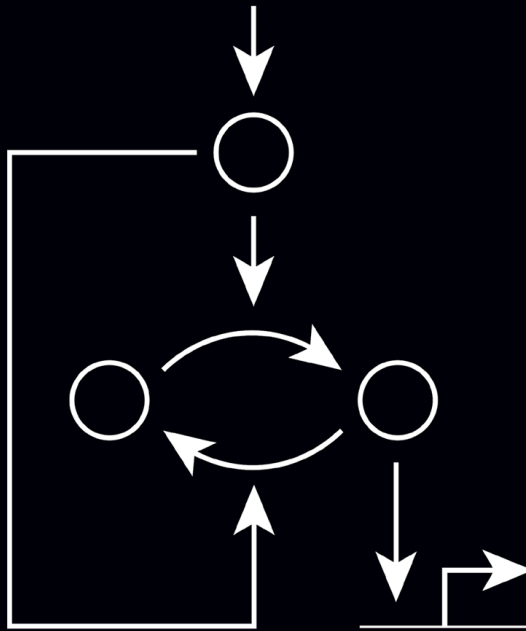


SECOND EDITION

An Introduction to Systems Biology

Design Principles of Biological Circuits



Uri Alon



CRC Press
Taylor & Francis Group

A CHAPMAN & HALL BOOK

7 New
Chapters

An Introduction to Systems Biology

Design Principles of Biological Circuits

Second Edition



Taylor & Francis

Taylor & Francis Group

<http://taylorandfrancis.com>

An Introduction to Systems Biology

Design Principles of Biological Circuits

Second Edition

Uri Alon



CRC Press

Taylor & Francis Group

Boca Raton London New York

CRC Press is an imprint of the
Taylor & Francis Group, an **informa** business
A CHAPMAN & HALL BOOK

CRC Press
Taylor & Francis Group
6000 Broken Sound Parkway NW, Suite 300
Boca Raton, FL 33487-2742

© 2020 by Taylor & Francis Group, LLC
CRC Press is an imprint of Taylor & Francis Group, an Informa business

No claim to original U.S. Government works

Printed on acid-free paper

International Standard Book Number-13: 978-1-4398-3717-7 (Paperback)
978-1-138-49011-6 (Hardback)

This book contains information obtained from authentic and highly regarded sources. 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, please access www.copyright.com ([http://www.copyright.com/](http://www.copyright.com)) or contact the Copyright Clearance Center, Inc. (CCC), 222 Rosewood Drive, Danvers, MA 01923, 978-750-8400. CCC is a not-for-profit organization that provides licenses and registration for a variety of users. For organizations that have been granted a photocopy license by the CCC, a separate system of payment has been arranged.

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: Alon, Uri, 1969- author.
Title: An introduction to systems biology : design principles of biological circuits / Uri Alon.
Description: Second edition. | Boca Raton, Fla. : CRC Press, [2019] |
Includes bibliographical references and index.
Identifiers: LCCN 2019003924 | ISBN 9781439837177 (pbk : alk. paper) |
ISBN 9781138490116 (hbk : alk. paper) | ISBN 9780429283321 (ebk)
Subjects: LCSH: Systems biology. | Computational biology. |
Biological systems--Mathematical models.
Classification: LCC QH324.2 .A46 2019 | DDC 570.1/13--dc23
LC record available at <https://lcn.loc.gov/2019003924>

Visit the Taylor & Francis Web site at
<http://www.taylorandfrancis.com>

and the CRC Press Web site at
<http://www.crcpress.com>

Contents

Introduction, xv

PART 1 Network Motifs

CHAPTER 1 ■ Transcription Networks: Basic Concepts	3
1.1 Introduction	3
1.2 The Cognitive Problem of the Cell	3
1.3 Elements of Transcription Networks	5
1.3.1 Separation of Timescales	7
1.3.2 The Signs on the Arrows: Activators and Repressors	8
1.3.3 The Numbers on the Arrows: Input Functions	9
1.3.4 Logic Input Functions: A Simple Framework for Understanding Network Dynamics	10
1.3.5 Multi-Dimensional Input Functions Govern Genes with Several Inputs	11
1.4 Dynamics and Response Time of Simple Regulation	13
1.4.1 The Response Time of Stable Proteins Is One Cell Generation	15
Further Reading	15
Exercises	16
Bibliography	18
CHAPTER 2 ■ Autoregulation: A Network Motif	21
2.1 Introduction	21
2.2 Patterns, Randomized Networks and Network Motifs	21
2.2.1 Detecting Network Motifs by Comparison to Randomized Networks	23
2.3 Autoregulation Is a Network Motif	23

2.4	Negative Autoregulation Speeds the Response Time of Gene Circuits	24
2.4.1	Rate Analysis Shows Speedup for Any Repressive Input Function $f(X)$	28
2.5	Negative Autoregulation Promotes Robustness to Fluctuations in Production Rate	29
2.6	Summary: Evolution as an Engineer	30
	Further Reading	31
	Exercises	31
	Bibliography	36
CHAPTER 3 ■ The Feedforward Loop Network Motif		37
3.1	Introduction	37
3.2	The Feedforward Loop Is a Network Motif	37
3.3	The Structure of the Feedforward Loop Gene Circuit	39
3.4	Dynamics of the Coherent Type-1 FFL with AND Logic	41
3.5	The C1-FFL Is a Sign-Sensitive Delay Element	42
3.5.1	Delay Following an ON Step of S_x	43
3.5.2	No Delay Following an OFF Step of S_x	44
3.5.3	The C1-FFL Is a Sign-Sensitive Delay Element	44
3.5.4	Sign-Sensitive Delay Can Protect against Brief Input Fluctuations	44
3.5.5	Sign-Sensitive Delay in the Arabinose System of <i>E. coli</i>	45
3.6	OR-Gate C1-FFL Is a Sign-Sensitive Delay for OFF Steps	46
3.7	The Incoherent Type-1 FFL Generates Pulses of Output	47
3.7.1	The Incoherent FFL Can Speed Response Times	48
3.7.2	Interim Summary: Three Ways to Speed Your Response Time	49
3.7.3	The I1-FFL Can Provide Biphasic Steady-State Response Curves	50
3.8	The Other Six FFL Types Can Also Act as Filters and Pulse Generators	51
3.9	Convergent Evolution of FFLs	51
3.10	Summary	52
	Further Reading	52
	Exercises	53
	Bibliography	59
CHAPTER 4 ■ Temporal Programs and the Global Structure of Transcription Networks		61
4.1	Introduction	61
4.2	The Single-Input Module (SIM) Network Motif	61

4.3	The SIM Can Generate Temporal Gene Expression Programs	62
4.4	The Multi-Output Feedforward Loop	64
4.5	The Multi-Output FFL Can Generate FIFO Temporal Programs	66
4.5.1	The Multi-Output FFL Also Acts as a Persistence Detector for Each Output	68
4.6	Signal Integration by Bi-Fans and Dense-Overlapping Regulons	68
4.7	Network Motifs and the Global Structure of Sensory Transcription Networks	70
4.8	Interlocked Feedforward Loops in the <i>B. subtilis</i> Sporulation Network	70
	Further Reading	73
	Exercises	73
	Bibliography	75
CHAPTER 5 ■ Positive Feedback, Bistability and Memory		77
5.1	Network Motifs in Developmental Transcription Networks	77
5.1.1	Positive Autoregulation Slows Responses and Can Lead to Bistability	78
5.1.2	Two-Node Positive Feedback Loops for Decision-Making	80
5.1.3	Regulating Feedback and Regulated Feedback	83
5.1.4	Long Transcription Cascades and Developmental Timing	84
5.2	Network Motifs in Protein–Protein Interaction Networks	85
5.2.1	Hybrid Network Motifs Include a Two-Node Negative Feedback Loop	85
5.2.2	Hybrid FFL Motifs Can Provide Transient Memory	87
5.2.3	Feedforward Loops Show a Milder Version of the Functions of Feedback Loops	87
5.3	Network Motifs in Neuronal Networks	88
5.3.1	Multi-Input FFLs in Neuronal Networks	89
5.4	Reflection	91
	Further Reading	91
	Exercises	92
	Bibliography	94
CHAPTER 6 ■ How to Build a Biological Oscillator		97
6.1	Oscillations Require Negative Feedback and Delay	97
6.1.1	In Order to Oscillate, You Need to Add a Sizable Delay to the Negative Feedback Loop	97

6.2	Noise Can Induce Oscillations in Systems That Have Only Damped Oscillations on Paper	101
6.3	Delay Oscillators	102
6.4	Many Biological Oscillators Have a Coupled Positive and Negative Feedback Loop Motif	103
6.5	Robust Bistability Using Two Positive Feedback Loops	107
	Further Reading	110
	Exercises	110
	Bibliography	112

PART 2 Robustness

CHAPTER 7 ■	Kinetic Proofreading and Conformational Proofreading	117
7.1	Introduction	117
7.2	Kinetic Proofreading of the Genetic Code Can Reduce Error Rates	118
7.2.1	Equilibrium Binding Cannot Explain the Precision of Translation	119
7.2.2	Kinetic Proofreading Can Dramatically Reduce the Error Rate	121
7.3	Recognition of Self and Non-Self by the Immune System	123
7.3.1	Equilibrium Binding Cannot Explain the Low Error Rate of Immune Recognition	124
7.3.2	Kinetic Proofreading Increases Fidelity of T-Cell Recognition	125
7.4	Kinetic Proofreading Occurs in Diverse Processes in the Cell	127
7.5	Conformational Proofreading Provides Specificity without Consuming Energy	128
7.6	Demand Rules for Gene Regulation Can Minimize Errors	129
	Further Reading	130
	Exercises	131
	Bibliography	136

CHAPTER 8 ■	Robust Signaling by Bifunctional Components	137
8.1	Robust Input–Output Curves	137
8.2	Simple Signaling Circuits Are Not Robust	138
8.3	Bacterial Two-Component Systems Can Achieve Robustness	140
8.3.1	Limits of Robustness	143
8.3.2	Remarks on the Black-Box Approach	143

8.3.3 Bifunctional Components Provide Robustness in Diverse Circuits	144
Further Reading	144
Exercises	145
Bibliography	150
CHAPTER 9 ■ Robustness in Bacterial Chemotaxis	153
9.1 Introduction	153
9.2 Bacterial Chemotaxis, or How Bacteria Think	153
9.2.1 Chemotaxis Behavior	153
9.2.2 Response and Exact Adaptation	155
9.3 The Chemotaxis Protein Circuit	156
9.3.1 Attractants Lower the Activity of X	157
9.3.2 Adaptation Is Due to Slow Modification of X That Increases Its Activity	158
9.4 The Barkai–Leibler Model of Exact Adaptation	159
9.4.1 Robust Adaptation and Integral Feedback	162
9.4.2 Experiments Show That Exact Adaptation Is Robust, Whereas Steady-State Activity and Adaptation Times Are Fine-Tuned	164
9.5 Individuality and Robustness in Bacterial Chemotaxis	165
Further Reading	166
Exercises	166
Bibliography	172
CHAPTER 10 ■ Fold-Change Detection	175
10.1 Universal Features of Sensory Systems	175
10.2 Fold-Change Detection in Bacterial Chemotaxis	176
10.2.1 Definition of Fold-Change Detection (FCD)	177
10.2.2 The Chemotaxis Circuit Provides FCD by Means of a Nonlinear Integral-Feedback Loop	178
10.3 FCD and Exact Adaptation	180
10.4 The Incoherent Feedforward Loop Can Show FCD	180
10.5 A General Condition for FCD	182
10.6 Identifying FCD Circuits from Dynamic Measurements	183
10.7 FCD Provides Robustness to Input Noise and Allows Scale-Invariant Searches	184

Further Reading	186
Exercises	186
References	188
CHAPTER 11 ■ Dynamical Compensation and Mutant Resistance in Tissues	191
11.1 The Insulin-Glucose Feedback Loop	191
11.2 The Minimal Model Is Not Robust to Changes in Insulin Sensitivity	193
11.3 A Slow Feedback Loop on Beta-Cell Numbers Provides Compensation	194
11.4 Dynamical Compensation Allows the Circuit to Buffer Parameter Variations	197
11.5 Type 2 Diabetes Is Linked with Instability Due to a U-Shaped Death Curve	200
11.6 Tissue-Level Feedback Loops Are Fragile to Invasion by Mutants That Misread the Signal	201
11.7 Biphasic (U-Shaped) Response Curves Can Protect against Mutant Takeover	202
11.8 Summary	203
Further Reading	204
Exercises	204
Bibliography	207
CHAPTER 12 ■ Robust Spatial Patterning in Development	209
12.1 The French Flag Model Is Not Robust	210
12.2 Increased Robustness by Self-Enhanced Morphogen Degradation	212
12.3 Network Motifs That Provide Degradation Feedback for Robust Patterning	214
12.4 The Robustness Principle Can Distinguish between Mechanisms of Fruit Fly Patterning	215
Further Reading	220
Exercises	220
Bibliography	223
PART 3 Optimality	
CHAPTER 13 ■ Optimal Gene Circuit Design	227
13.1 Introduction	227
13.2 Optimal Expression Level of a Protein under Constant Conditions	228

13.2.1 Cost of the LacZ Protein	229
13.2.2 The Benefit of the LacZ Protein	230
13.2.3 Fitness Function and the Optimal Expression Level	231
13.2.4 Cells Reach Optimal LacZ Levels in a Few Hundred Generations in Laboratory Evolution Experiments	232
13.3 To Regulate or Not to Regulate? Optimal Regulation in Changing Environments	234
13.4 Environmental Selection of the Feedforward Loop Network Motif	236
13.5 Inverse Ecology	238
Further Reading	239
Exercises	239
Bibliography	247
CHAPTER 14 ■ Multi-Objective Optimality in Biology	249
14.1 Introduction	249
14.2 The Fitness Landscape Picture for a Single Task	249
14.3 Multiple Tasks Are Characterized by Performance Functions	250
14.4 Pareto Optimality in Performance Space	251
14.5 Pareto Optimality in Trait Space Leads to Simple Patterns	252
14.6 Two Tasks Lead to a Line Segment, Three Tasks to a Triangle, Four to a Tetrahedron	253
14.7 Trade-Offs in Morphology	254
14.8 Archetypes Can Last over Geological Timescales	256
14.9 Trade-Offs for Proteins	257
14.10 Trade-Offs in Gene Expression	258
14.11 Division of Labor in the Individual Cells That Make Up an Organ	259
14.12 Variation within a Species Lies on the Pareto Front	260
Further Reading	263
Exercises	263
Bibliography	271
CHAPTER 15 ■ Modularity	273
15.1 The Astounding Speed of Evolution	273
15.2 Modularity Is a Common Feature of Engineered and Evolved Systems	273
15.3 Modularity Is Found at All Levels of Biological Organization	274
15.4 Modularity Is Not Found in Simple Computer Simulations of Evolution	275

15.5 Simulated Evolution of Circuits Made of Logic Gates	275
15.6 Randomly Varying Goals Cause Confusion	278
15.7 Modularly Varying Goals Lead to Spontaneous Evolution of Modularity	278
15.8 The More Complex the Goal, the More MVG Speeds Up Evolution	280
15.9 Modular Goals and Biological Evolution	281
Further Reading	283
Exercises	283
Bibliography	284
APPENDIX A ■ The Input Functions of Genes: Michaelis–Menten and Hill Equations	287
A.1 Binding of a Repressor to a Promoter	287
A.2 Binding of an Inducer to a Repressor Protein: The Michaelis–Menten Equation	289
A.3 Cooperativity of Inducer Binding and the Hill Equation	291
A.4 The Monod–Changeux–Wyman Model	292
A.5 The Input Function of a Gene Regulated by a Repressor	293
A.6 Binding of an Activator to Its DNA Site	294
A.6.1 Comparison of Dynamics with Logic and Hill Input Functions	295
A.7 Michaelis–Menten Enzyme Kinetics	295
Further Reading	297
Exercises	297
Bibliography	298
APPENDIX B ■ Multi-Dimensional Input Functions	299
B.1 Input Function That Integrates an Activator and a Repressor	299
Exercise	301
Bibliography	301
APPENDIX C ■ Graph Properties of Transcription Networks	303
C.1 Transcription Networks Are Sparse	303
C.2 Transcription Networks Have Long-Tailed Out-Degree Sequences and Compact In-Degree Sequences	303
C.3 Clustering Coefficients	305
C.4 Quantitative Measure of Network Modularity	305
Bibliography	306

APPENDIX D ■ Noise in Gene Expression	307
D.1 Introduction	307
D.2 Extrinsic and Intrinsic Noise	307
D.3 Distribution of Protein Levels	308
D.4 Network Motifs Affect Noise	309
D.5 Position of Noisiest Step	310
Further Reading	311
Bibliography	311
WORDS OF THANKS	313
INDEX	315



Taylor & Francis

Taylor & Francis Group

<http://taylorandfrancis.com>

Introduction

When I first read a biology textbook, it was like reading a thriller. Every page brought a new shock. As a physicist, I was used to studying matter that obeys precise mathematical laws. But cells are matter that dances. Structures spontaneously assemble, perform elaborate biochemical functions and vanish effortlessly when their work is done. Molecules encode and process information virtually without errors, despite the fact that they are under strong thermal noise. How could this be? Are there special laws of nature that apply to biological systems that can help us understand why they are so different from nonliving matter?

We yearn for laws of nature and simplifying principles, but biology is astoundingly complex. Every biochemical interaction is exquisitely crafted, and cells contain networks of thousands of such interactions. These networks are the result of evolution, which works by making random changes and selecting the organisms that survive. Therefore, the structures found by evolution are, to some degree, dependent on historical chance and are laden with biochemical detail that requires special description in every case.

Despite this complexity, scientists have attempted to discern generalizable principles throughout the history of biology. The search for these principles is ongoing and far from complete. It is made possible by advances in experimental technology that provide detailed and comprehensive information about networks of biological interactions.

Such studies led to the discovery that one can, in fact, formulate general laws that apply to biological networks. Because it has evolved to perform functions, biological circuitry is far from random or haphazard. It has a defined style, the style of systems that must function. Although evolution works by random tinkering, it converges again and again onto a defined set of circuit elements that obey general design principles.

The goal of this book is to highlight some of the design principles of biological systems, and to provide a mathematical framework in which these principles can be used to understand biological networks. The main message is that biological systems contain an inherent simplicity. Although cells evolved to function and did not evolve to be comprehensible, simplifying principles make biological design understandable to us.

This book is written for readers who have had a basic course in mathematics. Specialist terms and gene names are avoided, although detailed descriptions of several well-studied biological systems are presented in order to demonstrate the principles. This book presents one path into systems biology based on mathematical principles, with less emphasis on experimental technology. Other directions can be found in the sources listed at the end of each chapter.

The mathematical descriptions in the book can be solved on the blackboard or on a small piece of paper. We will see that it can be very useful to ask, “Why is the system designed in such a way?” and to try to answer with simplified models.

We conclude this introduction with an overview of the chapters. The first part of the book deals with transcription regulation networks. Elements of networks and their dynamics are described. We will see that these networks are made of repeating occurrences of simple patterns called network motifs. Each network motif performs a defined information processing function within the network. These building block circuits were rediscovered by evolution again and again in different systems. Network motifs in other biological networks, including signal transduction and neuronal networks, are also discussed. The main point is that biological systems show an inherent simplicity, by employing and combining a rather small set of basic building-block circuits, for specific computational tasks.

The second part of the book focuses on the principle of robustness: biological circuits are designed so that their essential function is insensitive to the naturally occurring fluctuations in the components of the circuit. Whereas many circuit designs can perform a given function on paper, we will see that very few can work robustly in the cell. These few robust circuit designs are nongeneric and particular, and are often aesthetically pleasing. We will use the robustness principle to understand the detailed design of well-studied systems, including bacterial chemotaxis and patterning in fruit fly development.

The final chapters describe how evolutionary optimization can be used to understand the optimal circuit design.

These features of biological systems – reuse of a small set of network motifs, robustness to component tolerances and constrained optimal design – are also found in a completely different context: systems designed by human engineers. Biological systems have additional features in common with engineered systems, such as modularity and hierarchical design. These similarities hint at a deeper theory that can unify our understanding of evolved and designed systems.

This is it for the introduction. Some of the solved exercises after each chapter provide more detail on topics not discussed in the main text. I wish you enjoyable reading.

INTRODUCTION TO THE SECOND EDITION

Thirteen years have passed since the first edition. In these 13 years, three daughters were born, Gefen, Tamar and Carmel. With their growth (Gefen is 11, Tamar is 9 and Carmel is 5), space was created to write this second edition.

In these 13 years, systems biology has become a field with conferences, journals, departments and institutes. Thousands of physicists, computer scientists and engineers have joined biology, and many biologists have learned theoretical approaches. Systems biology is now a part of research in all fields of biology.

It is a thrilling experience to survey the advances made in these 13 years, especially the conceptual advances and new design principles that have emerged. These advances account for new chapters in this book, devoted to paradoxical components, fold-change detection, dynamical compensation in tissues, multi-objective optimality in evolution and modularity.

