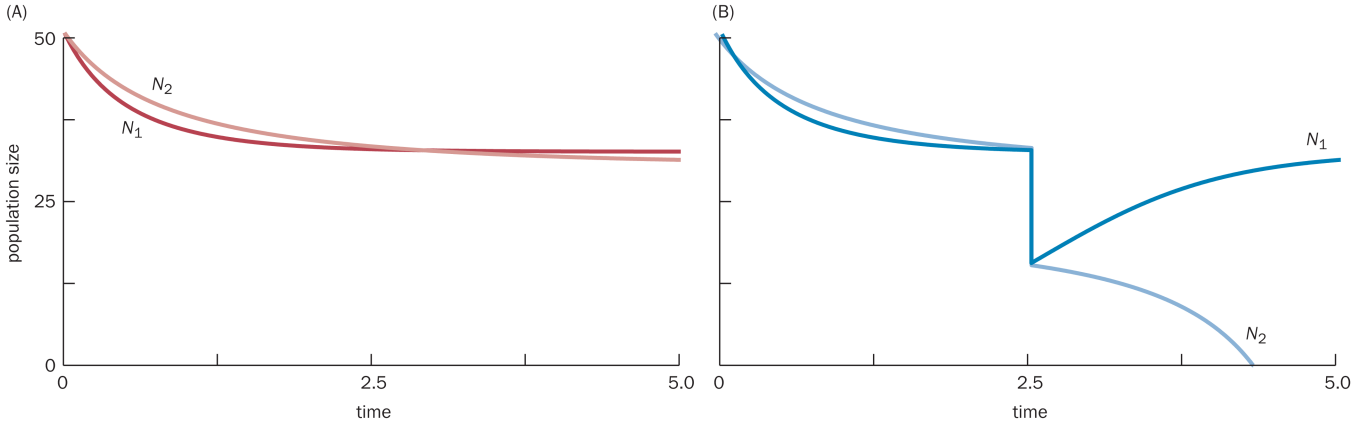


Figure 14.4 Comparison of two logistic growth processes exposed to different types of predation. Without perturbations (A), the two functions (14.9) and (14.10) are quite similar. However, if some external factor reduces both populations to a sufficiently small value, the population with proportional predation (N_1) recovers, while the population exposed to constant predation (N_2) becomes extinct. In (B), both populations are reduced to 15 at time 2.5.

computation of the perturbation threshold above which the second population recovers and below which it goes extinct.

In typical models of population growth, all parameters are constant, and the dynamics is therefore fully determined. In reality, the growth rate may depend on environmental factors, such as temperature, which of course is a function of time, or it may depend on the current population size [18]. Furthermore, the parameters usually fluctuate stochastically within some range. For very large populations, such variations are not problematic, but they can become detrimental if a population is very small. Modeling stochasticity is difficult with differential equations but can be achieved rather easily with analogous discrete models [19].

14.3 ANALYSIS OF SUBPOPULATIONS

While the growth of a **homogeneous** population is interesting in many respects, the range of likely dynamics is somewhat limited. More interesting are the dynamic behaviors of **stratified populations** containing different categories of individuals and interactions between the subpopulations. An example of the dynamics of subpopulations was rather extensively discussed in Chapters 2 and 11, where models described how susceptible individuals become infected, recover, and possibly become susceptible again. Uncounted variations of these SIR models have appeared in the mathematical modeling literature, and we will not discuss them again. A variation of such subpopulation models accounts for the different ages of individuals that are exposed to a disease vector; a very clear example was COVID-19, which was particularly troublesome for the elderly. Specific questions in such a study ask whether a disease outbreak can be controlled and at what age it is most efficacious to vaccinate individuals. If many age classes, along with different health and treatment states, are considered, these models can consist of hundreds of differential equations [20]. Interestingly, the dynamics of the actual disease-causing agents, such as viruses, is often not explicitly accounted for in these models [21]; however, see the main example in Chapter 11.

An interesting example of a simple growth model with age classes was proposed by Leslie [22] many decades ago. It is framed as a linear recursive model of the type we discussed in Chapter 4. The individuals in the population are categorized into subpopulations of different ages and different proliferation rates. All individuals have the same potential lifespan, but they may of course die earlier. For humans with a maximum age of 120, one could select $m = 12$ age classes of $\tau = 10$ years each and set a corresponding recording interval of τ years. In other words, population sizes are exclusively documented only every τ years but not in between.

The Leslie model is most easily formulated as a **matrix** equation, which for four age classes has the form

$$\begin{pmatrix} P_1 \\ P_2 \\ P_3 \\ P_4 \end{pmatrix}_{t+\tau} = \begin{pmatrix} \alpha_1 & \alpha_2 & \alpha_3 & \alpha_4 \\ \sigma_1 & 0 & 0 & 0 \\ 0 & \sigma_2 & 0 & 0 \\ 0 & 0 & \sigma_3 & 0 \end{pmatrix} \begin{pmatrix} P_1 \\ P_2 \\ P_3 \\ P_4 \end{pmatrix}_t . \tag{14.11}$$

Let us dissect this equation. The vector on the left contains $P_1, \ldots, P_4$ at time $t + \tau$. These quantities are the sizes of the subpopulations in age classes 1, 2, 3, 4, respectively. Almost the same vector appears on the far right, except that it captures the population sizes at time t. Thus, the recursive character of the equation is the same as in the simple case of exponential growth (Chapter 4), and writing (14.11) in vector and matrix notation yields

$$\mathbf{P}_{t+\tau} = \mathbf{L}\mathbf{P}_t . \tag{14.12}$$

Connecting the two vectors is the Leslie matrix **L**. It consists mainly of zeros, except for quantities α in the first row and quantities σ in the sub-diagonal. The α's represent the reproduction rate for each age class, given as the number of offspring produced per individual in this age class within the time period τ. The parameters σ represent the fractions of individuals surviving from one age class into the next and have values between 0 and 1. Writing out the matrix equation shows these features more intuitively. For instance, the equation for P_2 is obtained by multiplying the elements in the second row of the matrix with the vector on the right-hand side, which yields

$$P_{2,t+\tau} = \sigma_1 P_{1,t} + 0 \cdot P_{2,t} + 0 \cdot P_{3,t} + 0 \cdot P_{4,t} = \sigma_1 P_{1,t} . \tag{14.13}$$

Expressed in words, age class 2 at time $t + \tau$ contains exactly those individuals that were in age class 1 at time t and survived time period τ. Age classes 3 and 4 have a similar structure. (Write down the equations to convince yourself.) Now let us look at age class 1. There is no class 0, so that there cannot be survival into class 1. Instead, all individuals in class 1 come exclusively from reproduction, and the rate of this process differs for the various age classes. For instance, individuals in class 3 produce offspring at a rate of α_3 per time period τ. These birth processes are reflected in the first equation, describing $P_{1,t+\tau}$, which is

$$P_{1,t+\tau} = \alpha_1 P_{1,t} + \alpha_2 P_{2,t} + \alpha_3 P_{3,t} + \alpha_4 P_{4,t} . \tag{14.14}$$

How does this equation account for the fact that only a certain percentage (quantified by σ_1) of individuals of class 1 survive? Only indirectly. First, according to the set-up of the model, no individual can stay in class 1 (although this class can have offspring that again would be in the same class). Furthermore, the surviving individuals ($\sigma_1 P_{1,t}$) show up in class 2, and those dying (namely, $(1 - \sigma_1)P_{1,t}$ individuals) are no longer explicitly accounted for; they disappear from the population and from the model.

As a side note, the transition process from t to $t + \tau$, as it is represented in (14.11), uses column vectors, as it is commonly done, which however is different from the row vectors we used for Markov models (Chapter 4). The two notations are interchangeable as long as the system matrix is transposed.

Because the process is so rigidly structured, we can write the Leslie equation for any numbers of age classes:

$$\begin{pmatrix} P_1 \\ P_2 \\ \vdots \\ P_m \end{pmatrix}_{t+\tau} = \begin{pmatrix} \alpha_1 & \alpha_2 & \ldots & \alpha_{m-1} & \alpha_m \\ \sigma_1 & 0 & \ldots & 0 & 0 \\ 0 & \sigma_2 & \ldots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & & \sigma_{m-1} & 0 \end{pmatrix} \begin{pmatrix} P_1 \\ P_2 \\ \vdots \\ P_m \end{pmatrix}_t . \tag{14.15}$$

This **recursion** again moves forward by τ time units at a time. It is also possible to jump $n\,\tau$ steps at once. Namely, we simply use the n^{th} power of the matrix:

$$\mathbf{P}_{t+n\tau} = \mathbf{L}^n \mathbf{P}_t. \tag{14.16}$$

Leslie models of this type are interesting tools for exploring trends in subpopulations, even though they make many explicit and implicit assumptions (**Box 14.2**). For instance, there is no mention of population sizes at time points inside the intervals, that is, between $t + k\,\tau$ and $t + (k+1)\,\tau$ for $k = 0, 1, 2, \ldots$. For simplicity, males are often not included, because they do not bear children. Importantly, the growth process is independent of the current population size. In other words, the parameters α and σ do not depend on time or on the population size—which is at odds with many real populations. As a consequence, the population either always keeps growing or always keeps decreasing, possible after some initial phase that is not monotonic; the population size may also be constant, but this state is not stable unless all P_i are zero (see Exercise 14.10). Of course, it is mathematically possible to make α and σ dependent on time or population size, but then the model is no longer linear, and matrix computations become mathematically much more difficult. For pure simulation purposes, these extensions are easy to implement and explore.

BOX 14.2: TYPES OF AGE STRUCTURES WITHIN POPULATIONS

In the Leslie model, the birth rates within a population typically differ by the age of the mother but they are constant over time within each age class. Similarly, death rates are usually higher for older individuals but remain constant over time. In reality, neither birth nor death rates are constant. Representative "population pyramids" are shown in **Figure 1**. China's population structure is shown in **Figure 2**. It implies some of the problems the country is expected to have in the future.

The Leslie model uses average survival and reproduction rates and therefore is **deterministic**. As a consequence, every simulation with the same parameters and initial sizes of subpopulations yields exactly the same results. In a variation that is conceptually simple but not trivial in its implementation and analysis, one could consider randomness in survival and reproduction [12, 13]. For instance, one might imagine cells that are moving through the **cell cycle** and enter its different phases in a probabilistic fashion. Typical quantities to be assessed might include the numbers of cells in each phase and the distribution of transition times throughout the cell cycle. This seemingly harmless change from deterministic to **random** events moves the model into the realm of stochastic processes, which are much more complicated than the deterministic Leslie model. Good texts include [23–25]. Also recall the project in Box 4.1.

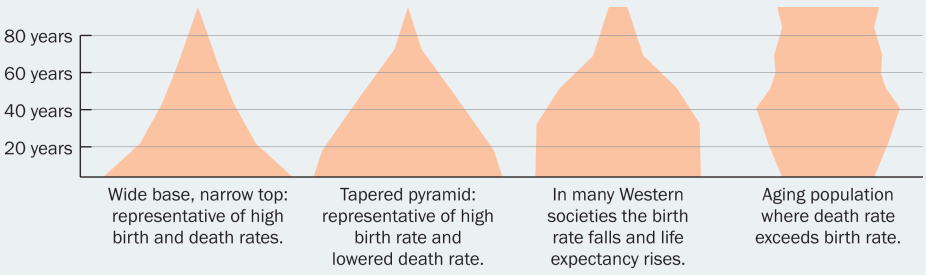


Figure 1 Types of population pyramids. Depending on the balance between birth and death rates, which changes over time in a population, the age distribution within a population assumes a different shape.

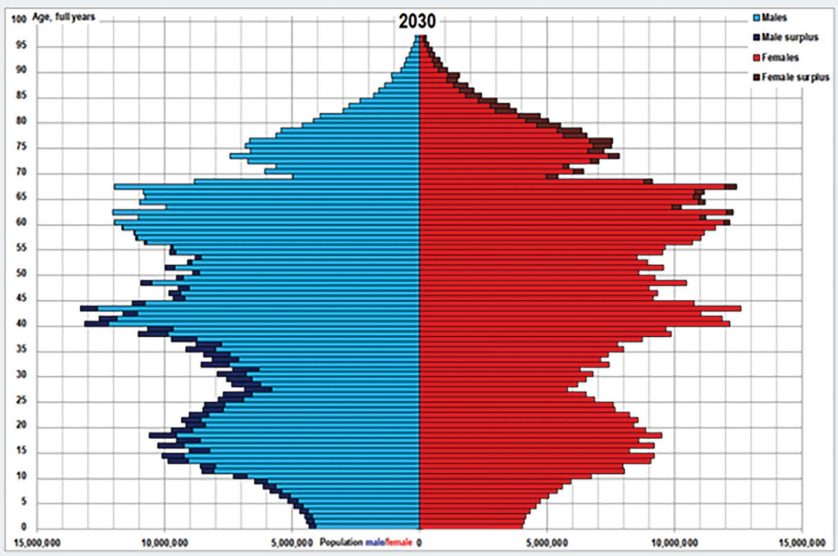


Figure 2 China's population pyramid projected for 2030. The effects of China's one-child policy, introduced in 1978—and the preference for male infants—are visible in the unnatural population balance between young and old. (From Rickky1409, published under CC BY-SA 4.0 license.)

14.4 INTERACTING POPULATIONS

14.4.1 General Modeling Strategy

In the simplest case of an interaction system, the individuals of two populations do not influence each other much, except that they might compete for the same resources. The simplest example may be a mixture of two exponentially growing bacterial populations with different growth rates. A more interesting example is a tumor growing within the tissue where it originated. Often the growth rate of tumor cells is not necessarily faster than that of healthy cells, but tumor cells undergo cell death at a much lower rate. We can easily model this situation with a system of healthy (H) and tumor (T) cells of the form

$$\begin{aligned} \dot{H} &= r_H H - m_H H^2, \\ \dot{T} &= r_T T - m_T T^2. \end{aligned} \tag{14.17}$$

The parameters r_H, r_T, m_H, and m_T represent growth and mortality rates of healthy and tumor cells, respectively. Suppose both growth rates are the same ($r_H = r_T = 1$), the mortality rates are $m_H = 0.01$ and $mT = 0.0075$, and initially only 0.1% of the cells are tumor cells ($H(0) = 1$, $T(0) = 0.001$). With these settings, it is easy to simulate the system. Even though the growth rates are the same and the number of tumor cells is at first miniscule, the lowered mortality allows the tumor eventually to take over (**Figure 14.5**).

More realistic than the coexistence in this example is the situation where two or more populations truly interact with each other. The most straightforward case is competition, which effectively reduces the carrying capacities of the populations. The usual approach is a model of logistic growth processes, which are expanded with interaction terms. If the interactions are very strong, one can imagine that not every population can survive. The typical questions asked in these situations are therefore: who will survive? Is coexistence possible? How long does it take to reach equilibrium? What do the transients from initiation to termination look like? Interestingly, some of these questions can be answered with a high degree of generality if only two populations are involved. The methods for this task are discussed next.

14.4.2 Phase-Plane Analysis

Every time we run a simulation study, we have to ask ourselves how much the results depend on the particular parameter values we had chosen. If we changed the values in some way, would the results be fundamentally different? For instance, if population A survives in a simulated scenario, is that a general outcome or is it merely the result of our current choice of initial numbers, growth rates, and other characteristics? For complicated models, these types of questions can be difficult to answer. However, if we are dealing with just two populations, qualitative **phase-plane analysis** is an effective tool. In principle, **qualitative analysis** could also be done for systems with more species, but a key contributor to our intuition would no longer be of help, namely a two-dimensional visualization. So, let us suppose we have two interacting populations whose dynamics is described with two ODEs. These equations consist of logistic growth functions of the type in (14.3) and (14.4), which furthermore contain one interaction term each. These interaction terms are typically of the same format but with different parameter values, because the two species respond differently to the effect of the interaction. Specifically, we use the following pair of equations as an example:

$$\begin{aligned} \dot{N}_1 &= r_1 N_1 \left(\frac{K_1 - N_1 - aN_2}{K_1} \right), \\ \dot{N}_2 &= r_2 N_2 \left(\frac{K_2 - N_2 - bN_1}{K_2} \right). \end{aligned} \tag{14.18}$$

In order to facilitate both mathematical and computational analyses, we select the following, more or less arbitrary, parameter values: $r_1 = 0.15$, $K_1 = 50$, $a = 0.2$, $r_2 = 0.3$, $K_2 = 60$, $b = 0.6$, $N_1(0) = 1$, $N_2(0) = 1$.

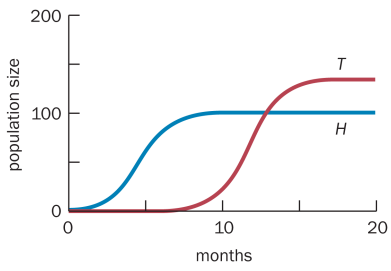


Figure 14.5 Dynamics of two coexisting populations. In the long run, a higher growth rate or lower mortality rate is more influential than a larger initial population size, as demonstrated here with a simplified model of healthy (H) and tumor (T) cells growing in the same tissue (see (14.17)).

It is straightforward to solve this system numerically for N_1 and N_2 as functions of time; a plot of the solution is shown in **Figure 14.6**. Some details are probably to be expected; others may be surprising. First, the dynamics of N_1 is similar to that of a logistic function that is not affected by outside forces. N_2 grows faster, owing to its larger growth rate, but it begins to fade as population N_1 grows. Due to the competition, neither population reaches its carrying capacity. In particular, N_2 only reaches a final size of about 34, while its carrying capacity is 60.

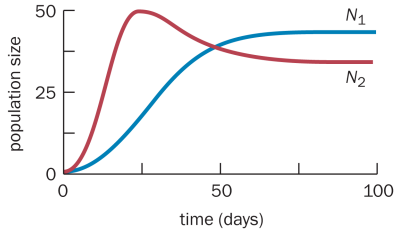


Figure 14.6 Two populations, N_1 and N_2, are competing for the same resources. N_2 grows faster, but competition from N_1 eventually leads to a decrease in population size. Neither population reaches its carrying capacity.

The key concept of the following type of qualitative analysis is that we do not take explicit account of time and instead plot the behavior of one population against the behavior of the other. Thus, we look at a coordinate system where the axes represent N_1 and N_2 (**Figure 14.7**). If we kept a record of N_1 and N_2 over time, the dataset would have the features shown in **Table 14.1**. We can easily see from Figure 14.7 that time is represented in a drastically different way than before, namely merely as labels along a curve that describes how N_2 depends on N_1. In fact, we could double the speed of the system by multiplying all rate constants in (14.18) by 2, and the plot would look exactly the same, except that the time labels would be changed. Expressed differently, time has moved to the background, and the focus is on the relationship between N_1 and N_2. Even though time no longer appears explicitly, this phase plot indicates the speed of the processes indirectly through the distance between consecutive time points. In Figure 14.7, the speed is greatest between $t = 10$ and $t = 20$, and the process almost comes to a halt beyond $t = 60$.

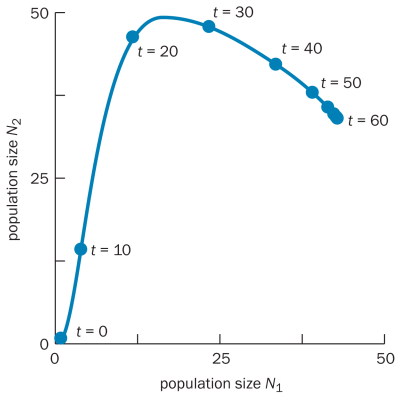


Figure 14.7 Phase-plane plot of two interacting populations. N_2 is plotted against N_1 for the 11 time points in Table 14.1. The population sizes between times $t = 70$ and $t = 100$, computed from (14.18), cluster close to those at $t = 60$.

Intriguingly, we can analyze this relationship generically with minimal information about the parameter values of the system. We begin with the steady states of the system. These are determined by setting both time derivatives in (14.18) equal to zero and solving for N_2 as a function of N_1 (or the other way around). The specific form of the equations makes this computation very easy. First, we can convince ourselves rather easily that $(N_1, N_2) = (0, 0)$ is a steady state. This makes sense, because if no individuals exist, there is no dynamics and everything stays the same. This "trivial" case is important to remember, but let us assume for a moment that N_1 and N_2 are strictly positive. If we divide the two equations by N_1 and N_2, respectively, which we are allowed to do because we assume that these variables are not zero, we obtain two simpler equations in two unknowns, which we can solve directly:

$$\begin{aligned} 0 &= r_1 N_1 \left(\frac{K_1 - N_1 - aN_2}{K_1} \right) \Rightarrow N_2 = \frac{K_1 - N_1}{a}, \\ 0 &= r_2 N_2 \left(\frac{K_2 - N_2 - bN_1}{K_2} \right) \Rightarrow N_2 = K_2 - bN_1. \end{aligned} \tag{14.19}$$

TABLE 14.1: SIZES OF TWO COMPETING POPULATIONS, N_1 AND N_2, OVER TIME

Time	N_1	N_2
0	1	1
10	4.06	14.40
20	11.96	46.38
30	23.58	47.96
40	33.76	42.27
50	39.34	38.04
60	41.70	35.81
70	42.61	34.80
80	42.96	34.37
90	43.10	34.20
100	43.15	34.13

These equations are valid for any positive parameter values. Using the numerical values from before, the nonzero steady-state solution is $(N_1, N_2) = (43.18, 34.09)$. The system contains two further steady states, where either N_1 or N_2 is zero and the other variable is not.

In addition to the four steady states, we can explore situations where only one of the time derivatives is zero. These situations are modeled by the two equations in (14.19) when they are considered separately. In the first case, the equation given by $\dot{N}_1 = 0$ is called the N_1-**nullcline**, where "null" refers to zero and "cline" comes from the Greek root for slope. In general, this function is nonlinear, but the N_1-nullcline in our example is represented by a linear function, namely $N_2 = (K_1 - N_1)/a$. It has a negative slope and intersects the vertical axis at K_1/a and the horizontal axis at K_1. Similarly, the N_2-nullcline is determined from setting the time derivative of N_2 equal to zero, and the result is $N_2 = K_2 - bN_1$, which is again a straight line with negative slope. It intersects the vertical axis at K_2 and the horizontal axis at K_2/b. The two nullclines intersect at the nonzero steady state, because at this point both time derivatives are zero. **Figure 14.8**, which is intentionally not drawn to reflect the earlier data to indicate the generic nature of the plot, shows the two nullclines and the two steady-state points. Note that the nullclines are straight because (14.18) has the format of a Lotka–Volterra system (see Chapter 5). In other cases, they would likely be nonlinear.

Even without specifying parameter values, we can find out more about the system by simply analyzing the phase plot with the two nullclines. For instance, for every point exactly on the N_1-nullcline, the derivative of N_1 is by definition zero. Therefore, any **trajectory** of the ODE system must cross this nullcline exactly vertically because the value of N_1 does not change on this line. Furthermore, $\dot{N}_1$ is less than zero on the right side of the nullcline and greater than zero on the left side. We can confirm this by moving from any point on the nullcline a tiny bit to the right: the value of N_1 increases very slightly, which makes the right-hand side of the first differential equation slightly negative. Thus, all derivatives $\dot{N}_1$ on the right of the N_1-nullcline are negative, and they are positive on the left. The analogous arguments hold for the N_2-nullcline, where all trajectories must cross horizontally. Taking these pieces of information together, the plot consists of four areas that are characterized by the signs of the two derivatives (see Figure 14.8). Every trajectory that satisfies the pair of differential equations (14.18) must move according to these derivatives, independent of specific parameter values. This insight allows us to sketch all possible trajectories on the phase plane. We do not know exactly what these trajectories look like, but we do know their most significant features, namely in what general direction they move and whether they cross a nullcline in a vertical or horizontal direction (**Figure 14.9**). Two special cases arise if one of the two variables is zero (see Exercises 14.20 and 14.21). Finally, the phase plot indicates that all trajectories move away from (0, 0): this trivial steady state is unstable.

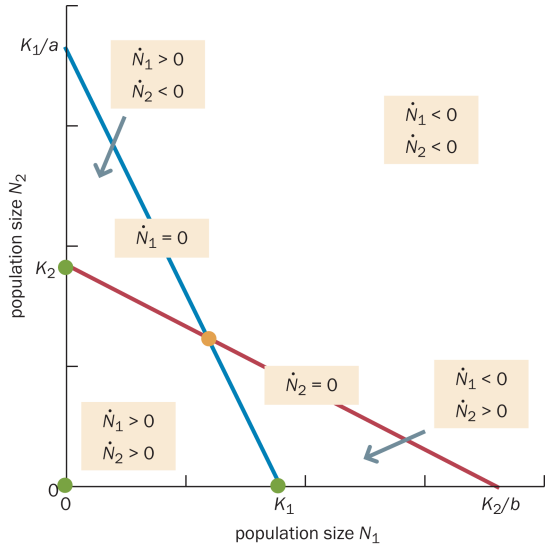


Figure 14.8 Phase-plane plot for two competing populations N_1 and N_2. The nullclines (red and blue lines) intersect at the nonzero steady state (yellow circle). At other steady states (green circles), N_1, N_2, or both populations are zero.

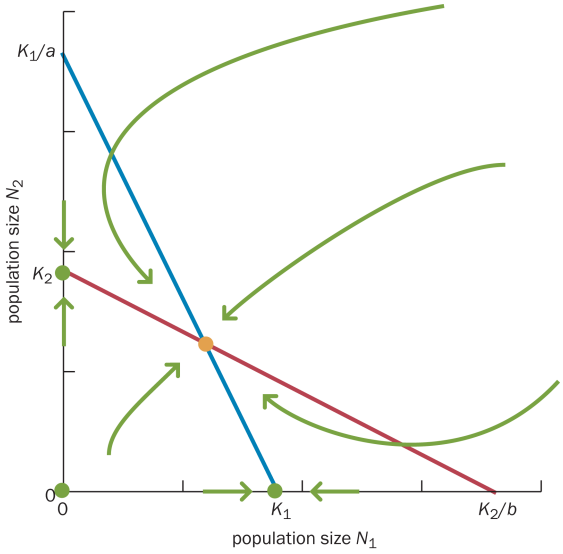


Figure 14.9 Coarse visualization of the dynamics of two competing populations. Sketching "generic" trajectories in the phase-plane plot shows that the two competing populations N_1 and N_2 approach the nonzero steady state (yellow circle), where both survive and coexist. Trivial mathematical exceptions are cases where N_1, N_2, or both are zero at the beginning.

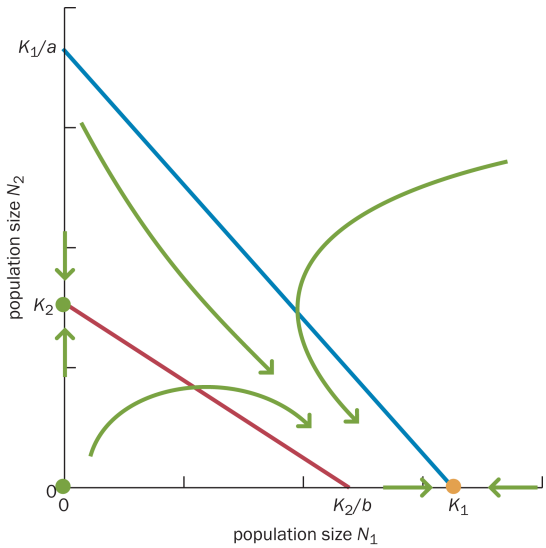


Figure 14.10 Depending on the parameter values, the two populations cannot coexist and one will eventually go extinct. If the parameter values are such that the nullclines do not intersect in the positive quadrant, one of the two populations (here N_1) survives at its carrying capacity, whereas the other (here N_2) disappears.

In the example we have just discussed, the two nullclines intersect at a single point, the nontrivial steady state. It may also happen that they do not intersect in the positive quadrant, which is the only area of interest. For instance, suppose $K_1 > K_2/b$ and $K_1/a > K_2$. The situation is shown in **Figure 14.10**. Now there are only three areas, with different combinations of signs of the two derivatives. Nonetheless, exactly the same arguments hold as before, but now all trajectories converge to $N_1 = K_1$, $N_2 = 0$, with the exception of trajectories that start exactly on the vertical axis (i.e., $N_1(0) = 0$). The interpretation is that, unless there are no individuals in population N_1 at all, N_1 will eventually survive and grow to its carrying capacity, while N_2 will become extinct. Examples are invasive species, which start with a few organisms but, if unchecked, lead to the extinction of the corresponding native species. The slopes and intersections of the two nullclines permit two additional situations, which are left as exercises.

Arguably the most famous class of examples where this type of analysis has been used concerns predator–prey systems. These systems were first discussed with mathematical rigor almost one hundred years ago by Alfred Lotka, a physical chemist and biologist, and Vito Volterra, a mathematician. Lotka used this approach originally for the analysis of chemical reactions but it is better remembered for population studies in ecology (see also Chapter 5). In its simplest form, the Lotka–Volterra (LV) model describes the interactions between one prey

species N and a predatory species P. The equations are nonlinear and quite similar to those we discussed earlier:

$$\begin{aligned}\dot{N} &= \alpha N - \beta NP,\\ \dot{P} &= \gamma NP - \delta P.\end{aligned} \tag{14.20}$$

All parameters are positive. In the absence of predators, N grows exponentially with growth rate α, whereas the presence of predators limits this growth through the term βNP. The form of a product of the two variables with a rate constant is based on arguments similar to those discussed in the context of chemical mass action kinetics (Chapter 9), where the product is used to describe the probability of two (chemical) species encountering each other in a three-dimensional space. The predator population P is assumed to rely exclusively on N for food; without N, the predator population decreases exponentially toward extinction. Again, the growth term is formulated as a product, but the rate γ is usually different from the rate β in the prey equation, because the magnitude of the effect of the same process (predation) is significantly different for prey and predators.With typical parameter values (α = 1.2, β = 0.2, γ = 0.05, δ = 0.3, $N_0 = 10$, $P_0 = 1$), N and P exhibit ongoing oscillations, in which P follows N with some delay and typically with a much smaller magnitude (**Figure 14.11**).

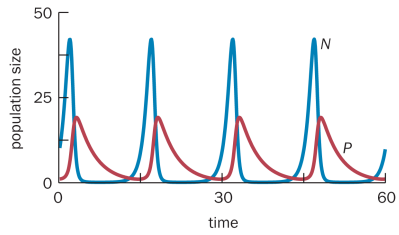


Figure 14.11 Idealized dynamics of predator (P) and prey (N) populations. The dynamics of the Lotka–Volterra system (14.20) consists of ongoing oscillations. In reality, such oscillations are only seen when the two species coexist in a complex environment.

Linearization and stability analysis demonstrates that (0, 0) is a saddle point and that the interior, non-trivial steady state is a center (see Chapter 5). Moreover, one can show that the general solution outside (0, 0) consists of closed orbits around the interior steady-state point, which correspond to sustained center oscillations with the same amplitude [26, 27]. These oscillations are no limit cycles, and every perturbation, even if it is very small, moves the system to a different orbit or causes a species to die; confirm this with simulations of (14.20).

Many important features of phase-plane analysis remain the same if the nullclines are nonlinear. In particular, the slope of a variable right on its nullcline is zero, so that trajectories again transect nullclines either horizontally or vertically. Also, the intersections of nullclines constitute steady-state points. The main difference is that the nullclines may intersect more than once. Thus, the system may have two interior intersection points, where neither of the variables is zero, one of which is typically stable whereas the other is unstable. Of course, there could be more than two intersections, depending on the model structure.

As an example, let's consider the system

$$\begin{aligned}\dot{N}_1 &= N_1\left(10 + 1.5N_1 - 0.1N_1^2 - N_2\right)/150\\ \dot{N}_2 &= N_2\left(50 + 0.3N_1^2 - 7N_1 - N_2\right)/150,\end{aligned} \tag{14.21}$$

where each population is augmented and reduced through two processes each. The nullclines are again computed by setting the first or the second equation equal to zero and expressing N_2 as a function of N_1; do this computation by hand. They are shown in Panel A of **Figure 14.12**. As before, the axes are additional nullclines, and the system allows for the extinction of either N_1 or N_2. Also, (0, 0) is a steady-state point, and we will see that it is unstable.

To explore the system behavior within the different areas created by the nullclines, it is useful to compute the **vector field** of the system. This representation consists of solving the equations many times for a very short time period, each time starting with initial values that are taken from a grid of relevant (N_1, N_2) pairs. Such a representation is shown in Panel B of Figure 14.12. Relatively long lines correspond to fast changes, whereas lines that almost look like dots represent very slow changes; rationalize why this is so. True dots are steady-state points. A sample of trajectories is shown in Panel C; the arrows were added to indicate the direction of each trajectory. The panel also contains a dashed line, called the **separatrix**, which is Latin for "she who separates." Why this line is allegedly female may be pondered but is not important here. What is important is that this line divides the phase plane into sections with different behaviors: Starting to the left of the separatrix, the system will eventually approach the steady state where N_1 becomes extinct and N_2 is 50. Starting to the right, the system will spiral into the coexistence

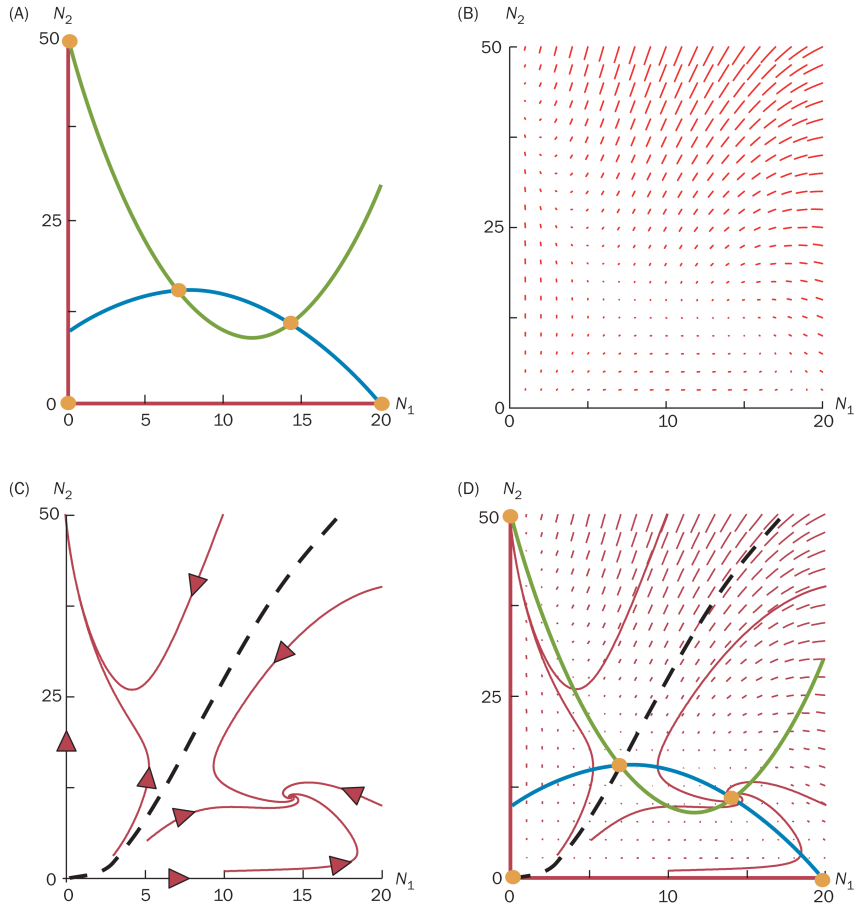


Figure 14.12 Phase-plane analysis of the system in (10.21). In contrast to the earlier systems, the nullclines in this case are nonlinear, thereby allowing two interior steady-state points. (A) shows the nullclines (including the axes) and the steady-state points (yellow). (B) depicts the vector field of the system, which summarizes the flow of all trajectories. A sample of trajectories is shown in (C), together with a separatrix (dashed), which separates trajectories ending either at (0, 50) or at (14.215, 11.115). (D) superimposes (A), (B), and (C) for a complete qualitative picture. Note that all trajectories transect the nullclines either in horizontal or vertical direction. The steady state at (7.035, 15.60) is a saddle point. Moving exactly on the separatrix toward this point would end in this point. However, deviating even slightly to the left or right of the separatrix pushes any trajectory further away.

state $(N_1, N_2) \approx (14.215, 11.115)$. The separatrix runs through the saddle-point steady state at (7.035, 15.60) and into the unstable point (0, 0). Of note is that every even so slight perturbation away from (0, 0) can lead to the extinction of either N_1 or of N_2 or to coexistence. This observation highlights the existential risks of small populations.

The simultaneous existence of alternative stable steady states and a separatrix is not just a mathematical curiosity. Varga et al. [28], for example, observed that the same mixtures of bacterial populations under antibiotic treatment can over time migrate to either one of two different steady states where one or the other set of populations survived.

It should be clear that all models discussed before are very much simplified and sometimes distant from reality. For instance, predators very rarely rely on a single prey species and are instead participants in complex food webs where many predator species feed on many prey species [29]. Other complications in population modeling result from complex dependences, such as the exclusive pollination of plants by specific insect species. Another issue of great import is the system dynamics if populations are very small. In a differential equation system, even 0.1 individuals of prey may recover and later provide food for the predators. In reality, the species would die out. Furthermore, all systems in nature are subject to environmental influences, some of which can greatly stabilize or destabilize the system. For instance, refuges for prey can drastically change the simple dynamics exhibited in Figure 14.11. Finally, as soon as the spatial expansion of a population becomes important, as is prominent in the spread of epidemics, one must resort to entirely different models, such as partial differential systems or agent-based models (Chapter 11). Nevertheless, simplified population models often distil the essence out of more realistic models and are therefore valuable and easier to understand. The following section describes some efforts toward more complicated situations.

14.4.3 Mixed Communities of Populations

No organism or population lives in isolation. We had already mentioned competition between two species, but that discussion was grossly simplified. Realistic environments contain mixed communities, sometimes consisting of thousands of populations, each of a different species, that live in the same environment and interact with each other. The diversity of these communities is widely known for biotopes like a tropical rainforest, but we don't have to go that far: A single handful of garden soil may contain a staggering number of different microbial species or OTUs (operational taxonomic units), along with fungi, nematodes, and other critters. A single gram of soil can contain enough microbial DNA to stretch almost 1,000 miles [30]. Even closer to home is our own body, which contains at the order of 40 trillion bacteria [31]. Different constellations of bacterial populations live on the skin, in different sections of the oral cavity, in the vagina, and essentially on every bodily surface. Huge numbers of bacteria live in the gut, where their density can reach many billion cells per milliliter. This number is enormous, but the bacteria are so small in size that the gut **microbiome** makes up only between one and two percent of a human's body mass. The human gut microbiome can easily contain hundreds or even thousands of different species, and the totality of different genes in this mixed community probably exceeds our own genome size by several orders of magnitude. Furthermore, the composition of the gut microbiome changes dynamically and drastically in response to age, diet, diseases, and, of course, antibiotics [32]. Common wisdom used to suggest that bacteria were enemies to be defeated and, of course, microbial communities may harm our bodies in various ways, as it is in the case if inflammation turns into sepsis. However, it has slowly become crystal clear that bacteria are indispensable for our health and well-being, for instance, by aiding our digestion. Indeed, problems with the gut microbiome have been related not only to digestive problems but even to depression and anxiety [33]. In contrast to our genome, which is static, microbiomes are very dynamic, as they respond to numerous perturbations from dietary changes to antimicrobial actions and replicate and mutate very quickly [34].

A deep understanding of mixed microbial communities is a fundamental requirement not only for maintaining or improving health and combatting disease but also for comprehending the uncounted effects of microbes on agriculture, forestry, and sustainable environments. As just one example, it is well known that termites are essential for eliminating wood debris and that the earth would suffocate without them. However, it is not really the termites who do the job but complex bacterial communities in their guts [35]. All in all, it has been estimated that 5×10^{30} prokaryotic cells inhabit the earth [36].

Microbial communities can form **biofilms** which, if given enough time, cover essentially all moist surfaces, from teeth to the insides of sewage pipes (**Figure 14.13**). Like gut microbiomes, biofilms may consist of hundreds or thousands of different species, and waste products from one species are used as nutrients by others. Intriguingly, the different species in mixed communities—and the individual organisms within these species—compete for the same physical space and the same resources, yet they depend on each other and often demonstrate a strict division of labor. Thus, in contrast to the typical competition in shared environments, different microbes assume distinctly different and complementary roles, to a point where a species does not survive without the others [32].

Understanding microbial communities faces significant challenges, the most restrictive of which might be the fact that the vast majority of species in these populations is not known. They cannot even be cultured in the lab and are therefore difficult to characterize. And even if a few representative species can be grown in the lab, their artificial environment is so deprived in comparison with their natural surroundings that the extrinsic validity of subsequent results often seems doubtful.

An interesting strategy for overcoming this problem was the introduction of **metagenomics** [37]. Here, samples of entire microbial ecosystems are taken directly from their natural environment and, without any attempt to culture or separate them, analyzed as if they belonged to a single species. While this procedure is now technically feasible, the challenge shifts to extracting reliable information from the mixture of millions of new genomic sequences that in some unknown

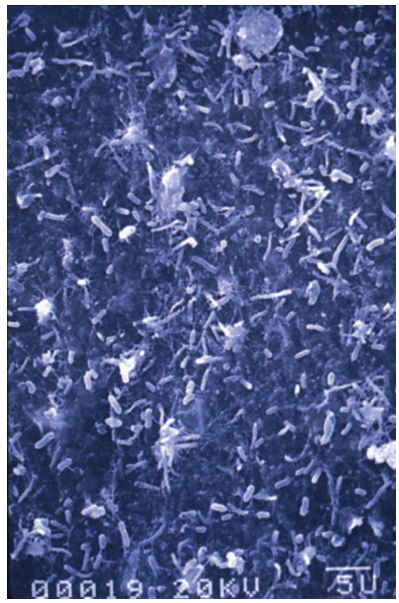


Figure 14.13 Scanning electron microscope image of a microbial biofilm. This mixed community developed on the inside of untreated water tubing within some dental equipment. Such water lines often contain bacteria, as well as protozoa and fungi. (From Walker JT & Marsh PD. *J. Dentistry* 35 [2007] 721–730. With permission from Elsevier.)

fashion belong to thousands of species. Needless to say, metagenomics stretches the current methods of bioinformatics to the limit. As a consequence, full assemblies of genomes are presently only possible for mixtures of a few species [36].

In addition to metagenomic studies, techniques are now becoming available for community proteomics, which has the potential of yielding unprecedented insights into the composition and function of mixed microbial communities. These techniques have already been applied to a variety of systems, including sludge, biofilms, soil, and gut microbiomes [38], but as for many other proteomics techniques, the experimental and computational analysis is challenging.

Due to these challenges, mathematical analyses of mixed communities have been relatively rare, but they are increasing in complexity and sophistication, based on advanced data generation [39–41]. In many cases, despite their numerous immediate and indirect caveats, the old tried-and-true Lotka–Volterra (LV) systems have often been used as the mathematical default approach [42–49], although some other avenues have been pursued [50–55]. As discussed in Chapter 5, the typical formulation for an LV system with n variables is

$$\dot{X}_i = a_i X_i \cdot \left(1 + \sum_{j=1}^{n} b_{ij} X_j\right). \tag{14.22}$$

As for the lower-dimensional case of a predator–prey or competition system, each right-hand side of this system of ODEs consists of a linear term and a sum of binary (second-degree) terms, which describe interactions between two species ($i \neq j$) or the crowding term within a species ($i = j$) that we encountered several times in logistic growth functions and competition models. It is also possible to include environmental factors in these types of models, either directly [56, 57] or through an integral term [58].

Higher-dimensional LV systems may have very interesting mathematical features. Surprisingly, despite their rigid structure, they can exhibit deterministic chaos (see Chapter 5). To recall, this term indicates that the system is entirely deterministic, and no randomness is included in the model structure. Nevertheless, the system oscillates in complicated patterns that are not predictable without the model and never repeat themselves. As a numerical example, consider a four-variable LV system of the type (14.22), proposed by Sprott and colleagues [59, 60], with the parameters in **Table 14.2**. It exhibits very complicated, **chaotic** dynamics (**Figure 14.14**).

While higher numbers of variables often make systems more complicated, other factors can contribute to the complexity of a model. For instance, some or all parameters could be made time-dependent. As so often, there is hardly any limit to the range of possibilities. An important question is the nature of the interactions between populations. In LV-systems, these always involve two populations, but it is easy to imagine that a third population, or some environmental condition, could affect some of the interactions [28, 61]. As an example, imagine a mixed bacterial community exposed to an antibiotic. Over time, the antibiotic may be degraded by some of the species or lose efficacy otherwise. If so, it would be reasonable to represent it as a dependent variable. It would be straightforward to formulate an equation for its dynamics in the LV system, but as soon as one wanted to account for its effect on the interactions between species, the LV format would become problematic. This challenge is presently under investigation. One mathematical extension is the generalization of the LV format to more than two interactants, possibly with exponents to characterize their contributions to the dynamics. The natural

TABLE 14.2: PARAMETER VALUES FOR THE LOTKA–VOLTERRA SYSTEM (14.22) PLOTTED IN FIGURE 14.12

a_1	1	b_{11}	−1	b_{21}	0	b_{31}	−2.33	b_{41}	−1.21	$X_1(0)$	1
a_2	0.72	b_{12}	−1.09	b_{22}	−1	b_{32}	0	b_{42}	−0.51	$X_2(0)$	1
a_3	1.53	b_{13}	−1.52	b_{23}	−0.44	b_{33}	−1	b_{43}	−0.35	$X_3(0)$	1
a_4	1.27	b_{14}	0	b_{24}	−1.36	b_{34}	−0.47	b_{44}	−1	$X_4(0)$	1

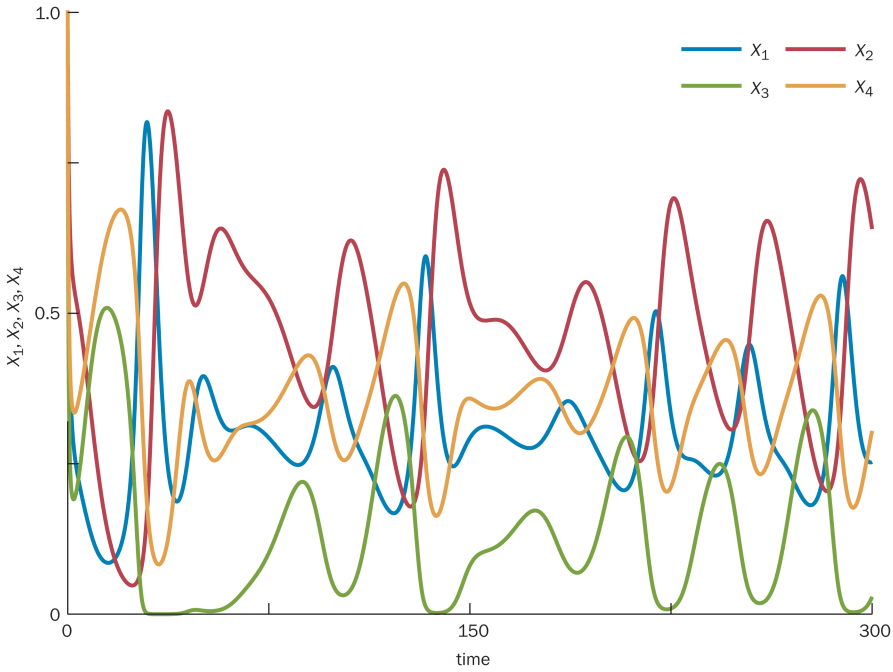


Figure 14.14 Even small dynamic systems may exhibit deterministic chaos. The form of a Lotka–Volterra model (14.22) is deceptively simple. In reality, such a system may exhibit very complicated dynamics, such as the chaotic behavior of the four-variable system shown here. (Adapted from Sprott JC, Vano JA, Wildenberg JC, et al. *Phys. Lett. A* 335 [2005] 207–212, with permission from Elsevier, and from Vano JA, Wildenberg JC, Anderson MB, et al. *Nonlinearity* 19 [2006] 2391–2404. With permission from IOP Publishing.)

result of this extension is a Generalized Mass Action (GMA) system, as we discussed in Chapter 5 (see also [62]).

Almost any change in the LV structure has severe drawbacks. The reason is that the term in parentheses in (14.22) is linear and therefore permits algebraic steady-state analyses and parameter estimation for arbitrarily large systems, which is a huge mathematical advantage that is lost in most generalizations (see Chapters 5 and 6 and [63, 64]).

14.5 EXERCISES

14.1 Using methods discussed in Chapter 6, fit the data from **Table 14.3** (taken from **Figure 14.1**) with Richards and Gompertz functions, as well as with the generalized growth equation proposed by Savageau [9], which has the form

$$\dot{N}_S = \alpha \cdot N_S^g - \beta \cdot N_S^h$$

with positive parameter values. Use fitting methods discussed in Chapter 5.

14.2 Derive the ODE format of the logistic growth law (14.3) from the explicit form in Figure 14.1, including the appropriate initial value. Begin by computing the derivative of the explicit function and then express it in terms of N and suitable parameters.

14.3 For small v, the Richards growth function can be represented approximately as the ODE

$$\dot{N}_R = \alpha N_R \ln\left(\frac{K}{N_R}\right) v.$$

Assess the quality of this approximation for different values of v.

14.4 It is sometimes said that the world population is growing "super-exponentially." Use **Table 14.4**, which was assembled from US Census Bureau datasets, to analyze this claim. Is it possible to model the data satisfactorily with an exponential model? Try to use methods from Chapter 6 to represent the data with one of the growth models described in this chapter. Can you predict from these data what the carrying capacity of the human population on Earth might be?

14.5 Use simulations or stability analysis to find the perturbation threshold above which the population with constant predation, (14.10), recovers and below which it goes extinct. Can you find a similar threshold for the population with proportional predation, (14.9)? Argue whether this is possible or not and/or execute simulations.

TABLE 14.3: GROWTH OF A BACTERIAL POPULATION*

Age of Colony (days)	*Colony Size (area in cm²)*
0	0.24
1	2.78
2	13.53
3	36.3
4	47.5
5	49.4

*Data from Lotka A. Elements of Physical Biology. Williams & Wilkins, 1924 [42].

14.6 Provide arguments and/or simulation results for why small populations are more strongly affected by (random) perturbations than large populations.

14.7 In the context of a Leslie model, discuss why it is important to use age classes of the same length and to match this length with the reporting time period τ.

14.8 For $n = 2$, $n = 3$, and $n = 5$, show that the matrix equation (14.16) generates the same numbers as using (14.15) twice, three times, or five times, respectively. Think before you execute the matrix equation five times.

14.9 Construct cases where the Leslie model shows either growth or decline.

14.10 Construct a Leslie model where the population size is constant but not zero. Is the solution unique? Execute simulations to show how robust this model is to changes in one of the age classes or in one of the parameters.

14.11 Select values for $\alpha_1, \ldots, \alpha_4$ and $\sigma_1, \ldots, \sigma_3$ and compute the total size of the population as a function of time.

14.12 In the Leslie model, make α and/or σ dependent on time. Perform simulations and study the consequences.

14.13 In the Leslie model, make α and/or σ dependent on population size. Perform simulations and study the consequences.

14.14 Population experts claim that delaying the first birth by a few years would have a significant impact on the human population growth explosion in developing countries. Set up a Leslie model to test this claim. If you find it helpful, ignore males. State which assumptions (realistic or unrealistic) you made.

14.15 The roly poly or pillbug, *Armadillidium vulgare* (**Figure 14.15**), is a terrestrial crustacean in the grasslands of California. It can live up to 4 or sometimes even 5 years. The world is a dangerous place for the roly poly: of roughly 10,000 eggs hatched in a year, only about 11% survive to enjoy their first birthday. Of these, about 48.5%are female, of which only about 23% breed. As a consequence, only about 126 breeding females enter the second year to produce offspring. This number continues to decrease, as indicated in **Table 14.5**, which was compiled from data in [65]. Fortunately for the species, the breeding females lay between 25 and 150 eggs per year.

TABLE 14.4: GROWTH OF THE HUMAN WORLD POPULATION

Year	*World Population P*	*log P*
1000	25,400,000	7.404833717
1100	30,100,000	7.478566496
1200	36,000,000	7.556302501
1300	36,000,000	7.556302501
1400	35,000,000	7.544068044
1500	42,500,000	7.62838893
1600	54,500,000	7.736396502
1700	60,000,000	7.77815125
1800	81,300,000	7.910090546
1900	155,000,000	8.190331698
1910	175,000,000	8.243038049
1920	186,000,000	8.269512944
1930	207,000,000	8.315970345
1940	230,000,000	8.361727836
1950	240,000,000	8.380211242
1960	3,042,389,609	9.483214829
1970	3,712,813,618	9.569703148
1980	4,452,686,744	9.648622143
1990	5,288,828,246	9.723359464
2000	6,088,683,554	9.784523403
2010	6,853,019,414	9.835881962
2020	7,597,238,738	9.880655774
2030	8,259,167,105	9.916936253

Figure 14.15 Roly poly. A typical population of the Californian roly poly or pillbug, *Armadillidium vulgare*, exhibits complex birth and death patterns. Courtesy of Franco Folini under the Creative Commons Attribution–Share Alike 3.0 Unported license.

TABLE 14.5: LIFE STATISTICS OF THE ROLY POLY

Age Class (years)	*Number of Eggs or Breeding Females**	*Number of Eggs Produced*
0	10,256	0
1	126	3,470
2	98	4,933
3	22	1,615
4	2	238

*The entry for age class 0 (<1 year old) represents the total number of eggs. Entries for later age classes are breeding females.

Use the information in Table 14.5 and the text to construct a Leslie model. Study whether the population will increase, decrease, or remain constant with these settings. Explore what happens if the survival rate of eggs is either 8% or 15%. Execute simulations in which 10 years of 8% and 10 years of 20% survival occur. Determine whether the order matters mathematically and/or biologically? Discuss and interpret your simulation results.

14.16 Formulate the transition from t to $t+\tau$ in the Leslie model (14.11) with row vectors.

14.17 Make predictions regarding the dynamics of two coexisting bacterial populations that are growing exponentially with different growth rates. To what degree do the initial population sizes matter?

14.18 What happens if the mortality rate of tumor cells in (14.17) is 0? Predict the outcome and confirm or refute the predictions with simulations. Adjust other model parameters if the results appear to be unrealistic. Summarize your findings.

14.19 Study what happens to the phase-plane plot if one multiplies both equations in the population system (14.18) by the same positive factor. Explore the same with a negative factor. In all cases, make predictions, study the phase-plane plot, run simulations, and discuss results and biological relevance.

14.20 Compute steady-state points in the population system (14.18) when either N_1 or N_2 is zero and the other variable is not. Interpret these points in the context of logistic growth functions.

14.21 Use conceptual arguments, rather than a formal mathematical derivation, to discuss the stability of the two steady states in (14.19) when one variable is zero and the other is not.

14.22 Discuss the two nullcline situations not analyzed in the text. Sketch the nullclines and trajectories in each case and make predictions about survival and extinction.

14.23 Select parameter values for the four combinations of nullclines and explore different trajectories with simulations.

14.24 Study what happens in the predator–prey model (14.20) if an external factor alters the value of N or P at time $t = 20$. Discuss your findings.

14.25 Perform a qualitative analysis of the predator–prey model (14.20). Discuss coexistence and extinction.

14.26 Linearize the predator–prey model (14.20) at the trivial and non-trivial steady states and locate their stability patterns on the diagram in Figure 4.17.

14.27 Implement the chaotic Lotka–Volterra system and study the effects of changing the initial value of X_1. Summarize your findings in a report.

14.28 Perform a phase-plane analysis for the following system:

$$\dot{N}_1 = 0.1N_1\left(\frac{100-2N_2-N_1}{15}\right)$$
$$\dot{N}_2 = 0.7N_2\left(\frac{60-N_1-N_2}{45}\right)$$

14.29 Perform a phase-plane analysis for the following system:

$$\dot{N}_1 = 0.6N_1\left(50-N_1-0.833N_2\right)$$
$$\dot{N}_2 = 0.2N_2\left(30-0.75N_1-N_2\right)$$

14.30 Perform a phase-plane analysis for the following system:

$$\dot{N}_1 = 3N_1\left(10-N_2+1.5N_1+0.1N_1^2\right)$$
$$\dot{N}_2 = 0.7N_2\left(20-N_1-N_2\right)$$

14.31 Create a figure like Figure 14.12 for the following system:

$$\dot{N}_1 = 0.3N_1\left(500-10N_2-70N_1+3N_1^2\right)$$
$$\dot{N}_2 = 0.7N_2\left(20-N_1-N_2\right)$$

14.32 Create a figure like Figure 14.12 for the following system:

$$\dot{N}_1 = 2N_1\left(10-N_2+1.5N_1-0.1N_1^2\right)$$
$$\dot{N}_2 = 2N_2\left(50-7N_1-N_2+0.3N_1^2\right)$$

14.33 The Florida Snakehead is an invasive, air-breathing fish that is native to parts of tropical Africa and Asia, occurring in China, Thailand, Sri Lanka, the Philippines, and India. It threatens the native bowfin (or mudfish) in Florida and the Southeast USA. To combat the problem, including the possible extinction of the bowfin, the Florida Fish and Wildlife Conservation Commission launched a program encouraging the catch-and-release for bowfin, but catch-and-keep (eat) snakehead. This program is popular because snakeheads are both big and tasty.

14.33.1 Design a competition model for snakehead (N_1) and bowfin (N_2) in the simple Lotka–Volterra format that was demonstrated in class and PSS. Include fishing for snakehead into the model.

14.33.2 Implement the model in PLAS or MATLAB® with the following parameter values:

Growth rates for N_1, N_2: 0.2, 0.4

Carrying capacities for N_1, N_2: 50, 25

Interaction parameters in equations for N_1, N_2: 0.2, 0.75

14.33.3 Design a phase-plane plot for the scenario without fishing that contains all typical elements: all nullclines, steady-state points (with indication whether they are stable or not), and representative trajectories. Submit a picture of your phase-plane plot. Explore different fishing rates and their effects on coexistence or extinction. Specifically, describe in words and through additional phase-plane plots how the system dynamics changes when fishing is included in the

model. Is coexistence of the two species possible? Is it possible that the bowfin becomes extinct? Is it possible that the snakehead becomes extinct?

14.33.4 Visualize different scenarios with time-course plots as well as vector fields or Monte Carlo simulations showing a sampling of trajectories. Mark the stable steady state point in each case.

14.33.5 Identify ranges for fishing rates that correspond to different outcomes. State for representative fishing rates how many time units (years) it takes until a species becomes extinct (indicated by a population size below 1) or when the system has essentially reached a steady state (simulation comes within 1% of the steady state).

14.34 Around the year 1918, the first black fire ant *Solenopsis richteri* reached the shores of the United States through the port of Mobile in Alabama. In the late 1930s, it was joined by the more aggressive red fire ant, *Solenopsis invicta*. Without significant predators, the two ant species spread quickly, honoring the red ant's Latin species name meaning "undefeated." In 1953, the United States Department of Agriculture found that 102 counties in ten states had been invaded. By 1996, about 300,000 acres throughout the South were infected [66]. How would you design one or more models describing the past spread and predicting future invasion into new territories? What kinds of data would be needed for what kind of model? Search the literature for model studies of the phenomenon.

FURTHER READING

Edelstein-Keshet L. *Mathematical Models in Biology*. McGraw-Hill, New York, NY, 1988.

Hassell MP. *The Dynamics of Competition and Predation*. Edward Arnold, London, UK, 1976.

Hoppensteadt FC. *Mathematical Methods of Population Biology*. Cambridge University Press, Cambridge, UK, 1982.

May RM. *Stability and Complexity in Model Ecosystems*. Princeton University Press, Princeton, NJ, 1973 (reprinted, with a new introduction by the author, 2001).

Murray JD. *Mathematical Biology I: Introduction*, 3rd ed. Springer, New York, NY, 2002.

Murray JD. *Mathematical Biology II: Spatial Models and Biomedical Applications*, Springer, New York, NY, 2003.

Ptashne M. *A Genetic Switch: Phage Lambda Revisited*, 3rd ed. Cold Spring Harbor Laboratory Press, Long Island, NY, 2004.

Renshaw E. *Modelling Biological Populations in Space and Time*. Cambridge University Press, Cambridge, UK, 1991.

Savageau MA. Growth of complex systems can be related to the properties of their underlying determinants. *Proc Natl Acad Sci USA*. 76 (1979) 5413–5417.

REFERENCES

1. Savageau, M.A., Growth of complex systems can be related to the properties of their underlying determinants. *Proc Nat Acad Sci USA,* 1979. **76**(11): p. 5413–5417.
2. Bunt, L.N.H., P.S. Jones, and J.D. Bedient, *The Historical Roots of Elementary Mathematics*. 1976, Englewood Cliffs, NJ: Prentice-Hall, p. 2.
3. Neugebauer, O. and A. Sachs, eds. *Mathematical Cuneiform Texts*. 1945, New Haven, CT: American Oriental Society and American Schools of Oriental Research, p. 35.
4. Segel, L.A., *Modeling Dynamic Phenomena in Molecular and Cellular Biology*. 1984, Cambridge, UK: Cambridge University Press. xx+300.
5. Malthus, T.R., *An Essay on the Principle of Population*. 1798: Anonymously published. (Reprinted by Cosimo Classics, New York, 2007).
6. Verhulst, P.-F., Notice sur la loi que la population poursuit dans son accroissement. *Correspondance mathématique et physique*, 1838. **10**: p. 113–121.
7. Richards, F.J., A flexible growth function for empirical use. *J Exp Bot*, 1959. **10**(2): p. 290–230.
8. Gompertz, B., On the nature of the function expressive of the law of human mortality, and on a new mode of determining the value of life contingencies. *Philos Trans Roy Soc London*, 1825. **115**: p. 513–585.
9. Savageau, M.A., Growth equation: A general equation and a survey of special cases. *Math Biosc*, 1980. **48**: p. 267–278.
10. Voit, E.O., Cell cycles and growth laws: The CCC model. *J Theor Biol*, 1985. **114**(4): p. 589–599.
11. Voit, E.O., Recasting nonlinear models as S-systems. *Mathl Comput Modelling*, 1988. **11**: p. 140–145.
12. Voit, E.O. and G. Dick, Growth of cell populations with arbitrarily distributed cycle durations; I. Basic model. *Mathem Biosci*, 1983. **66**: p. 229–246.
13. Voit, E.O. and G. Dick, Growth of cell populations with arbitrarily distributed cycle durations; II. Extended model for correlated cycle durations of mother and daughter cells. *Mathem Biosci*, 1983. **66**: p. 247–262.
14. Voit, E.O., Dynamics of self-thinning plant stands. *Ann Bot*, 1988. **62**: p. 67–78.
15. White, J., The allometric interpretation of the self-thinning rule. *J Theor Biol*, 1981. **89**: p. 475–500.
16. Bonabeau, E., Agent-based modeling: Methods and techniques for simulating human systems. *Proc Nat Acad Sci USA*, 2002. **14**(99 (Suppl.3)): p. 7280–7287.
17. Macal, C.M. and M. North. Tutorial on agent-based modeling and simulation part 2: How to model with agents, in *2006 Winter Simulation Conference*. 2006, Monterey, CA: IEEE.
18. Richter, O., *Simulation des Verhaltens ökologischer Systeme. Mathematische Methoden und Modelle*. 1985, Weinheim, Germany: VCH Verlagsgesellschaft. IX + 219.
19. Voit, E.O. and D.V. Olivenca, Discrete biochemical systems theory. *Front Mol Biosci*, 2022. **9**: p. 874669.
20. Hethcote, H.W., P. Horby, and P. McIntyre, Using computer simulations to compare pertussis vaccination strategies in Australia. *Vaccine*, 2004. **22**: p. 2181–2191.

21. Gutierrez, J.B., et al., From within host dynamics to the epidemiology of infectious disease: Scientific overview and challenges. *Mathem Biosc*, 2015. **270**: p. 143–155.
22. Leslie, P.H., On the use of matrices in certain population mathematics. *Biometrika*, 1945. **33**: p. 183–212.
23. Pinsky, M.A. and S. Karlin, *An Introduction to Stochastic Modeling*. 4th ed. 2011, Amsterdam: Academic Press.
24. Ross, S.M., *Introduction to Probability Models*. 1985, Orlando, FL: Academic Press.
25. Wilkinson, D.J., *Stochastic Modelling for Systems Biology*. Mathematical and Computational Biology Series. 2006, Boca Raton, FL: CRC Press (Taylor and Francis).
26. Hirsch, M.W. and S. Smale, *Differential Equations, Dynamical Systems, and Linear Algebra*. 1974, Orlando, FL: Academic Press.
27. Kaplan, D. and L. Glass, *Understanding Nonlinear Dynamics*. 1995, New York, NY: Springer-Verlag.
28. Varga, J.J., et al., Antibiotics drive expansion of rare pathogens in a chronic infection microbiome model. *mSphere (Microb Ecol),* 2022. **7**(5).
29. Hui, D., Food web: Concept and applications. *Nat Educ Knowledge*, 2012. **3**(12): p. 6.
30. Trevors, J.T., One gram of soil: A microbial biochemical gene library. *Antonie Van Leeuwenhoek,* 2010. **97**(2): p. 99–106.
31. Sender, R., S. Fuchs, and R. Milo, Revised estimates for the number of human and bacteria cells in the body. *PLoS Biol*, 2016. **14**(8): p. e1002533.
32. Turroni, F., et al., Human gut microbiota and bifidobacteria: From composition to functionality. *Antonie Van Leeuwenhoek*, 2008. **94**(1): p. 35–50.
33. Clapp, M., et al., Gut microbiota's effect on mental health: The gut-brain axis. *Clin Pract*, 2017. **7**(4): p. 987.
34. Gerber, G.K., The dynamic microbiome. *FEBS Lett*, 2014. **588**(22): p. 4131–4139.
35. Ali, H.R.K., N.F. Hemeda, and Y.F. Abdelaliem, Symbiotic cellulolytic bacteria from the gut of the subterranean termite *Psammotermes hypostoma* Desneux and their role in cellulose digestion. *AMB Express*, 2019. **9**(1): p. 111.
36. Wooley, J.C., A. Godzik, and I. Friedberg, A primer on metagenomics. *PLoS Comput Biol*, 2010. **6**(2): p. e1000667.
37. Rondon, M.R., et al., Cloning the soil metagenome: A strategy for accessing the genetic and functional diversity of uncultured microorganisms. *Appl Environ Microbiol*, 2000. **66**(6): p. 2541–2547.
38. Wilmes, P. and P.L. Bond, Microbial community proteomics: Elucidating the catalysts and metabolic mechanisms that drive the Earth's biogeochemical cycles. *Curr Opin Microbiol*, 2009. **12**(3): p. 310–317.
39. Medina, M. and J.L. Sachs, Symbiont genomics, our new tangled bank. *Genomics*, 2010. **95**(3): p. 129–137.
40. Vieites, J.M., et al., Metagenomics approaches in systems microbiology. *FEMS Microbiol Rev*, 2009. **33**(1): p. 236–255.
41. Narayana, J.K., et al., Mathematical-based microbiome analytics for clinical translation. *Comput Struct Biotechnol J*, 2021. **19**: p. 6272–6281.
42. Lotka, A., *Elements of Physical Biology* (reprinted as *Elements of Mathematical Biology*. 1956, New York: Dover). 1924, Baltimore: Williams and Wilkins.
43. Lotka, A.J., Analytical note on certain rhythmic relations in organic systems. *Proc Natl Acad Sci USA*, 1920. **6**(7): p. 410–415.
44. Volterra, V., Fluctuations in the abundance of a species considered mathematically. *Nature*, 1926. **118**: p. 558–560.
45. Mühlbauer, L.K., et al., gauseR: Simple methods for fitting Lotka-Volterra models describing Gause's "Struggle for Existence". *Ecol Evol*, 2020. **10**(23): p. 13275–13283.
46. Wangersky, P.J., Lotka-Volterra population models. *Annu Rev Ecol Syst*, 1978. **9**: p. 189–218.
47. May, R.M., *Stability and Complexity in Model Ecosystems*. 1973, Princeton, NJ: Princeton University Press.
48. Marino, S., et al., Mathematical modeling of primary succession of murine intestinal microbiota. *Proc Natl Acad Sci USA*, 2014. **111**(1): p. 439–444.
49. Xavier, J.B., et al., Mathematical models to study the biology of pathogens and the infectious diseases they cause. *iScience*, 2022. **25**(4): p. 104079.
50. Trosvik, P., N.C. Stenseth, and K. Rudi, Convergent temporal dynamics of the human infant gut microbiota. *ISME J*, 2010. **4**(2): p. 151–158.
51. Wang, J.M., D.J. Fleet, and A. Hertzman, *Gaussian Process Dynamical Models*. 2006, Vancouver: MIT Press, NIPS, p. 1441–1448.
52. Certain, G., F. Barraquand, and A. Gårdmark, How do MAR(1) models cope with hidden nonlinearities in ecological dynamics? *Meth Ecol Evol*, 2018. **9**(9): p. 1975–1995.
53. Ives, A.R., Predicting the response of populations to environmental change. *Ecology*, 1995. **76**(3): p. 926–941.
54. Olivenca, D.V., J.D. Davis, and E.O. Voit, Inference of dynamic interaction networks: A comparison between Lotka-Volterra and multivariate autoregressive models. *Front Bioinform*, 2022. **2**: p. 1021838.
55. Kumar, M., et al., Modelling approaches for studying the microbiome. *Nat Microbiol*, 2019. **4**(8): p. 1253–1267.
56. Dam, P., et al., Dynamic models of the complex microbial metapopulation of Lake Mendota. *Nature PJ Sys Biol Appl*, 2016. **2**: p. 16007.
57. Stein, R.R., et al., Ecological modeling from time-series inference: Insight into dynamics and stability of intestinal microbiota. *PLoS Comput Biol*, 2013. **9**(12).
58. Bucci, V. and J.B. Xavier, Towards predictive models of the human gut microbiome. *J Mol Biol*, 2014. **426**(23): p. 3907–3916.
59. Sprott, J.C., et al., Coexistence and chaos in complex ecologies. *Phys Lett A*, 2005. **335**: p. 207–212.
60. Vano, J.A., et al., Chaos in low-dimensional Lotka–Volterra models of competition. *Nonlinearity*, 2006. **19**: p. 2391–2404.
61. Wucher, B.R., et al., Breakdown of clonal cooperative architecture in multispecies biofilms and the spatial ecology of predation. *Proc Natl Acad Sci USA,* 2023. **120**(6): p. e2212650120.
62. Voit, E.O., Biochemical Systems Theory: A review. *Int Scholarly Res Network (ISRN—Biomathematics)*, 2013. **Article 897658**: p. 1–53.
63. Voit, E.O. and C.-H. Chou, Parameter estimation in canonical biological systems models. *Int J Syst Synth Biol*, 2010. **1**(1): p. 1–19.
64. Davis, J.D., D.V. Olivença, and E.O. Voit, Methods of quantifying interactions among populations using Lotka-Volterra models. *Front Syst Biol*, 2022. **2**: p. 1021897.
65. Paris, O.H. and F.A. Pitelka, Population characteristics of the terrestrial isopod *Armadillidium vulgare* in California grassland. *Ecology*, 1962. **43**(2): p. 229–248.
66. Lockley, T.C., *IPM World Textbook*, http://ipmworld.umn.edu/chapters/lockley.htm, 2004–2007.

15 Emerging Topics in Systems Biology

When you have read this chapter, you should be able to:

- Present an overview of the current state of the art in systems biology
- State some of the pressing challenges for systems biology in the near future
- Explain why some application areas are difficult to analyze and in need of systems analysis
- Explain where current modeling techniques fall short and what the field might do to advance them
- Discuss what a theory of biology would entail

This final chapter discusses where systems biology currently stands and where it might go in the near future. It presents some challenging application areas awaiting systems analysis and lists some of the foreseeable modeling needs and theoretical developments. After having read the chapter, you should be able to:

The famous baseball great Yogi Berra once said: "It's tough to make predictions, especially about the future." Indeed, looking into the future is tricky business. Many a forecaster has met with ridicule once the future arrived and turned out to be drastically different than even the wildest imagination had thought possible. The world has used many, many more than five computers, as allegedly predicted in the early 1950s by Thomas John Watson, the president of IBM. We have not run out of copper, as predicted by the Club of Rome in the 1970s, when the number of telephone lines was rapidly growing, apparently without end, but before optical fibers and cell phones had been invented [1]. People are not flying between high rises with individual jet packs, as has been envisioned in uncounted science fiction movies. "Junk" DNA is not really junk at all [2]. Confounding the challenge of prediction is the experience that truly significant changes are often caused by rare events, which in science are also called black swans, punctuated equilibria, or singularities [3–5].

The future of systems biology is particularly difficult to predict, even for a relatively short time period of the next ten or 20 years, because systems biology is still such a young discipline that any type of extrapolation from history into the unknown should immediately trigger warning signals [6]. If we take molecular biology as an example and try to imagine what forecasts might have looked like 50 years ago, we probably see very quickly how dramatically predictions can go astray. At a time where a doctoral degree was awarded for sequencing a single gene, who could have imagined microarrays with thousands of short DNA sequences,

DOI: 10.1201/9781003402770-15

spotted by robots? Who could have foreseen the incredible advances in molecular and cellular imaging? Who would have thought that it would almost become a routine task to analyze single cells? Who could have imagined today's computational capabilities and the rapidly growing role of machine learning and artificial intelligence? So, with respect to systems biology, it seems that we can only be certain that there is not much certainty when it comes to predicting the face of systems biology in 2100 or even in 2040. In spite of all this uncertainty, it might be useful to speculate about some application areas and methodological trends that are arising on the horizon.

15.1 OVERALL LONG-TERM GOALS OF SYSTEMS BIOLOGY

Systems biology is still in its infancy, but two long-term goals have begun to crystalize, and they point in diametrically opposite directions. The first is the creation of application models that encompass ever greater amounts of data and other information, thereby becoming increasingly more realistic. The first intermediate objective within this pursuit has been the assembly of realistic, reliable whole-cell models, which Masaru Tomita called "a grand challenge for the 21st Century." His group dedicated considerable effort to the development of a software platform, called E-Cell, that permits the user to define large models of processes within cells [7]. Why would those models be of interest, outside bragging rights? Carrera and Covert [8] explain that whole-cell models are beneficial for integrating heterogeneous datasets, identifying gaps in our knowledge, interpreting complex responses of cells, making predictions regarding different phenotypes of a cell, suggesting new experiments, and providing a safe and efficient framework for the design of genetically modified organisms with methods of synthetic biology. It is to be expected that the whole-cell models will be followed by whole-organ and whole-organism models, which eventually should permit reliable simulations of diseases and treatments, as well as systems simulators for agriculture and the management of sustainable environments. Some beginnings of these trends are already recognizable and will be discussed later.

The second long-term goal of systems biology is to understand and rationalize the design and operating principles that we observe time and again in natural systems. Knowing these principles would not only be a great academic achievement, but it would also be of enormous benefit for synthetic biology, because in modifying or creating new organisms, one could avoid structures or processes that nature had identified as inferior over the past 3.8 billion years. So, an important question is: why did nature select some designs and not others? We discussed these principles in Chapter 6 and will not revisit them here.

15.2 EMERGING APPLICATIONS

15.2.1 Models of Cells, Organs, and Organisms

The cell is often considered the smallest autonomous unit of life. After all, it has its own mechanisms of food uptake and excretion and in many cases produces all it needs for survival, parts recycling and replication. These features are most clearly visible in unicellular organisms but they also hold, to some degree, for multicellular organisms where, however, cells tend to specialize and focus on a few functions, while leaving other functions to different cell types. For instance, red blood cells carry oxygen but do not contain mitochondria or DNA and therefore cannot replicate. Then again, all multicellular organisms start with omnipotent stem cells.

Computational models targeting individual cells have been emerging over the past two decades. They pursue two distinct objectives, which both are in their infancy and will almost certainly be the subject of much investigation in the future. These objectives are **whole-cell modeling** and **single-cell analysis**. The following contains brief summaries of what has been accomplished and what is yet to be done.

The long-term goal of whole-cell modeling is the creation of models of single cells that realistically capture all processes occurring in a real cell [7]. That is a tall

order, but systems biology has already begun this pursuit with one of the simplest possible organisms, the bacterium *Mycoplasma genitalium*. Over a decade ago, Markus Covert and his team published a very comprehensive computational model of this organism [9]. It was synthesized from information in over 900 publications and included 28 modules comprising all molecular components and interactions that are necessary for survival. This groundbreaking model contained almost 2,000 experimentally measured parameters and was validated against a host of experimental data. In addition to reflecting known processes quite well, the model predicted some previously unknown processes associated with DNA replication. In spite of the large numbers of components and processes, the authors cautioned that the model is only a "first draft and that extensive effort is required before the model can be considered complete."

Additional contributions to the field followed soon; reviews include [10–13]. Covert's group [14, 15] developed and analyzed an expanded whole-cell model of *E. coli* that accounts for cell responses to changes in the environment. Atkinson et al. [16] applied whole-cell modeling to predict division of labor between two microbial strains that use agricultural waste as a low-cost feedstock. Thornberg and colleagues [17] proposed a 3D model that contained deterministic and stochastic components to connect genetic information with metabolism and cell growth. Various authors [18, 19] addressed the challenges of constructing real-life whole-cell models from experimental data, outlining the steps necessary to design such models, proposing computational and experimental methods of computer-aided design specifically adapted to synthetic biology [12] and outlining the challenges of the unsurprisingly enormous task of reliable parameter estimation [20]. The result of these efforts is a basis for whole-cell modeling, but the field is still wide open and awaits investigation in uncounted directions.

It is almost certain that the whole-cell efforts will be advanced and extended toward models of organs and, eventually, whole organisms. A prominent target is the creation of "medical digital twins," which are computer systems whose responses to perturbations resemble human responses so closely that they may eventually be able to help improve the diagnosis of diseases, prognostics, and personalized treatment for a wide range of medical conditions [21]. Current computational platforms addressing whole organs and organisms include the Virtual Metabolic Human database, which was connected to COBRA and FA platforms [22], the Physiome project [23], the Virtual Liver [24] and Virtual Brain project [25], and the Virtual Rat project [26]. As more and more reliable models of organs and organisms become available, they may be employed as disease simulators that will be used for training, diagnostics, and computer-generated suggestions for likely efficacious treatments. Early versions of such simulators already exist [27–29] and companies such as Applied Biomath [30] have been selling computer simulations of virtual patients and PK/PD models to the pharmaceutical industry for some while.

Outside human health and disease, biological systems simulators have been under development for agricultural purposes and to understand plant diseases. As an example, the Soybean Growth Simulation Model SoySim [31] simulates the growth of soybean plants under user-supplied input conditions, including weather, geographical location, evaporation of water from the soil, and water loss from plants through transpiration, as well as daily measurements of minimal and maximal temperatures, solar radiation, air humidity, rainfall, and details regarding crop management. SoySim integrates this information and predicts dates of the various developmental stages of the growing plants, seed development, yield potential, and water use and makes recommendations for future management.

More generic is windows intuitive model of vegetation response to atmosphere and climate change (WIMOVAC) [32], which is based on known biochemical and biophysical mechanisms of photosynthesis and simulates the responses of vegetation to environmental alterations or changes in light, temperature, CO_2, humidity, and other factors and provides an intuitive teaching and analysis tool for crop development. The software is linked to a database of parameter values that permit simulations of different plant species and genotypes without the user having to change the computer code.

Complementing these simulators for normal growth conditions are simulators of plant diseases. An example is LATEBLIGHT, which simulates the growth of the

water mold *Phytophthora infestans* on potato foliage in response to weather, host growth, host resistance, and fungicide use [33, 34].

15.2.2 Single-Cell Analysis

A very different focus on individual cells has developed into a whole new scientific field: single-cell analysis. The experimental basis is primarily the recent feasibility of characterizing the transcriptome of individual cells with sophisticated biomedical technologies. Some while ago, it had been assumed—quite naively by modern standards—that, for instance, breast cancer is breast cancer. Alas, during the past two decades, it became increasingly troubling that some breast cancers responded to some drugs, whereas other breast cancers, which appeared to be of the same type, did not. These insights led to the realization that tumors and other cellular assemblies more often than not consist of different cell types, all with their own features. These insights suggested that bulk experiments could be misleading [35].

New single-cell RNA sequencing (scRNA-seq) technologies were eventually accompanied with corresponding methods of extracting more granular information about epigenetic and proteomic features in individual cells [36–38]. In fact, it gradually became feasible to characterize the genome sequence, DNA methylation state, chromatin accessibility, histone modifications, and intracellular and surface bound proteins of an individual cell. All this information had to be integrated to yield reliable, comprehensive insights, which in turn spawned the need for novel computational approaches.

In a different pursuit, the field of immunology has been focusing on single-cell technologies, which became feasible due to the recent advances in characterizing the diversity and dynamics of T-cell and B-cell repertoires by single-cell sequencing of T-cell receptors (TCRs) and B-cell receptors (BCRs). It is now possible to monitor changes in the adaptive immune system, such as clonal expansion and contraction, over the course of an infection.

So far, most computational tasks in this context have been addressed with methods of statistics, machine learning, and bioinformatics, but it is only a matter of time before causal, dynamical systems models will be needed. As just one example of this line of thinking, differential equation methods were proposed to infer the progression of signaling processes by concatenating multiplexed molecular profiles from thousands of noisy single-cell snapshot data into compounded time series [39–42]. The potential of this methodology can be seen in its application to the so-called *epithelial mesenchymal transition* (EMT), a very important cellular program implicated in increased tumor invasiveness, heterogeneity, and drug resistance. It had been assumed that this transition required the rewiring of the governing signaling network, but a properly executed dynamic analysis of single-cell data demonstrated that a single reaction-based pathway model with constant structure and near-constant parameters was sufficient to represent differences in signaling throughout EMT [41].