In parallel, the principles that appeared in the first edition have been refined and applied to many more systems. Thus, every old chapter is extensively revised with new examples and mathematical techniques.

Additional improvements stem from the feedback of instructors and students who used the first edition of this book in courses in biology, physics, engineering and computer science. This feedback accounts for new chapters on biological oscillators and bistability and memory. It also helped me to introduce important methods such as nullclines and phase plane analysis.

As I wrote the second edition, I used it to teach a 2018 course, complete with guitar songs, which is video-documented online. The participants helped me explain things more clearly and take more deep sighs of relief.

The book did not grow much in size because I removed, trimmed and edited the chapters to keep things readable, to remove errors and to minimize topics which have proved to be less important than I originally thought. Throughout, I selected studies that help clarify the principles, especially studies that were pioneering.

I wrote at the kitchen table after the triple bedtime, and in flights and hotels while traveling. Keeping a steady pace of about an hour a day helped me finish in about a year, an overall joyous experience. I was encouraged by Michael Elowitz, by my life-friend Galia Moran and by my publisher. After years of working together in research, I discovered Avi Mayo as a devoted and creative partner for making the figures, which are all new. I'm grateful for being surrounded by family, friends, students and colleagues whom I love.

FURTHER READING

(Bialek, 2012) "Biophysics: searching for principles."

(Fall et al., 2004) "Computational cell biology."

(Fell, 1997) "Understanding the control of metabolism."

(Heinrich and Schuster, 1996) "The regulation of cellular systems."

(Klipp et al., 2014) "Systems biology in practice: concepts, implementation and application."

(Kriete and Eils, 2006) "Introducing computational systems biology."

(Palsson, 2006) "Systems biology: properties of reconstructed networks."

(Phillips et al., 2012) "Physical biology of the cell."

(Savageau, 1976) "Biochemical systems analysis: a study of function and design in molecular biology."

Videos of 2018 course and additional material: <http://www.weizmann.ac.il/mcb/UriAlon/download/systems-biology-course-2018>

BIBLIOGRAPHY

Bialek, W. 2012. 'Biophysics: searching for principles'. Princeton University Press.

Fall, C.P. et al. (eds) 2004. 'Computational cell biology'. New York, NY: Springer New York (Interdisciplinary Applied Mathematics). doi: 10.1007/b97701.

Fell, D.A. 1997. 'Understanding the control of metabolism'. Portland Press.

Heinrich, R. and Schuster, S. 1996. 'The regulation of cellular systems'. Springer US.

Klipp, E. et al. 2014. 'Systems biology in practice: concepts, implementation and application'. Wiley VCH; 2nd edition.

Kriete, A. and Eils, R. 2006. 'Computational systems biology', Academic Press.

Palsson, B. 2006. 'Systems biology: properties of reconstructed networks'. Cambridge University Press.

Phillips, R. et al. 2012. 'Physical biology of the cell'. Garland Science, 2nd edition.

Savageau, M.A. 1976. 'Biochemical systems analysis: a study of function and design in molecular biology'. Addison-Wesley Pub. Co.

Part 1

Network Motifs



Taylor & Francis

Taylor & Francis Group

<http://taylorandfrancis.com>

Transcription Networks

Basic Concepts

1.1 INTRODUCTION

The cell is an integrated device made of several thousand types of proteins. Each protein is a nanometer-size molecular machine that carries out a specific task with exquisite precision. For example, the micron-long bacterium *Escherichia coli* is a cell that contains a few million proteins, of about 4500 different types (typical numbers, lengths and timescales can be found in [Table 1.1](#)).

Cells encounter different situations that require different proteins. For example, when cells sense sugar, they begin to produce proteins that transport the sugar into the cell and utilize it. When damaged, the cell produces repair proteins. The cell therefore continuously monitors its environment and calculates the amount at which each type of protein is needed. This information-processing function, which determines the rate of production of each protein, is largely carried out by transcription networks.

The first few chapters in this book will discuss transcription networks. This opening chapter defines the elements of transcription networks and their dynamics.

1.2 THE COGNITIVE PROBLEM OF THE CELL

Cells live in a complex environment and can sense many different signals, including physical parameters such as temperature and osmotic pressure, biological signaling molecules from other cells, beneficial nutrients and harmful chemicals. Information about the internal state of the cell, such as the level of key metabolites and internal damage (damage to DNA, membrane or proteins), is also important. Cells respond to these signals by producing appropriate proteins that act upon the internal or external environment.

To represent these environmental states, the cell uses special proteins called **transcription factors** as symbols. Transcription factors are designed to transit rapidly between active and inactive molecular states, at a rate that is modulated by a specific environmental signal. Each

TABLE 1.1 Typical Biological Parameter Values (Biology by the Numbers, 2016)

Property	<i>E. coli</i>	Yeast	Human (Fibroblast)
Cell volume	1 μm^3	1000 μm^3	10,000 μm^3
Proteins/cell	$\sim 4 \times 10^6$	$\sim 4 \times 10^9$	$\sim 4 \times 10^{10}$
Mean size of protein	4–5 nm		
Size of genome	4.6 $\times 10^6$ bp 4500 genes	1.2 $\times 10^7$ bp 6600 genes	3.2 $\times 10^9$ bp 21,000 genes
Regulator binding site length	10–20 bp	5–10 bp	5–10 bp
Promoter length	~ 100 bp	~ 1000 bp	$\sim 10^4$ – 10^5 bp
Gene length	~ 1000 bp	~ 1000 bp	10^4 – 10^6 bp (with introns)
Concentration of 1 protein/cell	~ 1 nM	~ 1 pM	~ 0.1 pM
Diffusion time of protein across cell	~ 0.1 sec ($D = 10 \mu\text{m}^2/\text{sec}$)	~ 0.3 sec	~ 10 sec
Diffusion time of small molecule across cell	~ 1 msec ($D = 100 \mu\text{m}^2/\text{sec}$)	~ 3 msec	~ 0.1 sec
Time to transcribe a gene	< 1 min (80 bp/sec)	~ 1 min	~ 30 min (including RNA processing)
Time to translate a protein	< 1 min (20 aa/sec)	~ 1 min	~ 30 min (including mRNA export)
Typical mRNA lifetime	3 min	30 min	10 h
Typical protein lifetime	1 h	0.3–3 h	10–100 h
Cell generation time	20 min–several hours	2h–several hours	20 h–nondividing
Ribosomes/cell	10^4	10^7	10^8
Mutation rate	10^{-9} – 10^{-10} /bp/replication	10^{-9} – 10^{-10} /bp/replication	$\sim 10^{-9}$ /bp/replication

active transcription factor can bind the DNA to regulate the rate at which specific target genes are read (Figure 1.1). The genes are read (transcribed) into mRNA, which is then translated into protein, which can act on the environment. The activities of the transcription factors in a cell therefore can be considered an internal representation of the environment. For example, *E. coli* has an internal representation with about 300 degrees of freedom (transcription factors). These regulate the rates of production of *E. coli*’s 4500 proteins.

The internal representation by a set of transcription factors is a compact description of the myriad factors in the environment. Many different situations are summarized by a particular transcription factor activity that signifies “I am starving.” Many other situations are summarized by a different transcription factor activity that signifies “My DNA is damaged.” These transcription factors regulate their target genes to mobilize the appropriate protein responses in each case.

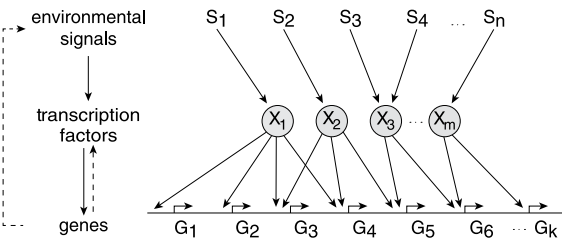


FIGURE 1.1

1.3 ELEMENTS OF TRANSCRIPTION NETWORKS

The interaction between transcription factors and genes is described by **transcription networks**. Let us begin by briefly describing the elements of the network: genes and transcription factors. Each gene is a stretch of DNA whose sequence encodes the information needed for production of a protein. The protein is produced in two steps, transcription and translation. First, the gene is copied into a disposable mRNA molecule by a protein machine called RNA polymerase (RNAP), a process called **transcription**. The mRNA is then translated into a protein (Figure 1.2).

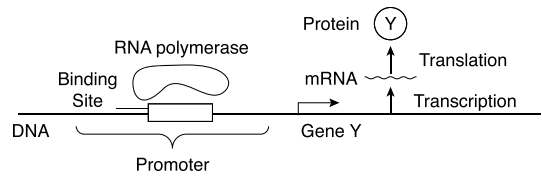


FIGURE 1.2

The rate at which the gene is transcribed, the number of mRNA produced per unit time, is controlled by a regulatory region of DNA that precedes the gene, called the **promoter** (Figure 1.2). RNAP binds a defined site (a specific DNA sequence) in the promoter (Figure 1.2). The precise DNA sequence of this site determines the chemical affinity of RNAP to the promoter, and specifies the transcription rate of the gene.

Whereas RNAP acts on all of the genes, changes in the expression of specific genes are due to transcription factors. Each transcription factor modulates the transcription rate of a set of target genes. Transcription factors affect the transcription rate by binding specific sites in the promoters of the regulated genes (Figures 1.3 and 1.4). When bound, they change the probability per unit time that RNAP binds the promoter and produces an mRNA molecule. The transcription factors thus affect the rate at which RNAP initiates transcription of the gene. Transcription factors can act as **activators** that increase the transcription rate of a gene (Figure 1.3), or as **repressors** that reduce the transcription rate (Figure 1.4).

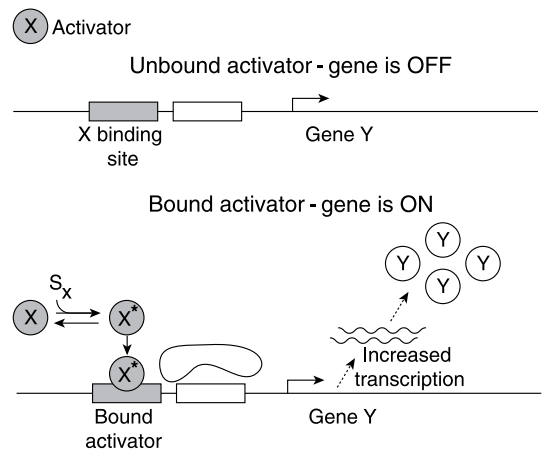


FIGURE 1.3

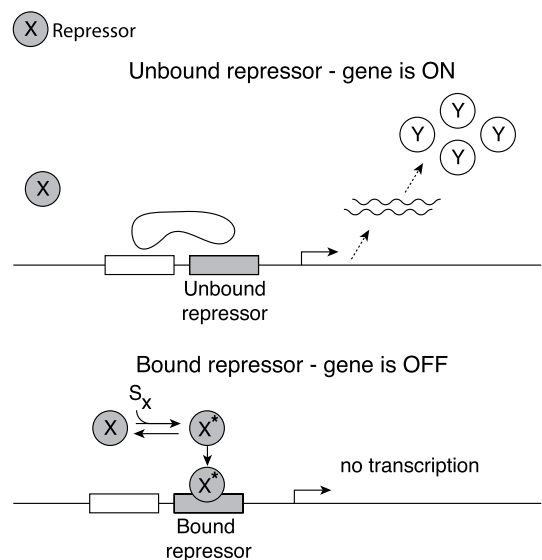


FIGURE 1.4

Transcription factor proteins are themselves encoded by genes, which are regulated by other transcription factors, which in turn may be regulated by yet other transcription factors, and so on. This set of interactions forms a transcription network (Figure 1.5). The transcription network describes all of the regulatory transcription interactions in a cell. In the network, the nodes are genes and arrows represent transcriptional regulation of one gene by the protein product of another gene. An arrow $X \rightarrow Y$ means that the product of gene X is a transcription factor protein that can bind the promoter of gene Y to control the rate at which gene Y is transcribed.

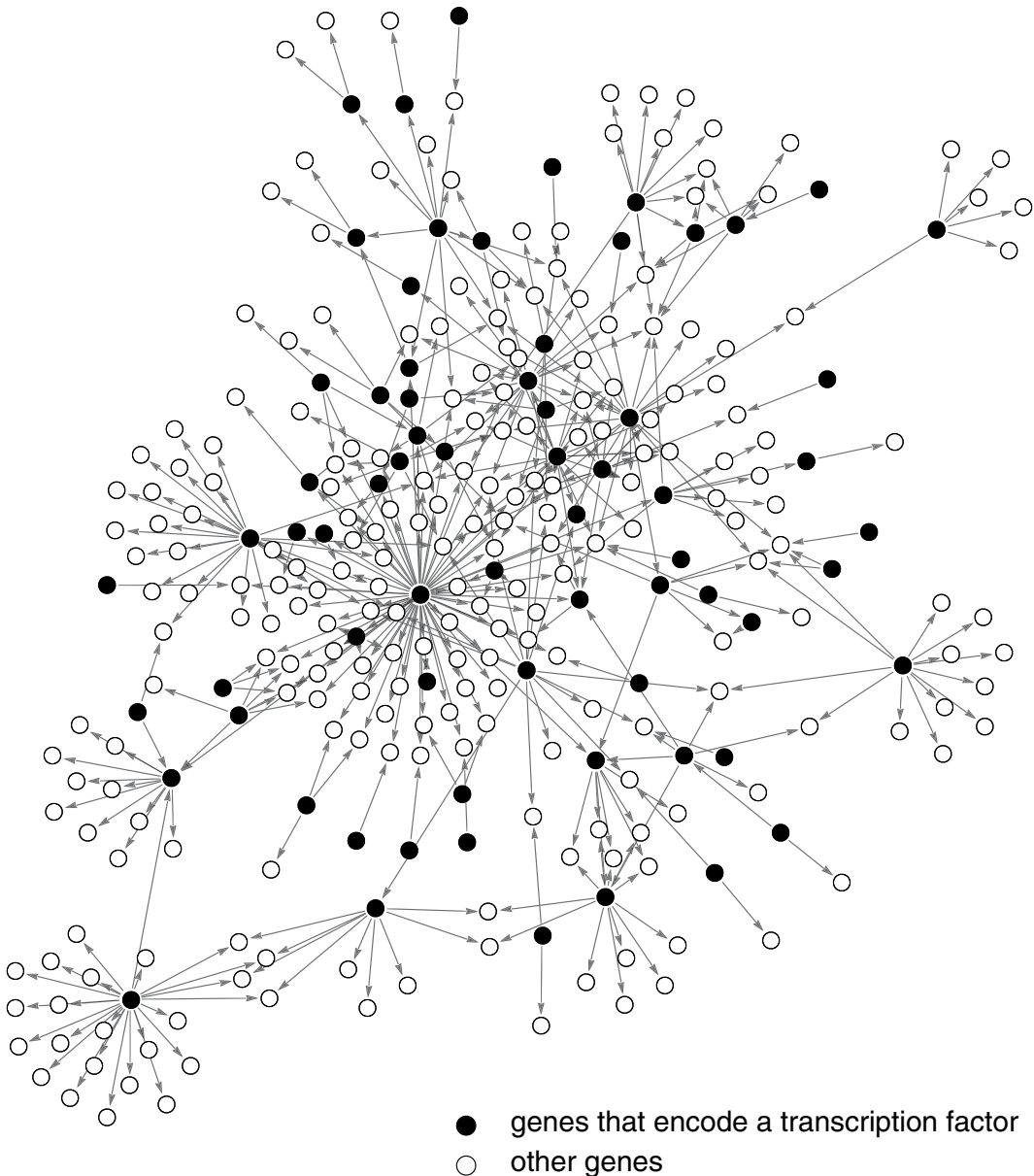


FIGURE 1.5

The inputs to the network are signals that carry information from the environment. Each signal is a small molecule, protein modification or molecular partner that directly affects the activity of one of the transcription factors. Often, external stimuli activate biochemical signal-transduction pathways that culminate in a chemical modification of specific transcription factors. In other systems, the signal can be as simple as a sugar molecule or hormone that enters the cells and directly binds the transcription factor. The signals usually cause a physical change in the shape of the transcription factor protein, causing it to assume an active molecular state. Thus, signal S_x can cause X to rapidly shift to its active state X^* , bind the promoter of gene Y and change the rate of transcription, leading to increased or decreased production of protein Y (Figures 1.3 and 1.4).

The network thus represents a dynamical system: after an input signal arrives, transcription factor activities change, leading to changes in the production rate of proteins. Some of these proteins are transcription factors that activate additional genes, and so on. The majority of the proteins are not transcription factors, but instead carry out the diverse functions of the cell, such as building structures and catalyzing reactions.

1.3.1 Separation of Timescales

Transcription networks are designed with a strong **separation of timescales**: the input signals usually change transcription factor activities on a sub-second timescale. Binding of the active transcription factor to its DNA sites often reaches equilibrium in seconds. Transcription and translation of the target gene takes minutes, and the accumulation of the protein product can take hours. Thus, the different steps between the input signal and the accumulation of the output protein have very different timescales. Table 1.2 gives typical timescales for *E. coli*.

Thus, transcription factor activity levels can be considered to be at steady state within the equations that describe network dynamics on the slow timescale of changes in protein levels.

In addition to transcription networks, the cell contains several other networks of interactions, such as signal-transduction networks made of interacting proteins, which will be discussed in later chapters. These networks typically operate much faster than transcription networks, and thus they can be considered to be approximately at steady state on the slow timescales of transcription networks.

Transcription factors regulate genes by a rich variety of mechanisms. Here, biology shows its full complexity. Transcription factors display ingenious ways to bind DNA at strategically placed sites. When bound, they block or recruit each other and RNAP (and, in human cells, many other accessory proteins) to control the rate at which mRNA is produced. However,

TABLE 1.2 Typical Timescales for the Reactions in the Transcription Network of *E. coli*

Binding of a small molecule signal to a transcription factor, causing a change in its activity	~1 msec
Binding of an active transcription factor to its DNA site	~1 sec
Transcription and translation of the gene	~minutes
Timescale for 50% change in concentration of the translated protein (stable proteins)	~1 hour (1 cell generation)

on the level of transcription network dynamics, and on the slow timescales in which they operate, we will see that one can usually treat all of these mechanisms within a unifying and rather simple mathematical description.

An additional remarkable property of transcription networks is the **modularity** of their components (Chapter 15). You can take the DNA of a gene from one organism and express it in a different organism. For example, you can take a piece of DNA with the gene for green fluorescent protein (GFP) from the genome of a jellyfish and introduce this gene into bacteria. As a result, the bacteria produce GFP, causing the bacteria to turn green. Regulation can also be added by adding a promoter region. For example, control of the GFP gene in the bacterium can be achieved by pasting in front of the gene a DNA fragment from the promoter of a different bacterial gene, say, one that is controlled by a sugar-inducible transcription factor. This causes *E. coli* to express GFP and turn green only in the presence of the sugar. Promoters and genes are generally interchangeable. This fact underlies the use of GFP as an experimental tool, employed in the coming chapters to illustrate the dynamics of gene expression.

Modular components make transcription networks very evolvable, because they can readily incorporate new genes and new regulation. In fact, transcription networks can evolve quite rapidly (here “rapidly” is on the scale of many generations): the arrows in transcription networks evolve on a much faster timescale than the coding regions of the genes. For example, mice and humans have very similar genes, but the transcription regulation of these genes, which governs when and how much of each protein is made, is different. In other words, many of the differences between animal species lie in the differences in the arrows of the transcription networks, rather than differences in their genes.

1.3.2 The Signs on the Arrows: Activators and Repressors

As we just saw, each arrow in a transcription network corresponds to an interaction in which a transcription factor directly controls the transcription rate of a gene. These interactions can be of two types. Activation occurs when the transcription factor increases the rate of transcription when it binds the promoter (Figure 1.3). Repression occurs when the transcription factor reduces the rate of transcription when it binds the promoter (Figure 1.4). Thus, each arrow in the network has a sign: + for activation, – for repression. Plus arrows are denoted by a regular arrow $X \rightarrow Y$, whereas minus arrows are denoted by a blunt-headed arrow $X \dashrightarrow Y$. Transcription networks often show comparable numbers of plus and minus arrows, with more positive (activation) interactions than negative interactions (e.g., about 60% activation in *E. coli* and 80% in yeast).

Each transcription factor acts primarily in one mode for its target genes, as either an activator or a repressor. In contrast, the input modes of regulation are often mixed: a typical gene is activated by some transcription factors and repressed by others. Thus, the signs on outgoing arrows (arrows that point out from a given node) are highly correlated, but the signs on incoming arrows (arrows that point into a given node) are not.¹

¹ A similar feature is found in neuronal networks, where $X \rightarrow Y$ describes synaptic connections between neuron *X* and neuron *Y*. In many cases, the signs (activation or inhibition) are more highly correlated on the outgoing synapses than the signs of incoming synapses. This feature, known as Dale’s rule, stems from the fact that each neuron primarily uses one type of neurotransmitter, which can be either excitatory or inhibitory.