The realization of different cell types within a tumor directly led to further questions of complexity. Is the composition of the mixed cell community, say in breast cancer, always more or less the same or is there a lot of variation? Do the participating cell types cooperate in some manner, for instance, through division of labor? Do they compete for the same resources? Will some of them eventually disappear or evolve based upon their microenvironment? Computational analysis of mixed communities might help answer some of these questions. For instance, if one would model the coexisting cell populations with a Lotka–Volterra model (Chapters 5 and 14), inferences of the signs of the interaction parameters could determine their type of coexistence [43]. In a different pursuit of complexity, experts in ecological systems and community dynamics have turned their attention to applying game theory of cooperation and cheating to these heterogeneous subpopulations within tumors [44, 45].

The growing body of information from experimental single-cell analyses has led to the idea of cataloging these data under a variety of experimental conditions. An early result was a draft *Human Cell Atlas* that defines all human cell types in terms of their distinctive molecular profiles and connects them with more

traditional features like morphology and location within the body [46]. Similar projects are envisioned for other model organisms. The information from the Human Cell Atlas is furthermore being extended toward "cell atlases as roadmaps in health and disease" [47]. Each atlas in this collection will identify all human cell types in terms of their specific molecular profiles and connect this information with other cellular features and physiological phenotypes. Application of these ideas has already led to the collation of an atlas of human tumors [48] and of Alzheimer's disease [49], among others.

15.2.3 Infection, Inflammation, and Sepsis

By their nature, complex diseases are puzzling. Systems biology has the potential to provide tools that will help us to understand and treat these diseases in the future. An overarching aspect of many different complex diseases is inflammation and the immune response that follows. This unifying theme ties together diseases that at first appear to be very different, including atherosclerosis, rheumatoid arthritis, asthma, cancer, AIDS, metabolic syndrome, Alzheimer's disease, chronically non-healing wounds, and sepsis [50, 51]. Inflammation is furthermore associated with natural aging processes. When seen in such a broad context, inflammation per se is not a detrimental process but the reflection of a complex communication network, with which the organism signals potentially harmful perturbations. Under healthy conditions, this system works remarkably well, but if it becomes unbalanced for an extended period of time, disease is the consequence.

In many cases, inflammation goes hand in hand with the body's immediate response to invading parasites, governed by the immune system. Indeed, mammals, including humans, possess not one but two systems that protect them from infecting pathogens: the **innate** and the **adaptive immune system**. As a very superficial distinction, the former recognizes unique molecular features of pathogens or foreign compounds with specific receptors and responds very quickly by activating countermeasures [52]. If the innate response is unable to contain the infection, the adaptive immune system is called up. This system responds on a time scale of days and ultimately equips the organism with a memory of earlier pathogen infections and immune responses. Although the adaptive system operates at a much slower speed, it directly interacts with the innate system: T-cells of the adaptive response and macrophages of the innate immune system are part of a positive feed-forward loop, in which they activate each other through soluble signaling proteins, called **cytokines**. The innate immune cells are needed to trigger responses in the adaptive system, while the activated adaptive immune cells effect the amplification of anti-pathogenic responses in the innate cells [53]. Positive control loops are genuinely dangerous, because they run the risk of spinning out of control. In the body, this risk is internally controlled by regulatory T-cells that normally suppress the innate response in order to preclude excessive responses.

To make matters more complicated, several cytokine receptors rely upon a shared subunit, known as the *common gamma chain*, in addition to a ligand-specific subunit that detects the presence of a cytokine. Systems modeling has helped understand how T-cells respond to circulating cytokine signals in the presence of multiple interleukins (IL-4, IL-7, IL-21) that might result in different instructions for adaptive immunity. Modeling the sequestration and competition for binding between the common gamma chain receptors revealed a hierarchy of responses and hidden crosstalk between signaling pathways that helped explain T-cell behavior in the presence of cytokines [54]. At present, we are only starting to understand the complexity of communication that occurs between cells by combinations of these soluble factors.

Infection is often the first trigger for calling up the immune system, which remedies the situation. While this system works splendidly in most cases of infections, it can also go awry. Indeed, the response of the immune system itself can cause damage and inflammation in associated tissues. It is possible that most of the infecting pathogens are killed but that a vicious cycle continues where tissue damage causes more inflammation, which in turn causes more damage. Again, this positive feedback is driven by cytokines and immune cells [50]. It can derail if one

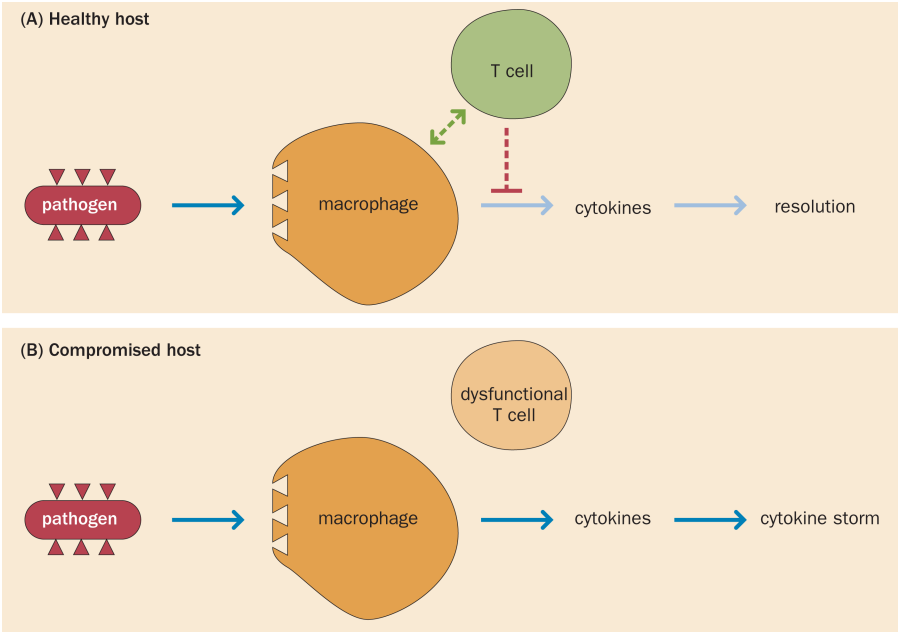


Figure 15.1 A brewing storm? In healthy hosts (A), T cells of the adaptive immune system interact with macrophages of the innate immune system and probably suppress the production of cytokines. In compromised hosts with insufficient functional T cells (B), the production of cytokines is uncontrolled and can result in a lethal cytokine storm. (Adapted from Palm NW & Medzhitov R. *Nat. Med.* 13 [2007] 1142–1144. With permission from Elsevier.)

of the many system components is compromised. A specific example is a CD4 T-cell count that is so low that the regulation of the innate immune system is lacking (Figure 15.1). If so, it is possible that inflammatory cytokines dramatically accumulate into a **cytokine storm**, which can be strong enough to kill the host through tissue damage and septic shock [55]. A cytokine storm in the lung, for instance, can become severe enough to block the airways through the accumulation of fluids and immune cells. Under normal conditions, cytokines move through the circulatory system in picomolar concentrations. However, these levels can increase 1,000-fold during infection or trauma, especially in otherwise young, healthy individuals. There is speculation that the high number of deaths among young adults during the 1918 influenza pandemic and during the 2005 bird flu might have been due to cytokine storms, and the same might have been true for some cases of COVID-19 [56, 57]. All in all, the innate and adaptive immune systems constitute an integrated defense system, where either component depends on the other for amplification and regulation, but this system needs to be controlled just right. If it is not, the body runs into problems.

Not surprisingly, infections and host responses involve uncounted physiological systems in the body, and their complexity and interactions can simply be overwhelming [58, 59]. The challenges for modeling inflammation are therefore manifold. The normal inflammation processes and their regulation are very complicated already, but to confound this challenge, components of the inflammatory response system also interact with non-inflammatory physiologic systems, which makes it difficult to deduce the architecture and control of the system from biological and clinical observations. Furthermore, individual inflammation responses depend critically on the preconditioning state of the organism, that is, on the current health status as well as the health history of the affected organism [51].

A good start toward tackling the situation through modeling is the establishment and ongoing refinement of a "disease map," which can serve—at the very least—as a tool for organizing the information underlying disease progression. Given the complexity of many diseases, the assembly and curation of such a map requires concerted community effort [60].

At this point, models addressing inflammation and the immune response are, broadly categorized, either based on much simplified conceptual diagrams, or they focus on one or two specific aspects, with an account of as much detail as feasible. In comparison with a comprehensive disease map [60], a simple immune response diagram like in Figure 15.1 is obviously vastly over-simplified but, if done right, it can nevertheless be a powerful tool for distilling the essence out of a

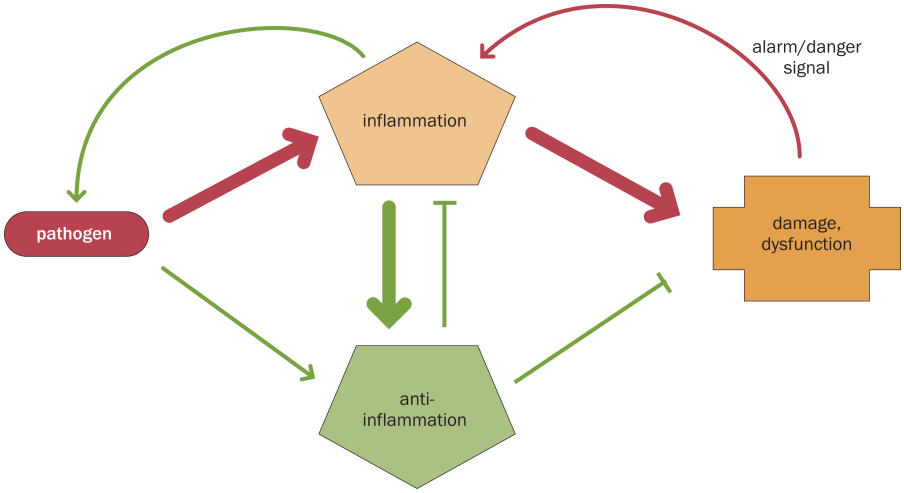


Figure 15.2 Inflammation as an imbalance. This very simplified diagram demonstrates the fine balance between counteracting processes leading either to inflammation or to recovery from an infection (red and green, respectively). (Adapted from Vodovotz Y, Constantine G, Rubin J, et al. *Math. Biosci.* 217 [2009] 1–10. With permission from Elsevier.)

complicated regulatory process. As an example, one may consider inflammatory responses as a homeostatic control system that can yield interesting insights through mathematical analysis [51, 61]. In this vein of simplification, the inflammatory response may be characterized as a balance of processes (Figure 15.2). The simplified representation accommodates several entry points into the acute inflammatory response and furthermore includes a damage signal that can feed back in a detrimental manner. Under normal conditions, a pathogen evokes a strong inflammation response and simultaneously a weaker anti-inflammation response. The inflammation triggers an attack on the pathogen and secondarily activates the anti-inflammation machinery. At the same time, pro-inflammatory agents, such as the tissue necrosis factor , can cause tissue damage or dysfunction. Anti-inflammatory agents such as TGF-β1 suppress inflammation and stimulate healing of damaged tissues. Tissue damage, if not controlled, can lead to further inflammation.

Following the logic of a simplified base structure as in Figure 15.2, it is not too difficult to imagine that the outcome of a pathogen attack is determined by the balance between the different processes in the system. For instance, if the stimulation of the anti-inflammatory machinery is sufficiently strong, inflammation will be suppressed and tissues have a chance to heal: in Figure 15.2, the processes in green overcome the processes in red. By contrast, if the processes shown in red are too powerful, the consequence is not only tissue damage but continuation of a vicious cycle between inflammation and further tissue damage, even after the pathogen has been defeated. The diagram also indicates that inflammation can be triggered by tissue damage, which may be due to trauma or disease.

Slightly more complicated models, specifically for viral infections, were recently reviewed by Zitzmann and Kaderali [59]. They are variations of a scheme that connects healthy and infected cells with the dynamics of virus replication and a very simplistic response by the innate and the adaptive immune system. A generic diagram, rewritten in the fashion proffered in this book, is shown in Figure 15.3.

Of course, conceptual and much simplified numerical models can only go so far. Then again, as soon as the specifics of some part of the immune system are taken into account, models quickly become large. Even restricted to the cellular level, the body produces numerous families of cytokines, including interleukins, chemokines, and interferons [57]. These diverse families form complex interaction networks, and it has only recently been realized that the early distinction between "pro-inflammatory" and "anti-inflammatory" cytokines is really no longer adequate in many instances, because a pro-inflammatory cytokine may trigger the production of an anti-inflammatory cytokine, so that the pro-inflammatory cytokine indirectly acts like an anti-inflammatory cytokine. Cytokine storms have been estimated to involve more than 150 different cytokines, coagulation factors,

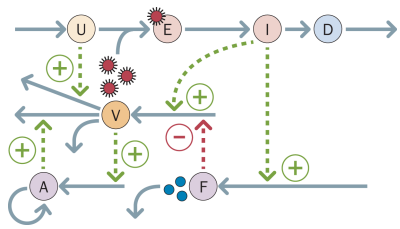


Figure 15.3 A model of virus infection and the body's response [51]. Uninfected, susceptible target cells (*U*) are infected by a virus (with load *V*) and become at first exposed (*E*) and eventually turn into virus-producing infected cells (*I*), some of which die and thus move to the pool *D*. The innate immune system is represented by interferon (*F*), whose production is triggered by the presence of infected cells. *F* impedes the production of virus by the infected cells, whereas the adaptive immune system (*A*), which is activated by the presence of virus, hastens its destruction. Solid arrows represent flow of material, whereas dotted arrows show signaling processes. The model was analyzed in detail by Handel et al. [62].

and radical oxygen species, and the ultimate response to an alteration of one of the cytokine species within such a complex network depends on much more than the type of cytokine. In fact, the preconditioning history of the network is of extreme importance in this situation [51]. Furthermore, cells of the innate immune system initiate both innate and adaptive immune responses [53] through the release of cytokines, which makes intuitive predictions of responses very difficult. To make matters even more complicated, there is plenty of cross-talk among different cell types and their signaling activities. These large numbers of different players and interactions clearly suggest that methods of systems biology, maybe combined with machine learning, will be needed to elucidate immune responses and cytokine storms [51].

Distinct approaches have been employed to model inflammatory processes in greater detail (for recent reviews see [51, 63]). They include the whole spectrum from mechanistically based ordinary and partial differential equation models and simulations to different stochastic and hybrid formulations, as well as rule- and agent-based models. A few examples of inflammation analyses are summarized in the next paragraphs.

A mechanistic mathematical model was constructed to elucidate processes of bacterial infection, cytokine dynamics, the acute inflammatory response, global tissue dysfunction, as well as therapeutic intervention [64]. The model was subsequently used to simulate a clinical trial designed to demonstrate the benefits of a therapy with different doses of a neutralizing antibody directed against the tumor necrosis factor TNF in patients with sepsis (see Chapter 13 for comments on virtual clinical trials). The simulation predicted cases with favorable and unfavorable outcome and also revealed why an actual clinical trial had failed and how the trial should have been designed to succeed.

An ODE model with 15 equations was developed to investigate whether different shock states following an infection could be explained by a single inflammatory response system [65]. Supported by different sets of experimental mouse data, the model demonstrated different trajectories of the inflammation process to either resolution or shock, which depended on the individual cellular and molecular state of each mouse. This modeling effort demonstrated three aspects. First, it proved that the complex biology of inflammation can indeed be coarsely captured by a mechanistic model. Second, it demonstrated that the shock response to different physiological challenges critically depends on the preconditioning state of the organism. And finally, the model presented trends in the progression of the inflammation process that were personalized and predictive with respect to survival.

Another example for the usefulness of simple ODEs in inflammation analysis is a model for the receptor mediated inflammation response in humans exposed to toxins that were produced by bacteria inside the body [66]. This model is interesting because it integrates genomic information and extracellular signaling processes into the bacterial infection process. By changing its parameter settings, the single model allows for the characterization of different outcomes, including healthy resolution responses, persistent infection responses, aseptic inflammation, and desensitization.

Agent-based inflammation modeling permits a hierarchical concatenation of interacting sub-models that collectively integrate basic scientific and clinical information on acute inflammation into a modular, multi-scale architecture [67]. In this case, the individual sub-models address pertinent cellular response mechanisms to infection and inflammation; the permeability of gut epithelia, which are the ultimate barrier against microbial pathogens; the inflammatory response of the gut to ischemia; and responses in the lung. The model successfully matched clinical observations on the pathogenesis of multiple organ failure and the underlying cross-talk between gut and lung. Another beautiful application of agent-based modeling is a thorough analysis of the immune response to tuberculosis that considers interactions between bacteria, macrophages, and T-cells in the lung microenvironment [68]. Recent years have seen enormous progress in this field, and we are gradually beginning to understand enough details of the inflammation network to think about initiating even more comprehensive systems biological investigations.

As discussed in Chapter 13, dynamic models of inflammation and sepsis, as well as other diseases, have been personalized and used as the basis for preclinical studies, virtual clinical trials, in-hospital care, and the long-term use of treatment [51].

15.2.4 Cancer

Due to the enormous importance of cancer, diverse models of **carcinogenesis** and **tumorigenesis** (that is, cancer development and tumor formation) have been proposed for a long time. Some of the early approaches still have relevance, at least conceptually. An influential early idea considered the process of carcinogenesis as a multi-hit or **multistage process** (Figure 15.4). The process is thought to begin with normal cells, some of which eventually become premalignant, due to some initiation event. The premalignant cells remain in this stage for a long time, possibly until death by other causes. A second transformation event converts one of the premalignant cells into a malignant cell, which replicates during a progression phase into a tumor. Radiation, exposure to toxicants, as well as a number of other processes can function as possible initiation and transformation events [69]. In this view of carcinogenesis, a single cell moving through all phases is seen as sufficient to lead to a tumor. Early mathematical models assessed the probabilities or rates of transitions within this multistage process and, in addition, allowed for cell replication and death in the various phases [70–72].

A more modern view of carcinogenesis still contains the same phases while recognizing full well that tumor formation is much more complicated [73]. First, it has become clear that cancer cell populations are heterogeneous and contain cells with different degrees of carcinogenic potential, as mentioned before [74, 75]. A surprising consequence is that killing much of the cancer is not necessarily the best strategy, because it is possible that the most virulent cells survive the treatment and subsequently form an even more aggressive tumor. Instead, a better strategy may call for controlling the tumor rather than attempting to remove it [76]. Second, several interesting studies (e.g., [77]) have demonstrated that essentially all middle-aged and older adults harbor microscopic tumors, most of which stay dormant for a long time, if not the entire life. As one illuminating example, relatively few adults are ever diagnosed with thyroid cancer, but up to 65% of adults have lesions [78]. Third, recent studies show that small tumors can have distinctly different fates (orange arrow in Figure 15.4). Some may grow into full-size tumors and may even metastasize, but other will grow and shrink in an oscillatory pattern, some will regress on their own, and some may stay dormant at a very small size that per se is not harmful [75, 79]. What determines these diverging trajectories is not entirely understood, but it seems that a change in the balance between stimulators and inhibitors may "flip a switch" during the progression phase, upon which the tumor starts growing and recruiting blood vessels. This recruitment process, called angiogenesis, apparently constitutes a significant checkpoint in tumor formation [80].

Future carcinogenesis models will probably retain the backbone of the multistage model but incorporate it into systems models that account for what we learn at the molecular level. For instance, these models might consider small RNA regulation (see Chapter 7), immune responses to the emergence of cancer cells, the recruitment of progenitor stem cells, the recruitment of blood vessels via angiogenesis, and the interactions between cancer cells and healthy tissue. It is clear that these models must either be of an integrating multiscale nature or address some aspect that can subsequently be embedded in a larger systems model.

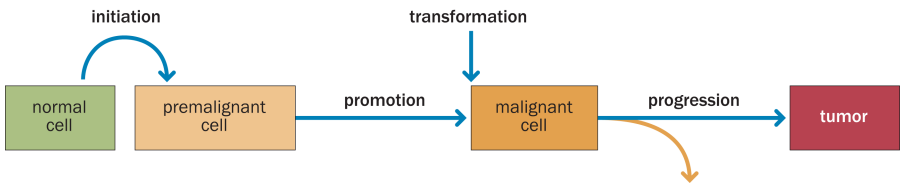


Figure 15.4 Conceptual phases and events during carcinogenesis. Over an extended period of time, a few of very many normal cells become initiated but still appear normal. A transformation event turns one of these cells malignant, and unless the body destroys this cell, it will eventually multiply and grow into a tumor.

The models will be useful for forecasting the dynamics of carcinogenesis and the efficacy of different treatment options. Ideally, they might even reveal overarching principles that differentiate tumor development from normal physiology. The models will complement biological and clinical studies and advance and refine our current understanding of processes leading to cancer and processes that keep cancer at bay [81].

15.2.5 Other Complex Diseases

The more we learn about any disease, the more complicated it becomes. Even diseases with a clear root cause, such as a single gene mutation, turn out to be multifaceted and, over time, often affect many physiological systems within the body. Well-known examples are sickle cell anemia and cystic fibrosis, which eventually manifest across many organs [82, 83]. Other diseases arise from multiple genetic contributions that layer a predisposition with environmental and lifestyle factors. For example, diabetes has relatively clear causes, outcomes, and symptoms, but it is a very complex, systemic disease that may be exacerbated by physiological and pathological responses of other organs and reactions by the immune system. An example is reduced renal clearance, which directly or indirectly affects virtually all aspects of physiology or pathology, beginning with a reduced removal of toxins and changes in blood pressure, volume, and flow, which may in turn affect cardiovascular functioning, easy bruising, and bleeding with a chance of infection, vomiting, headaches, and seizures. Mathematical models of diabetes go back a long way [84] and, coming from humble beginnings, they have reached a level where they are beginning to be predictive. Indeed, simple diabetes models have shed light on the kinetics of insulin in vivo and identified a particular composite model parameter, the disposition index, as representative of the pancreas' ability to compensate for insulin resistance [85].

While diabetes modeling has made great progress [86, 87], there is still a lot to be learned, and as we gather more biological and clinical information, models will become increasingly important tools of integration. The beginnings of this type of integration can be seen in sophisticated physiologically based simulation models of diabetes and metabolic syndrome [88–90] that capture the function of regulatory pathways in these diseases and have been used to connect diabetes to the risk of heart problems [91]. Given that Type 2 diabetes afflicts roughly 20 million patients in the US every year and is a primary cause of heart and kidney disease as well as blindness, the rewards of developing a deeper understanding of the disease and its sequelae are beyond any doubt.

In more generic terms of modeling human development, health and disease, some research teams have begun to consider disease, aging, and frailty as the consequence of a loss of complexity, arguing that some highly periodic processes become diseased if they are too regular and do not exhibit the characteristics of fractals or even chaos [92, 93]. An example is a very regular fetal heartbeat, which very often is cause for concern [94–97].

15.2.6 Neurons, Brains, and Neurodegenerative Disease

According to the Alzheimer's Association [98] roughly 6.7 million people in the US live with Alzheimer's Disease. Parkinson's disease is the second most common, age-related neurodegenerative disease, affecting 3% of individuals over 65 and 5% of people over 85 years of age [99, 100]. Ten percent of people over 65 are estimated to have some form of dementia. Even among younger individuals, neurological issues and mental health are becoming an increasingly troublesome problem. Yet, we are far from understanding the inner workings of the brain and the true etiology of neurological and neurodegenerative diseases.

Medical researchers, experimental biologists, and mathematical modelers have long been fascinated with nerves and brains. Intriguingly, individual neurons are not all that difficult to understand, and we also know a lot about the human brain in terms of anatomy and physiology. But in spite of intense research efforts, we still don't know how our memory works or what determines whether a person will be afflicted with Alzheimer's disease later in life.

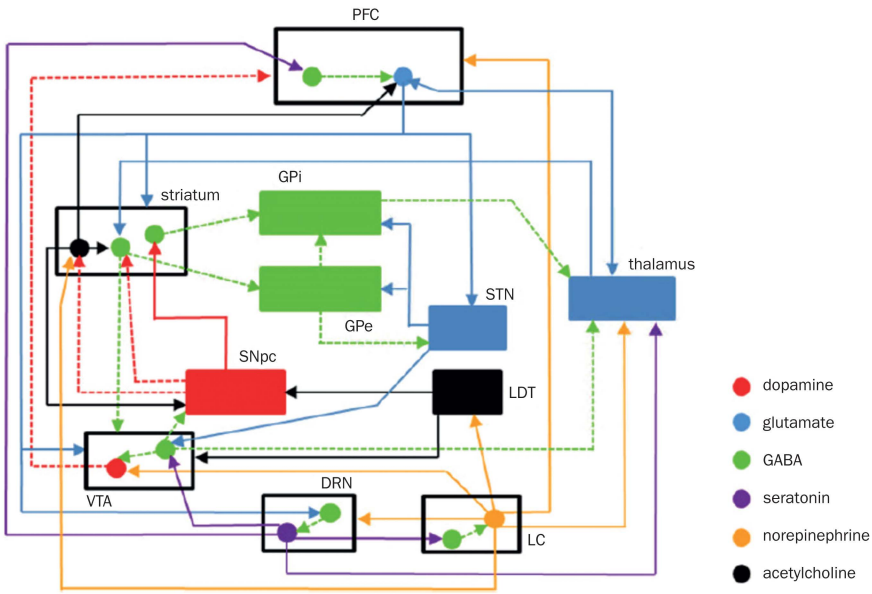


Figure 15.5 Balances among six neurotransmitter systems in multiple regions across the human brain. Colors of arrows represent different neurotransmitter projections. Solid and dashed arrows represent activation and inhibition, respectively. Arrows pointing to a circle show projections onto a specific type of neurons. Arrows pointing to boxes represent projections affecting all types of neurons or an entire region. The affected brain regions include: PFC: prefrontal cortex; GPi: globus pallidus internus of the striatum; GPe: globus pallidus externus; STN: subthalamic nucleus; SNpc: substantia nigra pars compacta in the midbrain; LDT: laterodorsal tegmental nucleus; VTA: ventral tegmental area; DRN: dorsal raphe nucleus; LC: locus coeruleus. From [113].

The field of neuroscience is obviously huge, and we will focus here exclusively on some computational systems modeling. The first major advance was the pioneering modeling work of Hodgkin and Huxley on action potentials in squid neurons, for which they received the Nobel prize (see [101] and Chapter 12). Outside the characterization of electrical features of neuronal signal transmission, most modeling approaches in neurobiology addressed either small systems in detail or large systems in a much coarser fashion [102, 103]. Uncounted variations on models of action potentials, individual cells, and synapses have been proposed, and careful mathematical analyses have been able to explain the basis of their functioning [104–106]; see also Chapter 12. In parallel, biochemical models were developed that describe the dynamics of **neurotransmitters** in individual neurons, as well as the transduction of signals across the synapse between two neurons (e.g., [107–110]). At a higher level of organization, models of neuronal signal transduction focused on the function and interactions of different brain modules but, in general, the more numerous and complicated any modules and their interactions became, the less detail could be included with respect to individual neurons and on intracellular events that are the basis for neuronal signaling [111].

Naturally, models of increased complexity that account for much more detail are expected to permit more realistic analyses than minimal models. Yet, even very simplified models can have strong appeal and explanatory power. As an example, Qi and colleagues proposed a model for alcohol dependence based on a simple mobile of a few interacting neurotransmitter balances [112]. A similar compromise between complexity and granularity was the basis of a model describing certain aspects of schizophrenia, in which the actions of neurons were entirely replaced by chemical balances among neurotransmitters in different brain regions (Figure 15.5) [113]. At this intermediate level of complexity, the model performed remarkably well.

The extreme in simplification is a neurologically inspired construct, called an artificial neural network (ANN). In such a network, neurons are arranged in a few layers and coded exclusively by their activation state as **on** or **off**, 1 or 0, and this state is transduced throughout the network in a discrete fashion. Namely, at each step, the 1's and 0's reaching each neuron at a given layer are multiplied with numerical weights, and the sum of the weighted inputs determines whether this neuron will be on or off in the next layer. Typically, an ANN contains an input and an output layer, as well as one or more "hidden" layers (Figure 15.6). Information regarding the output is fed back to the earlier layers during training of the ANN. This drastic simplification of a realistic network of biological neurons has the advantage that huge arrays of thousands of artificial neurons can be studied very effectively. Intriguingly, it was shown that, over time, sufficiently large ANNs can

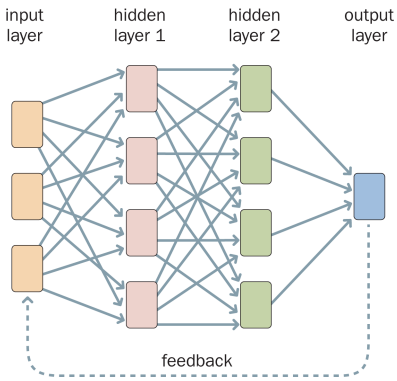


Figure 15.6 Basic structure of a typical artificial neural network. The signals between layers depend on the status of each neuron and on weights that are multiplied to them and iteratively changed during the training phase.

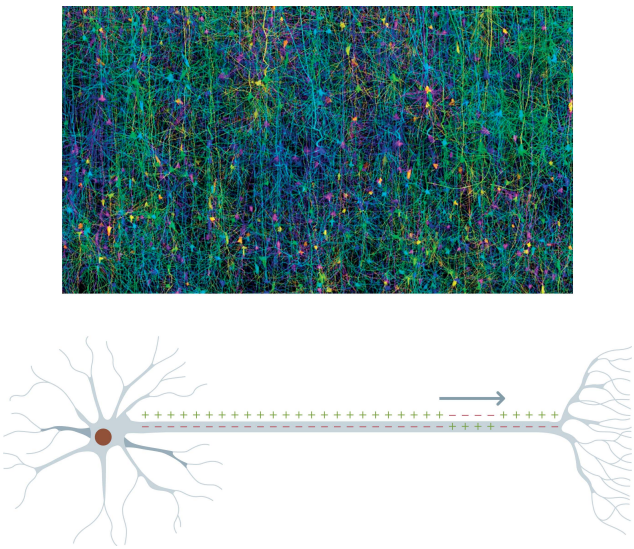


Figure 15.7 31,000 interacting neurons with 37 million synapses modelled on a supercomputer. Result of a Blue-Brain simulation showing the location of axons, dendrites, and synapses within a sand grain-sized chunk of a rat brain. (Reproduced with permission from [119]. From Blue Brain Project/EPFL. Licensed under CC BY 4.0.)

be trained to recognize any patterns by iteratively changing the weights with which each neuron processes its inputs [114, 115]. As a result, ANNs are sometimes called *universal functions* because they are capable of representing an unlimited spectrum of responses, and extensive research has led to applications of ANNs to all kinds of classification and optimization tasks that no longer have anything to do with neuroscience. For instance, ANNs can be trained to find and read zip codes on envelopes and efficiently distinguish a sloppily written 7 from a 1. Research on ANNs and their successors has been extremely successful in machine learning and artificial intelligence, and the future is very likely to see increasingly more sophisticated applications [116].

At the other end of the spectrum between simplicity and complexity, the culmination of simulating realistic interaction networks of neurons is presently the Blue Brain Project [117], for which a large group of scientists has been collecting detailed micro-anatomical, genetic, and electrical data of specific brain sections, such as the human neocortical column. The algorithm integrates this information into three-dimensional simulation models that account for very large populations of over 200 types of neurons (**Figure 15.7**). The sheer size of the project requires enormous supercomputing power, and the results are fascinating. A relatively new effort along these lines is the Human Connectome project [118]. A major issue with these complex approaches is that the simulation results have become so complicated that their analysis is a significant challenge in itself.

The different approaches toward understanding the brain highlight distinct aspects of neurons and neural networks and ignore others. The grand challenge for neuro-systems biology is now the creation of conceptual and computational multi-scale models that integrate electrical, biochemical, genetic, anatomical, developmental, and environmental aspects associated with brains in a manner that offers deep insights into fundamental functions like learning, memory, and adaptation, as well as the development and progression of neurodegenerative diseases and addiction [120]. The first step will be bridging the gap from single neuron models to detailed, realistically sized neural networks. This step is by no means trivial because it is known that neural networks in brains are not only very complex in structure but can dynamically change in activity, for instance, between wakefulness and sleep. Switches between these states are often rapid, and the rules guiding them are largely unknown [121]. Initial attempts in this direction are models that study bursting patterns in networks of maybe a dozen to one hundred neurons [122]. A complementary approach is the creation of computer languages that are custom-tailored to analyses of how anatomical features, synapses, ion channels, and electrochemical processes govern the functionality of the brain [107].

A good starting point for studying real-world neural networks has been the relatively simple, fascinating worm *Caenorhabditis elegans*, which consists of exactly 959 somatic cells, of which exactly 302 are neurons (**Figure 15.8**).

Figure 15.8 Accounting for all neurons. *Caenorhabditis elegans* is a simple worm with exactly 959 cells, 302 of which are neurons. (Courtesy of kdfj under the Creative Commons Attribution-Share Alike 3.0 Unported license.)

Uncounted studies have investigated various aspects of this organism and its nervous system [123]. As a case in point, researchers in Japan have created a database of synaptic connectivities in this organism [124] and *C. elegans* has been used for all kinds of studies; a surprising example is a study of nicotine addiction [125]. The little worm has even been to the international space station [126].

A more complex candidate is the California sea slug *Aplysia californica*, which has become a model for neurobiological studies, because it has only about 20,000 neurons but exhibits interesting responses to stimulation like its hallmark response: the ejection of red ink (**Figure 15.9**). Eventually, models of these and more complex neural networks will grow and become reliable enough to permit targeted interrogations and analyses of brains in higher organisms. They will lead to a deeper understanding and might help us explain neurodegenerative disease, addiction, epilepsy, and other menacing diseases, as well as normal, physiological phenomena like intelligence and conscience (e.g., [127]). An exciting domain of research is understanding how natural neural networks in brains can inform the creation of artificial neural networks. As systems models of interconnected neurons develop with increasing complexity and computational processing continues to grow, we can look forward to an age of neuromorphic computing [128], in which bio-inspired neural architectures mimic the plasticity and information processing of living systems.

Figure 15.9 The sea slug *Aplysia californica* has become a model organism for studying learning and memory. Here, the slug releases a cloud of ink to confuse an attacker. (Courtesy of Dibberi under the Creative Commons Attribution-Share Alike 3.0 Unported license.)

15.2.7 Quantitative Systems Pharmacology

We discussed in Chapter 13 how systems modeling is increasingly being used in drug development. Recent years have seen a strong increase in interest in this approach by the pharmaceutical industry. Indeed, an entire field of study has emerged: *Quantitative Systems Pharmacology* (QSP); for an extensive review, see [129]. It develops computational models of health and disease processes within the body, typically based on ODEs, and uses them to test the main effects, safety profiles and side effects of drug candidates. Specifically, these models simulate drug activity as disease-causing perturbations of healthy physiological systems and thereby can elucidate the possible pathological responses of a patient [130]. As discussed in Chapter 13, the prominence of such models was substantially elevated by the FDA Modernization Act 2.0 [131, 132], which allows computational modeling as an alternative to animal testing, at least in certain cases.

QSP is a direct extension of pharmacokinetics (PK), which by and large investigates what the body does to a drug (e.g., storing, degrading to other compounds, excreting) and pharmacodynamics (PD), which investigates what the drug does to the body [133]. An early proponent of the iterative interplay between experiments and mathematical models in QSP was Arthur Guyton, who in the 1970s outlined three important features and benefits of what is now known as QSP [129, 134]. First, not every detail of a health and disease system needs to be included in a model, as long as the overall behavior is appropriately captured. Second, a model allows us to comprehend the complex regulatory structure of a disease and the body's response. And third, a model may be used as a tool for assessing inconsistencies in data coming from different sources. He suggested utilizing models for informing the design of future experiments in an effort to fill gaps in knowledge.

One obvious benefit of QSP models to the pharmaceutical industry is their potential to increase the efficiency of drug development across all therapeutic indications and thereby to save large amounts of money [130]. Furthermore, as we discussed in Chapter 13, dynamic models are well suited for personalization and the design of virtual clinical trials. The field is developing rapidly and the combination of systems biology and PD models will increasingly account for a drug's multiple targets and for the effects of genomic, epigenomic, and posttranslational changes on drug efficacy [135]. Specific recent applications of QSP modeling include an analysis of the effect of different drugs on certain types of breast cancer [136]; a detailed QSP model of sickle cell disease, including inter-personal variations in responses to a promising treatment [129]; and a QSP model of idiopathic pulmonary fibrosis [137]. Models of this type have also been envisioned as tools for translating research from one model species (e.g., mice) to another (e.g., humans) [138].

15.2.8 Organoids for Personalized Medicine and Drug Screening

A relatively new approach related to QSP is the practice of growing tissues in vitro for testing drug toxicity for diverse sets of patients. The approach is facilitated by induced pluripotent stem cell (iPSC) technologies, which have matured to the point where somatic cells from an individual can be reprogrammed into stem cells and then differentiated into various organ tissues. These miniature organs, called **organoids**, can serve as testbeds for screening potential drug responses and toxicity. Specifically, simulating organoids that are derived from different genetic backgrounds and with different transcriptomic profiles helps characterize the heterogeneity of possible drug effects and can be performed far earlier and much more cheaply than expensive clinical trials.

As a different application, coaxing iPSC tissues into appropriate mature organ types is a way of enhancing our knowledge and testing mechanisms of human development [139–142]. For example, one might study how an aggregate of stem cells organizes into the brain cortex and why a single point mutation can disrupt this process [143]. With these platforms, it is also possible to rapidly test tissues derived from many patients and leverage banked iPSCs collected from rare diseases in order to study functional differences (such as electrical firing in the brain) that would not be apparent by studying a population of cultured cells in a dish. Modeling these multicellular processes requires multiscale tools and high-throughput computing to simulate assemblies of 10^5–10^7 dynamically interacting cells.

In addition to studying individual organoid phenotypes, the ability to derive multiple tissues allows in vitro screening for therapies that might impact multiple organs or tissue types. An example is an antibody that might work well against some eye disease but also causes cardiotoxicity. Coupled with microfluidic devices, micro-physiological "human-on-chip" systems permit soluble factors to flow between organoids to mimic inter-organ communication, such as the gut-brain axis or liver-kidney crosstalk. Agencies like NASA are investing in these multi-organ platforms for studying the effects of microgravity and deep space radiation on human physiology. For a comprehensive understanding, there is a need for computational models of these miniaturized systems in terms of directed differentiation into organoid form and function [144], as well as models capturing the dynamic communication between organ modules, for instance, through hormone-based signaling [145]. Eventually, "patient-on-chip" platforms that generate in vitro data for QSP modeling are expected to inform difficult personalized clinical decisions on what treatments may be most effective and at what dose [146].

15.2.9 Organisms and Their Interactions with the Environment

It is complicated enough to design models for systems within individual cells and organisms, but the degree of complexity jumps by another order of magnitude when it is our task to investigate organisms in their environments. Any comprehensive study of this type immediately requires multiple scales in time, space, and organization; deterministic processes mingle with stochastic effects; and data and other pieces of information are so diverse and heterogeneous that it often seems overwhelmingly difficult to merge and integrate them in our minds. It seems again that system approaches will eventually be our best bet.

A particularly grand challenge with unlimited ramifications is a deeper understanding of the oceans. They provide most of the oxygen we breathe and provide the ultimate storage for water and a lot of food. As with many sub-specialties of biology, many aspects of the dynamics of and in oceans has been addressed with reductionistic methods and models of specific details, yet it has so far not been possible to integrate the huge range of spatial, temporal, and organizational scales into reliable computational models. Nonetheless, we can see rudimentary beginnings of systemic analyses in many places.

A first step beyond observational and wet sciences is the rise of ecoinformatics, which uses methods of bioinformatics to manage, analyze, and interpret the enormous quantities of data that have relevance for ecological systems and, in particular, our oceans [147]. The challenges are great, not merely because of the sheer

volume of information but also due to the highly heterogeneous nature of this information as well as a reluctance to adopt world-wide data standards and experimental and analytical protocols. Such challenges are currently addressed with the creation of data warehouses, the collection and structuring of "metadata" (information describing the details of datasets), laboratory information management systems (LIMS), data pipelines, customized mark-up languages, specialized ontologies, and semantic web tools, including computer-based reasoning systems. However, efficient storage and access to data alone will not be sufficient. Beyond direct data analyses and automated inferences, computational means of information integration, simulation, and interpretation are needed. In other words, ecoinformatics must be complemented with dynamical systems ecology.

Recognizing this need, the US Department of Energy currently sponsors a program, Systems Biology for Energy and Environment, that identifies "the earth's living systems [as] a potential source of capabilities that we can put to use to meet national challenges" and has the ultimate scientific goal to "understand the principles underlying the structural and functional design of living systems" [148]. The program focuses on energy, environmental remediation, and carbon cycling and sequestration. Within the latter focus area, the role of the oceans clearly plays a crucial role, because marine blue-green algae are responsible for almost half of the global CO_2 fixation. There is little doubt that the well-being of these micro-algae depends directly on nutrients, pH, temperature, salinity, contamination, and the surrounding multi-species communities, but the exact parameters of optimal, suboptimal, and barely tolerable growth conditions are yet unknown.

An example of how molecular and systems biology have begun to reach out to investigations of global ocean processes is a study on single-celled marine blue-green algae of the phylum *cyanobacteria*, which are among the most abundant organisms on earth (**Figure 15.10**). These small creatures are able to convert solar energy into biomass and thereby account for 20 to 30% of the earth's photosynthetic productivity. Furthermore, they recycle about 40% of the entire carbon that cycles between oceans, the atmosphere, the earth's crust, soils, microorganisms, plants, and animals. Researchers at Harvard's Department of Systems Biology found that cyanobacteria of the genus *Synechococcus* fix carbon and convert it into sugar within spatially well-placed internal structures, called carboxysomes [149]. A single protein is responsible for the optimal spatial organization of these carboxysomes, and mutations in the corresponding gene lead to a photosynthetic productivity that is decreased to 50% of the wild type production. Thus, a systemic, causal connection has been elucidated between the action of a specific gene, the protein it codes for, the functionality of a microorganism, and a truly global phenomenon. This causal connection could become the starting point for the design of genetically modified algae or bacteria for the production of hydrogen, with methods of synthetic biology.

A different starting point for systemic studies of oceans is the careful investigation of coexisting marine populations with methods described in Chapter 14 (**Figure 15.11**). More and more detailed data for such studies are being collected

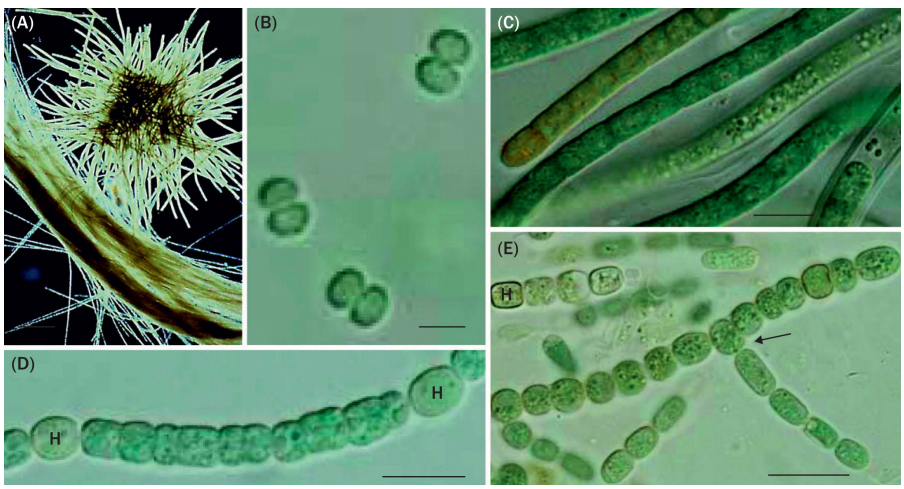


Figure 15.10 Different strains of cyanobacteria (blue–green algae) may have very diverse shapes. (A) Bloom-forming, filamentous, and colony-forming *Trichodesmium thiebautii* (scale bar 100 μm). (B) Unicellular *Synechococcus PCC 6301* (scale bar 1 μm). (C) Filamentous *Symploca PCC 8002* (scale bar 10 μm). (D) Filamentous, nonbranching, and heterocystous (H) *Nostoc PCC 7107* (scale bar 10 μm). (E) Filamentous, heterocystous (H) and branching (arrow) *Fischerella PCC 7521* (scale bar 15 μm). (From Cox PA, Banack SA, Murch SJ, et al. *Proc. Natl Acad. Sci. USA* 102 [2005] 5074–5078. With permission from National Academy of Sciences, USA.)

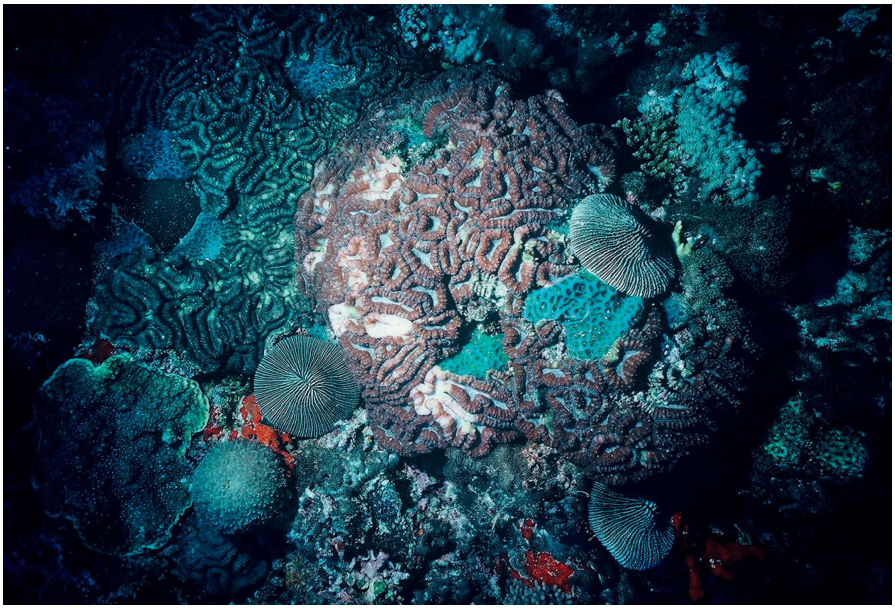


Figure 15.11 Corals and other invertebrates compete for space. Data characterizing interacting populations are currently being collected and create a starting point for systems-biological model analyses. (From Burkepile DE & Hay ME. Coral reefs. In Encyclopedia of Ecology [SE Jorgensen & B Fath, eds], pp 784–796. Elsevier, 2008. With permission from Elsevier.)

world-wide for different marine systems. For instance, Burkepile and Hay [150] measured the effects of single-species and mixed-species herbivorous fish populations on the colonization and progression of algal communities and their impact on coral growth. In carefully controlled experiments with otherwise comparable systems, they demonstrated convincingly that species diversity is critical for the health of coral systems and the resilience of our oceans following disturbances.

Confounding the enormous challenges toward understanding the dynamics of our oceans better is the fact that they are in constant transition, which has become a well-known issue in the context of climate change and human use of the oceans. Thus, the oceans are not only a very complex multi-layered system with dynamics of days or years, but their entire context is also changing on a time scale of centuries. There is no doubt that these changes are of utter importance, but it is also clear that we are far from understanding them.

15.3 MODELING NEEDS

To predict future modeling needs, it might be prudent to revisit the present state of the art, consider how we got here, and then try to extrapolate trends into the coming years. There is no doubt that computational modeling has come a long way. Thirty or 40 years ago, computing happened on main-frame computers that required the manual feeding of punch cards with instruction code. There was no internet, no parallel computing, no user-friendly software. Much has happened since these dark days. Computational power and speed have doubled every 18 months since the 1950s, access to computing has increased millions of times, and today's higher-level languages like MATLAB®, *Mathematica*®, Java, Python, R, SBML, PLAS, and COPASI facilitate the development and implementation of computational ideas among users, most of whom would certainly have been intimidated by the need to master computer languages like Fortran or XML, let alone Assembly code just a few decades ago.

While progress has been amazingly fast, it is clear that we still have a long way to go. But the bottleneck is not just computational speed and convenience [6]. In fact, outside certain areas, such as protein structure and function prediction and very large-scale simulation and optimization studies, it is not necessarily the technical aspects of computing that stand in the way of the next significant advance. New concepts and radically new modeling techniques might be needed. Consider a simple but puzzling example: rainwater runs down a pile of dirt in a very efficient fashion, yet capturing the process in a realistic mathematical model requires enormous effort. Why is that? Spiders with rather small brains are able to construct

complex webs, and animals respond to situations never encountered before with appropriate and effective actions, but we are hardly capable of representing such activities adequately in systems of differential equations. To realize model-aided personalized medicine, it is not sufficient to characterize the direct effects of a disease or treatment but also their secondary and tertiary effects, along with interactions between different treatments and the multitudinous personal characteristics of an individual.

These challenges are indeed great, and it might well be possible that mathematical modeling and systems biology are in need of a significant paradigm shift, comparable to the one that physics experienced it in the early 1900s when quantum theory opened the door to concepts unimaginable just a few decades before. We may need a new way of mathematical thinking, maybe with a new mathematical language, but at this point nobody knows what such a language or way of thinking might look like.

Before we throw our hands in the air and resign ourselves to waiting for a "new math" to appear, before we discard modeling as simplistic and powerless and instead suggest doing another set of lab experiments, let's just see what the alternatives are. It is undisputed that realistic systems consist of hundreds or thousands of important components and that these make modeling very complicated. But the same hundreds or thousands of important components make every other approach even more complicated. Yes, we may not be able to keep track of hundreds of dynamically changing systems components in a computer-aided model analysis right now, but we will *never* be able to manage them in our unaided minds. Statistical association models may be sufficient for identifying averages, but they will not suffice to address specific cases, as they are required in personalized medicine [151]. Beyond considerations of magnitude, many experiments simply cannot be executed because they are technically impossible, practically infeasible, or unethical. We cannot test a new drug on every human, and it is probably not a good idea to release a new strain of algae into the oceans on a trial basis.

Thus, while there is a blatantly clear mismatch between the grand challenges to be solved and the tools we have presently at our disposal, the magnitude of the tasks before us is not a call to pause systems biology until we have all ingredients together but rather to increase our efforts, because we have no choice. We need to search and re-search and re-search again, amassing incremental progress and growing our body of collective experience. These marching orders are hard, but it will be fascinating and fun to witness the field tackle these complicated issues. Step by step, sometimes evolutionary and sometimes revolutionary, sometimes elegantly and sometimes with brute force, maybe devising specialized AI, we will expand and strengthen our arsenal of modeling methods, address small problems, then larger problems, first connect two biological levels and later-on more. As a warm-up, we will study hundreds of isolated processes, then small systems, and ultimately realistic and relevant systems of interacting systems. The journey is open-ended, but it invites us to take the next steps toward a glorious horizon. And while we move along, new methods will allow us to increase speed and efficiency, and we will learn which types of techniques tend to work, which do not, and which are scalable with reasonable efficiency. Eventually, some smart student, uninhibited by tradition, may indeed design an entirely new math. It is moot to speculate what this new math might look like, but it is possible to assemble a partial wish list of topics that need to be addressed.

Interestingly, our current models are often either too simple or not simple enough. If our goal is to explore and manipulate a real-world system in a very specific manner, drastic simplifications are likely to make predictions unreliable. For instance, if the task is to increase product yield in a batch of microorganisms, accounting for as many details as possible is more likely to lead to success than simplifications of the model. Similarly, it seems that the consideration of a larger number of biomarkers will be advantageous in personalized medicine. At the same time, as models become larger and more complex, they become more difficult to analyze, understand, and predict, while simpler models often offer insights that would be masked by too much detail. Thus, starting from the level of complexity of today's models, some models will have to increase in scope and complexity, while others will decrease, depending on the purposes of the modeling effort. And so, the circle closes: the introduction to modeling in Chapter 2 made it clear that

the goals and purposes of a model dictate its structure and features, and this statement still holds. The following sections first discuss extensions of our current modeling techniques and then simplifications that might lead us one day to one or more theories of biology.

15.3.1 Data-Modeling Pipelines

In the past, there was a clear distinction between what biologists do and what mathematical modelers do. While division of labor has been one of humankind's greatest inventions, there is a real risk of losing information and insight when systems biology is strictly separated into experiments and model analyses; just recall the example of six blind people analyzing different parts of an elephant (Chapter 1). As a case in point, the biological literature contains thousands of diagrams showing how various components activate or inhibit some of the processes in a system. As soon as a modeler picks up the diagram and converts it into a model, the biologist's intuition and much undocumented tangential or contextual information are lost, which may cause substantial problems. What will be needed with increasing urgency is a semi-automated, carefully curated pipeline of methods from raw data to dynamic models, which permits the addition of tangential, qualitative information, maybe in the form of metadata. Some efforts in this direction are already underway [152]. A typical pipeline may contain the following phases [6], where Steps 5–11 mirror the steps of optimal model construction introduced in the second half of Chapter 6:

1. Upload typical raw data in an agreed-upon standardized format into a conveniently designed database.
2. Constantly improve methods for efficiently warehousing and accessing data.
3. "Clean" the data by removing errors and flag perceived outliers during preprocessing, based on solid statistical principles.
4. Extract significant information from the datasets, thereby distinguishing signal from noise.
5. Convert simple and complex associations among the data into valid chains of causes and effects, recognizing that feedbacks and other nonlinearities may feed effects back into causes.
6. Create diagrams reflecting the web of causes and effects.
7. Convert the diagrams into symbolic models.
8. Facilitate the choice of appropriate functional representations.
9. Infer values of model parameters from data.
10. Implement the model with parameter values into a computational structure.
11. Analyze this structure with analytical and simulation methods.

Some of these steps may appear to be straightforward but, in reality, pose significant technical challenges [6]. For instance, modern RNAseq and mass spectrometry require a lot of technical decisions before the output data can be fed into the pipeline as Step 1. In addition to bioinformatics and systems biology, the various steps are nowadays being investigated in the new field of data science, which combines expertise from statistics, machine learning, artificial intelligence, and computer science to manage and analyze large amounts of data [153].

A good illustration of a data pipeline arises in the nascent field of **exposome** research [154, 155], whose goal it is to measure and integrate all exposures of an individual—from the time as a fetus until the day of the analysis—and to characterize how these exposures combine to affect health. Obvious exposures are environmental, but they also include lifestyle choices and diet, and their effects are often modulated by an individual's genetic predisposition. Not surprisingly, the data for an exposome analysis can be very diverse in amount, content, accuracy, and format, and may characterize transcriptomics, proteomics, metabolomics, toxicological profiles, environmental monitoring data, remote sensing, local health risks, and many connections in between. Managing and analyzing such

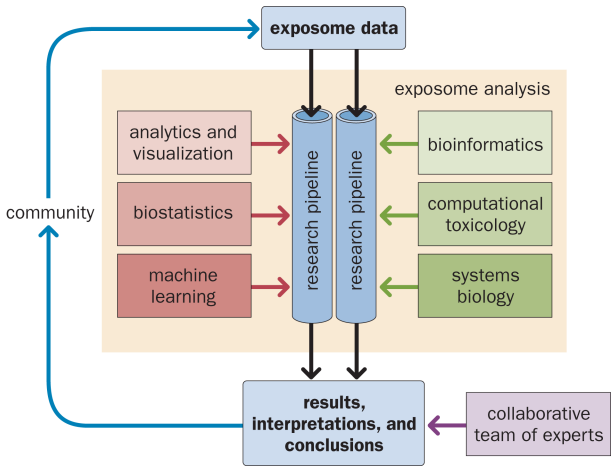


Figure 15.12 Pipeline of competencies involved in a full exposome data analysis. The pipeline here addresses research, teaching, and community involvement (provided by Dr. Lance Waller, Emory University).

heterogeneous data requires collaborative expertise in subject areas such as physiology, toxicology, and the clinical sciences, as well as data science including biostatistics, bioinformatics, informatics, machine learning, computer science, mathematical modeling, and systems biology. In addition, effective exposome research must include stakeholder involvement and regular feedback from the community (Figure 15.12).

The challenge for the foreseeable future will be to automate these phases as much and as quickly as possible, because the available datasets will be growing rapidly in number, size, and complexity and so will the models for their analysis. And who would want to design diagrams and models for hundreds of variables by hand? In fact, automation of mathematical construction, maybe with the help of chatbots or other artificial intelligence tools, should eliminate human errors and become preferable as the number of variable species expands [156]. Caution is advised, though, because every step, if automated, carries a risk of making assumptions that are biologically not tenable or at least sub-optimal or of omitting spurious but ultimately important information. Thus, the entire process will have to be "curated," at first by humans and eventually maybe by supervised artificial intelligence.

While automating the pipeline will be a significant advance, it will also be important to devise the different phases of the pipeline in such a manner that every trained biologist can readily understand them—at least in concept—and is able to intervene in the process. As a different aspect of the same issue, all biologists should be trained at least in the basics of modeling, while ideally having access to professional data scientists and modelers for complicated tasks.

15.3.2 The Role of Machine Learning and Artificial Intelligence in Systems Biology

Machine learning (ML) and artificial intelligence (AI) have been enjoying an astounding rise in prowess and publicity, so we should ask what their role for the future of systems biology might be. Three areas are expected to be supported and advanced by ML and AI.

The first domain is the extraction of information from large datasets, possibly as part of a data pipeline. With the rise of -omics technologies, an enormous amount of data is being generated, and many of these data are widely available in rapidly growing databases. The former MIT President Jerome Wiesner once noted that "Getting an education from MIT is like trying to get a drink from a firehose." Systems biology is in the same situation, mainly because quite a few of the available data are so noisy or unreliable that they are not suited as a solid basis for modeling. However, if trained and applied appropriately, ML and AI can offer very strong tools for distinguishing signal from noise within large datasets, thereby allowing the "cleaning" of raw data and the establishment of reduced yet much more informative datasets.

A promising second application area is the automated design of models, especially if system diagrams are already available or can be designed based on internet information. Here, AI might be able to integrate several available diagrams with additional information to generate a single diagram for human curation and approval, possibly along with a listing of useful information on rates and modulating effects. For setting up equations, there are no strict guidelines, as we discussed earlier, but two options could be implemented to initiate the process. One could use default representations like mass-action, Michaelis–Menten, or Hill rate laws for biochemical pathway systems or logistic functions for growth processes. A powerful alternative would be the use of canonical models which, as we discussed in Chapter 5, are very flexible and, at the same time, constructed according to simple, strict rules. The initial default for models of interacting populations would presumably be the Lotka–Volterra framework, whereas metabolic and physiological system might be best approached with mass action, generalized mass action, or S-system models. Once this choice is made, the actual formulation of equations is straightforward. The models could even be implemented automatically with default parameters for the kinetic orders and rate constants that would seem reasonable in a given situation.

The third area could be the way simulations are done. Inspiration comes from the gaming industry, which has been developing very powerful game engines in recent years. These tools for creating new games can, for instance, integrate 2D images into 3D constructs and help assemble physically realistic structures down to the resolution of hair movements. Typical game engines permit scripting capabilities, sound integration, networking, streaming, memory management, threading, localization support, scene graphs, and video support for cinematics. Just imagine what similarly powerful "bioengines" could do for simulating complex physiological systems [6]. There is little doubt that gaming approaches would also be very attractive educational tools.

Finally, the output of model simulations may prove to be useful for enhancing machine learning algorithms and classifier construction. As an example, the simulation output from a constraint-based model, consisting of steady-state flux values, was used as a feature for classifying concentration levels of a key variable in a decision tree analysis of signal transduction [157, 158]. Moreover, in analogy to the sophisticated modeling performed in weather forecasting, rudimentary mechanistic models might be developed in the future as an initial step to generate "data," based upon knowledge of the biological system, for instance, in an effort to train neural networks [159] or other machine learning tools [160].

Suppose we follow the steps of the data pipeline, possibly supported by AI, to the point where we have extracted clean data, ideally in the format of time series, that is, as a dense series of snapshots over time. In a simple case, such data will come from a single level of biological organization, such as gene expression or metabolism (**Figure 5.13**), but eventually, the data might be "multi-omic" and include measurements of gene expression, protein prevalence, metabolite profiles, and physiological markers, all from the same biological organism or system at

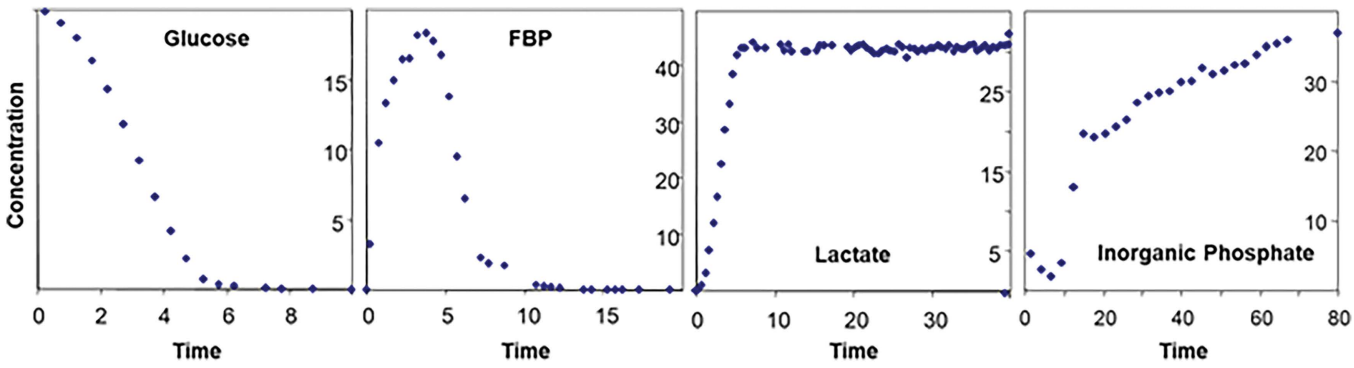


Figure 15.13 Metabolic datasets on fermentation by the bacterium Lactococcus lactis. Each panel shows a time series of concentrations of glycolytic metabolites (adapted from [166]).

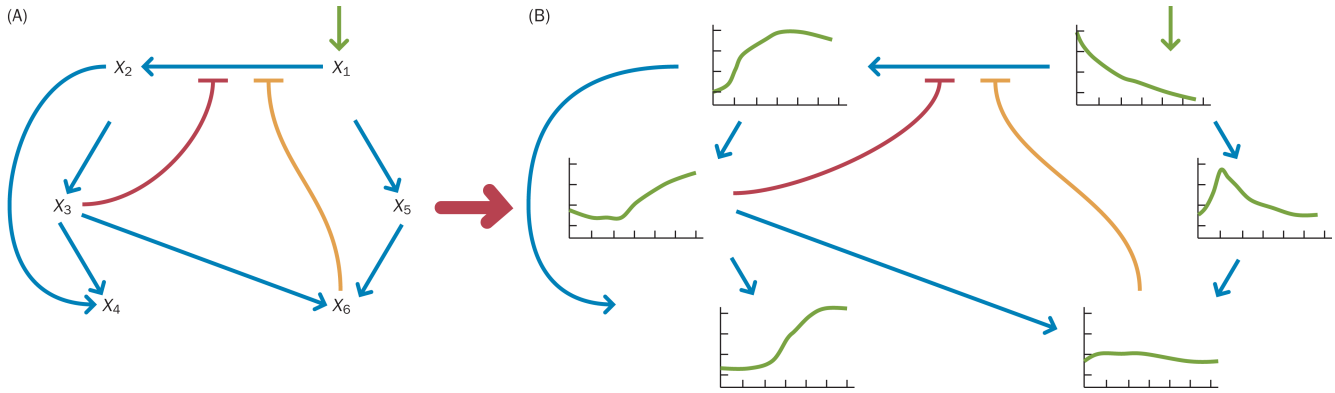


Figure 15.14 Concept-map modeling. (A) A typical diagram of a biological system is augmented by coarse, measured, or expected dynamic responses to a typical input (green arrow). (B) The result is a system of time courses (green lines) that can be formulated as symbolic equations. Its parameter values can be estimated using inverse methods (see Chapter 5). Blue arrows indicate flow of material. Red and orange blunted lines indicate known and hypothesized inhibition signals, respectively.

a sequence of time points (e.g., [161, 162]). These data can be an ideal starting point for designing and analyzing dynamic models (e.g., [163–165]).

Of course, experimental data are typically sparse, especially if they reflect the state of a system at different time points, because their creation requires time and effort. An important role for computational modeling is to weave these individual, sparse snapshots into cohesive storylines that explain what is happening in a biological system.

If a process diagram is known—or thought to be known—but time series data are scarce or not available at all, it may still be possible to design a dynamic model for testing hypotheses, at least in a qualitative or semi-quantitative manner. An interesting approach permitting this type of model construction and analysis is **concept map modeling** [167]. In the first step of this method, the components and processes are put into a typical diagram of boxes and arrows. Next, they are formulated as symbolic equations, based directly on the diagram. To some degree, this process can be automated if one uses mass action or GMA models as default representations; however, any formulations may be used. In the following step, a biologist familiar with the system sketches the known or expected response of each system component to a perturbation, such as an increase in a variable due to the knock-down of a gene (**Figure 15.14**). For instance, the biologist might expect some component to increase slowly, then speed up, and finally reach a saturation level in an S-shaped manner. This information might have been measured or reflect the biologist's experience and intuition, which is often more valuable than one specific dataset. At this point, AI could become very beneficial as it could add pertinent information from the internet, which would be curated by a subject area expert before use. The measured or alleged time course responses could be directly converted into numerical time series that show how much and how fast each component is expected to change and what the transient response might look like; this conversion may be done with software that extracts coordinates from graphs, such as [168]. Most experienced biologists will have a good intuition regarding such responses in systems that they have been investigating for many years. The alleged time courses are now converted into parameter values of the symbolic model that had been formulated from the components and the connectivity of the original diagram. The method for this conversion is parameter estimation, as it was discussed in Chapter 6. Additional information from other sources can be added to the model as well. Ideally, the result is a numerical model that captures the biologist's intuition of the features of all system components. It is now easy to perform model simulations. These might try to assess the likelihood of a hypothesized mechanism (such as the orange inhibition signal in Figure 15.14) or simulate tested and yet untested scenarios. The results are compared with the biologist's expectations or with new data. Confirmatory simulation results are of course welcome, but counterintuitive results are similarly valuable because they point

to an incomplete understanding of some detail. A thorough analysis of ill-fitting simulations often provides concrete hints of where the problems are most likely to lie. These problems might be due to a faulty model structure, for instance, in terms of missing or ill-placed processes or regulatory signals. They may also be numerical in a sense that rate constants or other parameters are assigned very different values than what they should be. The analysis may also suggest the replacement of the original canonical model formulation with more complicated functions. Importantly, the process generates a computational structure that mimics the biological phenomenon and is incomparably easier to interrogate than the actual biological system. This interrogation will often lead to new hypotheses that are to be tested in the laboratory. A detailed example of using a diagram combined with expert knowledge can be found in [109].

Modeling immediately becomes much more complicated if it spans several levels of biological organization. We discussed some approaches toward multiscale analysis in Chapter 11, but much more will need to be done in the future. For instance, we will need effective means of integrating coarse and fine-grained information. Models of phenomena like complex diseases will have to be modular, expandable, and connect different levels and scales. No models within the near future will capture realistic systems in their entirety, which means that we must develop techniques that functionally and validly connect the well-defined core of a modeled system with an ill-defined outside. We need to characterize local effects of a model on global responses in its environment and capture how these global effects feed back to the modeled system. We must also find ways of connecting different time scales in an effective manner. We must invent efficient techniques for merging deterministic and stochastic features that change dynamically and in spatially heterogeneous environments.

15.4 TOWARD A THEORY OF BIOLOGY . . . OR SEVERAL THEORIES?

According to the tenets of mathematical logic, a theory consists of a formal language that permits the formulation of inference rules and a set of sentences or theorems that are assumed or proved to be true. The assumed sentences are often called axioms, which form a set of common beliefs; an example is the axiom that every integer has a successor. If these axioms are not accepted as true, further inferences are usually moot, because one cannot decide whether they are true or false. Valid application of the inference rules to axioms and proven theorems leads to new theorems that can be proved to be true or refuted, as long as one accepts the axioms.

Faced with such rigor, does it even make sense to strive for a theory of biology? What could serve as the language, what would be the rules of inference, what kinds of axioms could one formulate? Are there even absolute truths in biology? Would the effort pay off in the end if we could actually reach our goal? Indeed, it might seem quite esoteric to pose a biological theory as a long-term goal, when so many practical and pressing questions await answers. But as the Prussian philosopher and psychologist Kurt Lewin said, "there is nothing so practical as a good theory" ([169], p. 288). To convince ourselves of the veracity of such a bold statement, just look at mathematics and physics. The laws of mathematics and physics have helped us understand much of the world, and the theories of physics are the foundations of engineering with all its enormous implications and applications. If we had similarly strong laws in biology, or at least in certain subspecialties, we would no longer have to question the logic of observed phenomena and could save plenty of time and effort if the laws would allow us to make reliable inferences.

What might such biological axioms or laws look like? Some of them we actually have already. A famous candidate for a biological axiom is the quote *omnis cellula e cellula* (every cell comes from a cell) of Rudolf Virchow, a German cell physiologist of the nineteenth century. Or consider the genetic code that assigns amino acids to certain codons and which is essentially ubiquitous among all

organisms on earth (see Chapter 8 for this code and a few exceptions). Without this "codon law" we would have to decode every new gene sequence from scratch! Every time we discovered a new organism, we would not only have to sequence its genome but also determine how this specific organism translated its gene sequences into peptides and proteins. We know about the Central Dogma, and even though details of this dogma have been challenged, for instance in the context of RNA viruses, it is a very powerful tool that we have learned to take for granted within its domain of applicability: if we knock out a coding gene, we do not expect the corresponding mRNA or protein to be present. Mendelian genetics is a good first approximation of the processes of inheritance, and it is simply necessary to define the boundaries of this "theory" a little bit more stringently than the Austrian monk Gregor Mendel did. Many biologists consider evolution a mature theory. Within a more limited scope, Michael Savageau's demand theory [170] has the character of a theory. Hiroaki Kitano suggested a theory linking disease to the robustness of a physiological system [88, 171]. There are certainly laws governing the development of an embryo, and we rely on these laws almost as much as on physical laws. There are even laws at higher levels, such as the famous phenomenon of behavioral conditioning that was studied by the Russian physiologist and psychologist Ivan Petrovich Pavlov. Much effort from different angles is currently being devoted to discovering fundamental laws that span several biological scales (e.g., see [172]).

Up to now, the search for general principles has required us to boil down complex, realistic systems and their models to essential features such as network motifs. This reduction of complexity presently is an art that has eluded strict general guidelines, which suggests that systematic methods of valid model reduction should be high on our list of priorities. In a parallel line of thinking, efforts are underway for creating mathematical concepts and techniques of topological data analysis that facilitate the detection and analysis of highly nonlinear and often hidden geometric structures in massive datasets [173]. Specific tasks that are to be addressed with these methods include how one might infer high-dimensional structures in data from low-dimensional representations and how discrete data points might be assembled into global structures [174].

In the near future, any budding theories of biological sub-disciplines might be modest in scope, but they will eventually span larger domains and later lead to a comprehensive understanding of processes like carcinogenesis or the dynamics of microbial communities. We may also discover that the number of hard, unbreakable laws in biology is limited but that it is quite possible to identify **fuzzy** or probabilistic laws, which are valid in most cases but not always. As we discussed in Chapter 6, several groups of researchers have begun focusing on the formulation of motifs, design principles, and operating principles, under which many organisms in the living world operate. Many of these will be encountered much more frequently than expected from a random design—but not in all cases. Thus, some of these principles may hold almost always, whereas other patterns may simply stand out more often than not and only in certain contexts. These limits in scope will become parts of theories and theorems, as it is common in mathematics, where certain statements are only true for certain sets of objects.

A strong theory of subareas of biology, for instance in gene regulation, would have tremendous implications. In analogy with the applications of physical theory in engineering, a biological theory of gene regulation would enormously streamline synthetic biology by permitting reliable predictions of which manipulations would actually lead to the desired goals. In many cases we are already making these predictions, and they are often correct. With a theory and with knowledge of its boundaries, the degree of certainty would come close to 100%.

Beyond all speculations regarding the future, a few trends are clear. First, we live in an unprecedented and incredibly exciting time for biology, modeling, and systems biology. Second, we have barely scratched the surface, and a book like this may cause a good chuckle 40 or 50 years from now. Hopefully by that time, our generation of systems biologists will have earned and deserved respect for how much we accomplished with the little we had. Above all, there is a huge and fascinating world out there that invites us to explore.

FURTHER READING

Loscalzo J, Barabási AL. Systems biology and the future of medicine. *Wiley Interdisc Rev Syst Biol Med.* 3(6) (2011) 619–627.

Lucas M. Future challenges in plant systems biology. *Meth Mol Biol.* (2021) 325–337.

Voigt CA. Synthetic biology 2020–2030: Six commercially-available products that are changing our world. *Nat Comm.* 11 (2020) 6379.

Voit EO. Perspective: Systems biology beyond biology. *Front Syst Biol.* 2 (2022) 987135.

Voit EO, Shah AM, Olivenca DV, Vodovotz Y. What's next for computational systems biology? *Frontiers Sys Biol.* 3 (2023), https://doi.org/10.3389/fsysb.2023.1250228.

REFERENCES

1. Meadows, D.L. and D.H. Meadows, *Toward Global Equilibrium*. 1973, Cambridge, MA: Wright-Allen Press, Inc.
2. Ohno, S., So much "junk" DNA in our genome. *Brookhaven Symp Biol*, 1972. **23**: p. 366–370.
3. Eldredge, N. and S.J. Gould, *Punctuated equilibria: An alternative to phyletic gradualism*, in *Models in Paleobiology*, T.J.M. Schopf, Editor. 1972, Freeman Cooper: San Francisco, CA. p. 82–115.
4. Taleb, N.N., *The Black Swan: The Impact of the Highly Improbable*. 2017, New York, NY: Random House.
5. Ulam, S., Tribute to John von Neumann. *Bull Amer Math Soc*, 1958. **64**(3): p. 1–49.
6. Voit, E.O., et al., What's next for computational systems biology? *Front Syst Biol*, 2024. in press.
7. Tomita, M., et al., E-CELL: Software environment for whole-cell simulation. *Bioinformatics*, 1999. **15**(1): p. 72–84.
8. Carrera, J. and M.W. Covert, Why build whole-cell models? *Trends in Cell Biol*, 2015. **25**(2): p. 719–722.
9. Karr, J.R., et al., A whole-cell computational model predicts phenotype from genotype. *Cell*, 2012: p. 389–401.
10. Goldberg, A.P., et al., Emerging whole-cell modeling principles and methods. *Curr Opin Biotechnol*, 2018. **51**: p. 97–102.
11. Feig, M. and Y. Sugita, Whole-cell models and simulations in molecular detail. *Annu Rev Cell Dev Biol*, 2019. **35**: p. 191–211.
12. Marucci, L., et al., Computer-aided whole-cell design: Taking a holistic approach by integrating synthetic with systems biology. *Front Bioeng Biotechnol*, 2020. **8**: p. 942.
13. Tang, L., Simulating a whole cell. *Nat Meth*, 2022. **19**: p. 271.
14. Sun, G., T.A. Ahn-Horst, and M.W. Covert, The *E. coli* whole-cell modeling project. *EcoSal Plus*, 2021. **9**(2): p. eESP00012020.
15. Ahn-Horst, T.A., et al., An expanded whole-cell model of *E. coli* links cellular physiology with mechanisms of growth rate control. *NPJ Syst Biol Appl*, 2022. **8**(1): p. 30.
16. Atkinson, E., et al., Resource-aware whole-cell model of division of labour in a microbial consortium for complex-substrate degradation. *Microb Cell Fact*, 2022. **21**(1): p. 115.
17. Thornburg, Z.R., et al., Fundamental behaviors emerge from simulations of a living minimal cell. *Cell*, 2022. **185**(2): p. 345–360, e28.
18. Bhat, N.G. and S. Balaji, Whole-cell modeling and simulation: A brief survey. *New Gener Comput* 2020. **38**: p. 259–281.
19. Singla, J. and K.L. White, A community approach to whole-cell modeling. *Curr Opin Syst Biol*, 2021. **26**: p. 33–38.
20. Babtie, A.C. and M.P.H. Stumpf, How to deal with parameters for whole-cell modelling. *J Roy Soc Interface*, 2017. **14**: p. 20170237.
21. Masison, J., et al., A modular computational framework for medical digital twins. *PNAS*, 2021. **118**(20): p. e2024287118.
22. Noronha, A., et al., The virtual metabolic human database: Integrating human and gut microbiome metabolism with nutrition and disease. *Nucleic Acids Res*, 2019. **47**(D1): p. D614–D624.
23. Physiome, *The Physiome Project*, http://www.physiome.org/Links/websites.html, 2011.
24. Virtual_Liver, *The Virtual Liver Network*, https://www.h-its.org/projects/virtual-liver/, 2011.
25. Virtual_Brain, www.thevirtualbrain.org. accessed 2023.
26. Virtual Rat Project, https://virtualrat.org/.
27. Stefanovski, L., et al., Bridging scales in Alzheimer's disease: Biological framework for brain simulation with the virtual brain. *Front Neuroinform,* 2021. **15**: p. 630172.
28. Kinyanjui, N.M. and F.D. Johansson, ADCB: An Alzheimer's disease simulator for benchmarking observational estimators of causal effects, in *Proceedings of the Conference on Health, Inference, and Learning*, F. Gerardo, et al., Editors. 2022, PMLR: Proceedings of Machine Learning Research. p. 103–118, https://proceedings.mlr.press/v174/kinyanjui22a.html.
29. Kansal, A.R., et al., Alzheimer's disease Archimedes condition-event simulator: Development and validation. *Alzheimers Dement (N Y)*, 2018. **4**: p. 76–88.
30. Grant, J., et al., Mechanistic PK/PD modeling to address early-stage biotherapeutic dosing feasibility questions. *MAbs*, 2023. **15**(1): p. 2192251.
31. SoySim, https://soysim.unl.edu/.
32. Song, Q., et al., A user-friendly means to scale from the biochemistry of photosynthesis to whole crop canopies and production in time and space—development of Java WIMOVAC. *Plant Cell Environ* 2017. **40**: p. 51–55.
33. Bruhn, J.A., et al., User's manual for LATEBLIGHT: A plant disease management game *Cornell Univ Dept Plant Pathol Mimeo*, 1981. **80**: p. 1–7.
34. Andrade-Piedra, J.L., et al., Qualification of a plant disease simulation model: Performance of the LATEBLIGHT model across a broad range of environments. *Phytopathology*, 2005. **95**: p. 1412–1422.
35. Di Carlo, D. and L.P. Lee, Dynamic single-cell analysis for quantitative biology. *Anal Chem*, 2006. **78**(23): p. 7918–7925.
36. Schmid, A., et al., Chemical and biological single cell analysis. *Curr Opin Biotechnol*, 2010. **21**(1): p. 12–20.
37. Stuart, T. and R. Satija, Integrative single-cell analysis. *Nat Rev Genet*, 2019. **20**(5): p. 257–272.
38. Adil, A., et al., Single-cell transcriptomics: Current methods and challenges in data acquisition and analysis. *Front Neurosci*, 2021. **15**: p. 591122.
39. Hasenauer, J., et al., ODE constrained mixture modelling: A method for unraveling subpopulation structures and dynamics. *PLoS Comput Biol*, 2014. **10**(7): p. e1003686.
40. Loos, C., et al., A hierarchical, data-driven approach to modeling single-cell populations predicts latent causes of cell-to-cell variability. *Cell Syst*, 2018. **6**(5): p. 593–603, e13.