1.3.3 The Numbers on the Arrows: Input Functions

The arrows not only have signs, but also carry numbers that correspond to the strength of the interaction. The strength of the effect of a transcription factor on a target gene is described by an **input function**. Consider the production rate of protein Y controlled by a transcription factor X . When X regulates Y , represented in the network by $X \rightarrow Y$, the number of molecules of protein Y produced per unit time is a function of the concentration of X in its active form, X^* :

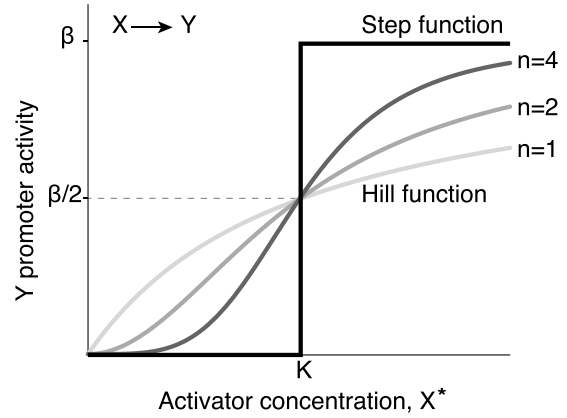


FIGURE 1.6

$$\text{rate of production of } Y = f(X^*) \quad (1.3.1)$$

Typically, the input function $f(X^*)$ is a monotonic function. It is an increasing function when X is an activator (Figure 1.6) and a decreasing function when X is a repressor. A useful function that realistically describes many gene input functions is called the **Hill function**. The Hill function can be derived from considering the equilibrium binding of the transcription factor to its site on the promoter (see Appendix A for further details). The Hill input function for an activator is a curve that rises from zero and approaches a maximal saturated level (Figure 1.6):

$$f(X^*) = \beta \frac{X^{*n}}{K^n + X^{*n}}, \quad \text{Hill function for an activator} \quad (1.3.2)$$

The Hill function has three parameters, K , β and n . The first parameter, K , is termed the **activation coefficient**, and has units of concentration. It defines the concentration of active X needed to significantly activate expression. From the equation it is easy to see that half-maximal expression is reached when $X^* = K$ (Figure 1.6). The value of K is determined by the chemical affinity between X and its binding site on the promoter, as well as additional factors. The second parameter in the input function is the **maximal promoter activity**, β . Maximal activity is reached at high activator concentrations, $X^* \gg K$, because at high concentrations, X^* binds the promoter with high probability and stimulates RNAP to produce many mRNAs per unit time. Finally, the **Hill coefficient** n determines the steepness of the input function. The larger n is, the more step-like the input function (Figure 1.6). Typically, input functions are moderately steep, with $n = 1-4$.

As do many functions in biology, the Hill function approaches a limiting value at high levels of X^* , rather than increasing indefinitely. This saturation of the Hill function at high X^* concentration is fundamentally due to the fact that the probability that the activator binds the promoter cannot exceed 1, no matter how high the concentration of X^* . The Hill equation often describes empirical data with good precision.

For a repressor, $X \rightarrow Y$, the Hill input function is a decreasing curve (Figure 1.7), whose shape depends on three similar parameters:

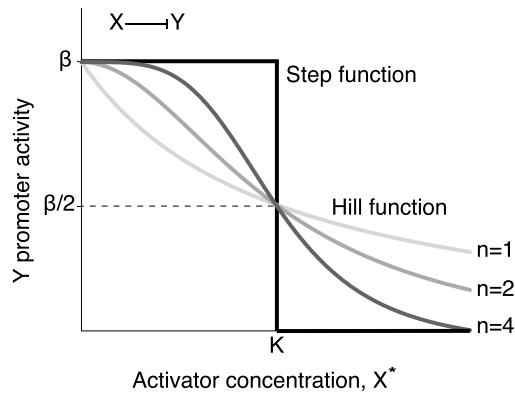


FIGURE 1.7

$$f(X^*) = \beta \frac{K^n}{K^n + X^{*n}} \quad \text{Hill function for a repressor} \quad (1.3.3)$$

Since a repressor allows strong transcription of a gene only when it is not bound to the promoter, this function can be derived by considering the probability that the promoter is unbound by X^* (see Appendix A). The maximal promoter activity β is obtained when the repressor does not bind the promoter at all (Figure 1.4), that is, when $X^* = 0$. Half-maximal repression is reached when the repressor activity is equal to K , the gene's repression coefficient. The Hill coefficient n determines the steepness of the input function (Figure 1.7).

Thus, each arrow in the network can be thought to carry at least three numbers, β , K and n . These numbers can readily be tuned during evolution. For example, K can be changed by mutations that alter the DNA sequence of the binding site of X in the promoter of gene Y . Even a change of a single DNA letter in the binding site can strengthen or weaken the chemical bonds between X and the DNA and change K . The parameter K can also be varied if the position of the binding site is changed, as well as by some changes in sequence outside of the binding site. Similarly, the maximal activity β can be tuned by mutations in the RNAP binding site or many other factors.

Laboratory evolution experiments show that when placed in a new environment, bacteria can accurately tune these numbers within several hundred generations to reach optimal expression levels (Chapter 13). Thus, these numbers are under selection pressure and can heritably change over many generations if environments change.

The input functions above go from a transcription rate of zero to a maximal transcription rate β . Many genes have a nonzero minimal expression level, called the gene's **basal expression level**. A basal level can be described by adding to the input function a term β_0 .

1.3.4 Logic Input Functions: A Simple Framework for Understanding Network Dynamics

Hill input functions are useful for detailed models. For mathematical clarity, however, it is sometimes useful to use even simpler functions that capture the essential behavior of these

input functions. The essence of input functions is a transition between low and high values, with a characteristic threshold K . In the coming chapters, we will sometimes approximate input functions using a **logic approximation** (Glass and Kauffman, 1973; Thieffry and Thomas, 1998). In this approximation, the gene is either OFF, $f(X^*) = 0$, or maximally ON, $f(X^*) = \beta$. The threshold for activation is K . Thus, logic input functions are step-like approximations for the smoother Hill functions. For activators, the logic input function can be described using a step-function θ that makes a step when X^* exceeds the threshold K :

$$f(X^*) = \beta\theta(X^* > K) \quad \text{logic approximation for activator} \quad (1.3.4)$$

where θ is equal to 0 or 1 according to the logic statement in the parentheses. The logic approximation is equivalent to a very steep Hill function with Hill coefficient $n \rightarrow \infty$ (Figure 1.6).

Similarly, for repressors, a decreasing step function is appropriate (Figure 1.7):

$$f(X^*) = \beta\theta(X^* < K) \quad \text{logic approximation for repressor} \quad (1.3.5)$$

We will see in the next chapters that by using a logic input function, dynamic equations become easy to solve.

1.3.5 Multi-Dimensional Input Functions Govern Genes with Several Inputs

We just saw how Hill functions and logic functions can describe input from a single transcription factor. Many genes, however, are regulated by multiple transcription factors. They are nodes in the network with two or more incoming arrows. Their promoter activity is thus a multi-dimensional input function of the different input transcription factors. Appendix B describes how input functions can be modeled by equilibrium binding of multiple transcription factors to the promoter.

Often, multi-dimensional input functions can be usefully approximated by logic functions, just as in the case of single-input functions. For example, consider genes regulated by two activators. Many genes require that *both* activator proteins bind to the promoter in order to show high expression. This is similar to an AND gate:

$$f(X^*, Y^*) = \beta\theta(X^* > K_x)\theta(Y^* > K_y) \sim X^* \text{ AND } Y^* \quad (1.3.6)$$

For other genes, binding of *either* activator is sufficient. This resembles an OR gate:

$$f(X^*, Y^*) = \beta\theta(X^* > K_x \text{ OR } Y^* > K_y) \sim X^* \text{ OR } Y^* \quad (1.3.7)$$

Not all genes have Boolean-like input functions. For example, some genes display a SUM input function, in which the inputs are additive (Kalir and Alon, 2004):

$$f(X^*, Y^*) = \beta_x X^* + \beta_y Y^* \quad (1.3.8)$$

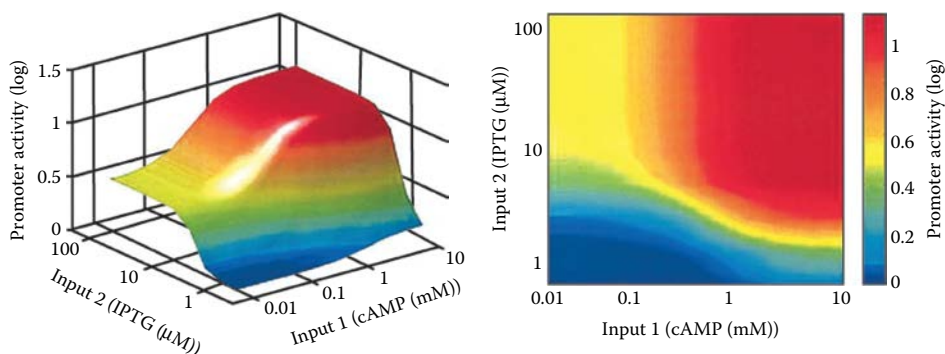


FIGURE 1.8 Adapted from (Setty et al., 2003).

Other functions are also possible. For example, a function with several plateaus and thresholds describes the *lac* system of *E. coli* (Figure 1.8). Genes in multi-cellular organisms can display input functions that calculate elaborate functions of a dozen or more inputs (Yuh, Bolouri and Davidson, 1998; Davidson et al., 2002; Beer and Tavazoie, 2004).

The functional form of input functions can be readily changed by means of mutations in the promoter of the regulated gene. For example, the *lac* input function of Figure 1.8 can be changed to resemble pure AND or OR gates with a few mutations in the *lac* promoter (Setty et al., 2003; Mayo et al., 2006; Figure 1.9). It appears that the precise form of the input function of each gene is under selection pressure during evolution.

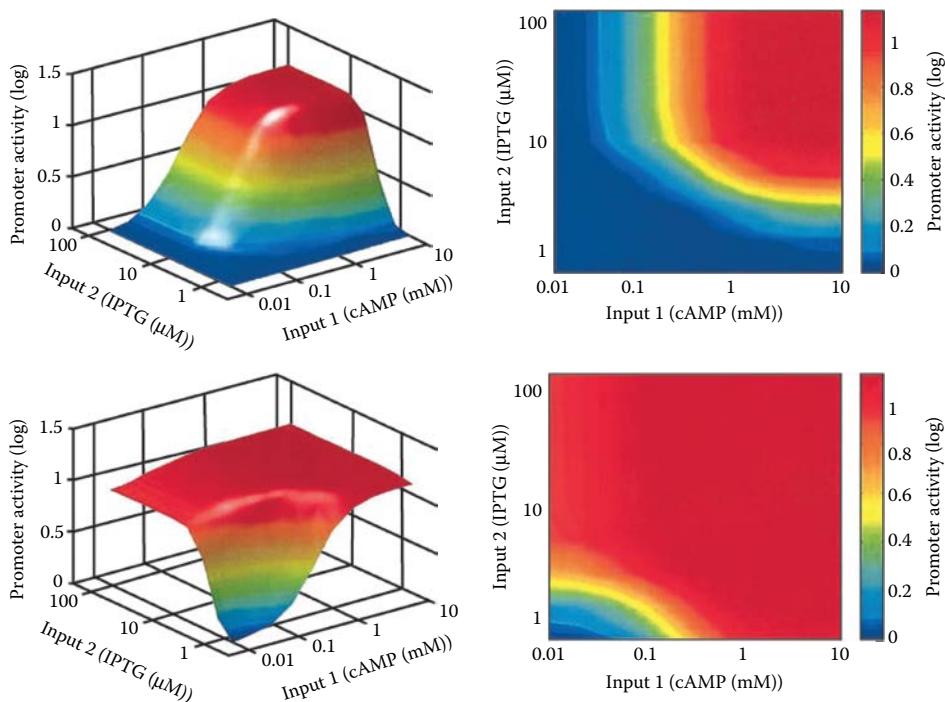


FIGURE 1.9 Adapted from (Setty et al., 2003).

1.4 DYNAMICS AND RESPONSE TIME OF SIMPLE REGULATION

We now turn to the dynamics of transcription networks. We begin with the dynamics of a single arrow in the network. Consider a gene that is regulated by a transcription factor with no additional inputs (or with all other inputs and post-transcriptional modes of regulation held constant over time²). This transcription interaction is described in the network by $X \rightarrow Y$ which reads “transcription factor X regulates gene Y .” Once X becomes activated by a signal, Y concentration begins to change. Let us calculate the dynamics of the concentration of the gene product, the protein Y and its response time.

In the absence of its input signal, transcription factor X is inactive and Y is not produced (Figure 1.3). When the signal S_x appears, X rapidly transits to its active form X^* and binds the promoter of gene Y . Gene Y begins to be transcribed, and the mRNA is translated, resulting in accumulation of protein Y . The cell produces protein Y at a rate β (units of concentration per unit time).

The production of Y is balanced by two processes, protein degradation (its specific destruction by specialized proteins in the cell) and dilution (the reduction in concentration due to the increase of cell volume during growth). The degradation rate is α_{deg} , and the dilution rate is α_{dil} , giving a total removal rate (in units of 1/time) of

$$\alpha = \alpha_{dil} + \alpha_{deg} \quad (1.4.1)$$

The change in the concentration of Y is due to the difference between its production and removal, as described by a dynamic equation³:

$$dY/dt = \beta - \alpha Y \quad (1.4.2)$$

The removal term in the equation αY is equal to the concentration Y times the probability per unit time that each protein Y is removed, α .

At steady state, Y reaches a constant concentration Y_{st} . The steady-state concentration can be found by solving for $dY/dt = 0$. This shows that the steady-state concentration is the ratio of the production and removal rates:

$$Y_{st} = \beta/\alpha \quad (1.4.3)$$

This makes sense: the higher the production rate β , the higher the protein concentration reached, Y_{st} . The higher the removal rate α , the lower is Y_{st} .

² Proteins are potentially regulated at every step of their synthesis process, including the following post-transcriptional regulation interactions: (1) rate of degradation of the mRNA, (2) rate of translation, controlled primarily by sequences in the mRNA that bind the ribosomes and by mRNA-binding regulatory proteins and regulatory RNA molecules and (3) rate of active and specific protein degradation. In eukaryotes, regulation also occurs on the level of mRNA splicing and transport in the cell. Other modes of regulation are possible.

³ The time for transcription and translation of the protein (minutes) is neglected because it is small compared to the response time of the protein-level dynamics (tens of minutes) (Table 1.2).

What happens if we now take away the input signal, so that production of Y stops ($\beta = 0$)? The solution of Equation 1.4.2 with $\beta = 0$ is an exponential decay of Y concentration (Figure 1.10):

$$Y(t) = Y_{st} e^{-\alpha t} \quad (1.4.4)$$

How fast does Y decay? An important measure for the speed at which Y levels change is the **response time**. The response time, $T_{1/2}$, is defined as the time to reach halfway between the initial and final levels

in a dynamic process. For the decay process of Equation 1.4.4, the response time is the time to reach halfway down from the initial level, Y_{st} , to the final level, $Y = 0$. The response time is therefore given by solving for the time when $Y(t) = Y_{st}/2$, which, using Equation 1.4.4, shows an inverse dependence on the removal rate:

$$T_{1/2} = \log(2)/\alpha \quad (1.4.5)$$

Note that the removal rate α directly determines the response time: fast removal allows rapid changes in concentration. The production rate β affects the steady-state level but not the response time.

Some proteins show rapid degradation rates (large α). At steady state, this leads to a seemingly futile cycle of protein production and destruction. To maintain a given steady state, $Y_{st} = \beta/\alpha$, such proteins require high production rate β to balance the high degradation rate α . The benefit of such a futile cycle is fast response times once a change is needed.

We have seen that loss of input signal leads to an exponential decay of Y . Let us now consider the opposite case, in which an unstimulated cell with $Y = 0$ is provided with a signal, so that protein Y begins to accumulate. If an unstimulated gene becomes suddenly stimulated by a strong signal S_x , the dynamic equation, Equation 1.4.2, results in an approach to steady state (Figure 1.11)

$$Y(t) = Y_{st} (1 - e^{-\alpha t}) \quad (1.4.6)$$

The concentration of Y rises from zero and gradually converges on the steady state $Y_{st} = \beta/\alpha$. Note that at early times, when $\alpha t \ll 1$, we can use a Taylor expansion⁴ to find a linear accumulation of Y with a slope equal to the production rate β (the tangent dashed line in Figure 1.11):

$$Y = \beta t \quad (1.4.7)$$

at early times, when $\alpha t \ll 1$. Later, as Y levels increase, the degradation term $-\alpha Y$ begins to be important and Y accumulation slows down as it converges to its steady-state level.

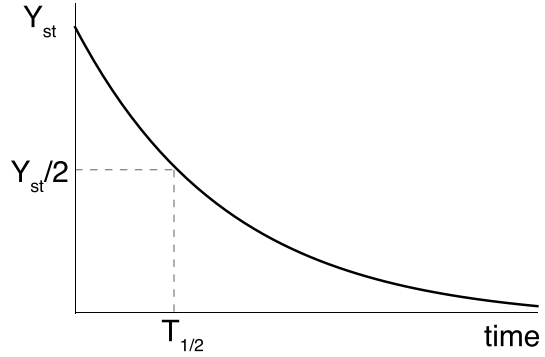


FIGURE 1.10

⁴ Using $e^{-\alpha t} \sim 1 - \alpha t$, and $Y_{st} = \beta/\alpha$.

The response time, the time to reach $Y_{st}/2$, can be found by solving for the time when $Y(t) = Y_{st}/2$. Using Equation 1.4.6, we find the same response time as in the case of decay:

$$T_{1/2} = \log(2)/\alpha \quad (1.4.8)$$

The response time for both increase and decrease in protein levels is the same and is governed only by the removal rate α . The larger the removal rate α , the more rapid the response time.

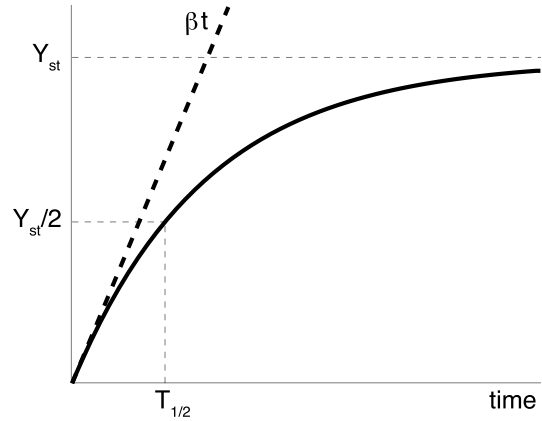


FIGURE 1.11

1.4.1 The Response Time of Stable Proteins Is One Cell Generation

Many proteins are not actively degraded in growing cells ($\alpha_{deg} = 0$). These are termed stable proteins. The production of stable proteins is balanced by dilution due to the increasing volume of the growing cell, $\alpha = \alpha_{dil}$. For such stable proteins, the *response time is equal to one cell generation time*. To see this, imagine that a cell produces a protein, and then suddenly production stops ($\beta = 0$). The cell grows and, when it doubles its volume, splits into two cells. Thus, after one cell generation time τ , the protein concentration has decreased by 50%, and therefore:

$$T_{1/2} = \log(2)/\alpha_{dil} = \tau \quad \text{response time is one cell generation} \quad (1.4.9)$$

This is an interesting result. Bacterial cell generation times are on the order of 30 min to a few hours, and animal and plant cell generation times are typically a day or longer. One would expect that transcription networks that are made to react to signals such as nutrients and stresses should respond much more rapidly than the cell generation time, otherwise only the cell's daughters can benefit. But for stable proteins, the response time, as we saw, is one cell generation time. Thus, *response time can be a limiting factor that poses a constraint for designing efficient gene circuits*.

In the next chapter, we will discuss simple transcriptional circuits that can help speed the response time.

FURTHER READING

Dynamics of Gene Networks

(Monod, Pappenheimer and Cohen-Bazire, 1952) "The kinetics of the biosynthesis of beta-galactosidase in *Escherichia coli* as a function of growth."

(Rosenfeld and Alon, 2003) "Response delays and the structure of transcription networks."

Molecular Mechanisms of Transcriptional Regulation

(Ptashne, 1986) "A genetic switch."

(Ptashne and Gann, 2002) "Genes and signals."

Overview of Transcription Networks

(Alon, 2003) “Biological networks: the tinkerer as an engineer.”

(Levine and Davidson, 2005) “Gene regulatory networks for development.”

(Thieffry et al., 1998) “From specific gene regulation to genomic networks: a global analysis of transcriptional regulation in *Escherichia coli*.”

EXERCISES

1.1 *A change in production rate:* A gene Y with simple regulation is produced at a constant rate β_1 . The production rate suddenly shifts to a different rate β_2 .

- Calculate and plot the gene product concentration $Y(t)$.
- What is the response time (time to reach halfway between the steady states)?

Solution (for part a):

- Let us mark the time when the shift occurs as $t = 0$. Before the shift, Y reaches steady state at a level $Y(t = 0) = Y_{st} = \beta_1/\alpha$. After the shift,

$$dY/dt = \beta_2 - \alpha Y \quad (\text{P1.1})$$

The solution of this linear differential equation is generally $Y = C_1 + C_2 e^{-\alpha t}$, where the constants C_1 and C_2 need to be determined so that $Y(t = 0) = \beta_1/\alpha$, and Y at long times reaches its new steady state, β_2/α . This yields the following sum of an exponential and a constant (Figure 1.12):

$$Y(t) = \beta_2/\alpha + (\beta_1/\alpha - \beta_2/\alpha)e^{-\alpha t} \quad (\text{P1.2})$$

Take the derivative with respect to time, dY/dt , and verify that Equation P1.1 is fulfilled.

1.2 *mRNA dynamics:* In the main text, we considered the activation of transcription of a gene (mRNA production) and used a dynamical equation to describe the changes in the concentration of the gene product, the protein Y . In this equation, $dY/dt = \beta - \alpha Y$, the parameter β describes the rate of protein production. In reality, mRNA needs to be translated to form the protein, and mRNA itself is also degraded by specific enzymes.

- Derive dynamical equations for the rate of change of mRNA and the rate of change of the protein product, assuming that mRNA is produced at rate β_m and degraded at rate α_m , and that each mRNA

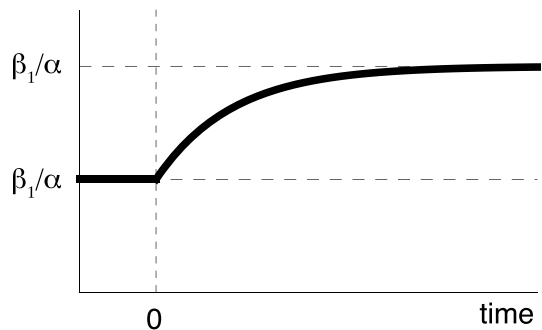


FIGURE 1.12

produces on average p protein molecules per unit time. The protein is removed at rate α .

- b. Note that mRNA is usually degraded at a much faster rate than the protein is removed, $\alpha_m \gg \alpha$. Can this be used to form a quasi-steady-state assumption that mRNA levels are at steady state with respect to slower processes? What is the effective protein production rate β in terms of β_m , α_m and p ? What would be the response time if the mRNA lifetime were much longer than the protein lifetime?

Solution:

- a. The dynamic equation for the concentration of mRNA of gene Y , Y_m , is:

$$dY_m/dt = \beta_m - \alpha_m Y_m \quad (\text{P1.3})$$

The dynamical equation for the protein product is due to production of p copies per mRNA per unit time and degradation/dilution at rate α :

$$dY/dt = pY_m - \alpha Y \quad (\text{P1.4})$$

- b. In the typical case that mRNA degradation is faster than the degradation/dilution of the protein product, we can assume that Y_m reaches steady state quickly in comparison to the protein levels. The reason is that the typical time for the mRNA to reach steady state is the response time $\log(2)/\alpha_m$, which is much shorter than the protein response time $\log(2)/\alpha$ because $\alpha_m \gg \alpha$. The steady-state mRNA level is found by setting $dY_m/dt = 0$ in Equation P1.3, yielding

$$Y_{m,st} = \beta_m/\alpha_m \quad (\text{P1.5})$$

Using this for Y_m in Equation P1.4 yields the following equation for the protein production rate:

$$dY/dt = p\beta_m/\alpha_m - \alpha Y \quad (\text{P1.6})$$

In other words, the effective protein production rate, which is the first term on the right-hand side of the equation, is equal to the steady-state mRNA level times the number of proteins translated from each mRNA:

$$\beta = p\beta_m/\alpha_m \quad (\text{P1.7})$$

- 1.3 *Time-dependent production and decay:* A gene Y with simple regulation has a time-dependent production rate $\beta(t)$ and a time-dependent degradation rate $\alpha(t)$. Solve for its concentration as a function of time.

Solution:

Verify by taking the time derivative that the following is correct:

$$Y(t) = e^{-\int_0^t \alpha(t') dt'} \left(Y(0) + \int_0^t \beta(t') e^{\int_0^{t'} \alpha(t'') dt''} dt' \right) \quad (\text{P1.8})$$

- 1.4 *Cascades:* Consider a cascade of three activators, $X \rightarrow Y \rightarrow Z$. Protein X is initially present in the cell in its inactive form. The input signal of X , S_x , appears at time $t = 0$. As a result, X rapidly becomes active and binds the promoter of gene Y , so that protein Y starts to be produced at rate β . When Y levels exceed a threshold K_y , gene Z begins to be transcribed. All proteins have the same degradation/dilution rate α . What is the concentration of protein Z as a function of time? What is its response time with respect to the time of addition of S_x ? What about a cascade of three repressors? Compare your solution to the experiments shown in Figure 1.13.

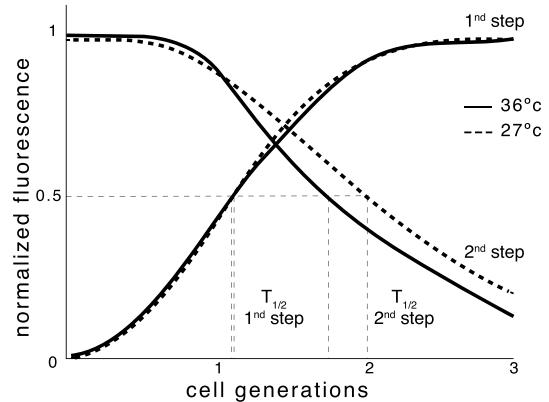


FIGURE 1.13 Adapted from (Rosenfeld and Alon, 2003).

- 1.5 *Fan out:* Transcription factor X regulates two genes, Y_1 and Y_2 . Draw the resulting network, termed a fan out with two target genes. The activation thresholds for these genes are K_1 and K_2 . The activator X begins to be produced at time $t = 0$ at rate β and is removed at rate α . The signal S_x is present throughout. What are the times at which the stable proteins Y_1 and Y_2 reach halfway to their maximal expression?
- 1.6 *Pulse of activation:* Consider the cascade of Exercise 1.4. The input signal S_x appears at time $t = 0$ for a pulse of duration D , and then vanishes.
- What is the concentration $Y(t)$?
 - What is the minimal pulse duration needed for the activation of gene Z ? Use a logic input function.
 - Plot the maximal level reached by the protein Z as a function of pulse duration D .

BIBLIOGRAPHY

- Alon, U. 2003. 'Biological networks: The tinkerer as an engineer', *Science*, 301(5641), pp. 1866–1867. doi: 10.1126/science.1089072.
- Beer, M.A. and Tavazoie, S. 2004. 'Predicting gene expression from sequence', *Cell*, 117(2), pp. 185–198. doi: 10.1016/S0092-8674(04)00304-6.

- Davidson, E.H. et al. 2002. 'A genomic regulatory network for development', *Science (New York, N.Y.). American Association for the Advancement of Science*, 295(5560), pp. 1669–1678. doi: 10.1126/science.1069883.
- Glass, L. and Kauffman, S.A. 1973. 'The logical analysis of continuous, non-linear biochemical control networks', *Journal of Theoretical Biology*, 39(1), pp. 103–129. Available at: <http://www.ncbi.nlm.nih.gov/pubmed/4741704>.
- Kalir, S. and Alon, U. 2004. 'Using a quantitative blueprint to reprogram the dynamics of the flagella gene network', *Cell*, 117(6), pp. 713–720. doi: 10.1016/j.cell.2004.05.010.
- Levine, M. and Davidson, E.H. 2005. 'Gene regulatory networks for development', *Proceedings of the National Academy of Sciences*, 102(14), pp. 4936–4942. doi: 10.1073/pnas.0408031102.
- Mayo, A.E. et al. 2006. 'Plasticity of the cis-regulatory input function of a gene', *PLoS Biology*, 4(4). doi: 10.1371/journal.pbio.0040045.
- Monod, J., Pappenheimer, A.M. and Cohen-Bazire, G. 1952. 'The kinetics of the biosynthesis of beta-galactosidase in *Escherichia coli* as a function of growth', *Biochimica et Biophysica Acta*, 9(6), pp. 648–660. Available at: <http://www.ncbi.nlm.nih.gov/pubmed/13032175>.
- Ptashne, M. 1986. *A genetic switch: gene control and phage lambda*. Cell Press.
- Ptashne, M. and Gann, A. 2002. *Genes & signals*. Cold Spring Harbor Laboratory Press. Available at: https://books.google.co.il/books/about/Genes_Signals.html?id=IvMQRwA3bEAC&redir_esc=y.
- Rosenfeld, N. and Alon, U. 2003. 'Response delays and the structure of transcription networks', *Journal of Molecular Biology*, 329(4), pp. 645–654. Available at: <http://www.ncbi.nlm.nih.gov/pubmed/12787666>.
- Setty, Y. et al. 2003. 'Detailed map of a cis-regulatory input function', *Proceedings of the National Academy of Sciences of the United States of America*, 100(13), pp. 7702–7707. doi: 10.1073/pnas.1230759100.
- Thieffry, D. et al. 1998. 'From specific gene regulation to genomic networks: A global analysis of transcriptional regulation in *Escherichia coli*', *BioEssays*, 20(5), pp. 433–440. doi: 10.1002/(SICI)1521-1878(199805)20:5<433::AID-BIES10>3.0.CO;2-2.
- Thieffry, D. and Thomas, R. 1998. 'Qualitative Analysis of Gene Networks', in *Proceedings of the Third Pacific Symposium on Biocomputing*, 1998, pp. 77–88.
- Yuh, C.H., Bolouri, H. and Davidson, E.H. 1998. 'Genomic cis-regulatory logic: Experimental and computational analysis of a sea urchin gene', *Science*, 279(5358), pp. 1896–902. doi: 10.1126/SCIENCE.279.5358.1896.



Taylor & Francis

Taylor & Francis Group

<http://taylorandfrancis.com>

Autoregulation

A Network Motif

2.1 INTRODUCTION

In the previous chapter, we considered the dynamics of a single interaction in a transcription network. Now, let's take a look at a real, live transcription network made of many interaction arrows. As an example, we will use a network from *E. coli* that includes about 20% of the organism's genes (Figure 2.1). This network looks very complex. Our goal will be to define understandable patterns of connections that serve as building blocks of the network. Ideally, we would like to understand the dynamics of the entire network based on the dynamics of the individual building blocks. In this chapter, we will:

1. Define a way to detect building-block patterns in complex networks, called **network motifs**.
2. Examine the simplest network motif in transcription networks, **negative autoregulation**.
3. Show that this motif has useful functions: speeding up the response time of transcription interactions and stabilizing them against noise.

2.2 PATTERNS, RANDOMIZED NETWORKS AND NETWORK MOTIFS

The transcription network of *E. coli* contains numerous patterns of nodes and arrows. Our approach will be to look for meaningful patterns on the basis of statistical significance.

To define statistical significance, we compare the network to an ensemble of **randomized networks**. The randomized networks are networks with the same characteristics as the real network (e.g., the same number of nodes and arrows as the real network), but where the connections between nodes and arrows are made at random. Patterns that occur in the real network significantly more often than in randomized networks are called **network motifs** (Milo et al., 2002; Shen-Orr et al., 2002).

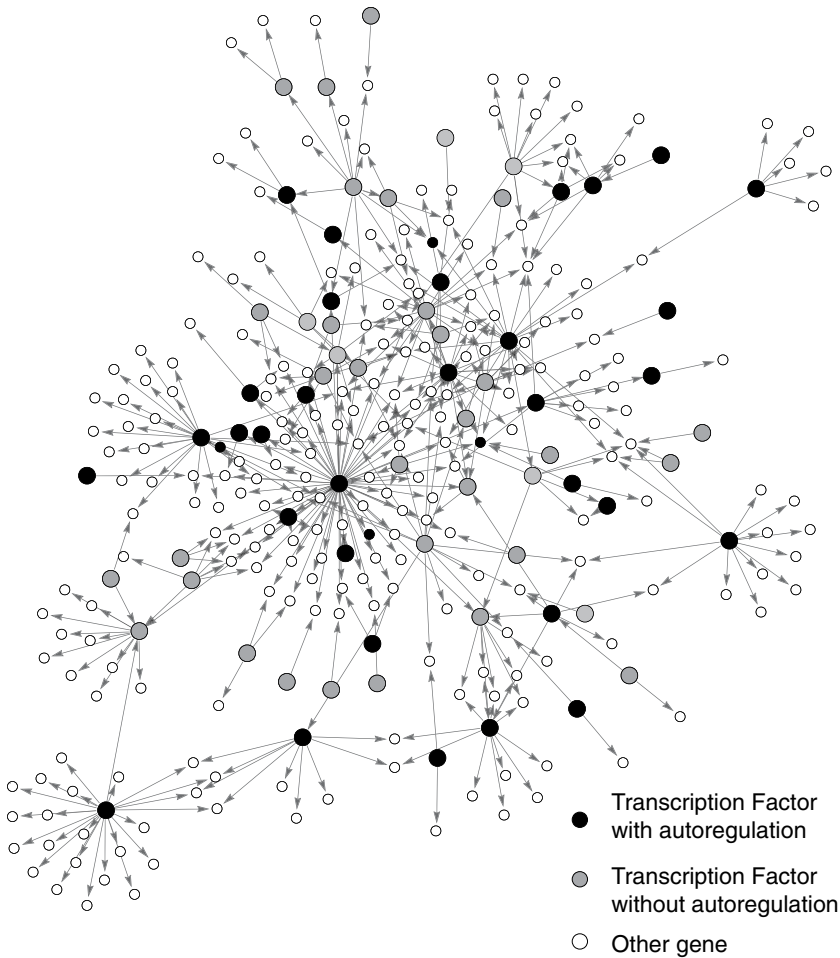


FIGURE 2.1

The basic idea is that patterns that occur in the real network much more often than at random must have been preserved over evolutionary timescales against mutations that randomly change arrows. To appreciate this, note that arrows are easily lost in a transcription network. A mutation that changes a single DNA letter in a promoter can abolish binding of a transcription factor and cause the loss of an arrow in the network.

Such mutations occur at a comparatively high rate, as can be appreciated by the following example. A single bacterium placed in a test tube with 10 mL of liquid nutrient grows and divides to reach a saturating population of about 10^{10} cells within less than a day. This population therefore underwent 10^{10} DNA replications. Since the mutation rate is about 10^{-9} per letter per replication, the population will include, for each letter in the genome, 10 different bacteria with a mutation in that letter. Thus, a change of any DNA letter can be rapidly reached in bacterial populations. A similar rate of mutations per generation per genome occurs in multi-cellular organisms (Table 1.1).

Similarly, new arrows can be added to the network by mutations that generate a binding site for transcription factor *X* in the promoter region of gene *Y*. Such sites can be generated, for example, by mutations or by events that duplicate or reposition pieces of a genome, or that insert into the genome pieces of DNA from other cells (Shapiro, 1999).

Hence, arrows in network motifs must be constantly selected in order to survive randomizing forces. This suggests that if a network motif appears in a network much more often than in randomized networks, it must have been selected based on some advantage it gives to the organism. Otherwise it would have been washed out.

2.2.1 Detecting Network Motifs by Comparison to Randomized Networks

To detect network motifs, we need to compare the real network to an ensemble of randomized networks. We will consider the simplest ensemble of randomized networks, introduced by Erdős and Rényi (1960; Bollobás and Thomason, 1985). This makes calculations easy and gives the same qualitative answers as more elaborate random network models.

For a meaningful comparison, the randomized networks should share the basic features of the real network. The real transcription network has *N* nodes and *A* arrows. To compare it to the Erdős–Rényi (ER) model, one builds a random network with the same number of nodes and arrows. In the random network, defined by the ER model, arrows are assigned at random between pairs of nodes. **Figure 2.2** compares a small network to a corresponding random ER network with the same number of nodes and arrows.

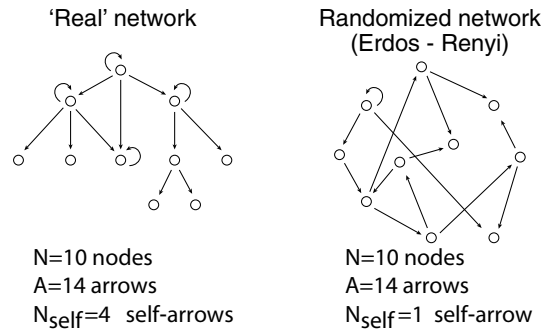


FIGURE 2.2

2.3 AUTOREGULATION IS A NETWORK MOTIF

We can now begin to compare features of the *E. coli* transcription network with the randomized networks. Let's start with self-arrows, arrows that originate and end at the same node. The *E. coli* network that we use as an example has 40 self-arrows (black nodes in **Figure 2.1**). These self-arrows correspond to transcription factors that regulate the transcription of their own genes.

Regulation of a gene by its own gene product is known as autogenous control, or **autoregulation**. Thirty-four of the autoregulatory proteins in the network are repressors that repress their own transcription: **negative autoregulation**.

Is autoregulation significantly more frequent in the real network than at random? To decide, we need to calculate the average number of self-arrows in an ER random network. To form a self-arrow, an arrow needs to choose its node of origin as its destination, out of the *N* possible target nodes. This probability is thus:

$$p_{\text{self}} = 1/N \quad (2.3.1)$$

Since A arrows are placed at random to form the random network, the average number of self-arrows is A times p_{self} , similar to tossing a coin A times with probability for heads of p_{self} :

$$\langle N_{\text{self}} \rangle_{\text{rand}} = A p_{\text{self}} = A/N \quad (2.3.2)$$

with a standard deviation that is approximately the square root of the mean (as in the coin tossing analogy):

$$\sigma_{\text{rand}} = \sqrt{A/N} \quad (2.3.3)$$

In the *E. coli* transcription network of [Figure 2.1](#), the numbers of nodes and arrows are $N = 424$ and $A = 519$. Thus, according to Equations 2.3.2 and 2.3.3, a corresponding ER network with the same N and A would be expected to have only about one self-arrow, plus or minus one:

$$\langle N_{\text{self}} \rangle_{\text{rand}} = A/N \sim 1.2, \quad \sigma_{\text{rand}} \sim \sqrt{1.2} \sim 1.1 \quad (2.3.4)$$

In contrast, the real network has 40 self-arrows, which exceeds the random networks by 35 standard deviations, which means they occur far more often than at random. Note that 35 standard deviations mark a very high statistical significance.

Thus, self-arrows, and in particular negatively autoregulated genes, are a network motif. A network motif is a recurring pattern in the network that occurs far more often than at random.

The next question is: Why is negative autoregulation a network motif? Does it have a useful function? To answer this, we will compare a negatively autoregulated gene to a simply (non-auto) regulated gene ([Figure 2.3](#)). Our criterion for comparison will be the response time of the system. As we saw in the previous chapter, the response time of a simply regulated gene is governed by its removal rate α :

$$T_{1/2} = \log(2)/\alpha \quad (2.3.5)$$

For stable proteins that are not appreciably degraded in the cell, the response time is equal to the cell generation time. We will now see how the negative autoregulation network motif can help speed up transcription responses.

2.4 NEGATIVE AUTOREGULATION SPEEDS THE RESPONSE TIME OF GENE CIRCUITS

Negative autoregulation occurs when a transcription factor X represses its own transcription ([Figure 2.3](#)). This self-repression occurs when X binds its own promoter to inhibit production of mRNA. As a result, the higher the concentration of X , the lower its production rate.

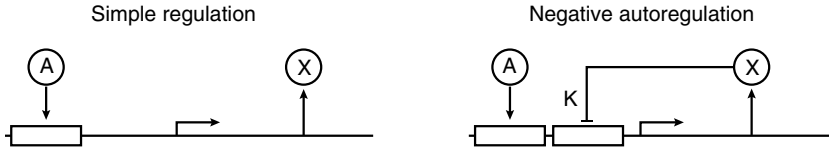


FIGURE 2.3

As we saw in the previous chapter, the dynamics of X are described by its production rate $f(X)$ and removal rate α :

$$\frac{dX}{dt} = f(X) - \alpha X \quad (2.4.1)$$

where $f(X)$ is a decreasing function of X .¹ As mentioned in [Chapter 1](#), a good approximation for many promoters is a decreasing Hill function:

$$f(X) = \frac{\beta K^n}{K^n + X^n} \quad (2.4.2)$$

In this input function, when X is much smaller than the repression coefficient K , the promoter is free and the production rate reaches its maximal value, β . On the other hand, when repressor X is at high concentration, no transcription occurs, $f(X) \sim 0$. The repression coefficient K has units of concentration, and equals the concentration at which X represses the promoter activity by 50%.