41. Wade, J.D., et al., Inference of continuous single-cell signaling responses from multiplex snapshot measurements. *bioRxiv*, 2020: p. 11, https://doi.org/10.11.15.383711.
42. Cho, H. and R.C. Rockme, Mathematical modeling with single-cell sequencing data. *bioRXiv*, 2019: p. 710640, https://doi.org/10.1101/710640.
43. Davis, J.D., et al., Methods of quantifying interactions among populations using Lotka-Volterra models. *Frontiers Systems Biol*, 2023. **2**, https://doi.org/10.10.3389/fsysb.2022.1021897.
44. Aktipis, C.A., et al., Cancer across the tree of life: Cooperation and cheating in multicellularity. *Philos Trans R Soc Lond B Biol Sci*, 2015. **370**(1673).
45. Dujon, A.M., et al., Identifying key questions in the ecology and evolution of cancer. *Evol Appl*, 2021. **14**(4): p. 877–892.
46. Regev, A., et al., The human cell atlas. *Elife*, 2017. **6**.
47. Regev, A., Cell atlases as roadmaps in health and disease. *Keio J Med*, 2020. **69**(4): p. 105.
48. Rozenblatt-Rosen, O., et al., The human tumor atlas network: Charting tumor transitions across space and time at single-cell resolution. *Cell*, 2020. **181**(2): p. 236–249.
49. Zhou, Y., et al., The Alzheimer's Cell Atlas (TACA): A single-cell molecular map for translational therapeutics accelerator in Alzheimer's disease. *Alzheimers Dement (N Y)*, 2022. **8**(1): p. e12350.
50. Vodovotz, Y., Translational systems biology of inflammation and healing. *Wound Repair Regen*, 2010. **18**(1): p. 3–7.
51. Vodovotz, Y., et al., Mechanistic simulations of inflammation: current state and future prospects. *Math Biosci*, 2009. **217**(1): p. 1–10.
52. Palm, N.W. and R. Medzhitov, Not so fast: Adaptive suppression of innate immunity. *Nat Med*, 2007. **13**(10): p. 1142–1144.
53. Zhao, J., et al., Do adaptive immune cells suppress or activate innate immunity? *Trends Immunol*, 2008. **30**(1): p. 8–12.
54. Gonnord, P., et al., A hierarchy of affinities between cytokine receptors and the common gamma chain leads to pathway cross-talk. *Sci Signal*, 2018. **11**(524).
55. Talaei, K., et al., A mathematical model of the dynamics of cytokine expression and human immune cell activation in response to the pathogen *Staphylococcus aureus*. *Front Cell Infect Microbiol*, 2021. **11**.
56. Reis, R.F., et al., A validated mathematical model of the cytokine release syndrome in severe COVID-19. *Front Mol Biosci*, 2021. **8**.
57. Yang, L., et al., Correction: The signal pathways and treatment of cytokine storm in COVID-19. *Signal Transduct Target Ther*, 2021. **6**(1): p. 326.
58. Ghazal, P., et al., Challenging molecular dogmas in human sepsis using mathematical reasoning. *EBioMedicine*, 2022. **80**: p. 104031.
59. Zitzmann, C. and L. Kaderali, Mathematical analysis of viral replication dynamics and antiviral treatment strategies: From basic models to age-based multi-scale modeling. *Front Microbiol*, 2018. **9**.
60. Ostaszewski, M., et al., COVID19 disease map, a computational knowledge repository of virus-host interaction mechanisms. *Mol Syst Biol*, 2021. **17**(10): p. e10387.
61. Reynolds, A., et al., A reduced mathematical model of the acute inflammatory response: I. Derivation of model and analysis of anti-inflammation. *J Theor Biol*, 2006. **242**(1): p. 220–236.
62. Handel, A., I.M. Longini, Jr., and R. Antia, Towards a quantitative understanding of the within-host dynamics of influenza A infections. *J R Soc Interface*, 2010. **7**(42): p. 35–47.
63. Vodovotz, Y., et al., Translational systems approaches to the biology of inflammation and healing. *Immunopharm Immunot*, 2010. **32**(2): p. 181–195.
64. Clermont, G., et al., In silico design of clinical trials: A method coming of age. *Crit Care Med*, 2004. **32**(10): p. 2061–2070.
65. Chow, C.C., et al., The acute inflammatory response in diverse shock states. *Shock*, 2005. **24**(1): p. 74–84.
66. Foteinou, P.T., et al., Modeling endotoxin-induced systemic inflammation using an indirect response approach. *Math Biosci*, 2009. **217**(1): p. 27–42.
67. An, G., Introduction of an agent-based multi-scale modular architecture for dynamic knowledge representation of acute inflammation. *Theor Biol Med Model*, 2008. **5**: p. 11.
68. Fallahi-Sichani, M., et al., Multiscale computational modeling reveals a critical role for TNF-alpha receptor 1 dynamics in tuberculosis granuloma formation. *J Immunol*, 2011. **186**(6): p. 3472–3483.
69. Feinendegen, L., et al., Systems biology and its potential role in radiobiology. *Radiat Environ Biophys*, 2008. **47**(1): p. 5–23.
70. Armitage, P. and R. Doll, The age distribution of cancer and a multi-stage theory of carcinogenesis. *Br J Cancer*, 1954. **8**(1): p. 1–12.
71. Moolgavkar, S.H. and A.G.J. Knudson, Mutation and cancer: A model for human carcinogenesis. *J Natl Cancer Inst*, 1981. **66**: p. 1037–1052.
72. Moolgavkar, S.H. and E.G. Luebeck, Multistage carcinogenesis and the incidence of human cancer. *Genes Chrom Cancer*, 2003. **38**: p. 302–306.
73. McCreery, M.Q. and A. Balmain, Chemical carcinogenesis models of cancer: Back to the future. *Ann Rev Cancer Biol*, 2017. **1**: p. 295–312.
74. Haupt, S., et al., Mathematical modeling of multiple pathways in colorectal carcinogenesis using dynamical systems with Kronecker structure. *PLoS Comput Biol*, 2021. **17**(5): p. e1008970.
75. Enderling, H., L. Hlatky, and P. Hahnfeldt, Migration rules: Tumours are conglomerates of self-metastases. *Br J Cancer*, 2009. **100**(12): p. 1917–1925.
76. Folkman, J. and R. Kalluri, Concept cancer without disease. *Nature*, 2004. **427**.
77. Black, W.C. and H.G. Welch, Advances in diagnostic imaging and overestimations of disease prevalence and the benefits of therapy. *N Engl J Med*, 1993. **328**: p. 1237–1243.
78. Grussendorf, M., I. Ruschenburg, and G. Brabant, Malignancy rates in thyroid nodules: a long-term cohort study of 17,592 patients. *Eur Thyroid J*, 2022. **11**(4).
79. Zahl, P.H., J. Maehlen, and H.G. Welch, The natural history of invasive breast cancers detected by screening mammography. *Arch Intern Med*, 2008. **168**(21): p. 2311–2316.
80. Abdollahi, A., et al., Transcriptional network governing the angiogenic switch in human pancreatic cancer. *Proc Natl Acad Sci U S A*, 2007. **104**(31): p. 12890–12895.
81. Altrock, P.M., L.L. Liu, and F. Michor, The mathematics of cancer: integrating quantitative models. *Nat Rev Cancer*, 2015. **15**(12): p. 730–745.
82. Kato, G.J., et al., Sickle cell disease. *Nat Rev Dis Primers*, 2018. **4**: p. 18010.
83. Olivença, D.V., et al., Mathematical models of cystic fibrosis as a systemic disease. *WIRE's Mech Dis*, 2023. **E1625**.
84. Bergman, R.N. and M.E. Refai, Dynamic control of hepatic glucose metabolism: Studies by experiment and computer simulation. *Ann Biomed Eng*, 1975. **3**(4): p. 411–432.
85. Bergman, R.N., Minimal model: Perspective from 2005. *Horm Res*, 2005. **64**(Suppl 3): p. 8–15.
86. Rathee, S. and Nilam, ODE models for the management of diabetes: A review. *Int J Diabetes Dev Ctries* 2017. **37**: p. 4–15.
87. Yang, B., et al., The progression of secondary diabetes: A review of modeling studies. *Front Endocrinol*, 2022. **13**.
88. Kitano, H., et al., Metabolic syndrome and robustness tradeoffs. *Diabetes*, 2004. **53** (Suppl 3): p. S6–S15.

89. Tam, J., D. Fukumura, and R.K. Jain, A mathematical model of murine metabolic regulation by leptin: Energy balance and defense of a stable body weight. *Cell Metab*, 2005. **9**(1): p. 52–63.
90. Visentin, R., et al., The UVA/Padova type 1 diabetes simulator goes from single meal to single day. *J Diabetes Sci Technol*, 2018. **12**(2): p. 273–281.
91. Eddy, D., et al., The relationship between insulin resistance and related metabolic variables to coronary artery disease: A mathematical analysis. *Diabetes Care*, 2009. **32**(2): p. 361–366.
92. Lipsitz, L.A., Physiological complexity, aging, and the path to frailty. *Sci Aging Knowledge Environ*, 2004. **2004**(16): p. pe16.
93. Lipsitz, L.A. and A.L. Goldberger, Loss of 'complexity' and aging. Potential applications of fractals and chaos theory to senescence. *Jama*, 1992. **267**(13): p. 1806–1809.
94. Ferrario, M., et al., Comparison of entropy-based regularity estimators: application to the fetal heart rate signal for the identification of fetal distress. *IEEE Trans Biomed Eng*, 2006. **53**(1): p. 119–125.
95. Williams, K.P. and F. Galerneau, Intrapartum fetal heart rate patterns in the prediction of neonatal acidemia. *Am J Obstet Gynecol*, 2003. **188**(3): p. 820–823.
96. Beard, R.W., et al., The significance of the changes in the continuous fetal heart rate in the first stage of labour. *J Obstet Gynaecol Br Commonw*, 1971. **78**(10): p. 865–881.
97. Goldberger, A.L., Is the normal heartbeat chaotic or homeostatic? *News Physiol Sci*, 1991. **6**: p. 87–91.
98. Alzheimer_Assoc, *Alzheimer's Association*, www.alz.org/media/Documents/alzheimers-facts-and-figures.pdf, accessed 2023.
99. Dexter, D.T. and P. Jenner, Parkinson disease: From pathology to molecular disease mechanisms. *Free Radic Biol Med*, 2013. **62**: p. 132–144.
100. Cerri, S., L. Mus, and F. Blandini, Parkinson's disease in women and men: What's the difference? *J Parkinsons Dis*, 2019. **9**(3): p. 501–515.
101. Hodgkin, A. and A. Huxley, A quantitative description of membrane current and its application to conduction and excitation in nerve. *J Physiol*, 1952. **117**: p. 500–544.
102. Izhikevich, E.M., *Dynamical Systems in Neuroscience*. 2007, Cambridge, MA: MIT Press.
103. Keener, J. and J. Sneyd, *Mathematical Physiology*. 2nd ed. 1998, New York NY: Springer Verlag.
104. Fitzhugh, R., Impulses and physiological states in theoretical models of nerve membrane. *Biophys J*, 1961. **1**: p. 445–466.
105. Herz, A.V., et al., Modeling single-neuron dynamics and computations: A balance of detail and abstraction. *Science*, 2006. **314**: p. 80–85.
106. Nagumo, J.A. and S. Yoshizawa, An active pulse transmission line simulating nerve axon. *Proc Inst Radio Engineers*, 1962. **50**: p. 2061–2070.
107. Gleeson, P., et al., NeuroML: A language for describing data driven models of neurons and networks with a high degree of biological detail. *PLoS Comput Biol*, 2010. **6**: p. 6 e1000815.
108. Kikuchi, S., et al., Kinetic simulation of signal transduction system in hippocampal long-term potentiation with dynamic modeling of protein phosphatase 2A. *Neural Netw*, 2003. **16**(9): p. 1389–1398.
109. Qi, Z., G.W. Miller, and E.O. Voit, Computational systems analysis of dopamine metabolism. *PLoS One*, 2008. **3**(6): p. e2444.
110. Qi, Z., G.W. Miller, and E.O. Voit, The internal state of medium spiny neurons varies in response to different input signals. *BMC Syst Biol*, 2010. **4**: p. 26.
111. Tretter, F., Mental illness, synapses and the brain—behavioral disorders by a system of molecules with a system of neuorns? *Pharmacopsychiatry*, 2010. **43**(Suppl. 1): p. S9–S20.
112. Qi, Z., F. Tretter, and E.O. Voit, A heuristic model of alcohol dependence. *PLoS One*, 2014. **9**(3): p. e92221.
113. Qi, Z., et al., A heuristic model for working memory deficit in schizophrenia. *Biochim Biophys Acta*, 2016. **1860**(11 Pt B): p. 2696–2705.
114. Hornik, K., M. Stinchcombe, and H. White, Multilayer feedforward networks are universal approximators. *Neural Netw*, 1989. **2**: p. 359–366.
115. McCulloch, W. and W. Pitts, A logical calculus of the ideas immanent in nervous activity. *Bull Math Biol*, 1943. **7**: p. 115–133.
116. Duval, F., *Artificial Neural Networks: Concepts, Tools and Techniques Explained for Absolute Beginners*. 2018, Scotts Valley, CA: CreateSpace Independent Publishing Platform.
117. Blue-Brain-Project, http://bluebrain.epfl.ch/, 2010.
118. Human-Connectome, https://neuroscienceblueprint.nih.gov/human-connectome/connectome-programs, accessed 2023.
119. de Garis, H., et al., A world survey of artificial brain projects, Part I: Large-scale brain simulations. *Neurocomputing*, 2010. **74**(1–3): p. 3–29.
120. Dhawale, A. and U.S. Bhalla, The network and the synapse: 100 years after Cajal. *HFSP J*, 2008. **2**(1): p. 12–16.
121. Destexhe, A. and D. Contreras, Neuronal computations with stochastic network states. *Science*, 2006. **314**: p. 85–90.
122. Rubin, J.E., Emergent bursting in small networks of model conditional pacemakers in the pre-Botzinger complex. *Adv Exp Med Biol*, 2008. **605**: p. 119–124.
123. Hobert, O., Neurogenesis in the nematode *Caenorhabditis elegans*, in *WormBook: The Online Review of C. Elegans Biology [Internet]*. Pasadena, CA, https://www.ncbi.nlm.nih.gov/books/NBK116086/.
124. Oshio, K., et al., *Database of Synaptic Connectivity of C. Elegans for Computation*, http://ims.dse.ibaraki.ac.jp/ccep/, 2010.
125. Engleman, E.A., S.N. Katner, and B.S. Neal-Beliveau, *Caenorhabditis elegans* as a model to study the molecular and genetic mechanisms of drug addiction. *Prog Mol Biol Transl Sci*, 2016. **137**: p. 229–252.
126. *Worms in Space*, https://www.nasa.gov/audience/forstudents/postsecondary/features/F_Worms_in_Space_prt.htm, accessed 2023.
127. Tretter, F., et al., Systems biology and addiction. *Pharmacopsychiatry*, 2009. **42**(Suppl 1): p. S11–S31.
128. Furber, S., Large-scale neuromorphic computing systems. *J Neural Eng*, 2016. **13**(5): p. 051001.
129. Azer, K., et al., History and future perspectives on the discipline of quantitative systems pharmacology modeling and its applications. *Front Physiol*, 2021. **12**.
130. Knight-Schrijver, V.R., et al., The promises of quantitative systems pharmacology modelling for drug development. *Comp Struct Biotechn J*, 2016. **14**: p. 363–370.
131. Modernization_Act, *CONGRESS.GOV. H.R.2617—Consolidated Appropriations Act, 2023. Became Public Law No: 117–328. December 29, 2022*, https://www.congress.gov/bill/117th-congress/house-bill/2617/text, accessed 2023.
132. Peterson, M.C. and M.M. Riggs, FDA advisory meeting clinical pharmacology review utilizes a quantitative systems pharmacology (QSP) model: A watershed moment? *CPT Pharmacometrics Syst Pharmacol*, 2015. **4**(3): p. e00020.
133. Marino, M., Z. Jamal, and P.M. Zito, Pharmacodynamics, in *StatPearls [Internet]*. Treasure Island, FL: StatPearls Publishing, accessed 2023.
134. Guyton, A.C., T.G. Coleman, and H.J. Granger, Circulation: Overall regulation. *Annu Rev Physiol*, 1972. **34**: p. 13–46.
135. Iyengar, R., et al., Merging systems biology with pharmacodynamics. *Sci Transl Med*, 2012. **4**(126): p. 126ps7.
136. Wang, H., et al., Quantitative systems pharmacology model predictions for efficacy of atezolizumab and nab-paclitaxel in

triple-negative breast cancer. *J Immunotherap Cancer*, 2021. **9**(2): p. e002100.
137. Longo, D., et al., Using quantitative systems pharmacology modeling to understand the pathophysiology of idiopathic pulmonary fibrosis, in TP126. *TP126 Struct FUNC Parenchy Lung Dis*, 2021. p. A4648–A4648.
138. Brubaker, D.K., et al., An interspecies translation model implicates integrin signaling in infliximab-resistant inflammatory bowel disease. *Sci Signal*, 2020. **13**(643).
139. Tewary, M., et al., A stepwise model of reaction-diffusion and positional information governs self-organized human peri-gastrulation-like patterning. *Development*, 2017. **144**(23): p. 4298–4312.
140. White, D.E., et al., Quantitative multivariate analysis of dynamic multicellular morphogenic trajectories. *Integr Biol (Camb)*, 2015. **7**(7): p. 825–833.
141. Libby, A.R.G., et al., Automated design of pluripotent stem cell self-organization. *Cell Syst*, 2019. **9**(5): p. 483–495 e10.
142. Glen, C.M., T.C. McDevitt, and M.L. Kemp, Dynamic intercellular transport modulates the spatial patterning of differentiation during early neural commitment. *Nat Commun*, 2018. **9**(1): p. 4111.
143. Lancaster, M.A., et al., Cerebral organoids model human brain development and microcephaly. *Nature*, 2013. **501**(7467): p. 373–379.
144. Norfleet, D.A., E. Park, and M.L. Kemp, Computational modeling of organoid development. *Curr Opin Biomed Eng*, 2020. **13**: p. 113–118.
145. Cyr, K.J., O.M. Avaldi, and J.P. Wikswo, Circadian hormone control in a human-on-a-chip: In vitro biology's ignored component? *Exp Biol Med (Maywood)*, 2017. **242**(17): p. 1714–1731.
146. Menshykau, D., Emerging technologies for prediction of drug candidate efficacy in the preclinical pipeline. *Drug Discov Today*, 2017. **22**(11): p. 1598–1603.
147. Jones, M.B., et al., The new bioinformatics: Integrating ecological data from the gene to the biosphere. *Annu Rev Ecol Evol Syst*, 2006. **37**: p. 519–544.
148. Genomicscience, http://genomicscience.energy.gov/benefits/index.shtml.
149. Savage, D.F., et al., Spatially ordered dynamics of the bacterial carbon fixation machinery. *Science*, 2010. **327**(5970): p. 1258–1261.
150. Burkepile, D.E. and M.E. Hay, Impact of herbivore identity on algal succession and coral growth on a Caribbean reef. *PLoS One*, 2010. **e8963**, https://doi.org/10.1371/journal.pone.0008963n.
151. Succi, S. and P.V. Coveney, Big data: the end of the scientific method? *Philos Trans A Math Phys Eng Sci*, 2019. **377**(2142): p. 20180145.
152. Martins dos Santos, V., et al., Systems biology in ELIXIR: Modelling in the spotlight [version 1; peer review: 1 approved, 2 approved with reservations]. *F1000Res*, 2022. **11**(1265).
153. Baumer, B.S., D.T. Kaplan, and N.J. Horton, *Modern Data Science with R*. 2nd ed. 2021, New York, NY: CRC Press.
154. Wild, C.P., Complementing the genome with an "exposome": The outstanding challenge of environmental exposure measurement in molecular epidemiology. *Cancer Epidemiol Biomarkers Prev*, 2005. **14**(8): p. 1847–1850.
155. Miller, G.W., *The Exposome: A Primer*. 2014, Amsterdam: Elsevier, Academic Press.
156. Faeder, J.R., M.L. Blinov, and W.S. Hlavacek, Rule-based modeling of biochemical systems with BioNetGen. *Methods Mol Biol*, 2009. **500**: p. 113–167.
157. Lewis, J.E. and M.L. Kemp, Integration of machine learning and genome-scale metabolic modeling identifies multi-omics biomarkers for radiation resistance. *Nat Commun*, 2021. **12**(1): p. 2700.
158. Hua, F., et al., Integrated mechanistic and data-driven modelling for multivariate analysis of signalling pathways. *J R Soc Interface*, 2006. **3**(9): p. 515–526.
159. Wang, S., et al., Massive computational acceleration by using neural networks to emulate mechanism-based biological models. *Nat Commun*, 2019. **10**(1): p. 4354.
160. Deist, T.M., et al., Simulation-assisted machine learning. *Bioinformatics*, 2019. **35**(20): p. 4072–4080.
161. Kaur, S., et al., Temporal ordering of omics and multiomic events inferred from time-series data. *NPJ Syst Biol Appl*, 2020. **6**(1): p. 22.
162. Subramanian, I., et al., Multi-omics data integration, interpretation, and its application. *Bioinform Biol Insights*, 2020. **14**: p. 1177932219899051.
163. Goel, G., I.C. Chou, and E.O. Voit, System estimation from metabolic time-series data. *Bioinformatics*, 2008. **24**(21): p. 2505–2511.
164. Dolatshahi, S., L.L. Fonseca, and E.O. Voit, New insights into the complex regulation of the glycolytic pathway in *Lactococcus lactis*. I. Construction and diagnosis of a comprehensive dynamic model. *Mol Biosyst*, 2016. **12**(1): p. 23–36.
165. Dolatshahi, S., L.L. Fonseca, and E.O. Voit, New insights into the complex regulation of the glycolytic pathway in *Lactococcus lactis*. II. Inference of the precisely timed control system regulating glycolysis. *Mol Biosyst*, 2016. **12**(1): p. 37–47.
166. Voit, E.O., A.R. Neves, and H. Santos, The intricate side of systems biology. *Proc Natl Acad Sci U S A*, 2006. **103**(25): p. 9452–9457.
167. Goel, G., I.C. Chou, and E.O. Voit, Biological systems modeling and analysis: a biomolecular technique of the twenty-first century. *J Biomol Tech*, 2006. **17**(4): p. 252–269.
168. WebPlotDigitizer, https://automeris.io/WebPlotDigitizer/, accessed 2023.
169. Lewin, K., *Resolving Social Conflicts & Field Theory in Social Science*. 1997, Washington, DC: American Psychological Association. p. v+422.
170. Savageau, M.A., Demand theory of gene regulation. I. Quantitative development of the theory. *Genetics*, 1998. **149**(4): p. 1665–1676.
171. Kitano, H., Grand challenges in systems physiology. *Front Physiol*, 2010. **1**: p. 3.
172. DARPA-FunBio, *Fundamental Laws of Biology*, http://www.darpa.mil/dso/thrusts/math/funmath/fundamental/index.htm, 2010.
173. DARPA, *Topological Data Analysis*, http://www.darpa.mil/dso/thrusts/math/funmath/tda/index.htm, 2010.
174. Ghrist, R., Barcodes: The persistent topology of data. *Bull Amer Math Soc*, 2008. **45**(1): p. 61–75.

Taylor & Francis
Taylor & Francis Group
http://taylorandfrancis.com

Glossary

The chapter numbers refer to the first substantial discussion of the topic.

Actin
A protein that is one unit of an actin filament in eukaryotic cells. Thin actin filaments and thicker **myosin** filaments are key components of the contractile machinery in skeletal and heart muscle cells. (Chapter 12)

Action potential
A dynamic pattern in voltage exhibited by the membranes of **excitable** cells, including neurons and heart cells. Typical action potentials consist of a fast **depolarization** and a slower repolarization phase. Excitable cells exhibit a **resting potential** between two action potentials. Action potentials are directly associated with neuronal signal transduction, muscle contraction, and heartbeats. (Chapter 12)

Ad hoc model
In contrast to a **canonical model**, a mathematical model that is constructed for a specific purpose and without adhering to a general **modeling** framework. (Chapter 4)

Adaptive
1. In immunology, a system of specialized cells, primarily B and T lymphocytes, that learn to recognize foreign antigens like pathogens, eliminate them or prevent their growth, and remember them in cases of re-infections. The system is activated by the **innate immune system**.
2. In systems theory, the property of a natural **system** or computational **model** of changing its characteristics in response to outside stimuli. (Chapter 10)

Affinity
A measure of the strength of an interaction between atoms or molecules or for the tendency of an atom or molecule to enter into a chemical reaction with unlike atoms or molecules. (Chapter 9)

Agent-based modeling (ABM)
A computational modeling method in which each component (agent) acts according to specific sets of rules. ABM is particularly useful for spatial modeling in heterogeneous environments. (Chapter 11)

Algorithm
A series of computational instructions designed for solving a specific task; usually implemented as computer code or software. (Chapter 6)

Allosteric (regulation)
Regulation of enzyme activity through binding of a ligand to the enzyme outside its active site. (Chapters 8 and 9)

Amino acid
One of 20 molecular building blocks used to generate **peptides** and proteins according to the **genetic code**. (Chapter 8)

Antagonism/Antagonistic
Literally, "working against each other." **Nonlinear** interaction between two **system** components that lessens both components' individual effects. Antonym to **synergism/synergistic**. (Chapter 1)

Antibody
See **Immunoglobulin**. (Chapter 8)

Apoptosis
A natural or induced cellular mechanism resulting in the self-destruction of a cell. Also called "programmed cell death." (Chapter 2)

Approximation
The replacement of a mathematical function with a simpler function. Usually the two have the same value, slope, and possibly higher derivatives at one chosen point, the **operating point**. (Chapter 2)

Arrhythmia
An irregular heartbeat or contraction pattern. (Chapter 12)

Artificial neural network (ANN)
A **machine learning algorithm**, inspired by natural neural networks, that can be trained to classify data and model complex relationships between inputs and outputs. For instance, upon successful training, an ANN may be able to distinguish healthy and cancer cells, based on images or **biomarker** profiles. (Chapter 13)

Atrioventricular (AV) node
A relatively small group of cells with **pacemaker** potential in the wall of the right **atrium** of the heart. Normally, the AV node receives electrical signals from the **sino-atrial node** and directs them through the **bundle of His** and the **Purkinje fibers** throughout the **ventricles**. (Chapter 12)

Atrium (pl. Atria)
One of the two upper, smaller chambers of the heart. See also **Ventricle**. (Chapter 12)

Attractor
In the field of dynamical systems, a point or set of points to which **trajectories** converge. The most typical attractors are stable steady-state points and stable **limit cycles**. In certain **chaotic** systems one talks of "strange attractors." The antonym is a **repeller**. (Chapter 5)

Automaticity
Term for the ability of **sino-atrial (SA) node** cells in the heart to **depolarize** spontaneously, thereby serving as **pacemaker cells**. (Chapter 12)

Autonomic nervous system
Part of the brain that, largely without conscious input, controls the function of most inner organs. (Chapter 12)

Auto-regulation
Regulation of an **adaptive** biological **system** by itself. An example is blood pressure regulation in the brain. (Chapter 10)

Base pair (bp)
A matching couple of two **nucleotides** on complementary DNA or RNA strands, such as cytosine–guanine (DNA and RNA) and adenine–thymine (DNA) or adenine–uracil (RNA), which are connected through hydrogen bonds. (Chapter 7)

Bayesian (network) inference
A statistical method for deducing from data the most probable connectivity (although not **causality**) between **nodes** in a **network** or **graph** that does not contain cycles. See also **Directed acyclic graph (DAG)**. (Chapter 3)

Behavior (of a system)
A collective term for changes in the **state** of a system, often over time and in response to a **perturbation** or **stimulus**. (Chapter 2)

Bi-fan
A **network motif** where two source **nodes** send **signals** to each of two target nodes. (Chapter 10)

Bifurcation (in a dynamical system)
A critical value of a **parameter** in a **system** (usually of **differential equations**), where the behavior of the system changes **qualitatively**, for instance, from a **stable steady state** to an unstable state surrounded by a stable **limit cycle oscillation**. (Chapter 5)

Biochemical systems theory (BST)
A **canonical** modeling framework that represents biological **systems** with **ordinary differential equations**, in which all processes are described as products of **power-law functions**. See also **Generalized mass action (GMA) system** and **S-system**. (Chapter 5)

Biofilm
A mixed microbial community consisting of many species and collectively forming a thin layer. (Chapter 7)

Bioinformatics
A field of study at the intersection of biology and computer science that organizes, analyzes, and interprets datasets and provides computational support for **-omics** studies. (Chapter 3)

Biologics
Pharmaceutically active molecules, such as **antibodies**, that are derived from naturally occurring biological materials. (Chapter 13)

Biomarker
A biological or clinical measurement that is associated with—or can be used to predict—a specific event (such as a disease). Some biomarkers are causative, while others are merely symptomatic. (Chapter 13)

Bistability
The property of a system to possess two **stable steady states** (and at least one unstable steady state). (Chapter 10)

Boolean model of a system
Typically a **discrete** model, in which the **state** of a **multivariate** system at time $t + 1$ is determined by a logical function of the states of the system components at time t. (Chapters 3, 7, 10)

Bootstrap
A statistical method for determining desired features of a dataset of size n. It is based on constructing and analyzing many samples of size n, whose data are randomly chosen one-by-one from the dataset and replaced before the next datum is chosen. See also **Jackknife**. (Chapter 6)

Branch-and-bound method
One of several sophisticated **search algorithms** that find **global optima** by systematically and iteratively subdividing the **parameter** space and discarding regions that cannot contain the solution. (Chapter 6)

Bundle of His
A collection of heart cells specialized for the rapid conduction of electrical signals, functionally connecting the **atrioventricular node** and the **Purkinje fibers**. (Chapter 12)

Calcium-induced calcium release (CICR)
An amplification mechanism in **cardiomyocytes**, in which a small amount of calcium input to the cell leads to a large efflux of calcium from the **sarcoplasmic reticulum** into the cytosol, where it leads to contraction. (Chapter 12)

Canonical model
A mathematical modeling format that follows strict construction rules. Examples are **linear** models, **Lotka–Volterra** models, and models within **biochemical systems theory**. (Chapter 5)

Carcinogenesis
The process of cancer development. (Chapter 15)

Cardiac
Related to the heart. (Chapter 12)

Cardiomyocyte
A contractible heart muscle cell. See also **Excitable cell** and **Pacemaker cell**. (Chapter 12)

Carrier protein
A protein transporting specific **ions** and other small molecules between the inside and outside of a cell. The required energy is often supplied in the form of ATP. Also called a pump or permease. (Chapter 12)

Carrying capacity
The number of individuals an environment can sustainably accommodate. See also **crowding** and **logistic growth**. (Chapter 14)

Causality analysis
A statistical assessment of the likelihood that one component (or condition) of a **system** directly controls or causes changes in another component (or condition). (Chapter 3)

cDNA
Complementary DNA; generated from mRNA by the action of the **enzymes** reverse transcriptase and DNA polymerase. (Chapter 7)

Cell cycle
The sequence of intracellular events that ultimately lead to the division and duplication of a cell. (Chapter 3)

Central Dogma (of molecular biology)
The general (and somewhat simplistic) concept underlying the unidirectional information transfer from DNA to mRNA (**transcription**) and then to protein (**translation**). (Chapter 7)

Central limit theorem
A fundamental law of statistics stating that the sum of many random variables typically approaches a normal (Gaussian) distribution (bell curve). (Chapter 12)

Channel protein
A protein spanning a cell membrane and forming a hydrophilic pore that permits very fast travel of specific **ions** across the membrane. (Chapter 12)

Chaos/chaotic
Without order or predictability. In the terminology of **systems** theory, **deterministic** chaos refers to a phenomenon of some **nonlinear dynamic systems** that are completely deterministic and numerically characterized yet exhibit erratic, unpredictable **behaviors**. Two **simulations** of these systems with different—but arbitrarily close—**initial values** eventually show no more similarities in their responses than two simulations with very different initial values. (Chapter 5)

Chaperone
A protein that assists in the correct folding of proteins and the assembly of macromolecular structures. Chaperones also protect the three-dimensional **conformation** of proteins in conditions of environmental stress. Heat-shock proteins are chaperones that protect proteins against misfolding due to high temperatures. (Chapter 8)

Chemical master equation
A detailed **ordinary differential equation** description of a chemical **reaction** that accounts for the **stochastic** nature of every molecular event associated with this reaction. (Chapter 9)

***Cis*-regulatory DNA**
A region of DNA that regulates the expression of a gene located on the same DNA molecule, for instance, by coding for a **transcription factor** or serving as operator. (Chapter 7)

Clique
A part of an undirected **network** or **graph** where all **nodes** are neighbors. That is, all nodes are directly connected to each other. (Chapter 3)

Closed system
In contrast to an **open system**, a mathematical **model** without input or loss of material. (Chapter 2)

Clustering coefficient
A quantitative measure for the connectivity of a **network**, either throughout the entire network or within the neighborhood of a given **node**. (Chapter 3)

Codon
A triplet of **nucleotides** that, when translated, represent an **amino acid**. (Chapter 7)

Coefficient
A (typically constant) multiplicative factor (number) in some term of an equation or a matrix. See also **Linear**. (Chapter 5)

Compartment
A defined space that is separated from its surroundings, for instance, by a membrane. See also **Compartment model**. (Chapter 5)

Compartment model
Usually a **linear ordinary differential equation model** of the **dynamics** of material moving between **compartments**, such as the organs and the bloodstream in an organism. See also **Pharmacokinetic model**. (Chapter 5)

Complexity
An ill-defined conglomerate of features of a **system** that includes its number of components, processes, and regulatory **signals**, as well as the degree of **nonlinearity** and unpredictability of the system. (Chapter 2)

Concept-map modeling
A strategy for converting biological diagrams and coarse input-response information into **dynamic models**. (Chapter 15)

Conductance (electrical)
A measure for how easily electricity can move through some material. (Chapter 12)

Conformation
The three-dimensional arrangement of atoms in a large molecule like a protein. The conformation can change with the molecule's chemical and physical environment. See also **Tertiary structure**. (Chapter 8)

Constraint (optimization)
Any type of threshold that a component, subject to **optimization**, is not permitted to cross; a condition that variables must satisfy in an optimization problem. (Chapter 3)

Continuous
Antonym to **discrete**. The feature of a function or set not to have holes or interruptions; in systems biology usually with respect to time and sometimes with respect to space. (Chapter 5)

Continuum approximation
An approximation of the mechanical features of an array of many similar objects with the mechanics of a continuous material; used, for instance, in modeling heart tissue. (Chapter 12)

Control coefficient
A quantitative measure of **steady-state sensitivity** in **metabolic control analysis**. (Chapter 3)

Controlled comparison
A mathematical technique for revealing **design principles**, based on comparing the **designs** of two similar **systems** in a minimally biased fashion. (Chapter 6)

Cooperativity
The phenomenon that **enzymes** or **receptors** may have multiple binding sites, which may increase (positive cooperativity) or decrease (negative cooperativity) their **affinity. Hill models** are often used to represent this phenomenon. See also **Allosteric** (regulation). (Chapters 8 and 9)

Coronaries
Arteries and veins supplying the heart muscle with blood circulation. (Chapter 12)

Correlative model
A **model** that relates one quantity to another, without explaining what causes the relationship or what the relationship means. See also **Explanatory model**. (Chapter 2)

Crosstalk
Information exchange between two processes or pathways, typically in **signal transduction**. (Chapter 10)

Crowding
A collective term for impeding interactions within a growing population, such as the competition for resources that lead to a slowing down of population growth, often until the **carrying capacity** is reached. (Chapter 14)

Curated (database)
Referring to the fact that information (in the database) was checked by an expert. (Chapter 9)

Cytokine
A signaling protein involved in responses to infections and inflammation. (Chapters 8 and 15)

Damped oscillation
An oscillation that subsides over time. See also **Limit cycle** and **Overshoot**. (Chapter 5)

Data mining
The process of extracting patterns from large datasets and forging them into useful information by combining methods from **machine learning**, artificial intelligence, statistics, and database management. See also **Text mining**. (Chapter 15)

Degree distribution
Statistical distribution of the number of **edges** per **node** in a **graph** or **network**. (Chapter 3)

Denature
To change or destroy the structure of a macromolecule, typically a protein or nucleic acid, by applying extreme physical or chemical stress. Denatured proteins are often removed by the **proteasome**. (Chapter 8)

Dependent variable
A **system** variable that is potentially affected by the **dynamics** of the system. See also **Independent variable**. (Chapter 2)

Depolarization
A rapid change, from negative to positive voltage, in the membrane potential of neurons, muscle cells, and heart cells during an **action potential**; it is followed by **repolarization**. (Chapter 12)

Design
1. Of a natural **system**. Similar to a network **motif**, the structural and regulatory composition of a **dynamic** system as it is observed. The observed design is sometimes analyzed in comparison with hypothetical alternatives, in order to reveal **design** and **operating principles**. (Chapter 2)
2. Of an artificial biological system. The core subject of **synthetic biology**. (Chapter 6)
3. Of an experiment. A set of statistical techniques assuring that results from the experiment will likely be statistically significant. (Chapter 6)

Design principle
A structural or **dynamic** feature of a **system** that, due to its functional superiority, is found more often than alternative designs. See also **Motif** and **Operating principle**. (Chapter 6)

Deterministic (model)
A model that does not allow or account for **random** events. See also **Stochastic** model and deterministic **Chaos**. (Chapter 2)

Diastole
Greek term for dilation and used to describe the relaxed state of the heart. See also **Systole**. (Chapter 12)

Difference equation
A **discrete** function in which the **state** of a **system** is defined as a function of its state at one or more previous time points. See also **Iteration** and **Boolean model**. (Chapter 4)

Differential equation
An equation or system of equations containing functions and their derivatives. Most **dynamical systems** are represented as differential equations. See also **Ordinary differential equation** and **Partial differential equation**. (Chapters 1, 2, and 5)

Differentiation
1. In developmental biology, the natural process of converting a cell with many possible roles or fates, such as a fertilized egg or stem cell, into a cell with one or a few specific functions. (Chapter 15)
2. In mathematics, the process of computing a derivative (slope) of a function. (Chapter 5)

Digital Twin
See Virtual Twin.

Directed acyclic graph (DAG)
A **graph** in which all **edges** have a unique direction and which does not allow **feedback** or the return to any **node**, once this node had been left. (Chapter 3)

Discrete
Antonym to **continuous**. The feature of a function or set to exhibit interruptions. For instance, a discrete timescale is defined for distinct time points t_1, t_2, t_3, . . . , but not in between, like the display of a digital watch. (Chapter 4)

DNA chip
Similar to a **microarray**, a tool for comparing gene expression in different cell types or under different conditions. In contrast to a microarray, a DNA chip contains artificially constructed, short nucleotide probes. (Chapter 7)

Dyad
A small space between the cell membrane and the **sarcoplasmic reticulum** in **cardiomyocytes**, where **calcium-induced calcium release** is controlled. (Chapter 12)

Dynamical system or model
A **system**—or **model** of a system—whose **state** and **behavior** changes over time. (Chapter 2)

Dynamics (of a model)
The collection of changes in **model** features over time. (Chapter 2)

Edge (in a graph)
The directed or undirected connection between two **nodes** in a **graph** or between a node and the **environment** of the **graph**. (Chapter 3)

Eigenvalue
One of a group of complex numbers that characterize various aspects of a linear or linearized **system**, such as its **stability** at a **steady state** and the different speeds with which different variables evolve over time. (Chapter 4)

Elasticity
A quantitative measure of the **sensitivity** of an **enzyme** or **reaction** to changes in the local environment, for instance, induced by changes in substrate; used in **metabolic control analysis**. Equivalent to the concept of a **kinetic order** in **biochemical systems theory**. (Chapter 9)

Electrocardiogram (ECG, EKG)
A reading from several pairs of electrodes, attached to the torso and limbs, of the electrical activity of the heart. (Chapter 12)

Electrochemical
Related to electrical events associated with **ions**. (Chapter 12)

Electrophysiology
Study of the normal functioning of electrical events in the body. (Chapter 12)

Elementary mode analysis (EMA)
A method for determining independent reaction paths within a **stoichiometric network**. (Chapter 9)

Emergent property
A property of a **system** that cannot be assigned to the features of a single component but only becomes possible from **nonlinear** interactions between the system components. See also **Superposition**. (Chapter 1)

Environment (of a model, graph or system)
Components and processes outside the explicit definition of a **model**, **graph**, or **system**. (Chapter 2)

Enzyme
A protein that catalyzes (facilitates) a biochemical **reaction**. (Chapter 8)

Epidemic
An infectious disease that sickens large percentages of a population. An extremely widespread epidemic is called a pandemic. (Chapter 14)

Epigenetics
The relatively new field of studying heritable features that are not due to changes in the **nucleotide** sequence of an organism's DNA. (Chapter 7)

Error (in modeling and statistics)
See **Residual**. (Chapter 6)

Evolutionary algorithm
One of several **machine learning** methods that, inspired by natural evolution, search for optimal and near-optimal solutions. See also **Genetic algorithm** and **Optimization**. (Chapter 6)

Excitable (cell)
A neuron or muscle cell that responds to **electrochemical** stimulation. See also **Action potential**. (Chapter 12)

Excitation–contraction (EC) coupling
A fundamental process in muscle physiology, where an electrical stimulus (usually an **action potential**) is converted into an increase in calcium concentration within the cytosol of a muscle cell, which causes a sliding motion between **actin** and **myosin filaments** and leads to contraction of the cell. (Chapter 12)

Exon
A segment of a gene sequence that is transcribed into mRNA and subsequently translated into protein. See also **Intron**. (Chapter 7)

Explanatory model
A conceptual or mathematical **model** that explains a phenomenon (e.g., development) based on events at lower levels of organization (e.g., gene expression). (Chapter 2)

Exponential growth
A model of growth—or **growth law**—that can be described by an exponential function with a positive **growth rate**. The corresponding process with a negative rate is called exponential decay. (Chapters 4 and 14)

Exposome
A new field of research attempting to assess health risks due to a combination of environmental factors, genetic predisposition, and lifestyle. (Chapter 15)

Expressed sequence tag (EST)
A gene fragment of about 300 nucleotides that has been sequenced from **cDNA**. (Chapter 7)

Extrapolation
The prediction of **model** responses outside the range for which actual data are available. (Chapter 2)

False negative
The result of a test indicating a negative outcome even though the outcome is in actuality positive. For instance, a test may suggest the absence of disease even though the person has the disease. See also **False positive**. (Chapter 13)

False positive
The result of a test indicating a positive outcome even though the outcome is in actuality negative. For instance, a test may suggest cancer even though the person does not have cancer. See also **False negative**. (Chapter 13)

Feature selection
A technique in statistics and **machine learning** for reducing the number of **variables** to a subset of most influential variables. (Chapter 6)

Feedback
A **signal** affecting a preceding reaction step or process in a generic pathway. (Chapter 2)

Feedforward
A **signal** affecting a downstream reaction step or process within a generic pathway. (Chapter 5)

Filament (muscle cell)
A microfiber composed of proteins like **actin** and **myosin**, which endows muscle cells with contractility. (Chapter 12)

Finite element approach
A set of methods for assessing the dynamics of fluids, materials, or solid bodies, based on subdividing these into very small units (elements) and studying the behavior of and between these elements. Often used to analyze **partial differential equations** as well as changes within bodies that have complicated shapes. (Chapter 12)

Fitness
1. In genetics, a quantitative measure of superiority of a gene, its gene product, or the organism possessing the gene.
2. In **Optimization**, a quantitative measure of superiority of one solution over another. (Chapter 6)

Fitting (of a model)
The automatic or manual adjustment of **parameter** values so that a **model** matches observation data as closely as possible. See also **parameter estimation**, **optimization**, **residual**, and **validation**. (Chapter 6)

Fixed point (in differential equations)
A time-invariant solution. See **steady state**. (Chapter 4)

Fluid mechanics
The study of the movement of and within liquids, often using **partial differential equations** and **finite element approaches**. (Chapter 12)

Flux
The amount of material (number of molecules) flowing through a pool (**variable**), **pathway**, or **system** during a defined time period. (Chapter 3)

Flux balance analysis (FBA)
A modeling approach for (large) metabolic **networks**, which at a **steady state** balances metabolic **fluxes** entering and leaving **nodes**—according to the **stoichiometry** of each reaction—and which assumes that the cell or organism optimizes some feature, such as its growth rate. See also **Metabolic flux analysis**. (Chapter 9)

Frank–Starling law
A relationship among blood volume, pressures, forces, and wall tensions in the heart. (Chapter 12)

Funny current
Relatively slow, mixed sodium–potassium current, entering a cell of the **sino-atrial node** and triggering spontaneous **depolarization**. (Chapter 12)

Fuzzy (logic)
A set of logical rules that are applied in a **probabilistic** (rather than a **deterministic**) fashion. (Chapter 15)

Gain
The **sensitivity** of a system response with respect to a small change in an **independent variable**. (Chapter 5)

Gap junction
A channel that connects the cytosols of two neighboring cells. The channel consists of two protein complexes that form half-channels spanning the cell membranes. The two half-channels meet in the extracellular space between the neighboring cells. See also **Transmembrane protein**. (Chapter 8)

Gate (logic)
A logical conjunction between two signals. In the AND gate, the signal is only propagated if both incoming signals are active. (Chapter 3)

Gene annotation
The assignment of a protein or function to a gene sequence. See also **Gene ontology**. (Chapter 7)

Gene ontology (GO)
A widely accepted, controlled vocabulary for **gene annotation**. (Chapter 7)

Generalized mass action (GMA) system
A modeling format within **biochemical systems theory**, based on **ordinary differential equations**, in which every process is represented with a product of **power-law functions**. See also **S-system**. (Chapter 5)

Genetic algorithm
A **machine learning** method that, inspired by the process of genetic inheritance, recombination, and mutation, searches for optimal and near-optimal solutions. See also **Evolutionary algorithm** and **Optimization**. (Chapter 6)

Genetic code
The assignment of each natural amino acid to one or more **codons** of **nucleotides** in DNA or RNA. Because of the near-universality of the genetic code, amino acid sequences in peptides and proteins can reliably be predicted from their coding genes. (Chapter 7)

Genetic engineering
The targeted manipulation of genes toward a goal of (industrial) interest. See also **Metabolic engineering** and **Synthetic biology**. (Chapter 7)

Genome
The totality of DNA in an organism. In addition to genes coding for proteins, much of the DNA is used for other purposes, including information regarding regulatory RNA. See also **Proteome** and **-ome**. (Chapter 7)

GFP
Green fluorescent protein. A protein whose DNA sequence can be inserted in frame with the protein-coding region of a target gene, permitting the localization of gene expression through a fluorescence assay. Fluorescent proteins with many other colors are available now. (Chapter 8)

Global optimum
The best possible point (with respect to some defined criterion or metric). Finding the global optimum for a **nonlinear system** is a difficult **optimization** problem of computer science. See also **Local optimum** and **Branch-and-bound method**. (Chapter 6)

Goldman–Hodgkin–Katz equation
An equation describing the equilibrium potential of a cell membrane as a function of internal and external **ion** concentrations and permeability constants. (Chapter 12)

G-protein
One of several specific proteins on the inside of the cell membrane that interact with **G-protein-coupled receptors** in numerous **signal transduction** processes. (Chapter 10)

G-protein-coupled receptor (GPCR)
A class of membrane-associated **receptors** that are involved in uncounted **signal transduction** and disease processes. (Chapter 10)

Graph
A visual or mathematical representation of a **network** consisting of **nodes** (**vertices**) and **edges** connecting some of the nodes with each other. (Chapter 3)

Graph theory
A branch of mathematics concerned with **graphs**. Graph theory has found rich applications in biological **network** analysis. (Chapter 3)

Growth law
A typically simple function describing changes in the size of a cell, organism, or population over time. The best-known examples are **exponential** and **logistic growth**, but many others exist. (Chapter 14)

Growth rate
Parameter in a **growth law** characterizing the speed of growth. (Chapter 14)

Half-life
The time it takes for the amount of some material to degrade from 100% to 50%, under the assumption of exponential decay. (Chapter 11)

Heat shock protein
One of a class of specific proteins that are activated under heat stress and protect other proteins from unfolding or **denaturing**. (Chapter 11)

Hill function
A type of **kinetic rate law** that is well suited for **modeling allosteric** processes and **cooperativity** involving n binding sites of an **enzyme**. The number n is called the Hill coefficient; $n > 1$ means positive cooperativity, $n < 1$ negative cooperativity. For $n = 1$, the Hill function becomes a **Michaelis–Menten rate law**. Often used to describe sigmoidal behavior. (Chapter 9)

Hit (drug development)
A natural or artificially created molecule, investigated during drug discovery and development, that interacts with a drug target and interrupts a disease process. (Chapter 13)

Hodgkin–Huxley model
The first mathematical **model** describing electrical activity, transmembrane currents, and action potentials in neurons and other excitable cells. (Chapter 12)

Homeostasis
Maintenance of the normal (steady) **state** of an organism or physiological **system**, which is achieved through the action of often complex internal control systems. See also **Allostasis**. (Chapter 13)

Homogeneous
Possessing the same features throughout. (Chapter 14)

Hub
A particularly well connected—and usually important—**node** within a **graph, network**, or **system**. (Chapter 3)

Hybridization
A core technique in numerous experimental methods for the identification of identical or similar DNA or RNA **nucleotide** sequences; based on the fact that nucleotide strands specifically bind to their complementary sequence through **base pairs**, such as cytosine and guanine. (Chapter 7)

Hysteresis
A property of **nonlinear systems** with two or more **stable steady states** where the **transient** toward one of these states depends on the recent history of the **dynamics** of the system. (Chapter 10)

Identifiability
A property of a **system** and a dataset indicating whether the system structure or **parameters** can be uniquely determined from the dataset. See also **Sloppiness**. (Chapter 6)

Immunoglobulin
Also called an **antibody**. A protein produced by white blood cells that is either attached to a B cell of the immune system or circulates through the blood and other bodily fluids. Immunoglobulins detect and neutralize invaders such as bacteria, viruses, and foreign proteins. (Chapter 8)

Immunoprecipitation
A technique for concentrating, isolating, and purifying proteins from a cell extract or protein mixture by means of specific **antibodies**. The protein–antibody complex is extracted from the solution, the antibodies are removed, and the protein can be analyzed. (Chapter 8)

Independent variable
A system variable that is not affected by the **dynamics** of the **system** itself but potentially by the environment or the experimenter. In many cases, an independent variable is constant during a given experiment. See also **Dependent variable**. (Chapter 2)

Initial value
A numerical start value that needs to be defined for every **dependent variable** in a system of **differential equations**, before the system can be solved numerically. (Chapter 5)

Innate (immune system)
A generic defense system in a variety of multicellular organisms that protects the host from infection by pathogens. In vertebrates, cells of the innate immune system, such as macrophages and neutrophils, rush to the site of infection and destroy invaders. The innate immune system activates the **adaptive immune system**, which learns to recognize and remember pathogens. (Chapter 15)

Intron
A segment of a gene sequence that is transcribed into RNA but later spliced out from the mature RNA. In protein coding genes, introns are not translated into protein. See also **Exon**. (Chapter 7)

Ion
An atom or molecule that carries an electrical charge due to a difference between its numbers of electrons and protons. (Chapter 12)

Irreversible (reaction)
Feature of a (biochemical) **reaction** not permitting a back-reaction. Absolutely irreversible reactions are rare, but the reverse direction is often negligible because of unfavorable energetics. See also **Reversible reaction**. (Chapter 9)

Isozyme
One of several different **enzymes** that catalyze the same biochemical **reaction**(s), sometimes with different efficiency and sensitivity to inhibitors or modulators. (Chapter 9)

Iteration
A recurring step or sequence of steps in an **algorithm** or **simulation**, usually with slightly changed settings. (Chapter 2)

Jackknife
Similar to the **bootstrap** method, the jackknife is a statistical method for determining desired features of a dataset. It is based on the repeated estimation of the desired feature when one or more data points are randomly left out. (Chapter 6)

Jacobian
A **matrix**, consisting of partial derivatives, that **approximately** represents the **dynamics** of a **nonlinear system** close to some **operating point**. (Chapter 5)

Kinetic order
1. In elemental chemical kinetics, the number of molecules of the same species undergoing a **reaction**.
2. In **biochemical systems theory**, the exponent of a **variable** in a **power-law function**, quantifying the effect of the variable on the process represented by the function; equivalent to the concept of **elasticity** in **metabolic control analysis**. (Chapter 5)

Kinetic rate law
A function describing how the **dynamics** of a biochemical **reaction** is affected by substrates, enzyme, and modulators. See also **Michaelis–Menten kinetics** and **Hill function**. (Chapter 9)

Kinetics (in biochemistry)
Pertaining to the rates or **dynamics** of a biochemical **reaction**. (Chapter 9)

Latin square or hypercube sampling
A statistical design and sampling method that circumvents large numbers of analyses that would be necessary if every combination of numbers in a complete grid were used. Instead, only one number per row and per column of the grid is selected. In higher dimensions, the grid is replaced with a cube or hypercube, where the latter is the analog to a cube in more than three dimensions. (Chapter 6)

Lead (drug development)
A potential drug molecule, such as a **hit** with improved potency, that interrupts a disease process and has desirable properties related to efficacy, **pharmacokinetics**, and side effects. (Chapter 13)

Least-squares method
A statistical **regression** technique that optimally matches a **model** to a given dataset. Optimality is defined based on the minimum of the sum of squared differences between each data point and the corresponding model value. See also **Residual**. (Chapter 6)

Limit cycle
An **attractor** or **repeller** in the form of an **oscillation** in a **dynamical system** whose amplitude does not change over time. In the case of a **stable** (unstable) limit cycle, nearby **oscillations** eventually converge to (diverge away from) the attractor (repeller). (Chapter 5)

Linear
The mathematical property of a function or **system** to be describable with sums of variables that are multiplied with constant **coefficients** (numbers). If f is a linear function of x, and x_1 and x_2 are any values of x, then the **superposition principle** $f(x_1 + x_2) = f(x_1) + f(x_2)$ holds. The antonym is a **nonlinear** function or system, where the aforementioned relationships do not hold. (Chapter 2)

Linearization
The **linear approximation** of a system at a chosen **operating point**. The linearization is characterized by the **Jacobian** matrix. (Chapter 5)

Local minimum (optimum)
A **state** of a **system** that is better (with respect to some defined criterion or metric) than all states close by. However, farther away, the system may have even better states; see also **Global optimum**. Local optima often prevent computer **algorithms** from finding the global optimum. (Chapter 6)

Logistic growth
A sigmoidal model of growth—or **growth law**—that is initially exponential but ultimately approaches an upper value, called the **carrying capacity**. The deviation from **exponential growth** is often attributed to **crowding**. (Chapter 14)

Lotka–Volterra model
A **canonical model**, based on **ordinary differential equations** and used primarily in ecological **systems** analysis, where all processes are represented as products of two variables and a rate **coefficient**. Each equation j may also contain just the jth variable with a coefficient. (Chapter 5)

Machine learning
A computational branch of the field of artificial intelligence, which trains **algorithms** to classify data according to given criteria and to reveal patterns within complex datasets. See also **Parameter estimation**, **Data mining**, and **Text mining**. (Chapter 6)

MAPK
Mitogen-activated protein kinase. One of a class of **enzymes** at the core of many **signaling cascades** in eukaryotes. Typically, phosphorylation events of different MAPKs occur at three hierarchical levels, thereby transducing a signal, for instance, from the cell membrane to the nucleus. (Chapter 10)

Markov model
Usually a **discrete**, **probabilistic** mathematical **model**, in which the **state** of a **system** is determined by the previous state of the system and a (fixed) set of transition probabilities for moving the system from any one state to any other state. (Chapter 4)

Mass spectrometry (MS)
An experimental technique used to characterize mixtures of molecules based on their mass and an applied electrical charge. (Chapter 8)

Mass-action (kinetics)
A simple quantitative description of the **dynamics** of chemical **reactions**, consisting of the product of a **rate constant** and all substrates involved in the reaction. If two substrate molecules of the same species are needed for the reaction, the substrate enters the mass-action function with a power of 2. See also **Kinetics** and **Generalized mass action system**. (Chapter 9)

Matrix (pl. Matrices)
1. In linear algebra: an array of numbers that typically correspond to the **coefficients** in a set of **linear** equations; see also **Vector**. Many aspects within the fields of linear algebra, **multivariate** calculus, **systems** analysis, and statistics are based on vectors and matrices. (Chapters 3 and 4)
2. In biology: extracellular proteins, such as collagen, laminin, and fibrinogen, that are deposited to form a network that structurally supports tissues. (Chapter 8)

Metabolic burden
The cost to the organism for maintaining certain levels of mRNAs, proteins, and metabolic products, especially if they are expendable. (Chapter 9)

Metabolic control analysis (MCA)
A framework of theorems and methods, based on **sensitivity analysis**, for the assessment of control in metabolic, genetic, and signaling **systems** at **steady state**. (Chapter 9)

Metabolic engineering
A set of techniques from molecular biology, genetics, and engineering for manipulating a culture of microorganisms toward a desired goal, such as the production of a valuable **amino acid**. (Chapter 9)

Metabolic flux analysis
A collection of experimental (labeling) and computational (**flux balance analysis**-like) techniques for the assessment of flux distributions in metabolic **networks**. (Chapter 9)

Metabolic pathway
Any prominent sequence of (typically **enzyme**-catalyzed) reaction steps, involving the synthesis or degradation of organic molecules. Metabolic pathways are often used to organize the complexity of metabolic **networks**. (Chapter 9)

Metabolic profile
A comprehensive set of metabolite concentrations characterizing a **metabolic pathway** or organism at one time point or a series of time points. (Chapter 9)

Metabolomics
The simultaneous study of many or all metabolites in a system. See also **-ome** and **-omics**. (Chapter 9)

Metagenomics
Methods of genomics applied to a mixed microbial community. A collective term for simultaneous **genome** analyses of many (coexisting) species in environments such as soil, the oceans, sewage pipes, or the human gut. (Chapter 15)

Michaelis–Menten mechanism (rate law)
A **conceptual** and mathematical **model** (**rate law**) of an **enzyme**-catalyzed **reaction**, in which substrate and enzyme reversibly first form an intermediate complex, before the product is generated and the enzyme recycled. A suitable **quasi-steady-state assumption** converts the describing system of **mass-action differential equations** into a simple function that quantifies the rate of product formation in terms of the substrate concentration. Its two parameters are V_{max}—the maximal rate of product formation—and the Michaelis constant KM, which corresponds to the substrate concentration for which this rate is $V_{max}/2$. A mathematical generalization is the **Hill function**. (Chapter 9)

Microarray (technology)
A set of methods for comparing gene expression in different cell types or under different conditions. Each microarray consists of a solid surface that holds very small probes of known DNA (or sometimes other molecules). See also **DNA chip** and **RNA-Seq**. (Chapter 7)

Microbiome
A mixed microbial community, usually living within the human body; for instance, in the oral cavity, vagina, or gut. (Chapter 15)

MicroRNA
A short **noncoding** RNA of between 20 and 30 nucleotides that serves a number of regulatory roles, including the silencing of genes. See also **Small RNA**. (Chapter 7)

Model (in biology)
1. In biological experimentation, the use of a particular species as a representative for a larger domain of organisms, such as a "mouse model."
2. Conceptual: a formalized idea of how components of a **system** might be related to and interact with each other.

3. Mathematical: a set of functions or (**differential**) **equations** that in some manner represent a biological phenomenon.
4. Computational: the use of **algorithms** supporting the **modeling** of a biological phenomenon. (Chapter 2)

Modeling (in biology)
The conversion of a biological phenomenon into a mathematical or computational construct and the subsequent analysis of this construct. (Chapter 2)

Module
A more or less confined or autonomous (sub-)**model** that is part of a larger model. (Chapter 2)

Molecular dynamics
A computationally intensive **simulation** technique for exploring the location and motion of each atom in a large molecule such as a protein. (Chapter 8)

Monte Carlo simulation
A statistical technique for exploring possible, likely, and worst-case **behaviors** of **complex systems** that is based on thousands of **simulations** with different randomized combinations of model settings. (Chapter 2)

Motif
A recurring structural feature of a **network** or **system** that is found more often than one would expect in a corresponding, **randomly** composed system. See also **Design principle**. (Chapter 6)

Multiscale modeling
The construction of **models** that simultaneously capture essential **system** features and **behaviors** at different temporal and organizational scales, such as gene expression, **physiology**, and aging. (Chapter 11)

Multistage cancer model
One of several mathematical **models** formalizing the assumption that a normal cell experiences several **discrete** changes before becoming a cancer cell. (Chapter 15)

Multivariate
Simultaneously depending on several **variables**. (Chapter 2)

Myocardium
Heart muscle. (Chapter 12)

Myocyte
Muscle cell. In the heart, it is called a cardiac myocyte or **cardiomyocyte**. See also **Myofibril**. (Chapter 12)

Myofibril
The basic contractile unit of a **myocyte**, consisting of **actin**, **myosin**, and other proteins. (Chapter 12)

Myosin
A motor protein that forms filaments in eukaryotic cells. Together with **actin** filaments, myosin filaments are key components of the contractile machinery in skeletal and heart muscle cells. (Chapter 12)

Na^+–$Ca2^+$ exchanger
A transport protein for ions, which exchanges calcium inside an **excitable cell** against sodium; in particular, after an **action potential**. (Chapter 12)

Na^+–K^+ pump
A transport protein that pumps Na^+ out of an **excitable cell**, while K^+ is pumped in. The pump can also run in reverse. (Chapter 12)

Nernst equation
An equation governing properties of **electrochemical** gradients. It describes free energy in terms of differences in **ion** concentrations across a membrane. (Chapter 12)

Network
A collection of **nodes** (**vertices**) and connecting **edges**. (Chapter 3)

Network motif
See **Motif**. (Chapter 6)

Neurotransmitter
A chemical such as dopamine or serotonin that is responsible for the transmission of specific **signals** between neurons across a synapse. (Chapter 15)

New chemical entity (NCE)
A potential drug molecule, intensely investigated in the early phases of drug development, that had not been approved by the FDA in any prior application. (Chapter 13)

NMR spectroscopy
Abbreviation for nuclear magnetic resonance spectroscopy. A technique that exploits the fact that atomic nuclei with an odd number of protons and neutrons spin on their own axes. This spin is assessed with strong magnets and can be interpreted to yield the structure of molecules, such as proteins. The same method can be used for noninvasive in vivo analyses of labeled metabolites. (Chapters 8 and 9)

Node
A **variable** or pool within a **network** or **graph**, which is connected by **edges** with other nodes in a graph or with the **environment** of the graph. Also called a **vertex**. (Chapter 3)

Noise
Summary expression for unexplained, apparently **random** errors in data. Similarly, unexplained, rapid fluctuations in a signal. See also **Residual**. (Chapters 6 and 10)

Noncoding DNA/RNA
DNA or RNA that is ultimately not translated into protein. (Chapter 7)

Nonlinear
Antonym of **linear**. The property of a function or **system** to have a curved appearance. If f is a nonlinear function of x and if x_1 and x_2 are values of x, then $f(x_1 + x_2)$ is generally not equal to $f(x_1) + f(x_2)$. See also **Superposition** and **Synergism**. (Chapter 2)

Nonlinear system
Any mathematical **model** that exhibits **nonlinear** features (possibly in addition to **linear** features). (Chapter 4)

Nucleoside
A building block of DNA or RNA, consisting of a base and a sugar (deoxyribose or ribose, respectively). If a phosphate group is attached as well, a nucleoside becomes a **nucleotide**. (Chapter 7)

Nucleotide
A building block of DNA or RNA, consisting of a base, a sugar (deoxyribose or ribose, respectively), and a phosphate group. DNA bases are adenine (A), cytosine (C), guanine (G), and thymine (T). In RNA, thymine is replaced by uracil (U). (Chapter 7)

Nullcline
A set of points in a **phase-plane plot** containing all situations where a particular **variable** has a slope ("cline") of zero ("null") and therefore does not change. Nullclines divide the space of **system** responses into sections of qualitatively similar **behaviors**. (Chapter 14)

Objective function
A function that is maximized (or minimized) in an **optimization** task. In **parameter estimation**, the objective function is often the sum of squared errors between a **model** and the modeled data. See also **Residual**. (Chapter 6)

-ome, -omics
Suffix added to words in order to indicate comprehensiveness or totality. Thus, the **proteome** consists of all proteins in a cell or organism. One of the oldest -omes is the rhizome, the totality of all roots of a plant. -ome is similar to the suffix -oma, which is used to describe tumors. Some new word concoctions using -ome and -omics are quite silly. See also **Genome**. (Chapter 7)

Ontology
A controlled vocabulary with generally accepted definitions of terms within a limited domain of knowledge. A well-known example is **gene ontology**. See also **Gene annotation**. (Chapter 7)

Open system
In contrast to a **closed system**, a mathematical **model** with input and/or output. (Chapter 2)

Operating point
The point at which a function and an **approximation** have the same value, slope, and possibly higher derivatives. (Chapter 5)

Operating principle
The dynamic analog to a **design principle** or **motif**: a natural strategy for solving a task that is superior to alternative strategies. (Chapter 6)

Operon
The clustered arrangement of (bacterial) genes under the control of the same promoter. Operons often lead to co-expression of functionally related proteins. (Chapter 7)

Optimization
The mathematical or computational process of making a **system** behave as closely as possible to a given objective, for example, by minimizing the error between some output of the model and the desired output. See also **Objective function**, **Parameter estimation**, and **Residual**. (Chapter 6)

Orbit
A closed **trajectory**. If a dynamical system starts at some point of an orbit, it will re-enter this point infinitely often again. An example is a **limit cycle**. (Chapter 5)

Ordinary differential equation (ODE)
A type of **differential equation**, or system of differential equations, in which all derivatives are taken with respect to the same **variable**, which is usually time. See also **Partial differential equation**. (Chapter 5)

Organoid
A small, three-dimensional assembly of cells, derived from stem cells and differentiated into cells of a specific tissue. (Chapter 15)

Oscillation
The behavior of a function (or **dynamical system**), where the value of the function (or of a system variable) alternates between phases with positive and negative slopes. See also **Damped oscillation**, **Overshoot**, and **Limit cycle**. (Chapter 5)

OTU (Operational Taxonomic Unit)
In cases where the nomenclature of "species" becomes problematic, such as in microbial communities analyzed through 16S RNA sequencing, OTU refers to a mathematically defined taxon. The unit is primarily used to describe functional traits. (Chapter 14)

Overfitting
The fact that a **model** with (too) many **parameters** permits a good representation of a **training set** of data but not of other datasets. (Chapter 6)

Overshoot
The **dynamics** of a function or **system** that temporarily exceeds its ultimate final value. (Chapter 5)

Pacemaker cell
Cells, primarily in the **sino-atrial node** of the heart, that **depolarize** spontaneously and therefore set the pace of the heartbeat. The signal from the sino-atrial node eventually reaches all **cardiomyocytes**. See also **Atrioventricular node**, **Bundle of His**, and **Purkinje fiber**. (Chapter 12)

Parameter (of a model)
A numerical quantity in a mathematical **model** that is constant throughout a given computational experiment or **simulation**. Its value may be altered in a different experiment. See also **Parameter estimation**. (Chapter 2)

Parameter estimation
A diverse set of methods and techniques for identifying numerical values of model **parameters** that render the **model** representation of a given dataset acceptable. (Chapter 6)

Partial differential equation (PDE)
An equation (or system of equations) that contains functions and derivatives with respect to two or more variables, which, for instance, may represent time and one or more space coordinates. See also **Differential equation** and **Ordinary differential equation**. (Chapter 5)

Path length
The number of consecutive **edges** connecting one **node** with another node within a **graph**. Of interest is often the longest minimal path length in a graph. (Chapter 3)

Peptide
A chain of up to about 30 **amino acids** that is deemed too short to be called a protein but otherwise has the same properties. (Chapter 8)

Peptide bond
The chemical (covalent) bond between **amino acids** in a **peptide** or protein, in which the carboxyl group of one amino acid reacts with the amino group of the neighboring amino acid. (Chapter 8)

Personalized medicine
An emerging approach to health and disease that uses **genomic**, **proteomic**, **metabolic**, and physiological **biomarkers** for the custom-tailored prevention and treatment of an individual's disease. (Chapter 13)

Perturbation
A usually rapid shift in conditions applied either to a biological **system** or to a mathematical **model**. (Chapter 2)

Petri net
A specific formalism for modeling **networks** and **systems**, consisting of two distinct types of **variables**: pools and reactions. (Chapter 3)

Pharmacogenomics
The study of interactions between gene expression and drugs. (Chapter 13)

Pharmacokinetics/Pharmacokinetic model
The **dynamics** of a drug within an organ, body, or cell culture; a mathematical representation of this dynamics. A notable special case is a **physiologically based pharmacokinetic (PBPK)** model, which is a hybrid between a **compartment model** and a set of **kinetic** models within the **compartments**. (Chapter 13)

Phase diagram
A rough sketch of domains in which a **system** exhibits different **qualitative behaviors**. For instance, the system may exhibit **oscillations** in one domain but not in another. See also **Phase-plane plot**. (Chapter 14)

Phase-plane analysis/plot
A graphical representation of a **dynamical system**, typically with two **dependent variables**, where one variable is plotted against the other, rather than plotting both variables as functions of time. The plot is divided by **nullclines** into domains of qualitatively similar **behaviors**. See also **phase diagram**. (Chapter 14)

Phylogeny, Phylogenetics
Pertaining to studies of the relatedness of species throughout evolution. (Chapter 7)

Physiologically based pharmacokinetic (PBPK) model
A **compartment model** with biologically meaningful **compartments**, such as organs, tissues, or the bloodstream. Often used to assess the fate of drugs within a higher organism. The dynamics of drugs within compartments may be represented with kinetic models. (Chapter 13)

Polysaccharide
A linear or branched polymer consisting of many sugar (carbohydrate) molecules of the same or different type. (Chapter 11)

Post-translational modification (PTM)
Any chemical change to a **peptide** or protein after it has been translated from mRNA and that is not encoded in the RNA sequence. (Chapter 8)

Power function/Power-law function
A mathematical function consisting of the product of a (non-negative) **rate constant** and one or more positive-valued **variables** that are each raised to some real-valued exponent, which in **biochemical systems theory** is called a **kinetic order**. See also **Canonical model**. (Chapter 5)

Power-law approximation
Linear approximation of a function within a logarithmic coordinate system, which always results in a (possibly **multivariate**) **power-law function**. (Chapter 5)

Power-law distribution
Specific **probability distribution** that follows a **power-law function** with negative exponent. An example of recent prominence is the distribution of **nodes** with certain numbers of **edges** in **networks** exhibiting the **small-world property**. (Chapter 3)

Prediction (of a model)
A feature of a **system**, such as a numerical or approximate value of a system **variable**, that has not been measured experimentally but is computed with the model. (Chapter 2)

Predictive health
A novel paradigm of health care whose goal is maintenance of health rather than treatment of disease. Predictive health heavily relies on personalized assessments of **biomarkers** that begin when the individual is still apparently healthy and follows him or her into old age and, possibly, chronic disease. (Chapter 13)

Probabilistic
Affected by **random** (**stochastic**) influences. (Chapter 2)

Probability distribution
A succinct, collective characterization of all values that a **random variable** may assume, along with their likelihoods. (Chapter 2)

Profile (of metabolites or proteins)
A collection of features characterizing many or all metabolites (or proteins) of a biological **system** at one time point or at a series of time points. See also **Time-series data**. (Chapter 9)

Proteasome
A cellular organelle consisting of a large protein complex and responsible for the disassembly of proteins that had been tagged by **ubiquitin**. The cell reuses the **peptides** and **amino acids** generated in the process. (Chapter 8)

Proteome
A collective term for all proteins in a cell, organism, or otherwise defined biological sample. See also **Genome** and **-ome**. (Chapter 8)

Pump (in a cell membrane)
A **carrier protein**, also called a permease, that transports specific **ions** across a cell membrane. (Chapter 12)

Purkinje fibers
Modified heart muscle cells that rapidly conduct electrical impulses from the **sino-atrial** and **atrioventricular nodes** to the excitable **cardiomyocytes** in the **ventricles**. See also **Bundle of His**. (Chapter 12)

Qualitative analysis
A type of mathematical analysis that depends only minimally on specific numerical settings—such as **parameter** values—and therefore tends to yield rather general results. (Chapter 14)

Qualitative behavior (of a model)
A **model** response that is not characterized by specific numerical features but rather as a member of a distinct class of features, such as **damped oscillations** or **limit cycles**. (Chapter 2)

Quantitative structure–activity relationship (QSAR)
A specific property of a molecule, such as drug activity, that is experientially or computationally inferred from its chemical composition. (Chapter 13)

Quantitative systems pharmacology (QSP)
A new field within pharmacology and pharmaceutical science using computational systems models understand the effects of drugs on the human body. (Chapter 15)

Quasi-steady-state assumption
The assumption that an intermediate complex between a substrate and an **enzyme** is essentially constant during a **reaction**. The assumption permits the explicit formulation of **Michaelis–Menten** and **Hill rate laws** as functions, rather than systems of **differential equations**. (Chapter 9)

Quorum sensing (QS)
A mechanism with which microbes collectively respond to their sensing that many other microbes in their environment send out chemical signals. (Chapter 10)

Random
Containing an aspect that is unpredictable, possibly because of limited information. (Chapter 2)

Random variable
Variable in a **stochastic** context, whose specific value is determined by a **probability distribution**. (Chapter 2)

Rate constant
A positive or zero-valued quantity, usually in chemical and biochemical **kinetics**, representing the turnover rate of a **reaction**. See also **Kinetic order**, **Elasticity**, and **Mass-action kinetics**. (Chapter 5)

Reaction (in biochemistry)
The conversion of a molecule into a different molecule, a process that is often catalyzed by an **enzyme**. See also **Reversible**, **Irreversible**, **Kinetic**, and **Rate constant**. (Chapter 9)

Receptor
A protein on the surface of (or spanning) a cell membrane that senses external signals which, if strong enough, trigger a **signal transduction** process. Prominent examples are **G-protein-coupled receptors**, **receptor tyrosine kinases**, and **Toll-like receptors**. See also **Transmembrane protein**. (Chapter 8)

Receptor tyrosine kinase (RTK)
A member of an important class of **transmembrane proteins** that transfer a phosphate group from a donor to a substrate and by doing so transduce a **signal** from the outside of a cell to the inside. (Chapter 8)

Recursion
In mathematics, a process characterized by a temporal sequence of **states** of a **system** that are defined in terms of the states of the system at one or more earlier time points. (Chapter 4)

Redox
Contraction of the terms reduction and oxidation. Used as an adjective referring to the **state** or change in state of a chemical **reaction** system with respect to losses or gains of electrons. For example, the physiological redox state is changed under oxidative **stress**. (Chapter 11)

Reductionism
The philosophy that the functioning of a **system** can be understood by studying its parts and the parts of parts, until the ultimate building blocks are revealed. **Systems biology** maintains that reductionism must be complemented with a reconstruction phase that integrates the building blocks into a functioning entity. (Chapter 1)

Refractory period
A time period, typically toward the end of an **action potential**, during which an **excitable cell** cannot be excited again. (Chapter 12)

Regression
A statistical technique for matching a **model** to experimental data in such a way that the **residuals** are minimized. (Chapter 6)

Repeller
In the field of dynamical systems, a set of points from which **trajectories** diverge. The most typical repellers are unstable steady-state points and unstable **limit cycles.** The antonym is **attractor**. (Chapter 5)

Repolarization
As part of an **action potential**, the return in voltage across the membrane of a neuron, muscle cell, or **cardiomyocyte** from a

depolarized state to the relaxed state, which in most cells corresponds to the **resting potential**. (Chapter 12)

Repressilator
A system of three genes (A, B, and C), where A represses B, B represses C, and C represses A. (Chapter 7)

Residual (error)
The remaining difference (error) between a **model** and data, after the model has been **fitted**. See also **Regression**, **Least-squares method**, and **Parameter estimation** (Chapter 6)

Resting potential
The electrical potential across the cell membrane of a relaxed, **excitable cell**, such as a neuron or **cardiomyocyte**. In neurons, the resting potential is about –70 mV, whereas it is about –90 mV in cardiomyocytes. See also **Action potential**. (Chapter 12)

Retrovirus
An RNA virus that needs to be reverse-transcribed by the host into complementary DNA, before it can be integrated into the host **genome**. See also **transcription**. (Chapter 7)

Reverse engineering
A diverse set of methods for inferring structures, processes, or mechanisms from data. An example is **Bayesian network inference**. (Chapter 3)

Reversible (reaction)
A (biochemical) **reaction** that may occur in forward and backward direction, usually with a different rate. See also **Irreversible reaction**. (Chapter 9)

Ribosome
A complex, consisting of specific RNAs and proteins, that facilitates the translation of mRNA into protein. (Chapter 8)

Riboswitch
An RNA-based regulatory element of gene expression in bacteria. (Chapter 8)

Ribozyme
A ribonucleic acid (RNA) that catalyzes certain chemical reactions. The word is composed from "ribonucleic acid" and **enzyme**. (Chapter 7)

RNA-Seq
A relatively recent high-throughput method for measuring gene expression, based on **cDNA** sequencing. See also **Microarray**. (Chapter 7)

Robust
An often vaguely defined term for the tolerance of a **system** toward **perturbations** or changes in **parameters**. See also **Sensitivity** and **Gain**. (Chapter 5)

RT-PCR
Abbreviation for:
1. **Reverse transcription followed by a quantitative polymerase chain reaction. A method** for quantifying mRNA and thus gene expression. See also **Northern blot**.
2. Real-time polymerase chain reaction. A fluorescence-based method for monitoring in real time how much DNA had been polymerized since the beginning of the experiment. Also called RT-qPCR, where q means quantitative. (Chapter 7)

Sarcoplasmic reticulum (SR)
A compartment in **cardiomyocytes** that can contain large quantities of calcium. Controlled calcium movement between the sarcoplasmic reticulum and the cytosol is ultimately responsible for cell contraction. (Chapter 12)

SBML (Systems Biology Markup Language)
A standardized computer language for **modeling** and sharing biological **systems** and data. (Chapters 1 and 15)

Scale-free
Property of a **graph** or **network** to have similar features, independent of the number of **nodes**. (Chapter 3)

Scatter plot
Graphical representation of two (or three) features of some entity as coordinates of points in a coordinate system, which often looks like a cloud of dots. An example consists of math and physics scores of a class of high-school students. For each student, the math score could be the x-coordinate and the physics score the y-coordinate of a unique point that represents this student. (Chapter 6)

Search algorithm
A set of computing instructions for finding an optimal solution, consisting of **iterations** of searching and comparing with the so-far-best solutions. See also **Optimization** and **Parameter estimation.** (Chapter 6)

Second messenger
A small molecule, such as nitric oxide, calcium, cAMP, inositol trisphosphate, or the sphingolipid ceramide, that propagates **signal transduction,** typically by diffusion and binding to other biomolecules. (Chapter 10)

Secondary structure (RNA)
The folding and partial base matching of a single-stranded RNA with itself, creating accurately matched regions, interrupted by loops and possibly ending in unmatched ends. (Chapter 7)

Sensitivity analysis
A branch of general **systems** analysis that quantifies how much a system feature changes if a **parameter** value is slightly altered. (Chapter 5)

Separation of timescales
A modeling technique for **systems** containing processes that run at distinctly different speeds. By assuming that very slow processes are essentially constant and that very fast processes are essentially at **steady state**, one restricts a **model** to the one or two most relevant timescales. (Chapter 11)

Signal
Any molecular, chemical, or physical event that can be sensed and transmitted by a cell or organism, usually through a **receptor**. (Chapter 10)

Signal transduction (signaling)
A collective term for the management and integration of **signals** received by the **receptors** of a cell. In many cases, **signaling cascades** relay an external signal to the **genome**, where specific genes are up- or down-regulated in response. (Chapter 10)

Signaling cascade
Usually a hierarchically arranged set of proteins, which, by becoming phosphorylated or dephosphorylated, transduce a chemical or physical **signal** from the cell surface to the **genome**. (Chapter 10)

Silencing (of a gene)
The ability of certain small RNAs to suppress gene expression. (Chapter 7)

Simulated annealing
A **parameter estimation** technique inspired by a method of heating and cooling metals in order to reduce impurities and defects. (Chapter 6)

Simulation
The process of using a mathematical **model** with different **parameter** and **variable** settings, with the goal of assessing its possible range of responses, including best, worst, and most likely scenarios or of explaining scenarios observed in—or predicted for—a biological **system**. (Chapter 2)

Single nucleotide polymorphism (SNP)
A point mutation in a specific **nucleotide** that distinguishes individuals. Because multiple **codons** can be translated into the same amino acid, the change is often neutral. (Chapter 13)

Sino-atrial (SA) node
A relatively small group of **pacemaker cells** located between the **atria** and **ventricles** of the heart, which **depolarize** spontaneously and thereby set the speed of a heartbeat. (Chapter 12)

SIR model
An established prototype **model** for assessing the spread of communicable diseases. *S*, *I*, and *R* stand for susceptible, infective, and (from the disease process) removed individuals. Thousands of variants have been proposed. (Chapter 2)

Sloppiness
The result of a **parameter estimation** process showing that many combinations of parameter values for a given **model** can **fit** the same datasets equally (or nearly equally) well. See also **Identifiability**. (Chapter 6)

Small RNA
Relatively short RNA sequences of 50–200 nucleotides that form specific loop structures and can bind to proteins or mRNAs, thereby affecting their function. See also **MicroRNA**. (Chapter 7)

Small-world (property of networks)
A special type of connectivity within networks, where a few **hubs** are heavily connected and the remaining **nodes** are more loosely connected. Thus, most nodes are not neighbors of one another, but most nodes can be reached from any other node by a small number of steps. The number of nodes with a given degree of connectivity follows a **power-law distribution** with negative exponent. (Chapter 3)

Smoothing
A mathematical technique for reducing **noise** in experimental data and for bringing the data closer to a **continuous** trend line. (Chapter 6)

Sphingolipid
A member of a specific class of lipids that are crucial constituents of membranes and also possess signaling function, for instance, during **differentiation**, aging, **apoptosis**, and a number of diseases, including cancer. (Chapter 11)

S-system
A **modeling** format within **biochemical systems theory**, in which every **ordinary differential equation** consists of exactly two products of **power-law functions** (one of which may be zero): one collectively representing all augmenting processes and one representing all degrading processes. See also **Biochemical systems theory** and **Generalized mass action system**. (Chapter 5)

Stability
1. The feature of a system to return to a **steady state** or **limit cycle** after small **perturbations**. (Local stability)
2. The **robustness** of the system **behavior** to slight, persistent changes in **parameter values** (structural stability). See also **Bifurcation**. (Chapter 5)

State (of a system or model)
A comprehensive collection of numerical features characterizing a **system** or **model** at a given point in time. (Chapter 2)

Static (model)
A **model** that describes a **network** or **system** without accounting for changes over time. (Chapter 3)

Steady state
The **state** of a **dynamical system** where no **dependent variable** changes in magnitude, although material may be flowing through the system. At a steady state, the time derivatives in all **differential equations** describing a system are equal to zero, and thus the equations become algebraic equations. See also **Homeostasis** and **Stability**. (Chapter 2)

Stimulus
An external **perturbation** to a **system** or **model**. Stimuli are often used to study possible **behaviors** of the system. (Chapter 2)

Stochastic, Stochasticity
Containing **probabilistic** or **random** aspects; often considered the antonym of **deterministic**. (Chapter 2)

Stochastic differential equation
An equation (or system of equations) that contains functions and derivatives as well as one or more **stochastic terms**, which, upon integration, lead to a collection of solutions that exhibit unpredictable randomness. Typically, the implementation of **stochasticity** is through a **noise** term added to the right-hand side of the **differential equation**. **(Chapter 10)**

Stochastic simulation
A **simulation** with a **model** that is subject to **random** events.