To solve the dynamics in the most intuitive way, let's use the logic approximation, where production is zero if $X > K$, and production is maximal, namely, $f(X) = \beta$, when X is smaller than K . This was described in [Chapter 1.3.4](#) using the step function θ :

$$f(X) = \beta \theta(X < K) \quad (2.4.3)$$

In [Exercise 2.3](#), we will also solve the dynamics with a Hill function, to find that the logic approximation is reasonable.

To study the response time, consider the case where X is initially absent, and its production starts at $t = 0$. At early times, while X concentration is low, the promoter is unrepressed and production is full-steam at rate β , as described by the production-removal equation:

$$\frac{dX}{dt} = \beta - \alpha X \quad \text{while} \quad X < K \quad (2.4.4)$$

¹ To understand the dynamics of a negatively autoregulated system, recall the separation of timescales in transcription networks. The production rate of X is governed by the probability that X binds the promoter of its own gene. The binding and unbinding of X to the promoter reaches equilibrium on the timescale of seconds. The concentration of protein X , on the other hand, changes much more slowly, on the timescale of hours. Therefore, it makes sense to describe the production rate by an input function, $f(X)$, equal to the mean promoter activity at a given level of X , averaged over many repressor binding events.

This results in an approach to a high steady-state value, as described in Section 1.4 of the previous chapter. At early times, in fact, we can neglect removal ($\alpha X \ll \beta$) to find a linear accumulation of X with time:

$$X(t) \sim \beta t \text{ while } X < K \text{ and } X \ll \beta/\alpha \quad (2.4.5)$$

However, production stops when X levels reach the self-repression threshold, $X = K$, because production is zero when X exceeds K (Figure 2.4). Small oscillations will occur around $X = K$ if there are delays in the system. Delays cause X to overshoot beyond K slightly, but then production stops and X levels decline until they decrease

below K , upon which production starts again, and so on. These oscillations are generally damped for realistic $f(X)$ unless delays are very long. Thus, X effectively locks into a steady-state level equal to the repression coefficient of its own promoter:

$$X_{st} = K \quad (2.4.6)$$

The resulting dynamics shows a rapid rise and a sudden saturation, as shown in Figure 2.5.

The response time, $T_{1/2}$, can be found by asking when X reaches halfway to steady state. For simplicity, let us calculate the response time using linear accumulation of X (Equation 2.4.5), in which $X = \beta t$. The response time, $T_{1/2}^{NAR}$, where NAR stands for negative autoregulation, is the time when X reaches half of the steady-state level,

$$\beta T_{1/2}^{NAR} = \frac{X_{st}}{2} = \frac{K}{2}, \text{ so that:}$$

$$T_{1/2}^{NAR} = K/2\beta \quad (2.4.7)$$

The stronger the maximal unrepressed promoter activity β , the shorter the response time. Negative autoregulation can therefore use a strong promoter to give an initial fast production, and then use autorepression to stop production at the desired steady state.

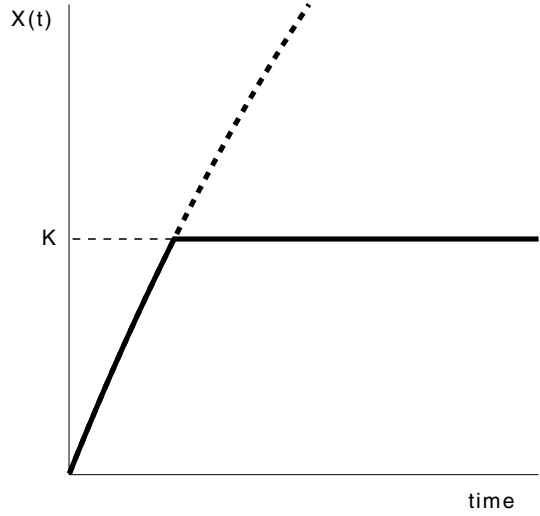


FIGURE 2.4

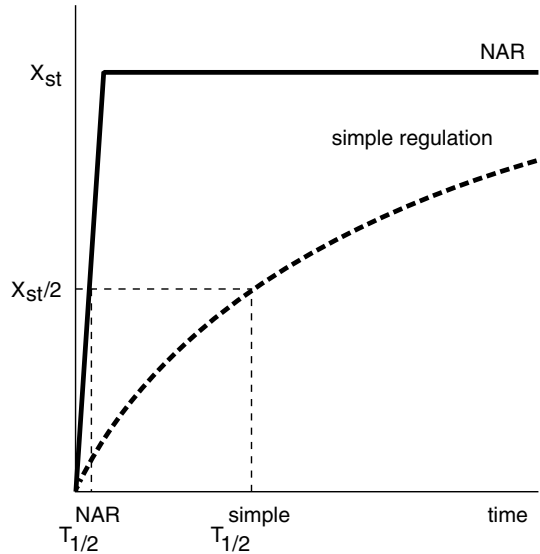


FIGURE 2.5

Note that evolutionary selection can easily tune the parameters β and K independently. The repression threshold K can be modified, for example, by mutations in the binding site of X , whereas β can be tuned by mutations in the binding site of RNAP in the promoter. Thus, the steady state ($X_{st} = K$) and the response time can be separately determined.

Let us compare this design with a simply regulated gene (a gene without negative autoregulation, as described in Section 1.4), which is produced at rate β_{simple} and removed at rate α_{simple} . To make a fair comparison, we should compare the two designs with the same steady-state levels. This is because the steady-state level of the protein is important for its optimal function. To achieve the same steady state, we must compensate for the repressive effect of NAR by providing it with a stronger promoter activity β than in simple regulation. In addition, the two designs should have as many of the same biochemical parameters as possible. For example, the two designs should have the same protein removal rate, $\alpha = \alpha_{\text{simple}}$. Such a fair comparison between biological circuits was called by Michael Savageau a **mathematically controlled comparison** (Savageau, 1976).

For a mathematically controlled comparison, we set K so that both designs reach the same steady-state expression level. Using the fact that in simple regulation $X_{st} = \beta_{\text{simple}}/\alpha_{\text{simple}}$ (Equation 1.4.3) and in NAR $X_{st} = K$ (Equation 2.4.6), equal steady state in the two circuits occurs when

$$K = \beta_{\text{simple}} / \alpha_{\text{simple}} \quad (2.4.8)$$

What is the response time of the two designs? The response time of simple regulation is governed by the removal rate as described in Chapter 1, so that $T_{1/2}^{\text{simple}} = \log(2)/\alpha_{\text{simple}}$. A much faster response can be achieved by the corresponding negative autoregulated circuit by making the promoter activity β large, because the response time, $T_{1/2}^{\text{NAR}} = K/2\beta$, is inversely proportional to β . Using Equation 2.4.8, we find that the ratio of the response times in the two designs can be made very small by making NAR have much stronger promoter activity than simple regulation:

$$\frac{T_{1/2}^{\text{NAR}}}{T_{1/2}^{\text{simple}}} = \frac{1}{2\log(2)} \frac{\beta_{\text{simple}}}{\beta} \quad (2.4.9)$$

An example is shown in Figure 2.5, in which the response time of the negative autoregulation design is about sevenfold faster than simple regulation. The intuitive reason for speedup is that NAR uses a strong promoter for a fast-initial rise, and then stops itself at the desired steady state K . Simple regulation with the same strong promoter would reach a steady state that is far too high, causing undesirable

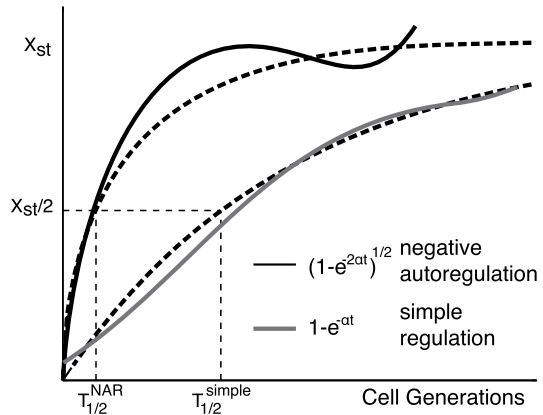


FIGURE 2.6 Adapted from (Rosenfeld, Elowitz and Alon, 2002).

over-expression of the gene product. Thus, you can be a fast driver if you have a strong motor and good breaks. The accelerated response of a negative autoregulatory circuit compared to simple regulation was experimentally demonstrated using high-resolution gene expression measurements (Figure 2.6, experiments are in full lines, and theory in dashed lines).

2.4.1 Rate Analysis Shows Speedup for Any Repressive Input Function $f(X)$

Speedup is also found when using Hill input functions instead of a step function. In fact, any shape of the input function $f(X)$, as long as it is a decreasing function of X , causes speedup in NAR. To see this, we introduce a useful tool for understanding circuits called the **rate plot**. Consider first simple regulation, $dX/dt = \beta - \alpha X$. In the rate plot, we plot the rates of production β and removal αX as a function of protein level X (Figure 2.7).

The first thing to notice is the value of X at which these two lines cross. At this point, production equals removal, and hence X levels don't change ($dX/dt = 0$). This is called a **fixed point** of the equation, namely the steady-state value ($X_{st} = \beta/\alpha$).

The rate plot can show us that this fixed point is globally stable: any value of X flows back to the fixed point. To see this, note that when X is higher than the fixed point, the removal curve is higher than the production curve. Hence, more X is removed than is produced, and X shrinks back to the fixed point. Similarly, when X is lower than the fixed point, production exceeds removal and X grows. X stops changing when it reaches the fixed point, its steady-state value X_{st} .

Now let's consider the response time. The speed at which X approaches the fixed point is given by the distance between the two curves, because the speed is the temporal derivative $dX/dt = \text{production} - \text{removal}$ (Figure 2.8).

Now let's do the mathematically controlled comparison with NAR

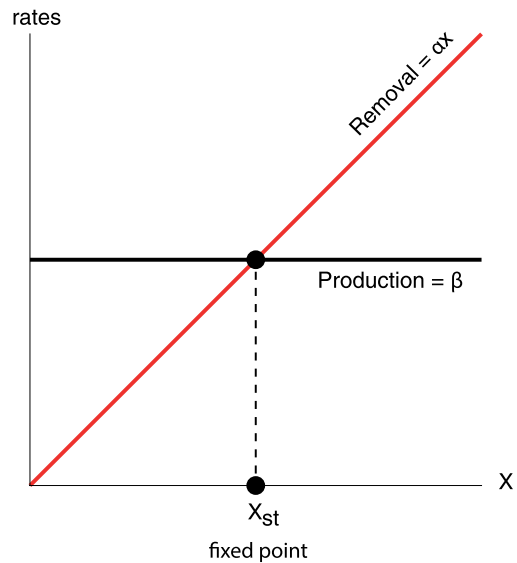


FIGURE 2.7

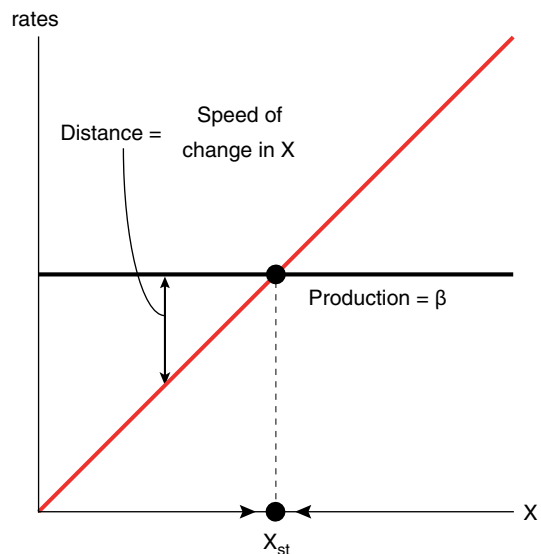


FIGURE 2.8

(Figure 2.9). We want the removal rate α to be the same in the two circuits, so the removal curve αX is the same. We also want the steady state, X_{st} , to be the same, so the production curve $f(X)$ must cross the removal curve at the same point as in simple regulation. We know that $f(X)$ is a decreasing function (negative autoregulation). The only way it can cross the removal line at the desired point is if it starts above the simple regulation production curve, crosses it at X_{st} and then drops below it. Due to this geometry, no matter what the exact shape of $f(X)$ is, we see that the distance between production and removal curves in NAR is bigger than in simple regulation. Thus, no matter what $f(X)$ is, protein level X will move faster to the fixed point (Figure 2.9). NAR speeds responses.

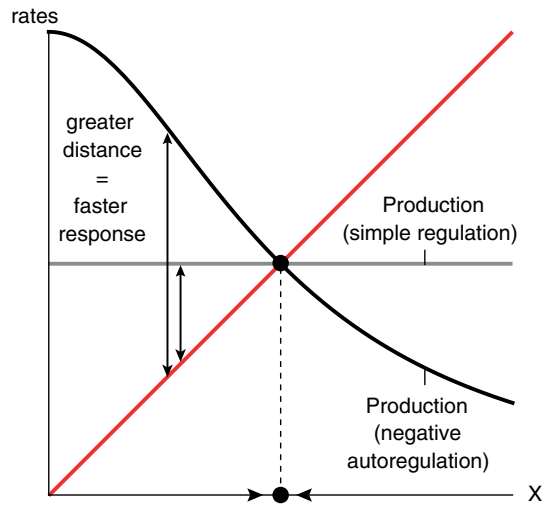


FIGURE 2.9

2.5 NEGATIVE AUTOREGULATION PROMOTES ROBUSTNESS TO FLUCTUATIONS IN PRODUCTION RATE

In addition to speeding the response time, negative autoregulation confers a second important benefit. This benefit is the increased robustness of the steady-state expression level with respect to fluctuations in the production rate β . This property was experimentally demonstrated using measurements of protein levels in individual cells (Becksel and Serrano, 2000). The production rate of a given gene, β , fluctuates over time due to variations in the metabolic capacity of the cell and its regulatory systems (see Appendix D). These cell–cell differences in β are typically on the order of tens of percents, and last over the entire generation time of the cells. Thus, a snapshot of genetically identical cells grown under identical conditions will show cell–cell differences in the expression of every protein. Noise is an unavoidable property of biological material.

Simple gene regulation is affected quite strongly by fluctuations in production rate β . The steady-state level is linearly dependent on the production rate:

$$X_{st} = \beta/\alpha \quad (2.5.1)$$

and therefore, a change in β leads to a proportional change in X_{st} . In contrast, negative autoregulation buffers fluctuations in the production rate. In the case of the sharp (step-like) autorepression that we have discussed, the steady-state level does not depend on β at all, and depends only on the repression threshold of X for its own promoter:

$$X_{st} = K \quad (2.5.2)$$

The repression threshold K is determined by hardwired factors such as the chemical bonds between X and its DNA site. Such parameters vary much less from cell to cell than production rates.

Moreover, NAR can make the steady-state robust to changes in removal rate α , such as those that occur when the growth rate of the cells changes. Removal rate affects the steady state in simple regulation quite strongly, as can be seen in the rate plot of Figure 2.10. In contrast, NAR with a steep regulation function has a steady state that depends only weakly on α , making protein levels less sensitive to changes in cell growth rate (Figure 2.11). This robustness to growth rate was experimentally demonstrated by Klumpp, Zhang and Hwa (2009). It is useful for making synthetic circuits in cells that can achieve a well-defined protein level (Shimoga et al., 2013).

Robustness of key properties of a biological system is a general design principle. We will study robustness in more depth in Part 2 of this book.

What about positive autoregulation? We will see in Chapter 5 that positive autoregulation acts in an opposite way: it slows down responses and can amplify noise in parameters. Such slowdown and stochasticity can be useful for processes that take many cell generations, as occurs when organisms develop from an egg to an embryo.

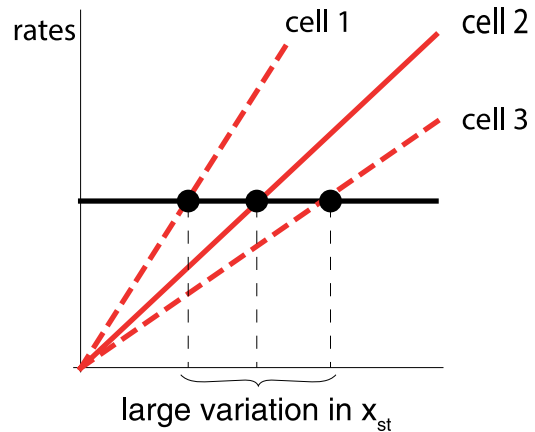


FIGURE 2.10

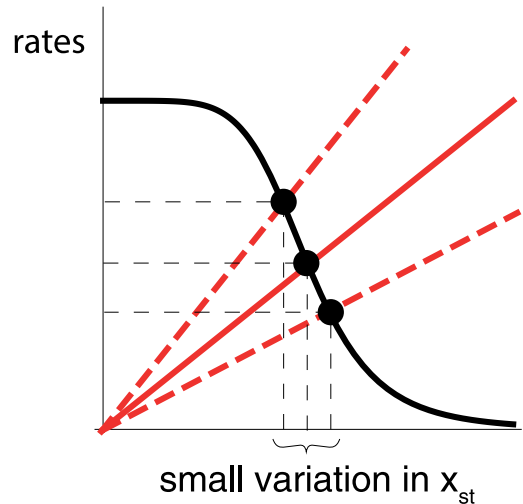


FIGURE 2.11

2.6 SUMMARY: EVOLUTION AS AN ENGINEER

Negative autoregulation is a network motif, a pattern that recurs throughout the network at numbers much higher than expected in random networks.

To understand why negative autoregulation is a network motif, we asked what functions it can perform. For this purpose, we analyzed its dynamic behavior. The dynamic analysis can be phrased as an engineering story. Think of evolution as an engineer working to design a gene circuit that reaches a desired steady-state concentration X_{st} . One possible design, design A, is simple regulation with a production rate set to reach X_{st} . Design B is negative autoregulation, with a stronger initial production rate, which, as X builds up, is suppressed to result in the desired steady state.

The second (NAR) design has the advantage that the goal, X_{st} , is reached faster. Furthermore, the fluctuations around X_{st} due to variations in production and removal rate are reduced in the second, autoregulated design. In an imaginary competition between two species are identical except that one uses circuit A, and the second uses circuit B, the latter would have a selective advantage. Over evolutionary times, structures that have engineering advantages would tend to be selected and appear as network motifs.

FURTHER READING

-
- (Becksel and Serrano, 2000) “Engineering stability in gene networks by autoregulation.”
 (Klumpp, Zhang and Hwa, 2009) “Growth-rate dependent global effects on gene expression in bacteria.”
 (Rosenfeld, Elowitz and Alon, 2002) “Negative autoregulation speeds the response time of transcription networks.”
 (Savageau, 1974) “Comparison of classical and autogenous systems of regulation in inducible operons.”
 (Savageau, 1976) “Biochemical systems analysis: a study of function and design in molecular biology.”
 (Shimoga et al., 2013) “Synthetic mammalian transgene negative autoregulation.”

EXERCISES

-
- 2.1 *Random networks*: (a) Write a computer program that produces a random Erdős–Rényi (ER) network with N nodes and A arrows. (b) Generate 100 ER networks with $A = 500$ and $N = 400$. What is the mean and standard deviation of the number of self-arrows?
- 2.2 *Self-arrows in random networks with transcription factors*: Consider the following random network model, which we will call modified-ER (MER). There are N nodes, of which only N_1 are allowed to send out arrows. These N_1 represent TFs which can regulate genes. Each arrow can choose any node out of the N as a target, including its node of origin. (a) Write a program to generate MER networks. (b) Generate 100 MER networks with $A = 500$, $N = 400$ and $N_1 = 100$ (corresponding approximately to the *E. coli* network used in this chapter). What is the average and standard deviation of the number of self-arrows? (c) Write a formula for the average number of self-arrows in a MER network as a function of A , N and N_1 . Are there more or less self-arrows than in an ER network? Explain.
- 2.3 *Autorepression with Hill input function*: What is the response time for a repressor that represses its own promoter as described by a Hill function with Hill coefficient n ?

$$\frac{dX}{dt} = \frac{\beta}{1 + (X/K)^n} - \alpha X \quad (\text{P2.1})$$

How much faster is the response than in non-autoregulated circuits? Use the approximation of strong autorepression, that is, $(X/K)^n \gg 1$.