Stochastic simulation algorithm
A specific **algorithm** for **modeling kinetic reaction** processes that involve small numbers of molecules and therefore do not permit methods of **statistical mechanics** and **kinetics**. (Chapter 9)

Stoichiometry
The connectivity of a metabolic network, augmented with an account for the relative quantities of reactants and products in each **reaction**. (Chapter 3)

Strange Attractor
The domain of a **chaotic** system to which **trajectories** converge and that they do not leave once they are in the attractor. See also **Attractor**. (Chapter 5)

Stress
A situation that moves a cell or organism away from its normal **physiological** state or range. See also **Allostasis**. (Chapter 11)

Stress response element
A short sequence of (maybe five) **nucleotides** in the promoter region of genes involved in responses to external **stresses**, such as heat, osmotic pressure, strong acid, or the depletion of required chemicals in a cell's environment. (Chapter 11)

Subpopulation
A subset of a population characterized by some common feature, such as age, gender, race, or a particular disease status. (Chapter 14)

Superposition (principle)
A feature characteristic of **linear**—but not **nonlinear**—**systems**: the response of a system to two simultaneous stimuli is equal to the sum of the responses to the same two stimuli administered separately. If the superposition principle holds, a system can be analyzed by performing many **stimulus**–response experiments and adding the observed stimulus–response relationships. See also **Synergism**. (Chapter 1)

Synergism/synergistic
Literally, "working together." The phenomenon that the combined action of two **system** components (or people) has a stronger effect than the sum of their individual actions. Synergism is a feature of **nonlinear** systems and violates the **superposition principle**. Antonym of **antagonism/antagonistic**. (Chapter 1)

Synthetic biology
A diverse set of experimental and computational manipulation techniques with the goal of creating new (usually microbial) biological **systems**. (Chapter 7)

System
An organized collection of dynamically interacting components.

Systems biology
A branch of biology that uses experimental and computational techniques to assess dynamic biological phenomena within larger contexts. (Chapter 1)

Systole
Greek term for contraction and used to describe the contracted state of the heart. See also **Diastole**. (Chapter 12)

Taylor approximation
A general type of **approximation** for functions of one or more **variables** around a specific **operating point**, resulting in a (possibly **multivariate**) **linear**, polynomial, or **power-law representation**. (Chapter 5)

Tertiary structure (of a protein)
The three-dimensional shape of a protein molecule. See also **Conformation**. (Chapter 8)

Text mining
Use of a **machine learning algorithm** to extract information from large amounts of text. See also **Data mining**. (Chapter 13)

Time-series data
Experimental or artificial data, consisting of values of **variables** at (many) subsequent time points. (Chapter 6)

Toggle switch (genetics)
A phenomenon or **model** where a gene is either fully expressed or silent but does not allow intermediates. (Chapter 7)

Toll-like receptors (TLRs)
A member of a class of **signaling** proteins that recognize a variety of foreign macromolecules—especially those on the cell walls of bacteria—and trigger the initiation of an immune response. See also **Receptor** and **Transmembrane protein**. (Chapter 8)

Training set
A portion of a dataset used by a **machine learning algorithm** to distinguish different classes of outputs and to establish classification rules. See also **Validation**. (Chapter 6)

Trajectory
The collective change in the **state** of a **system** over time; usually between a **stimulus** and some later state, such as a steady state. (Chapter 5)

Transcription
The process of creating a matching messenger RNA (mRNA) from a DNA sequence. (Chapter 7)

Transcription factor
A protein affecting the expression of a specific gene by binding to the corresponding DNA. (Chapter 7)

Transcriptome
The totality of RNAs transcribed from the **genome** of an organism. See also **-ome**. (Chapter 7)

Transfer RNA (tRNA)
One of several specific RNAs that facilitate the attachment of the correct amino acid to a growing **peptide** or protein, while it is being translated from mRNA. This **translation** process occurs within a ribosome. (Chapter 7)

Transient
A collective term for behaviors of a **system** between the time of a stimulus and the time the system reaches a **steady state**, a **stable oscillation**, or some other **attractor** or point of interest. (Chapter 5)

Translation
The process of generating a protein or **peptide** whose **amino acid** sequence corresponds to the **genetic code** of an mRNA. The process occurs within a ribosome and uses **transfer RNA**. See also **Post-translational modifications**. (Chapter 7)

Transmembrane protein
A member of a large class of proteins that span the membrane of a cell and thereby allow communication between the outside and inside of the cell. Transmembrane proteins are involved in **signal transduction** and a large number of diseases. See also **Gap junction** and **G-protein-coupled receptor.** (Chapter 8)

Transposon
A stretch of DNA that can move from one location in a **genome** to another. (Chapter 7)

Trial (stochastic process)
One of many evaluations of a **stochastic model**. (Chapter 2)

Two-component (signaling) system
While not exclusively limited to this definition, usually a specific type of **signaling** mechanism, with which microorganisms sense their environment. See also **Receptor** and **Signal transduction**. (Chapter 10)

Two-dimensional gel electrophoresis
An experimental method for separating proteins by their size and electrical charge. (Chapter 8)

Ubiquitin
A small protein which, attached to other proteins, tags them for disassembly in the **proteasome** and later recycling of the resulting **peptides** and **amino acids**. The tagging process is called ubiquitination. (Chapter 8)

Uncertainty
Degree to which an experimental result is not exactly known, owing to experimental inaccuracies or other challenges; see also **Variability**. Advanced measurement techniques might reduce uncertainty but not variability. (Chapter 6)

Validation
The process of testing (and confirming) the appropriateness of a **model** with data that had not been used for the construction of the model. See also **Training set**, **Identifiability**, and **Sloppiness**. (Chapter 2)

Variability
Degree to which an experimental result is affected by differences among individual cells, organisms, or other items under investigation; see also **Uncertainty**. Advanced measurement techniques might reduce uncertainty but not variability. (Chapter 2)

Variable
A symbol representing a component of a **system**, which may or may not change over time. See also **Dependent** and **Independent variable**. (Chapter 2)

Vector
1. In mathematics, a convenient arrangement of variables or numbers in a one-dimensional array. Often used in conjunction with a **matrix**. Many aspects within the fields of linear algebra, **multivariate** calculus, and statistics are based on vectors and matrices. (Chapter 5)
2. In genetic engineering, an artificially constructed virus or plasmid used to introduce foreign DNA into a bacterium or host cell. (Chapter 7)

Ventricle
One of the two larger chambers of the heart. See also **Atrium**. (Chapter 12)

Vertex (pl. Vertices)
Synonym of **node**. (Chapter 3)

Virtual Clinical Trial (in Systems Biology)
The computational analogue of a clinical trial, in which the "patients" with their variabilities are individually parameterized computer models representing a disease and its possible treatments and side effects with sufficient accuracy. (Chapter 13)

Virtual Twin (in Systems Biology)
A computer model simulating a human (or system within a human) that is ideally so close to reality that it can be used to explore diseases and treatments. Also called Digital Twin. (Chapter 13)

Voltage-gated
Property of a membrane channel to open and close in response to the membrane potential. (Chapter 12)

Western blot
A technique for measuring gene expression based on produced protein. See also **Northern blot**. (Chapter 7)

XML
Abbreviation for eXtensible Mark-up Language. Computer language with a defined vocabulary that permits the structuring of data and facilitates translation into different languages and software packages. See also **Ontology** and **SBML**. (Chapters 1 and 15)

X-ray crystallography
A technique for determining the structure of a protein. The protein must first be crystallized. An X-ray shot through the crystal scatters in a characteristic fashion that allows the deduction of the molecular structure of the protein. (Chapter 8)

Taylor & Francis
Taylor & Francis Group
http://taylorandfrancis.com

Index

Note: Page numbers in *italics* indicate a figure and page numbers in **bold** indicate a table on the corresponding page. Page followed by "b" refer to boxes.

3D structure of protein, 261–262

A

absorbing state, 104
actin, 265, 273, 384, *384*
action potential, 383, 395–397
  cardiomyocyte, 381, 395–397, *395*, *396*
  lengthening during repolarization, 402
  models, 397–400, *400*
  phases, 396, *396*
  prolongation, 408
  repeated, 400, *400*
  SA and AV node cells, 395–396, *395*
  sodium-calcium exchanger, 395
activator, metabolic systems, 293–294
activator module, 243
acute lymphoblastic leukemia (ALL), 421
adaptation, 337–338, *337*, *338*, **338**
adaptive immune system, 471–472, *472*
adenine, 215, *216*,
ad hoc models, 133–134, *134*
adjacency matrix, 63, *63*
  Markov matrix *vs.*, **105**
adrenaline, 383
affinity, 170
agent-based modeling (ABM), 365, 365, 474
  Netlogo, 365–370, 366–368b, *369*
  outlook, 372
  spatial features, *370*, 370–372
aging
  heart failure, 402
  inflammation, 471
  lens, 274
alanine (Ala), 256, **257**
albumin, serum, 268
algae, 481, *481*
algorithms for parameter estimation, 166
  nonlinear systems, 172–174, *174*
    appropriate function, 194–196, *195*, *196*
    comprehensive grid search, 174–176, *175*
    evolutionary, 173
    genetic, 177–179, *178*
    gradient, 176–177, *176*
    other stochastic, 179–180
allosteric inhibition, 266–267
allosteric regulation, 294
α-helix, 261
Alzheimer's disease, 14, 476
Akaike Information Criterion (AIC), 197
amino acids, 218, 256, *256*, **257**, 258
  codon, 218
  physico-chemical properties, 260–261
Amorpha-4,11-diene, 244–245, *244*
amorpha-4,11-diene, 244
AND gate, 74, *75*
ant colony optimization (ACO), 179, 180
antibody 271, *271*
  monoclonal, 428, *429–431*, 429–431, **431**
antibody binding, *429–431*, 429–431, **431**
anti-inflammatory cytokine, 473
antigen, immune cascade, 198–199, *198*
antimalarial drug, 244
anti-parallel strand, 215, *216*
antiporter, 393
Aplysia californica, neural networks, 479, *479*
apolipoprotein, 274
apoptosis, 30
application, emerging, 468–482
approximation, 31, *32*
  canonical, 133
  continuum, 382
  linearized ODE models, 116–119, *117*, 120–121b
  power-law, 137, 142b, 143b, 294, 305
  Taylor, *see* Taylor approximation theorem
aptamer, 428
aquaporin, 274, *274*
aquatic system, 3
  ocean, 480–482, *482*
archaea, 329
arginine (Arg), **257**, **258**
arrhythmia, 402
Artemisia annua, 244
artemisinin production, 244–245, *244*
artificial gene circuit, 241–244, *242*, *243*
artificial neural network (ANN), 421, 477, *477*
ascorbate pathway, 315
aspartic acid (Asp), **257**, *330*
ATH1 gene, 361, *357*
atomic force microscopy (AFM), *217*
atrial fibrillation, 402
atrioventricular (AV) node, 382
atrioventricular (AV) valve, *380*, 380
atrium, 379
attractor, 150–157
  chaotic, 154–157, *155–157*
  classes of, 150
  defined, 150
  limit cycles, 150–154, *151–154*
  Lorenz, 154–155, *155*
  strange, 155, *155*
  Thomas, 156, *156*
automaticity, 395
autonomic nervous system (ANS), 382–383
autophosphorylation, 324, 331
auto regulation, 74, *74*
AV (atrioventricular) node, 382
AV (atrioventricular) valves, *380*, 380
axiom, 488

B

Bacillus anthracis, 331
Bacillus subtilis, 240b
bacteria
  actin, 273
  biofilm, 460, *460*
  carrier protein, 392
  cyanobacteria, 265, 481, *481*
  cytokine, 271
  feedforward loop, 76–77
  flagellar motor, 269–270, *269*
  gene regulation, 199
  genome, 214
    organization, 52, 231
    size, 217
  glycoprotein, 274
  inflammation, 474
  keratin, 273
  lipo-oligosaccharide, *266*
  operon, 52, 229, 231
  plasmid, 214, 221
  population growth, 95, *95*, *447*, 447, 454
  predation, 450
  quorum sensing, 241
  riboswitch, 240b, *240*
  stress signal sensing, 76
  Toll-like receptor, 272, *272*
  transcription factor, 231–232
  transposon, 214
  tryptophan, 214
  two-component signaling system, 329–333, *330–333*, **332**
  "wild type" 17
barcoding, 218, *219*
base, DNA and RNA, 215, *216*, *217*
base pair, 215, *216*, *217*
basic reproduction number, 41
basin of attraction, 127, 326
Bayes' formula, 80
Bayesian Information Criterion (BIC), 197
Bayesian network inference, 79, 82–84
Bayesian reconstruction, interaction networks, 78–81, *80*
Bayes' rule, 80
Bayes' theorem, 80
behavior, of discrete *vs.* continuous models, 31
behavioral conditioning, 489
β-sheets, 261, *263*
bidirectional graph, 61, *61*
bifan motif, 74, *75*, *347*
bifurcation point, 147
bimolecular metabolic reaction, 288
binomial model, 33, 35b
biochemical reaction, metabolic system, 286–295
  background, 286–287, 287b
  mathematical formulation of elementary reactions, 288–289
  rate laws, 289–295, *290*, *291*, *292*
biochemical systems theory (BST), 136, 306
  generalized mass action (GMA) systems, 136–137, *137*
  metabolic pathways, 294
  S-system, 138–139
BioCyc, 74, 298, 299–300b, 300, 314
bioengineering, 11
biofilm, 241, 460, *460*
biofuel, 312
bioinformatics, 12, 314, 460
  DNA sequences, 221
  Markov models, 104
biological components, variations, 9
biological process, gene products, 219
biological revolution, 3
biological system, 1–17
  analyzing, 4, *4*
  capability to manipulate, 3
  complicated, 1, *2*
  confusing simple, 8–10, *8*, *9*
  conversion into computational models, 16
  design pattern, 51, 52, 53
  design principle/motif, 51, 52, 53
  emergent property, 11, 12
  engineering of, 3
  examples, 1, *2*
  goal of deeper understanding of, 3
  history of dealing with, 3
  nonlinear process in, 4, *5*, 7
  size *vs.* complexity of, 4, *5*
  static structure and dynamics of, 16
biological target, drug development, 426
biological theory, 488–489
biological trait, 218
biological variability, and disease, 415–416
  modeling, 416–419, *417*, **417**, *418*
biologics, 428, *429*
bioluminescence, 342–345, *343*, *344*
biomarkers, 419, 420–421, *420*
  disease, 262, 420–421, *420*
  normal range, 424, *424*
biomass, 312, 313
bionanotechnology, *13*
biotechnology, 12
bipartite graph, *62*, 69–70
bistability
  signaling systems, *32*, 325–326, *326*
bistable system, 150
black box model, oscillation, 291, 378, 386–389, *386–388*
blood sugar level, 421
Blue Brain Project, 379, 478, *478*
blue-sky catastrophe, 33, 33b, *34*
bolus experiment, 145–146, *146*
Boolean network, 72–74, *72*, *73*
bootstrap method, 185
bottom-up estimation, 40

bottom-up strategy, 17
  parameter estimation, 167
bradycardia, 402
brady–tachy syndrome, 402
brain disease, 476–478, *477*, *478*
brain, neuron, 372
brain tumor, 232
branch-and-bound methods, 175–176
branched pathways, 195, *196*, 200, *200*
BRENDA, 295, 295–296b, *296*, 298
  pathway screening, 438
BST, *see* biochemical systems theory (BST)
bundle of His, 382, 384, 402
Burkholderia pseudomallei, *219*
bursting oscillation, 154, *154*

C

Caenorhabditis elegans, neural network, 478, *478*
calcium adenosine triphosphatase (Ca^{2+} ATPase), plasma membrane, 393
calcium (Ca^{2+}) channel
  L-type, 395, 403, *403*
  T-tubule, 403, *403*
  T-type, 395
calcium dynamics, dyad, 403–406, *403–404*, *406*
calcium flux
  dyad, 403, *403*
  slowing down, 402
  Somogyi-Stucki model, 390, *390*
calcium-induced calcium release (CICR), 389, 396, 403
calcium release unit (CaRU), 407
cAMP, *see* cyclic adenosine monophosphate (cAMP)
cancer, 475–476, *475*
canonical models, 127, 133, 134–135
  concept map modeling, 487, *487*
  generalized mass action (GMA) systems, 136–137, *137*
  limit-cycle oscillation, 135
  lin–log model, 139
  Lotka-Volterra, 134, 135–136, *135*, 156, 189, 457
  power-law approximation, 137, 142b, 143b
  S-systems, 138–139, 141
  toggle switch model for SOS pathway of *E. coli*, 139, *140*
capacitance, 397
capillary electrophoresis (CE), 297
carcinogenesis, 475–476, *475*
cardiac cell, *see* cardiomyocyte
cardiac cycle, *403*
  blood flow, *380*, 381
  ECG, *385*
  metabolic responses to oxygen supply, 389
  movement of ions, 389, 391
  shape of heart, 382
cardiac output, 381
cardiomyocyte, 381, 382–383
  action potential, 381, 395–397, *395*, *396*
    model, 397–400, *400*
  aging, 402
  automaticity, 395
  contraction, 378, 385
  electrochemistry, 391–401
    biophysical description, 394–395
    models of action potential, 397–400, *400*
    repeated heartbeat, 400–401, *400*
    resting potential, 395–397, *395*, *396*
  excitable, 382, 383, 384, *384*
  funny current, 395
  heart failure, 401–408
    modeling based on molecular event, 403–408, *403–404*, *406*, *407*
  oxidative stress, 402
  resting potential and action potential, 383, 384, 395–397, *395*, *396*
  sarcolemma, 403, *403*
  sino-atrial node, 382
carrier protein, 267, 268, 392–393
carrier, 267–270, *269*
carrying capacity, 446
catabolite activator protein (CAP), *229*, 230
causality analysis, 77
cell cycle, 69
  genomic regulation, 338
  heat stress response, 355
  oscillator, 341
  regulatory network, 338
cell-free system, *345*, 345–346, *346*
cell membrane
  depolarization, 383
  gated channel, 399–400
  heart function and failure, 403
  ion movement, 391–392
  resting potential, 395
  T-tubules, 405
cell-to-cell communication, 268
cellular component, gene product, 219
Central Dogma, 213–215, *214*, 228, *279*
central limit theorem, 405
central nervous system, 11
channel protein, 392
chaos, 145
chaotic, 99
chaotic attractor, 154–157, *155–157*
chaotic dynamics, 461, **461**, *462*
chaotic process, 99
chaperone, *259*, 260
chaperonin, 259
charge gradient, 391
CheY protein, 329–330, *330*
children, growth curves, 54, *54*
chitin, 273
chloroplast, 214
chromatin, 221
chromosome, 221
cilia, 270
circadian clock, 339–341, *340–341*
cis-regulatory motif, 229
citric acid, 307
c-Jun N-terminal kinase (JNK), 334
clinical phase, drug development, 427
clinical trial, 427
clique, 69, 84
closed system model, 32
clustering coefficient, 63–64, *64*
Clostridium thermocellum, *263*
coding region, 220
codon, 218, 256–258, **258**
codon bias, 218
codon law, 489
codon usage bias, 258
coefficient, 116
collagen, 272, *273*
communication, 13–16, *14*
community proteomics, 461
compartment model, 116
  pharmacokinetics, 432–435, *432–434*
competitive inhibition, 294
complexity, Lotka-Volterra model, 135
comprehensive grid search, 174–176, *175*
comprehensive screening experiment, 147
computational power, 482
computational systems biology, 422, *422*
computer-aided approximation, 13
computer-aided design (CAD), 240–241
computer-based reasoning system, 481
computer simulation, 21, *41*, 42
  personalized medicine, 423–424
  population growth, 449
computing environment, 12, *13*
concentration gradient, 391
concept-map modeling, 487, *487*
conceptual model, 15
conditional probabilities, 80, *80*
conductance, 397, 399
conformation, 231, 258
congestive heart failure, *395*, 402
connection, 16
connectivity relationship, 303
connexin, 268, *268*
consistency, 42–44, *43*
constrained optimization, 307
constraint, 70, 307
continuous linear systems
  linear differential equation, 116
  linearized model, 116–119, *117*, 120–121b
continuous model, 94
continuous timescale models, 31
continuum approximation, 382
control coefficient, 302–303
cooperativity, 267, 292
coral growth, 482, *482*
co-regulating gene, 232
corn, 245
coronary, 381
correlative model, 29, *29*
Corynebacterium glutamicum, 232, 309
critical point analysis, 145, 147
cross-bridge, 384, *384*
crosstalk, MAPK cascade, 337
crowding effect, 447
cryo-electron microscopy (cryo-EM), 259, *259*, 261
curated database, 300
curved surface, linearization of, 117, *117*
customized mark-up languages (XML), 12
cyanobacteria, 265, 481, *481*
cyclic adenosine monophosphate (cAMP), 230, 324
  funny current, 395
  signaling systems, 324
cysteine (Cys), **257**
cystine, 256
cytochrome, 268–269
cytokine, 271–272, 471–474, *472*
cytokine storm, 272, *472*, 472, 473–474
Cytoscape, 60
cytosine, 215, *216*

D

damped oscillation, 135
data availability, 28
data mining, 315
data-modeling pipeline, 484–485, *485*
datasheet, 240
data warehouse, 481
degree distribution, 65, *66*
degree associated with node, 63–64, *63*
demand theory, 199, 231, 489
denaturation, 355
deoxyribonucleic acid, *see* DNA
deoxyribose, *216*
dependency among network components, 77–78, *78*
dependent variable, 36–37, *37*
depolarization, 383, 395–396, *396*
design pattern, 51, 52, 53
design principle, 51, 52, 53, 197–200, *198*, *199*
deterministic model, 32, 33b
  population dynamics, 453b
  recursive, 94–97, **95**, *96*
deterministic recursive model
  linear recursion, 94–95, **95**
  nonlinear recursion, 97–99, *98–99*
  red blood cell, 95–97, *96*
  steady state, *99*, 99–102, 99–102, *102*
diabetes, 32, 421, 428, 476
diagram, model development, 35–37, *36*, *37*
diastole, 379
dicer, 227
difference equation, 97
differential equation, 8
  linear, 114, 116
  ordinary, 113
    heart, 405
    metabolic system, 288
    population growth, 447
  parameter estimation for system of, 185–194, 186b, *187–188*, **189–190**, *191–194*, **192–193**
  partial, 278, 341
    heart, 380, 404
  stochastic, 325, 338
differentiation, 113
diffusion process, 267, 288
Digitalis purpurea, 393
directed acyclic graph (DAG), 82–83
directed graph, 61, *61*, 63–64, *63–64*, *85*
discrete biological system, 93–94
  deterministic recursive model
    linear recursion, 94–95, **95**
    nonlinear recursion, 97–99, *98–99*
    red blood cell, 95–97, *96*
    steady state, *99*, 99–102, 99–102, *102*

stochastic recursive model
Markov chain model, 103–107, **105**, *105*, **107–108**
probability tree, 103, *103*
random walk, *102*, 102–103
discrete model, 31, 94
discrete process, 45
discrete time point, 93
disease, *see also* biomarkers; infectious disease modeling
biological variability and, 415–416
modeling, 416–419, *417*, **417**, *418*
complex, 415, 476
delineation, *424*
-free equilibrium, 41
disulfide bond, 256, 259, 260
DNA (deoxyribonucleic acid)
adduct, 223
barcoding, 218, *219*
base pairs, 215, *216*, *217*
chemical and physical features, 215–217, *216–217*
codon, 218, 256–258, **258**
double-stranded, 214, 215, 217
epigenetics, 214, 223
eukaryotic packing, 221, *222*, *223*
exon and intron, 220, *221*
genetic variation, 213
hybridization, 215
"junk" 220, 467
key property, 215–223
noncoding, 218–221, *221*
nucleotide, 215, *216*
origin-binding domain (OBDs), *217*
random mutation, 33
replication, 215
sequencing, 218, 221
size and organization, 217–218, *219*
supercoiled, 217, *217*
synthetic biology, 239–241
transcription, 213, 215, 221, 223
hierarchical control, 232, *233*
translation, 219
DNA adduct, 223
DNA methyltransferases (DNMTs), *222*
dose, 426
Drosophila, 315
drought, plant response, *2*
drug development, 425
antimalarial, 244–245
biological target, 426
biologics, 428, *429*
clinical phase, 427
compartment model, 432–435, *432–433*
lead identification, 428–429
lead optimization, 426
model
compartment, 432–435, *432–433*
dynamic, 437–439, *437*
pharmacokinetic, 432–435, *432–434*, 435b
receptor dynamics, 429–432, *429–432*, **431**
new chemical entities (NCEs), 425–426
pipeline, 426–428, *426*, 437
post-approval phase, 427
post-translational modifications (PTMs), 260
preclinical phase, 425–426, *426*
process, 425–427, *426*
quantitative structure-activity relationships (QSARs), 426
role of systems biology, 427–440
computational target and lead identification, 428–429
emerging, 440
pathway screening with dynamic models, 437–439, *437*
pharmacokinetic modeling, 432
receptor dynamics, 429–432, *429–432*
virtual clinical trial, 439–440
toxicity testing, 426–427
drug sensitivity, 421
duplication-divergence (DD) model, 68
dyad, calcium dynamics, 403–406, *403–404*, *406*
dynamical feature, exploration and validation, 44–45, 45b, *46*
dynamic flux balance analysis (DFBA), 314
dynamic model, 30, 437–439, *437*
dynamic network, 71–72
Boolean network, *72*, 72–74, *73*
motifs, *74–76*, 74–77
dynamic system
attractor, 150
chaotic, 154–157, *155–157*
limit cycle, 150–154, *151–154*
of biological system, 16
model development, 26–27
dynamics motif, 147–150, *148*, *149*
dystrophin, 220, 286

E

E-Cell, 468
ecoinformatics, 480–481
edge-detection algorithm, 241
edge, 61, 62
effector cell, 198–199, *198–199*
efficacy, 426
eigenvalue
linear ODEs, 116
stability analysis, 99, 127–130
elasticity, 302, 303
elasticity coefficient, 302
elastin, 273
electrical charge, 397
electrical conductance, 382, 397–398
electrical conductor, 397
electrical gradient, 391
electrical potential, 397
electrical resistance, 397
electric current, 397
electrocardiogram (ECG, EKG), 385, *385*
electrochemical gradient, 393
electrochemical process, heart, 378
electrochemistry, cardiomyocyte, 391–401
biophysical description, 394–395
model of action potential, 397–400, *400*
repeated heartbeat, 400–401, *400*
resting potential and action potential, 395–397, *395*, *396*
electromotive force (EMF), 397
electrophysiology, heart, 378
elementary mode analysis (EMA), 310–313, *311*, *312*
embryonic cell cycle oscillator, 341
emergent property, 11
ensemble method, 196
entropy, 78, *78*
environment
interaction of organism with, 480–482, *481–482*
model development, 32
environmental pressure, 17
EnvZ two-component sensing system, 200, 332
ENZYME (database), 295
enzyme, 265–267, *266*, *267*
activity profile, 60
catalyzed reaction, 134, 169, *170*, 289, *290*
control of pathway system, 301–302
dicer, 227
drug development, 244, *244*, 426, 438
flux analysis, 298
functional loop, 8–9, *8*
gene circuit, 241
gene regulation, 229
heat stress response, 355–357, *355*
hexokinase, 285, 286
immune system, 272
independent variable, 36, 130
kinase, 270
kinetics, 134, *134*, 296–297
lac operon, 229–230, *229*
metabolic control analysis, 302–303, *303*
metabolic system, 286, 287, 287b, 295b
reconstruction, 314, *315*
Michaelis-Menten rate law, 290–293, *291*, *293*
mitogen-activated protein kinase cascades, 333–337, *334–337*, **335**
personalized medicine, 420, 422
perturbation, 418, 438
population growth, 445
power-law approximations of multivariate function, 143b
protein activity and dynamics, 274
protein folding, 260
rate law, 290–293
ribozyme, 226, *226*
sensitivity analysis, 130
signal transduction, 324, 329
small RNAs, 227
substrate conversion, 39
synthetic biology, 239, 313
transcription factor, 232
tryptophan pathway, 214
enzyme-catalyzed reaction, *134*, 134, 286, *290*, 293, *310*, 315, 393
Enzyme Commission (EC) number, *267*
epidemics, 459
epidermal growth factor receptor, 270, *270*
epigenetic modification, 214
epigenetics, 214, 220, *222*, 223, 416
epinephrine, 383
epithelial mesenchymal transition (EMT), 470
ERG9 gene, 244
ERGO, 300
error function, 168, 176–177
Escherichia coli
barcode, 218, *219*
computation of volume, 30b
CheY signal transduction, 329
elementary mode analysis, 310–313, *312*
EnvZ two-component sensing system, 200
gene regulation, 229, 234
lac operon, 229
metabolic flux distribution, 310, *310*
modeling variability and disease, 418
network motif, 76
single mutant, 11
SOS pathway, 139, *140*
supercoiled DNA, 217, *217*
synthetic biology, 241
toggle switch model, 139, *140*, 241–242, *242*, 243
transcription factor, 232
two-component signaling system, 329–330, 332
ethanol, 266, *267*, 297, 312, 313
Euprymna, 342
evolutionary algorithm, 173, 177, *178*
evolutionary programming, 179
excitable cardiomyocyte, 383, 384, *384*, 395
excitation-contraction (EC) coupling, 384
exclusive Or (XOR), 337
exon, 220, *221*
expected value, 81
experimental systems biology, 421–422, *422*
explanatory model, 29, *29*
exploration, mathematical models, 26, 42, 44–45, 45b, *46*
exponential decay, 114
exponential growth, 94, 95, 114, 445
exposome, 419, 484, *485*
external validation, 44
extracellular matrix, 265, 269
extracellular-signal-regulated kinase (ERK), 334
extraction, metabolite analysis, 297, 316
extrapolation, 21

F

Factor VIII, 428
false-negative outcome, 425
false-positive outcome, 425
farnesyl pyrophosphate (FPP), 244, *244*
feature selection, structure identification, 197
feedback
metabolic pathway, 301, *301*, *302*
nested, 301, *302*
feedback inhibition, 36, *36*, 53
metabolic pathway, 301, *301*, *302*
feedback loop, 76–77, *76*
feedforward loop, static network models, 76–77, *76*
feedforward modulation, 149
fermentation, 27
fibroblast, 273
fibronectin, 265, 269
filament, heart, 381
finite element approach, heart, 381
first-degree heart block, 402
first-order process, 170

fish population, 482
fitness, 173
fitting, 168, 184, 185
FitzHugh-Nagumo equation, 401
fixed point, 98
flagellar motor, 269–270, *269*
flow chart, 24, *24*
flow of information, 36, *36*
flow of material, 36, *36*
fluid mechanics, heart, 381
flux, 60, 70
  metabolic system, 296, 298
flux analysis, metabolic, 298
flux balance analysis (FBA), 307–310, *308–310*, 313
flux network, 70–71, *70–71*
flux profile, 296
food, transgenic, 245
Food and Drug Administration (FDA), 425, *426*, 427, *429*, 439, 479
fractal, 99, 150, 476
Frank–Starling law, 380
free radical, 402
fructose, 1,6-bisphosphate, 8, 149
fructose, 6-phosphate (F6P), 310, *310*
Fugu rubripes, 217
functional loop, 8–9, *8*, *9*
funny current, 395
future development, 16–17
Fuzzy law, 489

G

gain analysis, 130
galactose, 229
galactose transacetylase, 229
β-galactosidase, 229, *229*
β-galactoside permease, 229, *229*
gamblers ruin, *102*, 102–103
Game of Life (Conway), 73
gap junction, 268, *268*, 384
gas chromatography with mass spectrometry (GC-MS), 297
gate, 74
gel electrophoresis, 262
gene
  arbitrary assignment, 72
  biological variability, 415–416
    modeling, 416–419, *417*, **417**, *418*
  Boolean network, 72
  Central Dogma, 213–215, *214*
  coding, 219
  defined, 219
  human, 6, 220
  information transfer, 214
  noncoding DNA, 218–221, *221*
  open reading frame, 220
  phenotype, 219
  transcription network, 53, 149
gene annotation, 220
gene expression
  control of metabolic pathway, 301–302
  structural pattern, 53
gene interaction network, 234, 236
Gene Ontology (GO) Consortium, 219
generalized mass action (GMA) system, 136–137, *137*, 357; *see also* biochemical systems theory (BST)
general stoichiometric model, 303–306, *305*, *306*
gene regulation, 228
  lac operon, 229–230, *229*, **230**
  model, 233–236, *234–237*
  mode, 231
  transcription factors, 213, 231–232, *233*
gene regulatory network, 84
  as signaling system, 338
gene systems, 213–245
  Central Dogma, 213–215, *214*
  DNA
    chemical and physical features, 215–217, *216–217*
    epigenetics, 214, 223
    eukaryotic packing, 221, *222*, *223*
    gene and noncoding, 218–221, *221*
    key properties, 215–223
    size and organization, 217–218, *219*
  RNA
    chemical and physical features, *216*, 224, *224*
    key properties, 224–228
    messenger, 224
    ribosomal, 226, *226–227*
    small, 226–228, *227*
    transfer, 224–225, *225*
    viruses, 228, *228*
  synthetic biology, 239–241
    artificial gene circuits, 241–244, *242*, *243*
    case studies, *244*, 244–245
genetic algorithms (GAs), 173, 177–179, *178*
mating process, 173, 178
parameter estimation, 173, 177–179, *178*
mutation, 178–179, *178*
genetic code, 218, 488
genetic variation, 84, 416
genome
  bacterial, 52–53, 231
  human, 6, 214, 227
  size and organization, 217–218, *219*
  structural patterns, 52, *52*
  transposon, 214
genome comparison, 315, *315*
genomic networks, 84
genomic regulation, cell cycle, 338
genomics
  meta-, 220, 460–461
  pharmaco-, 421
Gibbs sampler, 83
Gln (glutamic acid), **257**, **258**
global optimum, 175
Glu (glutamine), **257**
glucose, 285, *286*
  bacteria, 186
  ^{13}C-labeled, 316
  diabetes, 32
  enzymes, 265
  lac operon, 229–230, *229*
  metabolic engineering, 313
  metabolic flux distribution, 310, *310*
  metabolic networks, 69, 285, *286*, 287
  simple system, 8, 10
  small metabolite, 213
glucose, 6-phosphate (G6P)
  Escherichia coli, 310, *310*
  glycolysis, 8
glutamic acid (Gln), **257**
glutamine (Glu), **257**
glycan, 274
glycine (Gly), **257**, 272
glycogen synthase, 359
glycolysis
  enzymes, 265
  simple systems, 8
glycoprotein, 273–274
GMA systems, *see* generalized mass action (GMA) systems
goals, model development, 26
Golden Rice, 245
Goldman–Hodgkin–Katz equation, 394, 407
Gompertz growth law, 447
gout, 437
G6P, *see* glucose, 6-phosphate (G6P)
G-protein-coupled receptor (GPCR), 323–324, *324*, 395
G-proteins, 323–324
gradient methods, parameter estimation, 173, 176–177, *176*
Granger causality, 77
graph
  adjacency matrices, 63, *63*
  bidirectional, 61, *61*
  bipartite, *62*
  clique, 69
  clustering coefficient, 63–64, *64*
  defined, 61
  degree distribution, 65, *66*
  degrees, 63, *63*
  directed acyclic, 82
  directed *vs.* undirected, 61, *61*, 63–64, *63–64*, 73
  edge, 61, 62
  hub, 64–65, *65*
  interaction, 61–71
  node, 61–63, *63*, 64b
  properties, 63–66, *63–66*, 64b
  random, 65
  signaling networks as, 325
  theory, 62, *62*
  vertex, 61
green fluorescent protein (GFP), 242, *242–243*, 263
grid search, comprehensive, 174–175, *175*
GroEL/ES protein complex, 259
growth curve, children, 54, *54*
growth dynamics, 446
growth law, 446–448
growth rate, 446
guanine, 215, *216*
guanine-nucleotide-binding, 323
guanosine triphosphate (GTP), 323
guestimate, 126, 176–177
gut microbiome, 460–461

H

half-life, 115, 427
α-helix, 261
health
  delineation, *424*
  predictive, 419–425
heart, 377–408
  aging effects, 402
  anatomy, 377, *378*, 379–380, *380*
  electrochemistry in cardiomyocytes, 391–401
    biophysical description, 394–395
    models of action potential, 397–400, *400*
    repeated heartbeat, 400–401, *400*
    resting potential and action potential, 395–397, *395*, *396*
  failing, 401–408
    modeling based on molecular events, 403–408, *403–404*, *406*, *407*
  hierarchy of scales and modeling approach, 379–385
    basics of anatomy, 379–380, *380*
    at cell level, 383–385, *384*, *385*
    at organ level, 380–382, *381*
    at tissue level, 382–383
  outlook for multiscale modelling, 372
  overview, 377–378, *378*
  pacemaker, 383, 388, *388*, 400–401
  simple models of oscillations, 385–391, *386*
    black box, 291, 378, 386–389, *386–388*
    meaningful, 389–391, *390*, *391*
heartbeat
  repeated, 400–401, *400*
  simulation of perturbed, *387*
heart block, 402
heart failure, 401–408
  modeling based on molecular events, 403–408, *403–404*, *406*, *407*
heat-shock proteins (HSPs), 260, 356
heat stress response
  multiscale modelling, 356, 357–361, *359–360*
heme, *269*
hemoglobin, 268, *269*
Henri-Michaelis-Menten rate law, *see* Michaelis–Menten rate law
hepatocyte, 372
heterogeneous information merged into conceptual model, 15
hexokinase, 285, 286
hexose, 312, 313, *359*
high-density lipoproteins (HDLs), 274
high-performance liquid chromatography (HPLC), 297
high-throughput methods, of molecular biology, 11, *12*
hill-climbing methods, 173, 176–177, *178*
Hill function, 134
  heart, 391
  metabolic systems, 295
  parameter estimation, 39, 40, *40*, 165
  signaling systems, *326*, 328B
Hill rate law (HRL), 293, *293*
  pharmacokinetics, 434
histidine (His), **257**
histidine kinase, 330, 331
histone, 221, *222*
histone deacetylases (HDACs), *222*
hits, small-molecule, 426
Hodgkin-Huxley model, 397, 399, 400, *400*, *401*
homeostasis, 70, 96, 123, 420

homogeneous model, 15, 32
homogeneous population, 451
hormone, 270, 271
HRL, *see* Hill rate law
hub, 64–65, *65*
Human Cell Atlas, 470–471
human genome sequencing, 6
human-on-chip, 480
Human Protein Atlas, 255
hybridization, 215
hyperuricemia, 437–438
hypothesis, 15
hypoxanthine-guanine phosphoribosyltransferase (HGPRT), 438
hysteresis,
  signaling systems, 326–328, *327*, **327**, 328–329b

I

identifiability, 196
Ile (isoleucine), **257**
immune cascade, 197–200, *198*, *199*
immune system
  innate and adaptive, 471–473, *472*
  protein, 271–272, *271*, *272*
immunoglobulin, 271, *271*
independent variable, 36–37, *36*
induced pluripotent stem cell (iPSC), 480
inducer, 229, 230, 231
infection, 471–474, *473*
infectious disease modeling, 26
  asymptomatics, 47–49, *48*, *49*
  canonical, 134–135
  consistency and robustness, *43*, 44
  diagram, *38*
  model selection, 28
  model structure, 29
  parameter estimation, 40
inflammation, 471–475, *472*, *473*
information
  coarse *vs*. fine-grained, 488
  mutual, 77–78, *78*
information theory, 78
inhibition
  allosteric, 266–267, 294
  competitive, uncompetitive, and noncompetitive, 294
inhibitor, metabolic systems, 293–294
initial value, 115
innate immune system, 471–472, *472*, 474
input, 8–9, *9*
  model development, 26
  persistent changes, 145
insulin
  crystal structure, *261*
  diabetes modeling, 476
  drug development, 428
  model design, 32, 35
interacting population, 454–462
  general modeling strategy, 454, *454*
  mixed communities, 449b, 460–462, *460*, **461**, *462*
  phase-plane analysis, 454–459, *455–459*, **455**
interaction, 16
interaction graph, 61–71
  edge, 61, 62
  node, 61–63, *63*, 64b
  properties, 63–66, *63–66*, 64b
  stoichiometric networks, 69–71
  theory, 62, *62*
  vertex, 61
interaction networks, Bayesian reconstruction, 78–81, *80*
intriguing method, 263
intron, 220, *221*
inversion recombination element, 240
ion, 379, 392
ion channels
  heart, 392, 395, 397, 399, 408
ion transport, 267, 383, 394
iPSC, *see* induced pluripotent stem cell
isoleucine (Ile), **257**
isoprenoid pathway, 244
isozyme
  control of metabolic pathways, 302
iteration, 50

J

jackknife method, parameter estimation, 185
Jacobian matrix, 102, 119–122, *122*, *123*
Jmol software, 261, *261*
Julia sets, 99
junk DNA, 220, 467

K

Kai system, 345, *345*
KEGG database
  metabolic network reconstruction, 314–315
  metabolic systems, 298, 298b, *299*, 300, 310
KidneySim project approach, 372
kinetic models, 309, 345
kinetic order, 136
Kirchhoff's laws, 398
Kyoto Encyclopedia of Genes and Genomes, *see* KEGG database