Solution:

In the limit of strong autorepression, we can neglect the 1 in the denominator of the input function, and we have:

$$\frac{dX}{dt} = \frac{\beta K^n}{X^n} - \alpha X \quad (\text{P2.2})$$

To solve this equation, multiply both sides by X^n and switch to the new variable, $u = X^{n+1}$. Note that $du/dt = (n+1)X^n dX/dt$. The equation now reads:

$$\frac{du}{dt} = (n+1)\beta K^n - (n+1)\alpha u \quad (\text{P2.3})$$

The solution of this linear equation is simple exponential convergence to steady state, the same as in [Chapter 1](#):

$$u = u_{st}(1 - e^{-(n+1)\alpha t}) \quad (\text{P2.4})$$

Switching back to the original variable X , we have:

$$X = X_{st}(1 - e^{-(n+1)\alpha t})^{\frac{1}{(n+1)}} \quad (\text{P2.5})$$

The response time is found by $X(T_{1/2}) = X_{st}/2$. This yields:

$$T_{1/2} = \frac{1}{(n+1)\alpha} \log \left(\frac{2^{n+1}}{2^{n+1} - 1} \right) \quad (\text{P2.6})$$

The response time decreases with n . For $n = 1, 2, 3$, the response time of NAR is 0.2, 0.06 and 0.02 of that of simple regulation. See [Figure 2.5](#) for the dynamics of a strongly autoregulated gene with $n = 1$. The sharper the negative autoregulation (higher n), the more the system approaches the sharp logic function limit discussed in this chapter, and the faster it responds.

When is this approximation valid? Note that the steady state is, according to Equation P2.2, $X_{st} = K(\beta/\alpha K)^{1/(n+1)}$. Thus, when the unrepresed steady state is much larger than the repression coefficient, that is, when $\beta/\alpha \gg K$, we have $X_{st} \gg K$.

- 2.4 *Parameter sensitivity:* Analyze the robustness of the steady-state level of X with respect to cell-cell variations in the production rate β for the system of Problem 2.3. To do this, we calculate the **parameter sensitivity** coefficient (Savageau, 1976; Goldbeter and Koshland, 1981; Heinrich and Schuster, 1998) of the steady-state concentration X_{st} with respect to β . The parameter sensitivity coefficient of property A with respect

to parameter B , denoted $S(A, B)$, is defined as the relative change in A for a given small relative change in B , that is, S :

$$S(A, B) = \frac{\Delta A}{A} \bigg/ \frac{\Delta B}{B} = \frac{B}{A} \frac{dA}{dB} \quad (\text{P2.7})$$

Solution:

The steady-state level is found from Equation P2.2 using $dX/dt = 0$, yielding:

$$X_{st} = K \left(\frac{\beta}{\alpha K} \right)^{\frac{1}{n+1}} \quad (\text{P2.8})$$

The parameter sensitivity coefficient, which describes relative changes in steady state due to changes in production rate, is:

$$S(X_{st}, \beta) = \frac{\beta}{X_{st}} \frac{dX_{st}}{d\beta} = \frac{1}{n+1} \quad (\text{P2.9})$$

Thus, sensitivity decreases with Hill coefficient n . The higher n , the weaker the dependence of the steady state on β . In other words, robustness to variations in production rates increases with the Hill coefficient.

For Hill coefficient $n = 4$, for example, $S(X_{st}, \beta) = 1/5$, which means that a 10% change in β yields only a 2% change in X_{st} . In the limit of very high n , the steady state does not depend at all on production or degradation rates, $X_{st} = K$. This is the steady-state solution found in the main text for the logic input function. Simple regulation is equivalent to $n = 0$, so that $S(X_{st}, \beta) = 1$. This means that a small change of $x\%$ in production leads to the same change of $x\%$ in steady state.

2.5 *Autoregulated cascade:* Gene X encodes a repressor that represses gene Y , which also encodes a repressor. Both X and Y negatively regulate their own promoters.

- a. Draw the circuit diagram.
- b. At time $t = 0$, X begins to be produced at rate β , starting from an initial concentration of $X = 0$. What are the dynamics of X and Y ? What are the response times of X and Y ? Assume logic input functions, with repression thresholds K_{XX} , K_{XY} for the action of X on its own promoter and on the Y promoter, and K_{YY} for the action of Y on its own promoter.
- c. At time $t = 0$, production of X stops after a long period of production, and X concentration decays from its initial steady-state level. What are the dynamics of X and Y ? What are the response times of X and Y ?

- 2.6 *Linearized positive autoregulation*: What is the effect of positive autoregulation on the response time?

Use as a model the following linear equation:

$$\frac{dX}{dt} = \beta + \beta_1 X - \alpha X$$

with $\beta_1 < \alpha$. Explain each term and solve for the response time. When might such a design be biologically useful? What happens when $\beta_1 > \alpha$?

- 2.7 *Turning off autoregulation*: What are the dynamics of a negatively autoregulated gene at steady state once its maximal promoter activity is suddenly reduced to $\beta = 0$? What is the response time, and how does it compare to simple regulation?

- 2.8 *Positive autoregulation with a step function*: Consider a step-function model for positive autoregulation, in which production of X is at level β_0 when $X < K$, and rises to β_1 when $X \geq K$. The removal rate is α .

i. Design the circuit such that production rate crosses removal rate three times in the rate plot.

ii. Solve the dynamics $X(t)$ for any initial value of X .

iii. What is the response time?

- 2.9 *Two-node positive feedback for decision-making*: During development from an egg to an embryo, cells need to make irreversible decisions to express the genes appropriate to their designated tissue types and repress other genes. One common mechanism is positive transcriptional feedback between two genes. There are two types of positive feedback made of two transcription factors. The first type is of two positive interactions $X \rightarrow Y$ and $Y \rightarrow X$. The second type has two negative interactions $X \rightarrow Y$ and $Y \rightarrow X$. What are the stable steady states in each type of feedback? Which type of feedback would be useful in situations where genes regulated by both X and Y belong to the same tissue? Which would be useful when genes regulated by X belong to different tissues than the genes regulated by Y ?

- 2.10 *NAR can increase input dynamic range*: Input dynamic range is the range input signal S_x that elicits a sizable change in output, X_{st} . For example, input dynamic range R can be defined as the ratio of input levels S_x needed for 90% and 10% response (i.e., for X_{st} to reach 90% and 10% of its maximal level).

a. Consider simple regulation with production rate is regulated by an upstream TF such that $\beta = \beta(S_x) = S_x/(K + S_x)$. What is the input dynamic range R ?

b. Add negative autoregulation to this circuit,

$$\frac{dX}{dt} = \frac{\beta(S_x)}{1 + \left(\frac{X}{K}\right)^n} - \alpha X.$$

- c. Explain the production term in this equation.
 - d. Compute the input dynamic range R (assume $(X/K)^n \gg 1$). Explain why NAR is said to increase the input dynamic range (Nevozhay et al., 2009; Madar et al., 2011).
- 2.11 *Linear analysis of stability*: Linear stability analysis assumes small perturbations around a fixed point, so that nonlinear functions can be linearized using their local slope (derivative). The resulting linear equations can be solved to find out whether the perturbation shrinks to zero (indicating a flow back to the fixed point and hence a stable fixed point), or grows (indicating an unstable fixed point).
- Consider an autoregulated circuit $dX/dt = f(X) - \alpha X$ which has a fixed point X_{st} . Assume a small perturbation $X(t) = X_{st} + \delta X(t)$. The function $f(x)$ can be expanded using its local slope, $f(X) = f(X_{st}) + (df(X_{st})/dX)\delta X + \text{higher order terms}$.
- a. Explain why the dynamics of the perturbation $\delta x(t)$ can be approximated by $d\delta X/dt = \beta_1 \delta X - \alpha \delta X$, with $\beta_1 = df(X_{st})/dX$.
 - b. Explain why negative autoregulation always shows a stable fixed point.
 - c. Explain why positive autoregulation can have either stable or unstable fixed points.
- 2.12 *Simple regulation as a filter for high-frequency noise, linear analysis*: Consider a simple regulation circuit $dX/dt = \beta - \alpha X$. Noise can be said to be composed of many components at different temporal frequencies. We will analyze these components one by one, in the limit of small noise in which linear equations can be used. Assume that production rate shows small amplitude fluctuations with frequency ω , so that $\beta = \beta_0 + a \sin(\omega t)$.
- a. Show that the protein X concentration oscillates around its steady state with frequency ω , by showing that that $X = X_{st} + C \sin(\omega t + \phi)$ solves the dynamic equation.
 - b. Plot the amplitude of the variation of X around its steady state, C/a , as a function of ω .
 - c. Explain why simple regulation can be said to filter out high-frequency noise, at frequencies higher than $\omega \sim 1/\alpha$.
- 2.13 *Negative and positive autoregulation as linear filters*: Repeat the linear filter calculation of Exercise 2.12 for linearized versions of positive autoregulation (PAR) and negative autoregulation (NAR). In these versions, the dynamic equation is $dX/dt = \beta + \beta_1 - \alpha X$ where β_1 is negative for NAR (because X reduces its own expression), and positive for PAR (with $\beta_1 < \alpha$).

What is the effect of linearized autoregulation on the noise filtering properties of these circuits? Why can it be said that NAR helps filter out low-frequency noise, whereas PAR can help filter out high-frequency noise?

BIBLIOGRAPHY

- Becksel, A. and Serrano, L. 2000. 'Engineering stability in gene networks by autoregulation', *Nature*, 405(6786), pp. 590–593. doi: [10.1038/35014651](https://doi.org/10.1038/35014651).
- Bollobás, B. and Thomason, A. 1985. 'Random graphs of small order', *North-Holland Mathematics Studies*, 118, pp. 47–97. doi: [10.1016/S0304-0208\(08\)73612-0](https://doi.org/10.1016/S0304-0208(08)73612-0).
- Erdős, P. and Rényi, A. 1960. 'On random graphs', *Publicationes Mathematicae*. doi: [10.2307/1999405](https://doi.org/10.2307/1999405).
- Goldbeter, A. and Koshland, D.E. 1981. 'An amplified sensitivity arising from covalent modification in biological systems', *Proceedings of the National Academy of Sciences*, 78(11), pp. 6840–6844. doi: [10.1073/pnas.78.11.6840](https://doi.org/10.1073/pnas.78.11.6840).
- Heinrich, R. and Schuster, S. 1998. 'The modelling of metabolic systems. Structure, control and optimality', *BioSystems*, 47(1–2), pp. 61–77. doi: [10.1016/S0303-2647\(98\)00013-6](https://doi.org/10.1016/S0303-2647(98)00013-6).
- Klumpp, S., Zhang, Z. and Hwa, T. 2009. 'Growth rate-dependent global effects on gene expression in bacteria', *Cell*, 139(7), pp. 1366–1375. doi: [10.1016/j.cell.2009.12.001](https://doi.org/10.1016/j.cell.2009.12.001).
- Madar, D. et al. 2011. 'Negative auto-regulation increases the input dynamic-range of the arabinose system of *Escherichia coli*', *BMC Systems Biology*, 12(5), p. 111. doi: [10.1186/1752-0509-5-111](https://doi.org/10.1186/1752-0509-5-111).
- Milo, R. et al. 2002. 'Network motifs: Simple building blocks of complex networks', *Science*, 298(5594), pp. 824–827. doi: [10.1126/science.298.5594.824](https://doi.org/10.1126/science.298.5594.824).
- Nevozhay, D. et al. 2009. 'Negative autoregulation linearizes the dose-response and suppresses the heterogeneity of gene expression', *Proceedings of the National Academy of Sciences*, 106(13), pp. 5123–5128. doi: [10.1073/pnas.0809901106](https://doi.org/10.1073/pnas.0809901106).
- Rosenfeld, N., Elowitz, M.B. and Alon, U. 2002. 'Negative autoregulation speeds the response times of transcription networks', *Journal of Molecular Biology*, 323(5), pp. 785–793. doi: [10.1016/S0022-2836\(02\)00994-4](https://doi.org/10.1016/S0022-2836(02)00994-4).
- Savageau, M.A. 1974. 'Comparison of classical and autogenous systems of regulation in inducible operons', *Nature*, 252(5484), pp. 546–549. doi: [10.1038/252546a0](https://doi.org/10.1038/252546a0).
- Savageau, M.A. 1976. 'Biochemical systems analysis. A study of function and design in molecular biology'. Available at: <https://ucdavis-pure-elsevier-com.ezproxy.weizmann.ac.il/en/publications/biochemical-systems-analysis-a-study-of-function-and-design-in-mo>
- Shapiro, J.A. 1999. 'Genome system architecture and natural genetic engineering in evolution', *Annals of the New York Academy of Sciences*, 870(1), pp. 23–35. doi: [10.1111/j.1749-6632.1999.tb08862.x](https://doi.org/10.1111/j.1749-6632.1999.tb08862.x).
- Shen-Orr, S.S. et al. 2002. 'Network motifs in the transcriptional regulation network of *Escherichia coli*', *Nature Genetics*, 31(1), pp. 64–68. doi: [10.1038/ng881](https://doi.org/10.1038/ng881).
- Shimoga, V. et al. 2013. 'Synthetic mammalian transgene negative autoregulation', *Molecular Systems Biology*, 9, p. 670. doi: [10.1038/msb.2013.27](https://doi.org/10.1038/msb.2013.27).

The Feedforward Loop Network Motif

3.1 INTRODUCTION

In this chapter, we will continue to discover network motifs in transcription networks and discuss their function. The main point is that **out of the many possible patterns that could appear in the network, only a few are found significantly – the network motifs**. Network motifs have defined information-processing functions. The benefit of these functions may explain why the same network motifs are rediscovered by evolution again and again in diverse systems.

To find network motifs, we will calculate the number of appearances of patterns in real and random networks. We focus in this chapter on patterns with three nodes (such as triangles). Patterns with two nodes and patterns with more than three nodes will be discussed in the next chapters.

3.2 THE FEEDFORWARD LOOP IS A NETWORK MOTIF

In the previous chapter, we discussed the simplest network motif, self-regulation, a pattern with one node. Let us now consider larger patterns of nodes and arrows, called subgraphs. Two examples of three-node subgraphs are shown in [Figure 3.1](#), the feedforward loop and the feedback loop. In total, there are 13 possible ways to connect three nodes with directed arrows, shown in [Figure 3.2](#). There are 199 possible four-node subgraphs, 9364 five-node subgraphs, and so on.

To find which of these subgraphs are significant, we need to compare the subgraphs in the real network to those in randomized networks. The numbers of subgraphs in randomized networks can be computed analytically or by computer simulations. For example, in Erdős–Rényi

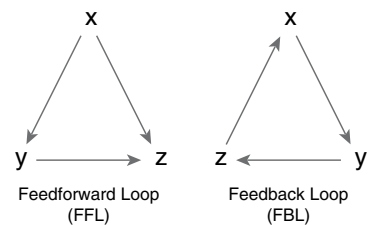


FIGURE 3.1

random networks, the mean number of feedforward loops is equal to the mean connectivity λ (mean number of arrows per node, $\lambda = A/N$) raised to the third power, λ^3 (see solved Exercise 3.2). The third power arises because the feedforward loop has three arrows.

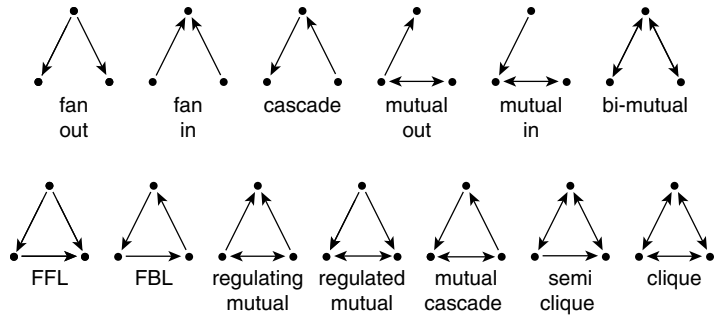


FIGURE 3.2

How do the numbers of three-node patterns in transcription networks compare with random networks? In the *Escherichia coli* transcription network that we use as an example, there are 42 feedforward loops and no feedback loops made of a cycle of three nodes. Nodes that participate in feedforward loops are shown in black in Figure 3.3. In contrast, in the corresponding randomized networks with the same mean connectivity $\lambda = 500/400 \sim 1.2$, there are only about 2 feedforward loops on average (Exercise 3.2)

$$\langle N_{FFL} \rangle_{\text{rand}} = \lambda^3 \sim 1.7$$

and the mean number of feedback loops is smaller than 1,

$$\langle N_{FBL} \rangle_{\text{rand}} = \lambda^3/3 \sim 0.6$$

The standard deviations of these numbers are generally the square roots of the means in such random networks.

We see that the feedforward loop is a strong network motif. It occurs much more often than at random. Its frequency exceeds its frequency in the ensemble of randomized networks by more than 30 standard deviations. In contrast, the three-node feedback loop is not a network motif (it is actually an anti-motif in many biological networks). The same conclusions apply also when comparing transcription networks to more stringent ensembles of randomized networks that closely preserve the properties of the real network.

In fact, in sensory transcription networks such as those of *E. coli* and yeast (Lee et al., 2002; Milo et al., 2002), the feedforward loop is the only significant network motif of the 13 possible three-node patterns. In this sense, these networks are much simpler than they could have been.

The massive overabundance of feedforward loops raises the question: Why are they selected despite randomizing forces? Do they perform a function that confers an advantage to the organism? To address this question, let's now analyze the structure and function of the feedforward loop network motif.

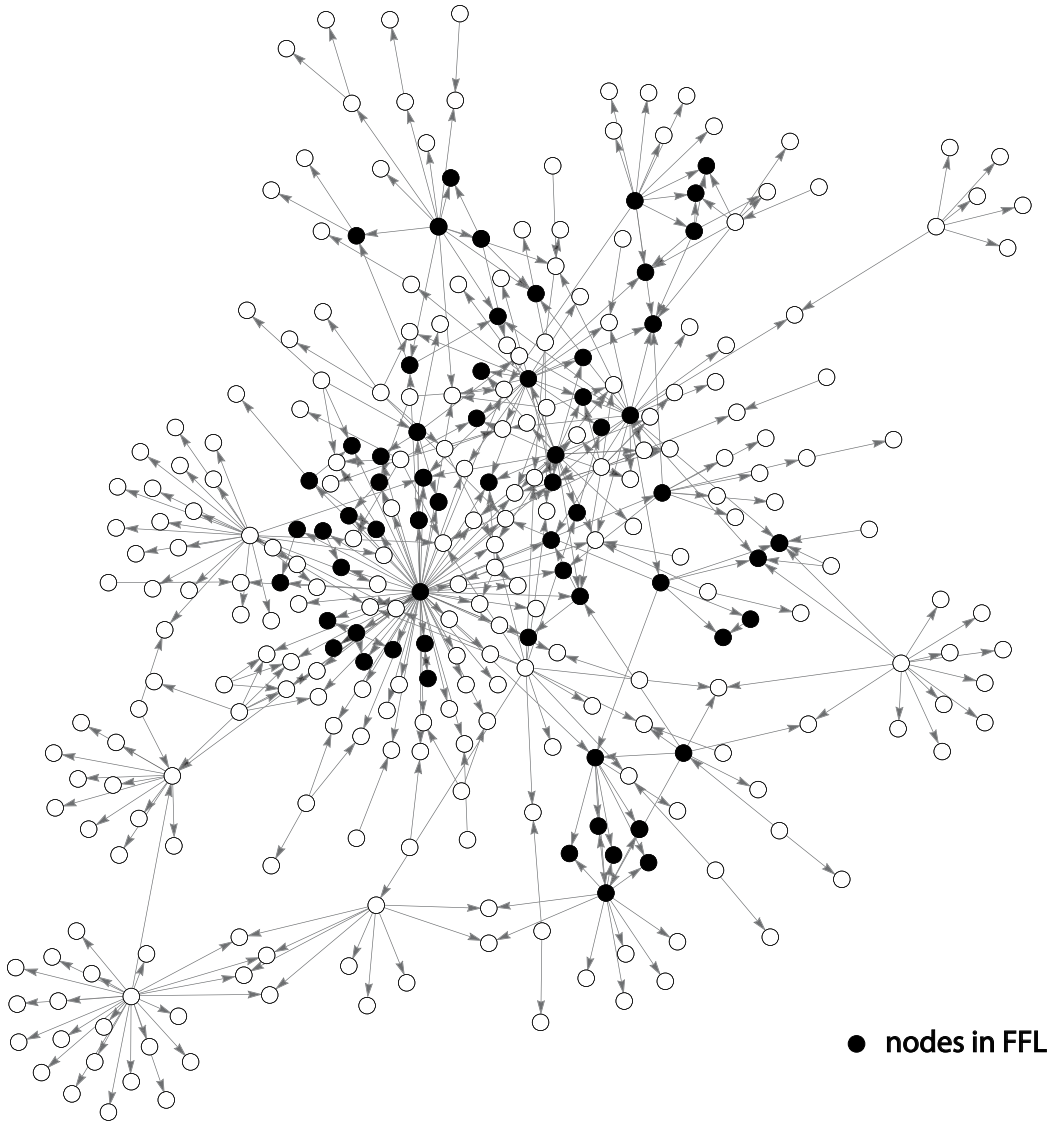


FIGURE 3.3

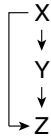
3.3 THE STRUCTURE OF THE FEEDFORWARD LOOP GENE CIRCUIT

The feedforward loop, which we will henceforth abbreviate **FFL**, is composed of transcription factor X that regulates a second transcription factor, Y , and both X and Y regulate gene Z (Figure 3.1). Thus, the FFL has two parallel regulation paths, a direct path from X to Z and an indirect path that goes through Y .