L

lacI regulator gene, *229*, 230, 242–244, *243*
lac operon, 229–230, *229*, **230**
Lactococcus lactis, 316
lactose, 229–230
Laplace transform, 116
large-scale model assessment, 49–51, *51*
Latin hypercube technique, 147, 175, 371
Latin square technique, 147
lattice-based model, 370
lattice-free, 371
law, biological, 32, 54, 488
lead, 426
lead identification, computational, 428–429
lead optimization, 426
least-squares method, parameter estimation, 166
left ventricular hypertrophy, 402
lens, 274, *274*
Leslie model, population dynamics, 452–453, 453
leucine (Leu), **257**
leukemia, acute lymphoblastic, 421
Levinthal's Paradox, 259
life cycle, 16
ligand binding, 431
light organ, 342
limit-cycle oscillation, 135, 152, *154*, 243
limit cycle, 150–154, *151–154*
limiting distribution, 106
linear algebra, 124
linear chain, 8, *8*
linear differential equations, 114
linearization, 117
linearized ODE models, 116–123
  approximation, 116–119, *117*, 120–121b
  curved surface, 117, *117*
  example of, 114–115
  Jacobian, 119–122, *122*, *123*
  operating point, 117
  perturbation, 117
linear models, 93–94
linear programming, 308
linear recursion models, 94–95, *95*
linear regression, 29, *29*
  misuse, 168–169, *168*
  parameter estimation, 167–171
    several variables, 169–172, *169–172*, **170–171**
    single variable, 167–168, *167–168*
linear systems, 113
  linear differential equation, 114
  linearized ODE models, 116–123, *117–118*, 120–121b, *122–123*
  parameter estimation, 167–171
    several variables, 169–172, *169–172*, **170–171**
    single variable, 167–168, *167–168*
lin–log model, 139
Lipid Maps project, 300
lipo-oligosaccharide, *266*
lipoprotein, 274
liver
  compartment model, 432–435, *432–433*
  drug development, 427, 432–435, *432–434*, 435b
  enzyme activity profile, 60
  lipoprotein, 274
  multiscale modeling, 372
  Virtual Liver Network, 372, 379
local minimum, 173
local motif, 261
logistic function, graph, 130–131, *131*, 133
logistic growth function, 445–446
logistic map, 98–99, *98*
Lorenz attractor, 155, *156*
Lotka-Volterra (LV) model, 134, 135–136, *135*, 156
  interacting populations, 457–458, *458*, **461**, 461
  mixed communities of populations, 461–462
low-density lipoproteins (LDLs), 274
L-type calcium ($Ca2^{+}$) channels (LCCs), 395, 403–408, *403–404*
lung
  and heart, 379, 380
  infection and inflammation, 472, 474
  pharmacokinetic modeling, 432, 434, 435
  pneumonia, 447
lung cancer, 419
lung physiome, 379
lymphocytes, suppressor, 198, *198*, 199
lysine (Lys), **257**, 309

M

machine learning, 12, 179
malaria, 244–245
Mandelbrot sets, 99
map
  concept, 487, *487*
  logistic, 98–99, *98*
  metabolic, 314, *315*
MAPK cascades, *see* mitogen-activated protein kinase cascades
Markov chain Monte Carlo (MCMC) algorithms, 83
Markov matrix, 104, 107
  vs. adjacency matrix, **105**
Markov models, 94, 103–107, *105*, 107–108b
mark-up language, 12
mass action kinetics, 289
  heart function, 406–407, *407*
mass-action models, 136, 331; *see also* generalized mass action (GMA) systems
mass action systems, generalized, 136–137, *137*
mass spectrometry (MS), 262
  metabolite analysis, 297
Mathematica, 482
mathematical models, 21–55
  analysis, 41–46
  approximation, 31, *32*
  binomial, 33, 35b
  blue-sky catastrophe, 33, 33b
  computer simulation, 21, 41, *41*
  consistency and robustness, 42–44, *43*
  correlative *vs*. explanatory, 29, *29*
  data availability, 28
  defined, 21
  deterministic model, 32, 33b
  diagnostics, 41–43, *43*
  diagram, 35–37, *36*, *37*
  discrete *vs*. continuous, 31
  equations, 38–39
  exploration and validation of dynamical features, 42, 44–46, 45b, *46*
  extensions and refinements, 47–49, *48*, *49*
  external validation, 44
  flow chart of process, 24, *24*
  flow of information, 36, *36*
  flow of material, 36, *36*
  goals, inputs, and initial exploration, 26–28, *27*
  issues to address, 24–25, **25**
  large-scale model assessments, 49–51, *51*
  of MAPK signaling cascades, *335–337*, 335–337
  open *vs*. closed, 32
  overview of, 21–26, *22*, *24*, **25**
  overfitting, 44, 196
  parameters, 35, 37, 39–40, *40*
  Poisson, 33, 35b
  probabilistic or stochastic, 32–33, 34–35b
  processes and interactions, 35
  question of design and operation, 51–53, *51–52*
  question of scale, 26–27, *27*

selection and design, 28–40, *28*
sensitivity, 42–43, *43*
simplicity *vs*. complexity, 53–55, *54–55*
spatial aspects, 32
stability, 42
static *vs*. dynamic, 30, 30–31b
steady state, 42
structure, 29–33
system components, 35–38
of TCS systems, *331–333*, 331–333
use and applications, 46–55
variables, 32, 35–36, *36*
MATLAB, 373
matrix, 63, 93–94, 113
matrix equation
population dynamics, 452–453
red blood cell dynamics, 97
proteins, 265
MBDs (methyl-CG-binding domains), *222*
MCA (metabolic control analysis), 302–303, *303*
MCMC (Markov chain Monte Carlo) algorithms, 83
mechanistic mathematical models, 474
medical digital twins, 469
medicine, 415–440
biological variability and disease and, 415–416
modeling, 416–419, *417*, **417**, *418*
personalized (precision), 419–425
data needs and biomarkers, 420–421, *420*
mathematical models, 421–425, *422*
traditional *vs*. 422, *422*
predictive health, 419–425
membrane-bound receptors, drug development, 429, *429*
membrane potential, 394–396
Mendelian genetics, 489
messenger protein, 270–271, *214*
messenger RNA (mRNA), 224
metabolic burden, 307
metabolic control analysis (MCA), 302–303, *303*
lin–log model, 139
metabolic engineering, 306–307, 310–313, *312*
elementary mode analysis, 310–313, *312*
metabolic flux analysis, 298
metabolic map, 314, *315*
metabolic network
stoichiometric, 69–71, *70–71*
metabolic pathway, structural patterns, 53
metabolic profile, 296
metabolic reaction
mathematical formulation, 288–289
rate laws, 289–295, *290*, *291*, *292*
metabolic syndrome, 476
metabolic systems, 285–316
biochemical reactions, 286–296
background, 286–287, 287b
mathematical formulation of elementary reactions, 288–289
rate laws, 289–295, *290*, *291*, *293*
control pathway systems, 300, *301*, 302
metabolic control analysis, 302–303
database for, 298, **298**–**300**, 300
inference
dynamic models, 316
network reconstruction, 314–316, *315*
metabolite, 285–286, *286*
metabolome data generation methods, 297–298
overview, 285–286
pathway and pathway systems, 296–303
biochemistry and metabolomics, **295**–**296**, 296–297
control, 300–302, *301*, *302*
elementary mode analysis, 310–313, *311–312*
flux balance analysis, 307–310, *308–310*
general stoichiometric model, 303–306, *305–306*
methods for generation of metabolome data, 297–298
optimization, 313–314
steady state analysis, 306–307
metabolism, defined, 286
metabolite, 285–286, *286*
metabolomics, 286, 296–297
data generation methods, 297–298
MetaCyc database, 314
metadata, 481
metagenomics, 220, 460
methicillin-resistant *Staphylococcus aureus* (MRSA), 214
methionine (Met), **257**
method of mathematically controlled comparison (MMCC), 198–200, *199*
methotrexate (MTX), 435–436b
methyl-CG-binding domains (MBDs), *222*
methyl groups, 214
Metropolis-Hastings method, 83
mevalonate pathway, 244, *244*
Michaelis–Menten function
ad hoc, 134
MAPK cascade, *334*, 335
parameter estimation, 39, 169, 171
PBPK models, 434
power-law approximations, 143b
Michaelis–Menten mechanism, 335, 338
Michaelis–Menten process
biological variability and disease, 416–419, *417*, **417**, *418*
mathematical models, 289–290
pharmacokinetics, 434
power-law approximations, 143b
Michaelis–Menten rate law (MMRL), 196
metabolic systems, 289–295, *290*, *291*
parameter estimation
linear systems, 169, *169–171*, **170**, **171**
nonlinear systems, 174, *175*
microbial biofilm, 460, *460*
microbial community, 460
microbial systems, 241, 245
microbiomes, 449, 460, 461
microRNAs (miRNA), 220, 226, 227–228, 240
middle-out analyses, 17
miRNA (micro-RNA), 226, 227–228
mitochondria, 1
mitogen-activated protein kinase (MAPK) cascade, 333–337, *334–337*, **335**
adaptation, *337*, 337–338, *338*
features, 333–334, *334*
mathematical models, *335*, 335–337, *336*, *337*
MMRL, *see* Michaelis–Menten rate law
model
defined, 21
mathematical, *see* mathematical models
static network, *see* static network models
model extension, 47–49, *48*, *49*
modeling
defined, 21
needs, 482–488
modeling approach, 16
modeling tools, 16
module, 22
modulon, 232
molecular biology
DNA sequences, 220
high-throughput methods of, 11, *12*
molecular chaperone, 259–260, *259*
molecular dynamics, 262
molecular function, gene product, 219
molecular inventories, 16, 17
molecular systems, 78, 423
monoclonal antibody, 428, *429–431*, 429–431, **431**
Monte Carlo method/simulation 49–51, *51*, 147
iterations, 50
personalized medicine, 418, 423
motif, 21, 51, 197
design principles and, 51, 52, 53
dynamic, 147–150, *148*, *149*
network, 61, *74–76*, 74–77
static network models, 74–77, *74*, *75*
motor protein, *110*
multidisciplinary team, 15
multiple linear regression, parameter estimation, 169–172, *169–172*, **170**–**171**
multiple organ failure, 474
multiscale modeling, 27, 354–364, 372
accounting for host response, 362, *362*
case studies, 354–357, *355–357*, 361
heat stress response, 357–361, *359–360*
molecular *vs*. cellular scales, 354, *355*
two-scale model design, 362–365, *363*, *364*
multi-stage cancer model, 475
multistage process, 475
multivariate functions, power-law approximation, 143b
multivariate ODE system, linearization of, 122–123
multivariate systems, 35
muscle cell, 211, 273, 381, 384
Mus musculus, 315
mutation
analysis of system dynamics, 146
biologic variability, 416
cyanobacteria, 481
drug development, 244
evolutionary programming, 179
genetic algorithms, 178–179, *178*
immune system, 271
metabolic systems, 285
probability, 34b, 35b
random, 33
rate, 35b
sensitivity analysis, 130
sickle cell disease, 258, 419, 476
signaling systems, 345
simulated annealing, 180
small-world networks, 69
transposon, 214
vector, 173
mutual information, 77–78, *78*
Mycoplasma genitalium, 469
Mycoplasma pneumoniae, 315
myocardial cell, *see* cardiomyocyte
myocardial infarction (MI), 385
myocardium, 382
myocyte, heart, *see* cardiomyocyte
myofibril, 384
myofilament, 384, *384*
myosin, 384, *384*
Myxococcus xanthus, 333

N

Na^+-$Ca2^+$ (sodium-calcium) exchanger, 395–396
Na^+-K^+ ATPase (sodium-potassium adenosine triphosphatase), 396
Na^+-K^+ (sodium-potassium) pump, 393
nanomedicine, *13*
negative feedback, 148
negatively inducible operon, 231
negatively repressible operon, 231
Nernst equation, 394
nested feedback, 301, *302*
Netlogo, 365–370, **366**–**368**, *369*
network, 53, 59
Bayesian reconstruction, 78–81, *80*
dynamics, 71–72
Boolean network, *72*, 72–74, *73*
network motif, *74–76*, 74–77
flux, 70–71, *70–71*
gene regulatory, 84
metabolic, 314–316, *315*
motif, 61, 74–77, *74*, *75*
neural, 478–479, *478*
protein-protein interaction (PPI), 61, *65*, 84, *89*, 264, 265
random, 67b
regress function, 169
regular, 67b
regulated, 62
scale-free, 66, *66*
signaling, Bayesian analysis, 81–84, *82*
small-world, 67–69, 67–68b, *68*
static, 59–89
Bayesian reconstruction, 78–84
dependency among components, 77–78, *78*
interaction graphs, 61–71
metabolic, 60
small-world, 67–69, 67–68b, *68*
stoichiometric, 69–71, *70*, *71*
strategies of analysis, 60–61
stoichiometric, 69–71, *70–71*
types of generic, 61, *61*
undirected, 77
network motif, 61, 74–77, *74–76*
neural networks, 478–479, *478*
neurobiology, 477
neurodegenerative disease, 476–478, *477*, *478*
neuron disease, 476–478, *477*, *478*

neuron
action potential, 11, 397
multiscale modeling, 372
neural networks, 478–479, *478*
neuronal signal transduction, 477
neurotransmitter, 477
new chemical entities (NCEs), 425–426
nlinfit, 189, 191
NMR spectroscopy, *see* nuclear magnetic resonance (NMR) spectroscopy
node, 61
degrees associated with, 63–64, *63*
number of paths between, 64b
noise, parameter estimation, 182, *182*, 191, *191–192*, **192**
noisy channel, reliability of communication through, 77–78
noncoding DNA, 218–221, *221*
noncoding RNA, 240
noncompetitive inhibition, 294
nonlinear dynamics
models, 94
ad hoc, 133–134, *134*
canonical, 134–135
generalized mass action, 136–137, *137*
Lotka–Volterra, 135–136
S-system, 138–147, *140*, *141*, **142–145**, *146*
nonlinear partial differential equation, heart, 380
nonlinear recursion models, 97–99, *98–99*
nonlinear regression, 29, *29*
parameter estimation, 173, 176, *176*
nonlinear systems, 94
models
ad hoc models, 133–134, *134*
canonical models, 127, 134–135, *135*
parameter estimation, 172–185
appropriate functions, 194–196, *195*, *196*
comprehensive grid search, 174–176, *175*
design and operating principles, 197–200, *198*, *199*, *200*
gradient method, *176*, 176–177
genetic algorithms, 173, 177–179, *178*
nonlinear regression, 173, 176, *176*
other stochastic algorithms, 179–180
search algorithms, 173, *174*
typical challenges, 182–185, *182–184*, **183**, **185**
noradrenaline, 383
NTH1/2 gene, 361
nuclear magnetic resonance (NMR) spectroscopy, 261
nucleo-bases, DNA and RNA, 215, *216*
nucleoside, 215, *216*
nucleosome, 221, *222*, *223*
nucleotide, 215, *216*
nullcline, 342
interacting populations, 456–458, *456*, *457*
numerical integrator, 145
numerical mathematics, 12

O

objective function, parameter estimation, 166
oceans, 480–482, *482*
ODEs, *see* ordinary differential equations (ODEs)
Ohm's law, 398
open reading frame (ORF), 220
open system model, 32
operating point (*OP*), 117, 121
operating principle, 200, *200*; *see also* design principle
operational taxonomic units (OTUs), 460
operon, 52, *52*, 229
lac, 229–230, *229*, **230**
regulation, 231
structural patterns, 52, *52*
über- 52, *52*, 232
optimization, 62
constrained, 307
lead, 426
parameter estimation, 166
orbit, 150
ordinary differential equations (ODEs), 113
conversion of, 144–145b
defined, 113
heart, 378
metabolic systems, 288
population growth, 447–448
signaling systems modeled with, 338
steady-state analysis
determination of, 123–127, *125*, *126*
local stability analysis, *127*, 127–130, *128*, *129*, *130*
sensitivity analysis, 130–133, *131*
stability analysis, *127*, 127–130, *128*, *129*, *130*
theory of, 115
organism, interactions with environment, 480–482, *481–482*
organoid, 480
OR gate, 74, *75*, *236*
origin-binding domains (OBDs), *217*
oscillation pattern
chaotic, 155, 157
circadian clock, 339–341, *340–341*
damped, 9, *9*, 135, 190
gene circuit, 242, *242*
heart, 378, 382, 384–389, *386*
limit-cycle, 135, 152, *154*, 243
periodicity, 99
predator-prey, 457, *458*
sine-cosine, 115
stable, S-system, 386, *386*
unpredictable deterministic model, 33b
oscillations models, 385–391, *386*
black box, 291, 378, 386–389, *386–388*
meaningful, 389–391, *390*, *391*
repeated heartbeat, 400–401, *400*
overfitting, structure identification, 44, 196
oxidative stress, 402

P

p53 protein, 30
pacemaker, 388, *388*
pacemaker system abnormalities, 402
parameter
defined, 35
identifiability, 196–197
model development, 35
population, 446
sensitivity analysis, 130–131, *131*
parameter estimation, 165–200
bottom-up strategy, 167
identifiability, 196
branch-and-bound method, 175–176
evolutionary algorithms, 173, 177, *178*
linear systems, 167–171
several variables, 169–172, *169–172*, **170–171**
single variable, 167–168, *167–168*
in model selection and design, 28–40, *28*
noise, 182, *182*, 191, *191–192*, **192**
nonlinear systems, 172–185
appropriate function, 194–196, *195*, *196*
comprehensive grid search, 174–176, *175*
design and operating principles, 197–200, *198*, *199*, *200*
gradient method, *176*, 176–177
genetic algorithms, 173, 177–179, *178*
nonlinear regression, 173, 176, *176*
other stochastic algorithms, 179–180
search algorithms, 173, *174*
typical challenges, 182–185, *182–184*, **183**, **185**
overview, 166–167
simulated annealing, 180
sloppiness, 184
SSE (sum of squared errors), 166, 174, *175*
structure identification, 196
systems of differential equations, 185–194, 186b, *187–188*, **189–190**, *191–194*, **192–193**
time-series-based, 167, 186–194
uncertainty, 185
Parkinson disease, 476
partial differential equations (PDEs), 341, 380
heart, 380, 381, 404
particle swarm optimization (PSO), 179
path lengths, 67
pathogen
antibodies, 271
immune system, 471–474
lipo-oligosaccharides, *266*
quorum sensing, 241
Toll-like receptors, 272, *272*
pathogenicity, 241
pathway screening, dynamic models, 437–439, *437*
pathway systems
generic, *36*
metabolic, 286, 296–298
analysis, 60–61
elementary modes, 310–313, *311*, *312*
flux balance analysis, 307–310, *308*, *309*, *310*
general stoichiometric model, 303–306, *305*, *306*
genome size, 217
optimization, 313–314
steady state analysis, 306–307
screening with dynamic models, 437–439
simple, 8
patient-on-chip, 480
PDEs, *see* partial differential equations (PDEs)
pentose, 312, 313
pentose phosphate pathway (PPP), *88*, *318*, *356–357*
peptide, 258
peptide bonds, 256, *256*
period doubling, 99
permease, 393
personalized medicine, 419–425
data needs and biomarkers, 420–421, *420*
mathematical models, 421–425, *422*
traditional *vs.* 422, *422*
perturbation, 117
analysis of systems dynamics, 147
Bayesian reconstruction of interaction networks, 79
bistability and hysteresis, 326
data-modeling pipeline, 487
static network models, 199, 242, 243
design principles, 199
drug development, 437–438
heart, 386, 291, 407
inflammation, 471
limit cycles, 151, 152, *152*
linearized ODE models, 117–118
metabolic control analysis, 302–303
modeling variability and disease, 418
other biological networks, 84
parameter sensitivity, 130, 131
phase-plane analysis, 458, 459
population dynamics under external, 450–451, *450*, *451*
recovery from, 41
signaling networks, 84, 345, *346*
stability analysis, 127, *127*, 129–130
static network models, 59
steady-state analysis, 123
Petri net analysis, 62
stochastic, 328
pharmacodynamics (PD), 479
pharmacogenomics, 421
pharmacokinetic modeling, 432–437b, *432–434*
pharmacokinetics, 435–437b, 479
pharmacokinetic tests, 427
phase-plane analysis, 454–459, *455–459*, **455**
phenotype, 219
phenylalanine (Phe), **257**
phosphate-deoxyribose backbone, 215, *216*
phosphoribosylpyrophosphate synthase (PRPPS), 438
photosynthesis, 27, 481
phylogeny, 215
physiologically based pharmacokinetic (PBPK) models, 432, *434*, 434–435, 435–436b
physiological modeling of heart, 377–408
electrochemistry in cardiomyocyte, 391–401
biophysical description, 394–395
models of action potential, 397–400, *400*
repeated heartbeat, 400–401, *400*
resting potential and action potential, 383, 384, 395–397, *395*, *396*
failing heart, 401–408
modeling based on molecular events, 403–408, *403–404*, *406*, *407*
hierarchy of scales and modeling approaches, 379–385
basics of heart anatomy, 379–380, *380*
at cell level, 383–385, *384*, *385*

at organ level, 380–382, *381*
at tissue level, 382–383
outlook for multiscale modelling, 372
overview, 377–378, *378*
simple models of oscillation, 385–391, *386*
black box, 291, 378, 386–389, *386–388*
meaningful, 389–391, *390*, *391*
Physiome Project, 379, 401, 439, 469
piwi-interacting RNAs (piRNAs), 228
plasma membrane calcium adenosine triphosphatase (PMCA), 393
plasmid, 214, 221
Plasmodium falciparum, 244, 245
platform-based design (PBD), 240
pneumonia, 447
Poisson model, 33, 35b
Polychaos dubium, 217
population
homogeneous, 451
interacting, 454–462
general modeling strategy, 454, *454*
mixed communities, 449b, 460–462, *460*, **461**, *462*
phase-plane analysis, 454–459, *455–459*, **455**
of subpopulations, 451, 543, 453b
population dynamics, 38
ad hoc models, 133
under external perturbations, 450–451, *450*, *451*
mixed communities, 449b, 460–462, *460*, **461**, *462*
nullcline, 456–457, *456*, *457*
predator-prey systems, 457, 459, *see also* Lotka-Volterra (LV) model
population growth, 445–448
exponential, 445
logistic, 445
complex phenomena, 448, 448–449b
stochastic events, 446
traditional models, 446–448
trends, 448b
population parameters, 446
positively inducible operon, 231
positively repressible operon, 231
post-approval phase, drug development, 427
post-transcriptional modifications (PTMs), 260
potassium channel, 395, 399
potassium leak channel, 392
power function, 95
power-law approximation, 137
of multivariate function, 143b
of nonlinear function, 142b
power-law distribution, 65, *66*
power-law function, 136
pharmacokinetics, 434
population growth, 449b
power-law model, biological variability and disease, 417, **417**, *418*
precision, adaptation, 337, *337*
precision medicine, *see* personalized medicine
preclinical phase, drug development, 425–426, *426*
predation, population dynamics, 450
predator-prey systems, 457, *458*, 459, 461
prediction, 16, 21
predictive health, 419–425
PR interval, 396, *396*
pro (proline), **257**, 272–273
probabilistic laws, 440
probabilistic models, 32, 34b, 35b
probability, conditional, 80, *80*
probability distribution, binomial, 34b
probability tree, 103
for cancer diagnosis, 103, *103*
pro-inflammatory cytokine, 473
prokaryote
codon usage bias, 258
flagella, 270
gene, 232
MAPK cascades, 333
synthetic biology, 239
transcription factors, 232
prokaryotic cell, 460
proline (Pro), **257**, 272–273
promoter, 220, *222*
protease, 140, 265, 428, *429*
proteasome, 260
protein, 255–278
activity and dynamics, 274–278, *275–278*, **277**
chemical and physical features, 256–260
amino acids, 256, *256*, **257**, 258
conformation (folding), 258–259
peptide bond, 256, *256*
posttranscriptional modifications, 256
primary structure, 261
quaternary structure, 261
secondary structure, 261
shape, 261, *261*
size, 258
tertiary structure, 256, 261
turnover, 260
overview, 255
experimental assessment, 260–225
green fluorescent protein, 263
interaction, 263–264, *263*, *264*
localization, 263
proteomics, 262
structure, 260–262, *261*, 262b
roles and functions, 264–274
enzymes, 265–267, *266*, *267*
immune system, 271–272, *271*, *272*
signaling and messenger, 270–271, *270*
structure, 272–274, *273*, *274*
transporter and carrier, 267–270, *268*, *269*
Protein Data Bank (PDB), 261, *261*, **262**
protein disulfide isomerases (PDIs), 260
protein folding, 13, 260–262
protein-protein interaction (PPI) networks, 61, *65*, 84, *89*, 264, *264*, *265*
protein scaffolds, 221, *222*
proteome, 262
proteomics, 262
community, 461
metabolic pathways, 297
Pseudomonas aeruginosa, 341
pulmonary artery, *380*, 380
pulmonary vein, *380*, 380
pump, 393
purine, DNA and RNA, 215, *216*
purine metabolism, 437, *437*
Purkinje fibers, 382, 384, 400
P wave, 396, *396*
pyrimidine, DNA and RNA, 215, *216*
Pyrococcus furiosus, *219*
pyrrolysine, 256, **257**
pyruvate, 8, 266, *267*

Q

QRS complex, 396, *396*
QT interval, 396, *396*
qualitative analysis, interacting populations, 454–459, *455–459*, **455**
qualitative behavior, validation of model, 44
quantitative structure-activity relationships (QSARs), 426
quantitative systems pharmacology (QSP), 479
quasi-steady-state assumption (QSSA), 290, 295
quaternary structure, 261
quenching of metabolic activity, 297
quorum sensing (QS), 241, 341–345, *342–344*

R

radical oxygen species, 474
radioactive decay, 288
random event, 453b
random graph, 65
random mutagenesis, 33
random mutation, 33
randomness, 32–33
in dynamical systems, 108–109
random network, 67b
random variable, 32
random walk, *102*, 102–103
rate constant, 116, 288
rate laws, 289–295, *290*, *291*, *292*
Reactome, 300
receptor, 270
receptor dynamics, drug development, 429–432, *429–432*, **431**
receptor-gated channel, 392
receptor tyrosine kinases (RTKs), 270, 324
recursion, 453
recursive models
deterministic
linear recursion, 94–95, **95**
nonlinear recursion, 97–99, *98–99*
randomness with, 107
red blood cell, 95–97, *96*
steady state, *99*, 99–102, 99–102, *102*
stochastic
Markov chain model, 103–107, **105**, *105*, **107–108**
probability tree, 103, *103*
random walk, *102*, 102–103
red blood cell
case studies, 95–97, *96*
drug development, 244
heart, 379
trends over time, 95–97, *96*, 316
reductionism, 6–7, *6*, *7*
refractory period, 382
regular network, 67b
regulated network, 62
regulatory signal, 9, *8*, *9*
regulon, 231–232
structural pattern, 52, *52*
relative sensitivities, 131
repeller, 150
replication, 215, *217*
repolarization, 385, 395, 396, *396*, 397
lengthening of action potential, 402
repressilator system, 242, *242–243*
repressor, 229, 230
repressor module, 243
residual error, 126
resistance, 397
resting potential, 384, 395–397
reverse engineering, 78
rheumatoid arthritis, 428, 471
ribonucleic acid, *see* RNA (ribonucleic acid)
ribose, *216*
ribosomal RNA (rRNA), 226, *226*, *227*
ribosome, 1, 224–225, 226, *227*, 259
riboswitch, 240, 240b, *240*
Richards function, 447, 448
RISC (RNA-induced silencing complex), 227
RNA (ribonucleic acid)
chemical and physical features, *216*, 224, *224*
double-stranded, 226–228, *227*
key properties, 224–228
messenger, 224
micro- 220, 226–228
noncoding, 240
piwi-interacting, 228
ribosomal, 226, *226–227*
short-interfering, 226
silencing, 226–227, *227*, 240
single-stranded, 224, *225*, 225, 228
small, 226–228, *227*
small activating, 228
small-interfering, 226–227, *227*, 240
small temporal, 227
transfer, 224–225, *225*
virus, 228, *228*
RNA-induced silencing complex (RISC), 227
RNA interference (RNAi) pathways, 226–227, 440
RNA polymerase, 224, 230
RNA virus, 228, *228*
robotics, 11
robust models, 130
robustness
biological networks, 53
mathematical models, 42–44, 43, 120
root finding algorithm, 125
RuBisCo, 264
Ryanodine receptor (RyR), 403–405, *404*, 407–408

S

Saccharomyces cerevisiae, *65*, 229, 244, 298, *313*, 354
sampling, metabolite analysis, 297

sarcolemma, 403, *403*
sarcoplasmic reticulum (SR), 389
sarcoplasmic reticulum (SR) calcium ($Ca2^+$) adenosine triphosphatase (ATPase) (SERCA) pumps, 403
scaffold protein, 221, 270
scale, 10
  model development, 26–27, *27*
scale-free network, 66, *66*
scatter plot, 167
screening experiment, comprehensive, 147
search algorithm, 126
search methods, 166
secondary structure, RNA, 224, *224*
second messengers, 324
semantic web tools, 481
semilunar outflow valve, 380
sensitivity
  adaptation, 337–338, *337*
  mathematical model, 42–43, *43*
  parameters, 130–131
separation of timescales, 354, 405
sepsis, 471–475, *472*, *473*
serine (Ser), **257**
serum albumin, 268
shock state, 474
short-interfering RNA (siRNA), 226–227, *227*
sickle cell anemia, 419, 476
sick sinus syndrome, 402
sigmoidal function, 130, *131*
signal, 323
signaling cascade
  defined, 81
  MAPK, 333–334
  metabolic pathway analysis, 300
  model design, 36, *36*
  network motif, 74, 75
  Toll-like receptors, 272, *272*
signaling molecule
  cAMP, 230
  DNA splicing, 220
  drug development, 426
  gene regulation, 234
  G-protein, 323, 324
  immune response, 271
  quorum sensing, 241
  simple system, 8
signaling networks, *see* signaling systems
signaling pathways, structural patterns, 53
signaling protein, 270, *270*
signaling systems, 323–346
  Bayesian analysis, 81–84, *82*
  G-protein-coupled receptor (GPCR), 323–324, *324*
  modeled with differential equations, 325–346
    adaptation, 337–338, *337*, *338*, **338**
    bistability, 325–326, *326*
    hysteresis, 326–328, *327*, 328–329b
    mitogen-activated protein kinase cascades, 333–337, *334–337*, **335**
    other, 338–346, *340–346*, **346**
    two-component, 329–333, *330–333*, **332**
  overview, 323–325, *324*
  second messenger, 324
signal transduction systems, 323–346
  cell-free systems, *345*, 345–346, *346*
  circadian clock, 339–341, *340*, *341*
  differential equations models
    bistability, 325–326, *326*
    hysteresis, 326–328, *327*, **328–329**
  defined, 270, 323
  enzyme, 324
  extracellular matrix, 265
  G-protein-coupled receptor, 323–324, *324*
  G-protein, 324
  heart, 384, 392, 393, 372
  importance, 81
  lipids, 300
  MAPK cascade, 333, 336, 337
    adaptation, *337*, 337–338, *338*
    features, 333–334, *334*
    mathematical models, *335*, 335–337, *336*, *337*
  metabolic systems, 285
  network motif, 74, 75
  networks, *see* signaling systems
  neuronal, 477
  neurotransmitter, 267
  overview, 323–325
  personalized medicine, 422
  protein states, *36*
  quorum sensing, 341–345, *342*, *343*, *344*
  receptor tyrosine kinases, 270, 324
  second messengers, 324
  two-component
    mathematical models, *331*, 331–333, *332*, *333*
    overview, 329–331, *330*, *331*
signal transduction pathways, 325
signal transduction protein, 329
silencing RNA (siRNA), 226–227, *227*, 240
simulated annealing (SA), 180
simulation, 21, *41*, 42, 44
sine–cosine oscillation, 115, 150, *151*
single-cell analysis, 468
single-cell RNA sequencing (scRNA-seq) technology, 470
single nucleotide polymorphism (SNP), 416
sinoatrial (SA) node, 382
SIR model, 38, 41, 45b
  case studies, 361
  linearization, 121–122, *123*
  Netlogo, 366, **366–368**
  population dynamics, 451
SIRS-VE model, *362*, 362–364
size scale, 10
sliding filament model, 385
sloppiness, parameter estimation, 184
small activating RNA (saRNA), 228
small-interfering RNA (siRNA), 226–227, *227*, 240
small-molecule hits, 426
small RNAs, 226–228, *227*
small temporal RNAs (stRNAs), 227
small-world networks, 67–69, 67–68b, *68*, 418
smoothing, parameter estimation, 188
sodium-calcium (Na^+-$Ca2^+$) exchanger, 395
sodium-potassium adenosine triphosphatase (Na^+-K^+ ATPase), 396
sodium-potassium (Na^+-K^+) pump, 393
Somogyi–Stucki model, 390, *390*, *391*
source antigen, 198–199
SoySim model, 469
spatial scale, 10
sphingolipids, 355
SR $Ca2^+$ ATPase (SERCA) pumps, 403
S-systems, 138–139, *140–141*, 141, 142–145b, *146*
  oscillations, 386, *386*
stability analysis, *127*, 127–130, *128–130*
stable, 99
stable limit cycle oscillations, 145
stable oscillations, 135
stable spiral, visualization of, 128, *128*
Staphylococcus aureus, 341
  methicillin-resistant, 214
START codon, **258**
state, mathematical model, 42
static condition, 60
static models, 30, 30–31b
static network models, 59–84
  causality analysis, 77
  dependencies among network components, 77–78, *78*
  graph
    adjacency matrix, 63, *63*
    bipartite, *62*
    clique, 69, 84
    clustering coefficient, 63–64, *64*
    defined, 61
    degree distribution, 65, *66*
    degree, 63, *63*
    edge, 61, 62
    hub, 64–65, *65*
    interaction, 61–71
    node, 61–63, *63*, 64b
    properties, 63–66, *63–66*, 64b
    theory, 62, *62*
    vertex, 61
  mutual information, 77–78, *78*
  networks, 61
    Bayesian reconstruction, 78–81, *80*
    dynamics, 71–77
    generic, 61, *61*
    regulated, 62
    scale-free, 66, *66*
    small-world, 67–69, 67–68b, *68*
    stoichiometric, 69–71, *70*, *71*
  optimization, 62
  overview, 59–60
  Petri net, 62
  strategies of analysis, 60–61
static structure of biological systems, 16
statistical association analysis, 421
statistical association models, 483
steady state
  defined, 123–124
  determination of, 123–127, *125*, *126*
  local stability analysis, *127*, 127–130, *128–130*
  mathematical model, 42
  metabolic pathway systems, 306–307
  population growth, 446
  sensitivity analysis, 130–133, *131*
  trivial, 124
steady-state analysis, 123–124, *125*, *126*
steepest-descent methods, 173
stimulon, 232
stimulus, validation of model, 44
stochastic event, 446
stochasticity, 108, 238, 365
stochastic models, 32–33, 34b
stochastic process, 404
stochastic recursive models, 94
  Markov chain model, 103–107, **105**, *105*, **107–108**
  probability tree, 103, *103*
  random walk, *102*, 102–103
stochastic simulation, 49
stoichiometric analysis, 70, 303
stoichiometric matrix, 70, 71, *71*
stoichiometric networks, 69–71, *70–71*
stoichiometry, 70
STOP codon, **258**
strange attractor, 155, *155*
stratified population, 451
stress response elements (STREs), 356
stress response system, 6–8, *7*, 356
structural patterns, 51–52, *51*
structure, persistent change, 145
structure-activity relationships (SARs), quantitative, 426
structure identification, 196
structure protein, 272–274, *273*, *274*
ST segment, 396, *396*
subpopulation
  analysis, 451–453, 453b
sudden cardiac death, 408
sugar, DNA and RNA, 215, *216*
sum of squared errors (SSE), 166, 174, *175*
supercoiled DNA, 217, *217*
superposition, 113–114
suppressor lymphocyte, 198, 199
symbolic equation, 38
symporter, 393
Synechococcus, 345, 481, *481*
synergism, 11, 15
synthetic biology, 214, 239–241, 240b
  artificial gene circuits, 241–244, *242*, *243*
  case studies, *244*, 244–245
system components, model development, 35–38
system matrix, 127
systems analysis, collecting information in, 5–6, *6*
systems biology
  application of, 16–17
    cancer, *475*, 475–476
    cell, organ, and organism, 468–470
    infection, inflammation, and sepsis, 471–475, *472*, *473*
    single-cell analysis, 470–471
  communicating, 13–15, *14*
  complex disease, 476
  design and operation questions, 52
  drug development, 427–440
  emerging applications, 468–482
  emerging topics, 467–489
  environment organisms, 480–482, *481*, *482*
  history of, 10–13, *12*, *13*
  long-term goals, 468
  machine learning and artificial intelligence, 485–488, *486*, *487*
  in mainstream, 10–13, *12*, *13*
  medicine, 415–440

modeling needs, 482–488
data-modeling pipelines, 484–485, *485*
neuron, brain, and neurodegenerative disease, 476–478, *477*, *478*
personalized medicine and drug screening organoids, 480
quantitative systems pharmacology, 479
reductionism and, 6–7, *6*, *7*
theories, 488–489
Systems Biology for Energy and Environment, 481
systems biology mark-up language (SBML), 12, 482
systems ecology, 69
systole, 379

T

tachyarrhythmia, 402
target identification, computational, 426
Taylor approximation theorem, 131, 120b
canonical models, 141, 142b, 143b
linearized ODE models, 116–119, 120–121b, *122*
parameter sensitivity, 131
T cell receptor signaling, 75, *75*
template-and-anchor model, 372
temporal scale, 10
termite mound, 1, *2*
tertiary structure, protein, 256, 261
text mining, 428
T helper cell, 271
theorem, 488
theory of biology, 488–489
thermodynamic hypothesis, 258
Thomas oscillator, 156, *156*
threonine (Thr), **257**
threshold, 149
thymine, 215, *216*
timescale, 10
separation of, 405
time-series-based parameter estimation, 167, 186–194, 186–188b, *188–194*
tissue necrosis factor (TNF), 473
toggle switch model
gene circuit, 241–242, *242*, 243
SOS pathway in *E. coli*, 139, *140*
Toll-like receptors (TLRS), 272, *272*
top-down analyses, 17
top-down estimation, 40
toxicity, 426–427
TPS1/2 gene, 357
training set, parameter estimation, 185
trait, biological, 218
trajectory, 150
transcription, 213, 215, 221, 223
hierarchical control, 232, *233*
transcription factors (TFs), 194, 213, 231–232, *233*
transcriptome, 470
transfer RNA (tRNA), 224–225, *225*
transforming growth factor β (TGF-β), 473
transgenic food, 245
transient cell cycle arrest, 355
transient decrease, translation initiation, 355
translation, 219
translation initiation, transient decrease, 355
transmembrane protein, 267
transporter, 267–270, *268*, *269*
transposable element, 214
transposon, 214
trehalose, 355, 356, *356*
trehalose cycle
multiscale analysis
heat stress response, 357–361, *359–360*
trehalose, 6-phosphate (T6P), 355
trial, 32
trivial steady state, 124, 125
troponin, 384, 389
tryptophan (Trp), 213–214, **257**
T-tubules, 403
tumor cell, 445, 454
tumor development, 475–476, *475*
epigenetics, 223
proteins, 265
tumor growth, 371, 454, 475
tumorigenesis model, 475
tumor suppression protein, 30
tumor suppressor gene, *222*
turn, 261
T wave, 396, *396*
two-component signaling (TCS) systems, 329–333, *330–333*, **332**
background, 329–331, *330–331*
mathematical models, *331*, 331–333, *332–333*
two-scale model
multiscale approach, 362–365, *363*, *364*
tyrosine (Tyr), **257**

U

über-operon, 52, *52*, 232
ubiquitin, 260
ubiquitination, 262, 264, 301
UCP2 gene, 244
ultrasensitivity, 342
uncertainty, parameter estimation, 185
uncompetitive inhibition, 294
undirected graph, 61, *61*, 63–64, *63–64*, 73
uniporter, 393
uracil, 215, *216*, 224
uric acid, 437, 438
uricase, 438

V

validation, 44–45, 45b, *46*, 185
valine (Val), **257**, 273
van der Pol oscillator, 151, 152, *152*, 386
variability, 28, 32
and disease, 415–419, *417*, **417**, *418*
variable, 32–33
dependent, 36–37, *37*
independent, 36–37, *36*
random, 23
variation in biological components, 9
vector, 93–94, 173, 458
field, 458
vena cava, 380, *380*
ventricle, 379–380, *380*
ventricular fibrillation, *381*
ventricular tachycardia, *381*, 402
vertex, 61
Vibrio fischeri, 241
virtual clinical trial
for drug development, 439–440
Virtual Liver Network, 372, 379
virtual metabolic human database, 469
Virtual Physiological Rat Project, 372, 403
virulence, 331, 341, 475
virus, 1
vitamin deficiency, 245
voltage, 398
voltage-gated channels, 392

W

whole-cell modeling, 468–469
"wild type" bacteria, 17
windows intuitive model of vegetation response to atmosphere and climate change (WIMOVAC), 469
Wobble, 218, 258

X

XML (customized mark-up languages), 12
X-ray crystallography, 261

Y

yeast
alcohol fermentation, 27
artemisinic acid production, 244–245, *245*
cell cycle, 69, 338
drug development, 244–245, *244*
gene regulation, 229, 234
intracellular metabolites, 297
metabolic pathway, 298b, 299b
signaling cascade, 337
sphingolipid, 232
transcription regulatory network, 64, *65*
two-hybrid assay, 264
Young-Laplace equation, 380

Z

ZWF1 gene, *357*