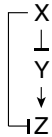
Each of the three arrows in the FFL can correspond to activation (plus sign) or repression (minus sign). There are, therefore, $2^3 = 8$ possible types of FFLs (Figure 3.4).

Coherent FFL

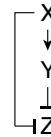
Coherent type 1



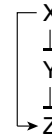
Coherent type 2



Coherent type 3

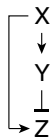


Coherent type 4

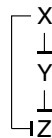


Incoherent FFL

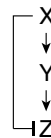
Incoherent type 1



Incoherent type 2



Incoherent type 3



Incoherent type 4

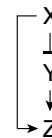


FIGURE 3.4

The eight FFL types can be classified into two groups: coherent and incoherent. In **coherent FFLs**, the indirect path has the same overall sign as the direct path. The overall sign of a path is given by multiplying the signs of the arrows on the path, so that two minus signs give an overall plus sign. For example, in coherent type-1 FFLs, X activates Z , and also activates an activator of Z , so that both paths are positive.

In **incoherent FFLs**, the sign of the indirect path is opposite to that of the direct path. For example, in the incoherent type-1 FFL, the direct path is positive and the indirect path is negative. The two paths have antagonistic effects. Incoherent FFLs have an odd number of minus arrows (one or three), and coherent FFLs have an even number.

Since the *E. coli* network has about equal numbers of positive and negative arrows, one might expect all 8 types of FFLs to appear with about equal frequency. This is not the case. Two types are much more frequent than the others, and together make up more than 80% of the FFLs (Figure 3.5). These two are the coherent type-1 FFL (C1-FFL) and the incoherent type-1 FFL (I1-FFL) (Mangan and Alon, 2003; Ma et al., 2004; Mangan et al., 2006).

In addition to the signs on the arrows, to understand the dynamics of the FFL we must also know how the inputs from the two regulators X and Y are integrated at the promoter of gene Z . That is, we need to know the input function of gene Z . We will consider two biologically reasonable logic

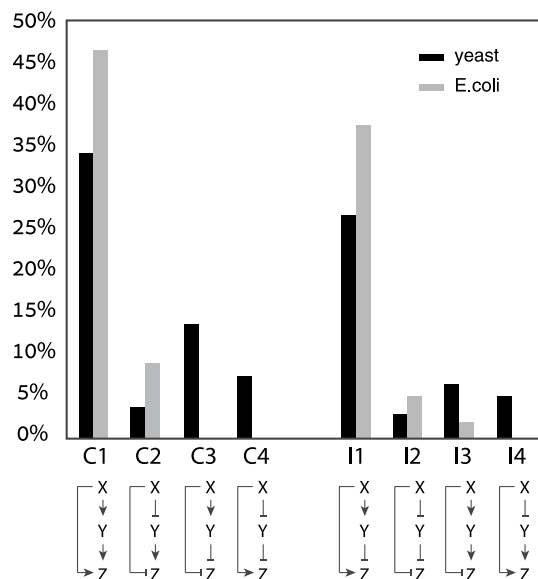


FIGURE 3.5

functions: AND logic, in which both X and Y are needed to turn on Z expression, and OR logic, in which either X or Y is sufficient.

We also need to consider the input signals to this circuit. The transcription factors X and Y in the FFL respond to the signals S_x and S_y . In some cases, these signals are molecules that directly bind the transcription factors, and in other systems the signals are modifications of the transcription factor caused by signal-transduction pathways activated by external stimuli. The effect of the signals usually operates on a much faster timescale than the transcriptional interactions in the FFL.

We can now study the dynamics of the proteins that make up the FFL as a function of time following a change in an external signal. We will begin with the common coherent type-1 FFL (C1-FFL). In this FFL, all three interactions are positive. As for the input function of the Z promoter, we will first consider AND logic, in which both activators X and Y need to bind the promoter of Z in order to initiate the production of protein Z (Figure 3.6).

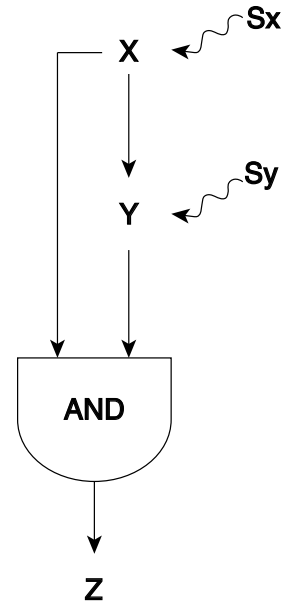


FIGURE 3.6

3.4 DYNAMICS OF THE COHERENT TYPE-1 FFL WITH AND LOGIC

Suppose that the cell expresses numerous copies of protein X , the top transcription factor in the FFL. The input to X is the signal S_x (Figure 3.7). Without the signal, X is in its inactive form. Now, at time $t = 0$, the signal S_x appears and triggers the activation of X . This is known as a **step-like stimulation** of X . As a result, the transcription factor X rapidly transits to its active form X^* . The active protein X^* binds the promoter of gene Y , initiating production of protein Y , the second transcription factor in the FFL. In parallel, other copies of X^* bind the promoter of gene Z . However, since the input function at the Z promoter is AND logic, X^* alone cannot activate Z production.

Production of Z requires binding of both X^* and Y^* . Z activation thus requires that the second input signal, S_y , is present, so that Y is in its active form, Y^* (Figure 3.7). Moreover,

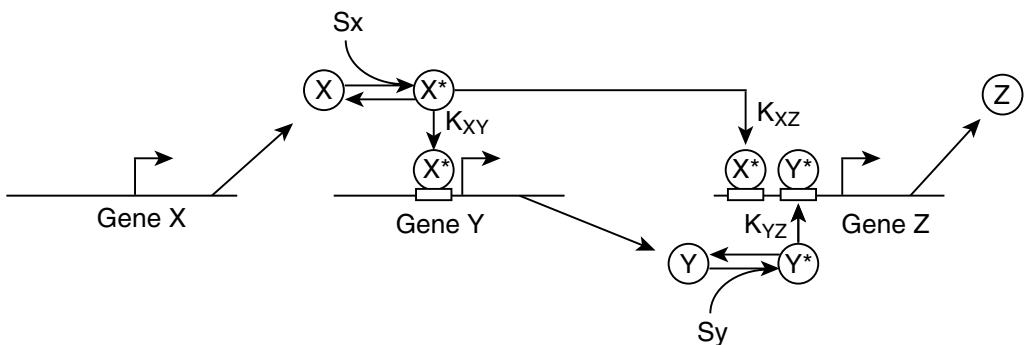


FIGURE 3.7

the concentration of Y^* must build up to sufficient levels to cross the activation threshold for gene Z , denoted K_{YZ} . This results in a delay in Z production.

We will now mathematically describe the FFL dynamics, in order to see how a simple model can be used to gain an intuitive understanding of the function of a gene circuit. We'll use logic input functions. Production of Y occurs at rate β_Y when X^* exceeds the activation threshold K_{XY} , as described by the step function θ :

$$\text{production rate of } Y = \beta_Y \theta(X^* > K_{XY}) \quad (3.4.1)$$

When the signal S_x appears, X rapidly shifts to its active conformation X^* . If the signal is strong enough, X^* exceeds the activation threshold K_{XY} and rapidly binds the Y promoter to activate transcription. Thus, Y production begins shortly after S_x . The accumulation of Y is described by our familiar dynamic equation with a term for production and another term for removal:

$$\frac{dY}{dt} = \beta_Y \theta(X^* > K_{XY}) - \alpha_Y Y \quad (3.4.2)$$

The promoter of Z is governed by an AND-gate input function. The AND gate can be described by a product of two step functions, because both regulators need to cross their activation threshold:

$$\text{production of } Z = \beta_Z \theta(X^* > K_{XZ}) \theta(Y^* > K_{YZ}) \quad (3.4.3)$$

Thus, the C1-FFL gene circuit has three activation thresholds (numbers on the arrows). The dynamics of Z are the balance of a production term with an AND input function and a removal term:

$$\frac{dZ}{dt} = \beta_Z \theta(X^* > K_{XZ}) \theta(Y^* > K_{YZ}) - \alpha_Z Z \quad (3.4.4)$$

We now have the equations needed to study the C1-FFL.

3.5 THE C1-FFL IS A SIGN-SENSITIVE DELAY ELEMENT

To analyze the dynamics of the C1-FFL, we will consider the response to steps of S_x , in which the signal S_x is absent and then saturating S_x suddenly appears (ON steps). We will also consider OFF steps, in which S_x is suddenly removed. For simplicity, we assume throughout that the signal S_y is present, so that the transcription factor Y is in its active form:

$$Y^* = Y \quad (3.5.1)$$

3.5.1 Delay Following an ON Step of S_x

Following an ON step of S_x , Y begins to be produced at rate β_Y . Hence, as we saw in [Chapter 1](#), the concentration of Y begins to exponentially converge to its steady-state level $Y_{st} = \beta_Y/\alpha_Y$ ([Figure 3.8](#)):

$$Y^* = Y_{st}(1 - e^{-\alpha_Y t}) \quad (3.5.2)$$

What about Z ? Production of Z is governed by an AND input function, in which one input, X^* , crosses its threshold as soon as S_x is added. But one input is not enough to activate an AND gate. The second input, Y^* , takes some time to accumulate and to cross the activation threshold, K_{YZ} . Therefore, Z begins to be expressed only after a delay ([Figure 3.8](#)).

The delay, T_{ON} , is the time needed for Y^* to reach its threshold. It can be seen graphically as the time when Y^* concentration crosses the horizontal line at height K_{YZ} . The delay, T_{ON} , can be found using Equation 3.5.2:

$$Y^*(T_{ON}) = Y_{st}(1 - e^{-\alpha_Y T_{ON}}) = K_{YZ} \quad (3.5.3)$$

which can be solved for T_{ON} , yielding:

$$T_{ON} = \frac{1}{\alpha_Y} \log \left(\frac{1}{1 - K_{YZ}/Y_{st}} \right) \quad (3.5.4)$$

This equation describes how the duration of the delay depends on the biochemical parameters of the protein Y ([Figure 3.9](#)). These parameters are the removal rate of the protein, α_Y , and the ratio between its steady-state level Y_{st} and its activation threshold K_{YZ} . The delay can, therefore, be tuned over evolutionary timescales by mutations that change these biochemical parameters.

Note that the delay T_{ON} diverges when the activation threshold K_{YZ} exceeds the steady-state level of Y , because protein Y can never reach its threshold to activate Z ([Figure 3.9](#)). Recall that Y_{st} is prone to

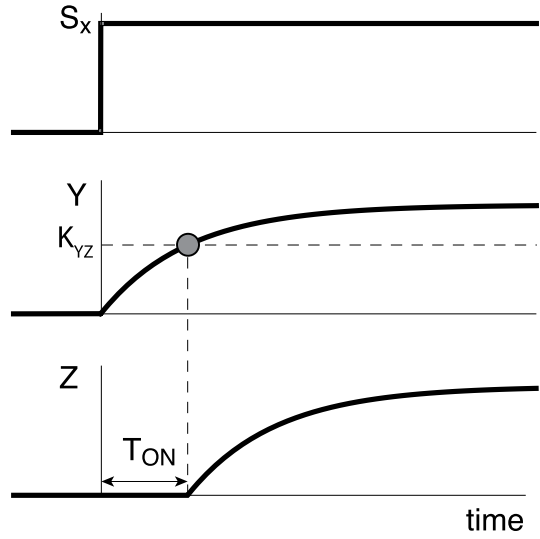


FIGURE 3.8

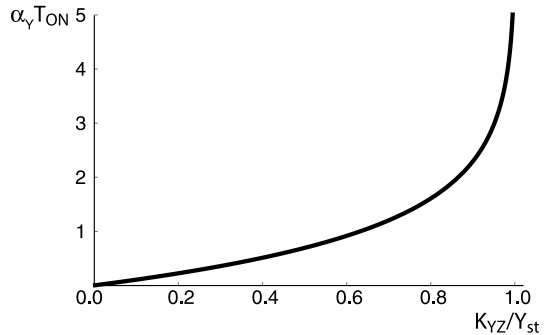


FIGURE 3.9

cell-cell fluctuations due to variations in protein production rates. Hence, a robust design will have a threshold K_{YZ} that is significantly lower than Y_{sp} to avoid these fluctuations. In bacteria, typical parameters provide delays T_{ON} that range from a few minutes to a few hours.

3.5.2 No Delay Following an OFF Step of S_x

We just saw that Z shows a delay following ON steps of S_x . We now consider OFF steps of S_x , in which S_x is suddenly removed (Figure 3.10). Following an OFF step, X becomes inactive and unbinds from the promoters of genes Y and Z . Because Z is governed by an AND gate, it only takes one input to go off in order to stop Z expression. Therefore, after an OFF step of S_x , Z production stops at once. There is no delay in Z dynamics after an OFF step (Figure 3.10).

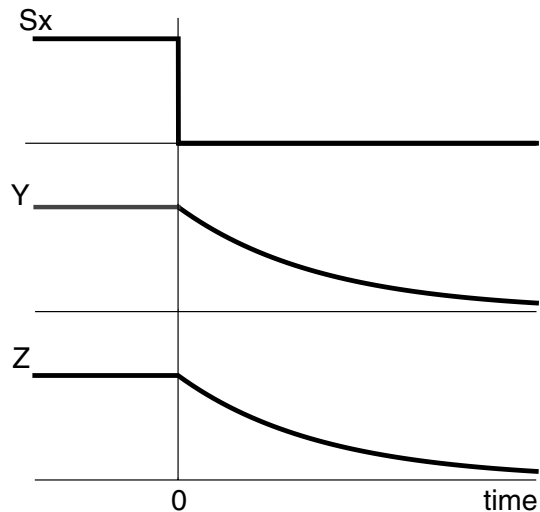


FIGURE 3.10

3.5.3 The C1-FFL Is a Sign-Sensitive Delay Element

The C1-FFL with AND logic shows a delay following ON steps of S_x , but not following OFF steps. This type of behavior is called **sign-sensitive delay**, where sign-sensitive means that the delay depends on the sign of the step, ON or OFF.

A sign-sensitive delay element can also be considered as a kind of asymmetric filter. For example, consider a pulse of S_x that appears only briefly (an ON pulse) (Figure 3.11). An ON pulse that is shorter than the delay time, T_{ON} , does not lead to Z expression in the C1-FFL. That is because Y does not have time to accumulate and cross its activation threshold during the pulse. Only persistent pulses (longer than T_{ON}) result in Z expression. Thus, this type of FFL is a **persistence detector** for ON pulses. On the other hand, it responds immediately to OFF pulses. In contrast to the FFL, simple regulation (with no FFL) does not filter out short input pulses, but rather shows production of Z that lasts as long as the input pulse is present.

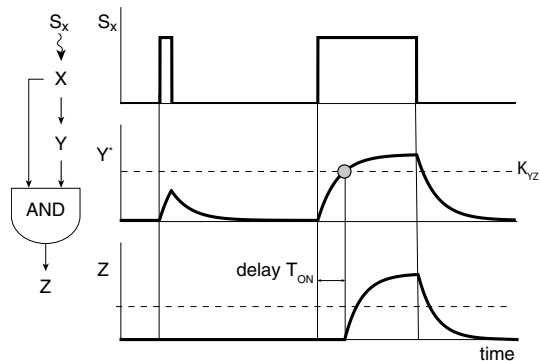


FIGURE 3.11

3.5.4 Sign-Sensitive Delay Can Protect against Brief Input Fluctuations

Why might sign-sensitive delay be useful? For clues, we can turn to the uses of sign-sensitive delays in engineering. Engineers use sign-sensitive delay when the cost of an error is not symmetric. A familiar example occurs in elevators: consider the beam of light used to

sense obstructions in the elevator door. If you obstruct the light with your hand, the doors immediately open. If you remove your hand for only a brief pulse, nothing happens (i.e., a brief pulse of light is filtered out). Only if you remove your hand for a persistent length of time do the doors close (a persistent pulse of light leads to a response). Put your hand back in and the doors open immediately. The cost of an error (doors closing or opening at the wrong time) is asymmetric: the design aims to respond quickly to a person in the beam and make sure that the person has moved away for a persistent period of time before closing the doors. The sign-sensitive delay thus serves a protective function.

In transcription networks, evolutionary selection may have placed the C1-FFL in diverse systems in the cell that require such a protection function. Indeed, the environment of cells is often highly fluctuating, and sometimes stimuli can be present for brief pulses that should not elicit a response. The C1-FFL can offer a filtering function that is advantageous in these types of fluctuating environments. The conditions for the natural selection of the FFL based on its filtering function will be discussed in more detail in [Chapter 13](#).

3.5.5 Sign-Sensitive Delay in the Arabinose System of *E. coli*

Experiments have demonstrated that sign-sensitive delays are carried out by the C1-FFL in cells of different organisms. For example, an FFL was experimentally studied in the well-characterized gene system that allows *E. coli* to grow on the sugar arabinose. The arabinose system consists of proteins that transport arabinose into the cell and break it down for use as an energy and carbon source. Arabinose is only used by the cells when the sugar glucose is not present: glucose is a superior energy source and is used in preference to most other sugars. Thus, the arabinose system needs to make a decision based on two inputs, arabinose and glucose. The proteins in this system are only made when the sugars in the environment of the cell meet the condition: arabinose AND NOT glucose.

The absence of glucose is symbolized within the cell by the production of a small molecule called cAMP. To make its decision, the arabinose system has two transcription activators, one called CRP that senses cAMP, and the other called AraC that senses arabinose. These regulators are connected in a C1-FFL with an AND input function ([Figure 3.12](#)). Thus, the input signals are $S_x = \text{cAMP}$ and $S_y = \text{arabinose}$.

Experiments on this system used steps of S_x and monitored the dynamics of the promoter of the arabinose utilization genes that act as node Z in the FFL, using green fluorescent protein (GFP) as a reporter. A delay was found after ON steps of S_x , but not after OFF steps ([Figure 3.12](#)). The delay following ON steps is $T_{\text{ON}} \sim 20$ min under the conditions of the experiment. In contrast, a simple-regulation system (with no FFL) that responds to the same input (the *lac* promoter) showed no delay for either ON or OFF steps, beyond the ~ 5 min needed to transcribe and translate the GFP reporter.

The observed delay in the arabinose FFL is on the same order of magnitude as the duration of spurious pulses of the input signal S_x in the environment of *E. coli*. These spurious pulses occur when *E. coli* transits between different growth conditions. Thus, the FFL in this system seems to have “learned” the typical timescale of short fluctuations in the input signal and can filter them out. It responds only to persistent stimuli, such as persistent periods of glucose starvation that require utilization of the sugar arabinose.

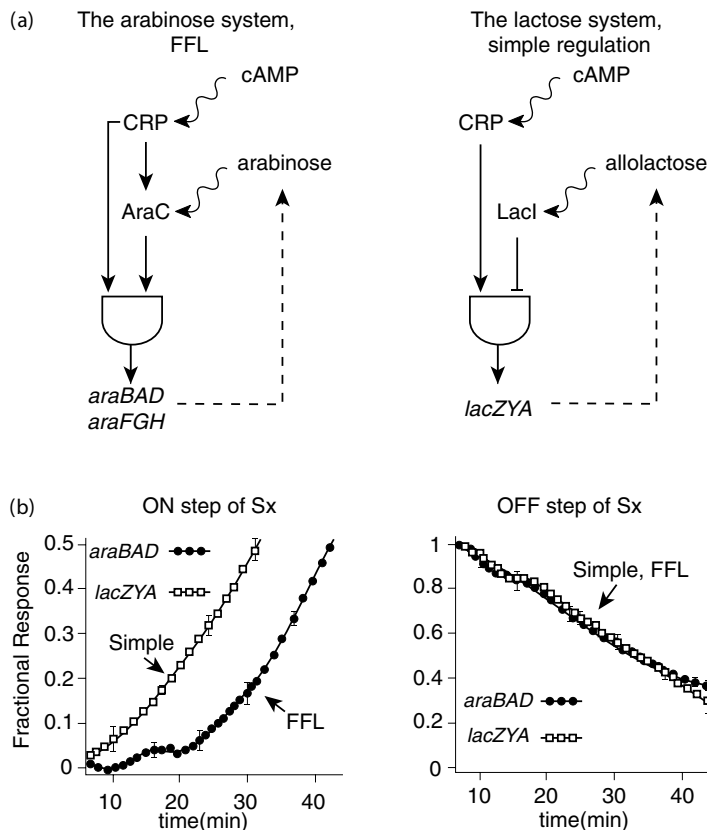


FIGURE 3.12 Adapted from (Mangan, Zaslaver and Alon 2003).

There is an asymmetry in this system: starting to make the arabinose utilization proteins when not needed uses up resources and carries a fitness cost (bacteria grow more slowly). In contrast, stopping protein production too soon is not as bad because the proteins are still around and can perform their function, if needed, for a generation or so, until they are diluted away by cell growth.

Note that the FFL in the arabinose system shows sign-sensitive delay despite the fact that this circuit is embedded in additional interactions, such as protein-level feedback loops (dashed lines in Figure 3.12, that represent the change in input signal due to transport of the sugar into the cell by the pumps, and its degradation by the enzymes of each system). Thus, although the theory we have discussed concerns a three-gene FFL circuit in isolation, the arabinose FFL shows the expected dynamics also when connected to the rest of the interaction networks of the cell. This modularity of function is generally found in experiments on network motifs (Atay et al. 2016; Bulcha et al. 2019).

3.6 OR-GATE C1-FFL IS A SIGN-SENSITIVE DELAY FOR OFF STEPS

What happens if the C1-FFL has an OR gate at the Z promoter instead of an AND gate? With an OR gate, Z is activated immediately upon an ON step of S_x , because it only takes one input

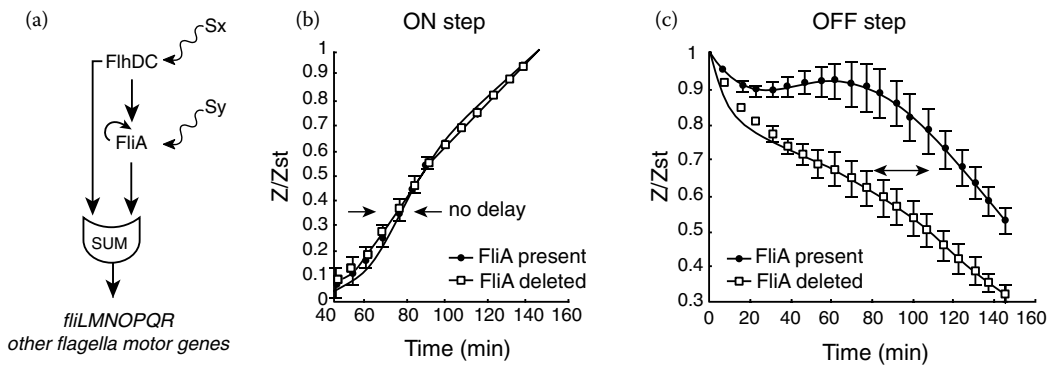


FIGURE 3.13 Adapted from (Kalir, Mangan and Alon 2005).

to activate an OR gate. Thus, there is no delay following an ON step of S_x . In contrast, Z is deactivated at a delay following an OFF step, because both inputs need to go off for the OR gate to be inactivated, and it takes time for Y^* to decay away after an OFF step of S_x . Thus, the C1-FFL with an OR gate is also a sign-sensitive delay element, but with signs opposite to those of the AND version (the delay is calculated in solved Exercise 3.7). As a result, the OR gate C1-FFL can maintain expression of Z even if the input signal is momentarily lost.

Such dynamics were demonstrated experimentally in the flagella system of *E. coli* (Figure 3.13; Kalir, Mangan and Alon, 2005). This FFL controls the production of proteins that self-assemble into a motor that rotates the flagella that allow *E. coli* to swim. We will discuss this system in more detail in Chapter 9. The delay in this FFL after removal of S_x is about one cell generation time, which is about 1h under the conditions of the experiment. This delay is on the same order of magnitude as the time it takes to assemble a flagella motor. The OR gate FFL provides continued expression for about an hour after the input signal goes off and can thus protect this gene system against transient loss of input signal.

3.7 THE INCOHERENT TYPE-1 FFL GENERATES PULSES OF OUTPUT

We can use the same approach to understand the function of the other prevalent type of FFL, the incoherent type-1 FFL (I1-FFL). In this circuit, X activates gene Z , and also activates Y which represses Z (Figure 3.14). Let's consider the case in which Y represses Z strongly. After an ON step of input signal S_x , X binds the promoters of Y and Z . As a result, Z concentration rises. The repressor Y also accumulates, and when it crosses its repression threshold, K_{YZ} , it shuts off Z production and Z levels begin to drop.

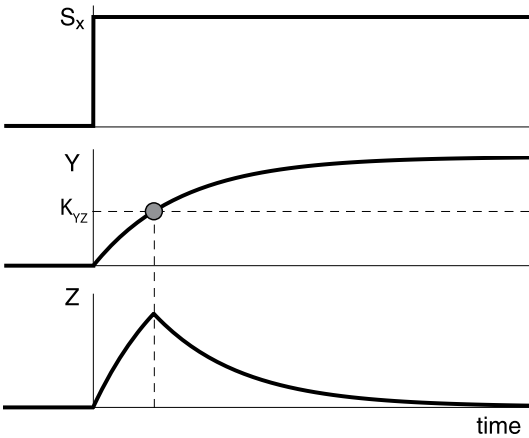


FIGURE 3.14

Thus, the I1-FFL generates a pulse of output in response to a step of input (Figure 3.14).

Such a pulse is seen, for example, in the system that signals mammalian cells to divide in response to the proliferation signal EGF (Amit et al., 2007). In response to a step stimulation of EGF, an I1-FFL generates a pulse of early response genes (Z_1 in Figure 3.15). Interestingly, the I1-FFL is linked with a C1-FFL that causes other genes to turn on at a delay relative to the early genes (Z_2 in Figure 3.15). The temporal pulse generated by the I1-FFL in response to a step input can be thought of as being similar to a temporal derivative of the input signal. In fact, the I1-FFL can respond to relative changes in input, a property that we will explore in Chapter 10.

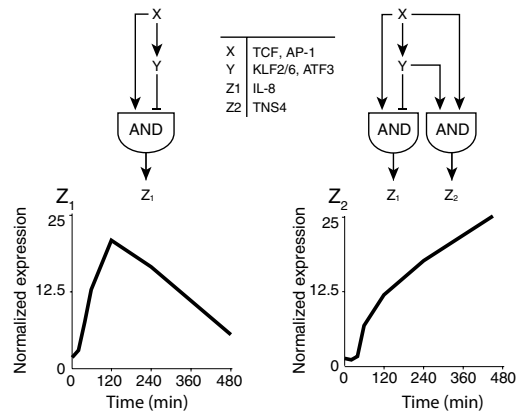


FIGURE 3.15 Adapted from (Amit et al. 2007).

3.7.1 The Incoherent FFL Can Speed Response Times

A pulse is found when Y strongly represses Z . If Y only partially represses Z , the pulse does not go down to baseline but instead converges to a steady state, Z_{st} , determined by the relative strengths of the activation by X and the repression by Y . In this case, the I1-FFL can be compared to simple regulation that reaches the same Z_{st} . The I1-FFL speeds responses relative to simple regulation (Figure 3.16). This is because it allows a rapid rise (when Y is still low), which is stopped by Y to reach the desired Z_{st} . The principle is similar to the strong motor and strong breaks we saw for negative autoregulation. The speedup is greater the stronger the repression by Y , as calculated in solved Exercise 3.13.

An experimental study of I1-FFL speedup is shown in Figure 3.17. This experiment employed the system which enables *E. coli* to grow on the sugar galactose as a carbon and energy source. As in other sugar systems, the genes in the galactose system are not highly expressed in the presence of glucose, a superior

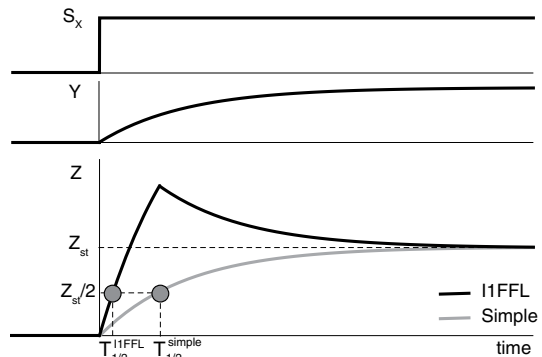


FIGURE 3.16

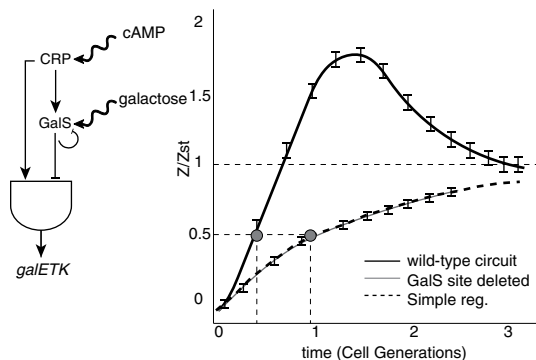


FIGURE 3.17 Adapted from (Mangan et al. 2006).

energy source. The galactose utilization genes are expressed at a low but significant level when both glucose is absent and galactose is absent, allowing the cell to grow rapidly on galactose should it appear in the environment. When galactose appears, the genes become fully expressed. The galactose genes are regulated by an I1-FFL, with the activator CRP and the repressor GalS.

Measurements show that the response of the output genes is accelerated upon glucose starvation (an ON step of S_x) compared to simply regulated genes (Figure 3.17). Removal of the repressor interaction in the I1-FFL, achieved by deleting the repressor binding site in the promoter of Z , abolishes this acceleration.

In addition to studying this network motif within a natural context, one can study it by making a synthetic I1-FFL made of well-characterized regulators. Weiss and colleagues constructed an I1-FFL using the activator LuxR as X , the repressor C1 of phage lambda as Y and green fluorescent protein as the output gene Z (Basu et al., 2005). This **synthetic circuit** in *E. coli* showed pulse-like responses to steps of the input signal S_x (the inducer of LuxR). The synthetic construction of gene circuits is a powerful approach for isolating and studying their properties (Elowitz and Lim, 2010).

3.7.2 Interim Summary: Three Ways to Speed Your Response Time

We have seen three different ways to speed the response time of transcription networks. The basic problem is that the transcriptional response time is slow, on the order of a cell generation time for proteins that are not degraded. This is a drawback for networks that need to respond rapidly to external signals. The three ways to speed response times are:

1. *Increased degradation rate*: As we saw in Chapter 1, the response time of simple gene regulation is inversely proportional to the removal rate, $T_{1/2} = \log(2)/\alpha$, where α is a sum of the rate of degradation of the protein and the rate of dilution by cell growth: $\alpha = \alpha_{deg} + \alpha_{dil}$. Therefore, increasing the degradation rate α_{deg} yields faster responses. However, there is a cost to this strategy: to maintain a given steady state, $X_{st} = \beta/\alpha$, one needs to increase the production rate β to balance the effects of increased degradation α . This creates a futile cycle, where the protein is rapidly produced and rapidly degraded. This cycle can be selected by evolution in some systems, despite the costs of increased production, due to the benefit of faster response.
2. *Negative autoregulation*: As we saw in Chapter 2, negative autoregulation can speed responses by a large factor. This speedup is due to the ability to use a strong promoter (large production rate β) to give rapid initial production, and then to turn production off by self-repression when the desired steady state is reached. The negative autoregulation strategy works only for proteins that can repress themselves, namely, only for transcription factor proteins.
3. *Incoherent FFL*: The incoherent type-1 FFL can speed up ON-responses, as we saw in the previous section. This is due to initially rapid production that is later turned off by a delayed repressor, to achieve a desired steady state. This speedup applies to the low-induction state in the presence of S_y . It can be used to speed the response time of any target protein, not only transcription factors.

Designs 2 and 3 can work together with 1: a large degradation rate further speeds the response of negative autoregulation and incoherent FFLs.

3.7.3 The I1-FFL Can Provide Biphasic Steady-State Response Curves

So far, we considered the dynamical response of the I1-FFL to steps of input signal. An additional feature is seen when we consider the steady-state response, Z_{st} , as a function of the input signal strength S_x . Simple regulation $X \rightarrow Z$ generates monotonic response functions, in which the output Z_{st} rises with the input S_x . The I1-FFL can generate a **non-monotonic** response in which Z_{st} first rises with S_x and then, at high levels, decreases, resulting in an inverse U-shape response curve. Because the response has a rising phase and a decreasing phase, it is called a **biphasic response**.

Biphasic responses arise in the I1-FFL when X activates Z at a lower threshold than it activates Y . Thus, when S_x is low, the repressor Y is not activated and only the $X \rightarrow Z$ arm is felt, resulting in a rising response curve. At high signal levels, the repressor is produced and Z_{st} drops with signal.

A biphasic response was studied experimentally in the galactose system of *E. coli* by Shai Kaplan et al. (Kaplan et al., 2008; Figure 3.18).

This biphasic response is due to the I1-FFL in the system, because deleting the repressor gene Y (*galS*) made the response curve monotonic (dashed line in Figure 3.18).

A combination of biphasic response to input dose and a temporal pulse is seen in experiments by Sheng-Hong Chen and Keith Yamamoto (Chen et al., 2013) on certain genes regulated by the glucocorticoid receptor. This receptor responds to cortisol, a central hormone regulator of the mammalian stress response. Cortisol is well known to have different effects depending on whether it is

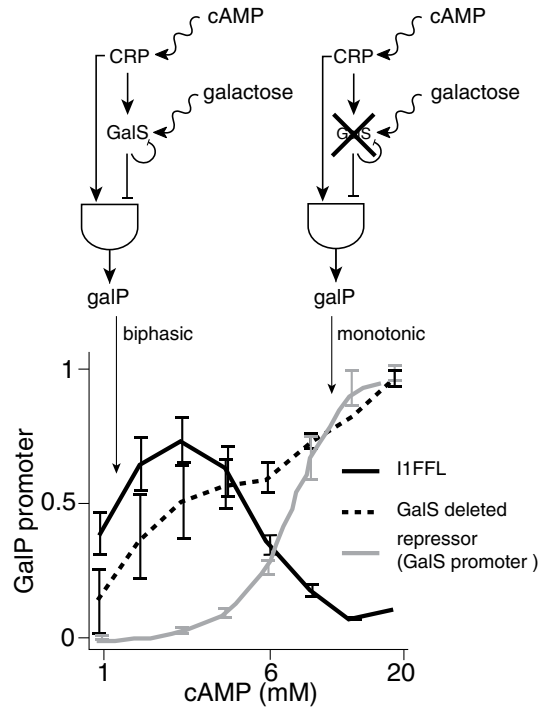


FIGURE 3.18 Adapted from (Kaplan et al. 2008).

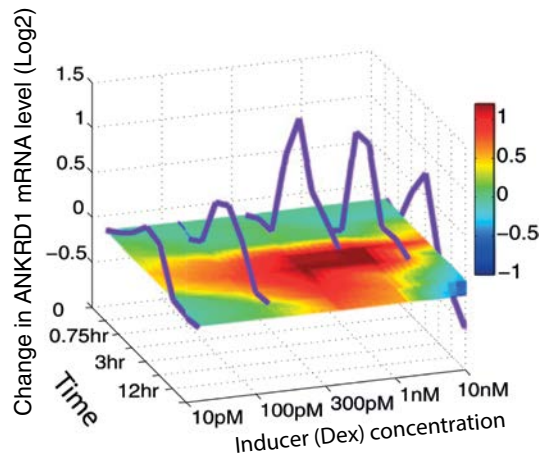


FIGURE 3.19 Adapted from (Chen et al. 2013).

TABLE 3.1 Response of Coherent FFLs to S_x in the Presence of S_y

Coherent FFL Type	1		2		3		4	
	AND	OR	AND	OR	AND	OR	AND	OR
ON step	delay	–	–	delay	–	–	delay	delay
OFF step	–	delay	delay	–	delay	delay	–	–

TABLE 3.2 Response of Incoherent FFLs to S_x in the Presence of S_y

Incoherent FFL Type	1		2		3		4	
ON step	pulse/accelerate		–		–		pulse/accelerate	
OFF step	–		pulse/accelerate		pulse/accelerate		–	
Biphasic	inverse U-shape		U-shape		U-shape		inverse U-shape	

present for a short or a long time (acute versus chronic stress). Cortisol concentration also has a biphasic effect, with high and low levels both causing unwanted physiological effects, with an optimum at intermediate levels. The studied output genes show pulses in time after a step of input (dexamethasone, an analog of cortisol). They also show a biphasic dose-response, with an amplitude that peaks at intermediate input dose (Figure 3.19). Based on this behavior, the researchers hypothesized that an I1-FFL is at play. Indeed, I1-FFLs were later experimentally identified in the cortisol system (Sass et al., 2013; Chinenov et al., 2014).

3.8 THE OTHER SIX FFL TYPES CAN ALSO ACT AS FILTERS AND PULSE GENERATORS

Using the approach we discussed in this chapter, you can figure out what the other six types of FFLs can do. Such analysis shows that all four types of coherent FFLs can act as sign sensitive delays (Table 3.1), with delays after ON or OFF steps of input. All four types of incoherent FFLs can generate pulses and biphasic responses (Table 3.2). The details vary, though. For example, an incoherent type-3 FFL can show a biphasic response which first drops with input signal and then rises, giving a U-shape rather than an inverse U-shape biphasic response.

Because all FFLs can have potentially useful functions, it is an open question why two types, C1-FFL and I1-FFL, appear much more often than the other six. Some clues are considered in Exercises 3.12 and 3.19.

3.9 CONVERGENT EVOLUTION OF FFLs

We end this chapter by thinking about the evolution of FFLs. How did FFLs evolve? The most common form of evolution for genes is **conservative evolution**, where two genes with similar function stem from a common ancestor gene. The genes, therefore, share a significant degree of DNA sequence similarity. Such genes are said to be **homologous**.

Did FFLs evolve in a similar way, where an ancestor FFL duplicated to gave rise to the present FFLs? It appears that the answer is no in most cases. For example, homologous genes Z and Z' in two species are often both regulated by FFLs in response to the same

environmental stimuli. If the two FFLs shared a common ancestor FFL, the regulators X and Y in the two FFLs would also be homologous. However, the regulators are usually not homologous in such FFL pairs. The DNA sequence of the regulators is so dissimilar that they belong to completely different transcription factor families.

Thus, evolution **converged** independently on the same regulation circuit (Conant and Wagner, 2003; Babu et al., 2004). Presumably, the FFL is rediscovered by evolution because it performs an important function in the different organisms. More about gene circuit evolution will be discussed in [Chapter 13](#).

3.10 SUMMARY

Of the 13 three-gene patterns, only the FFL is a network motif in the transcription networks of *E. coli* and yeast as well as in the sensory transcription networks of higher organisms. Of the eight possible sign combinations of the FFL, two are most commonly found, the C1-FFL in which all interactions are positive, and the I1-FFL in which an activator activates both a gene and its repressor. The C1-FFL acts as a persistence detector, filtering out brief pulses of input. The incoherent type-1 FFL (I1-FFL) can act as a pulse generator in time and in input dose, and a response accelerator. This acceleration can be used in conjunction with the other mechanisms of acceleration, such as increased degradation and negative autoregulation. Evolution converged again and again on the FFLs in different gene systems and in different organisms. Thus, this recurring network motif is an example of a pattern that may have been selected for its specific dynamical functions. As we will see in the next chapters, the FFL is also a network motif in several other types of biological networks.

FURTHER READING

Convergent Evolution of Network Motifs

(Babu et al., 2004) “Structure and evolution of transcriptional regulation networks.”

(Conant and Wagner, 2003) “Convergent evolution of gene circuits.”

Experimental Study of the Dynamics of Coherent FFLs

(Kalir, Mangan and Alon, 2005) “A coherent feedforward loop with a SUM input function protects flagella production in *Escherichia coli*.”

(Mangan, Zaslaver and Alon, 2003) “The coherent feedforward loop serves as a sign-sensitive delay element in transcription networks.”

Experimental Study of the Dynamics of Incoherent FFLs

(Amit et al., 2007) “A module of negative feedback regulators defines growth factor signaling.”

(Bleris et al., 2011) “Synthetic incoherent feedforward circuits show adaptation to the amount of their genetic template.”

(Basu et al., 2004) “Spatiotemporal control of gene expression with pulse-generating networks.”

(Chen et al., 2013) “Incoherent feed-forward regulatory logic underpinning glucocorticoid receptor action.”

(Kaplan et al., 2008) “The incoherent feed-forward loop can generate non-monotonic input functions for genes.”

(Mangan et al., 2006) “The incoherent feedforward loop accelerates the response time of the Gal system of *Escherichia coli*.”

(Sasse and Gerber, 2015) “Feedforward transcriptional programming by nuclear receptors: regulatory principles and therapeutic implications.”

Feedforward Loop Network Motif

(Mangan and Alon, 2003) “Structure and function of the feedforward loop network motif.”

(Shen-Orr et al., 2002) “Network motifs in the transcriptional regulation network of *Escherichia coli*.”

Synthetic Circuits

(Elowitz and Lim, 2010) “Build life to understand it.”

EXERCISES

Subgraphs in Random Networks

3.1 *Sparseness of biological networks*: A sparse network is a network in which the density of arrows is low. To be specific, we will consider an Erdős–Rényi (ER) network as a basic model for random networks. In an ER network, A arrows are placed at random between N nodes.

- a. How many places are there to put an arrow, including self-arrows?

Answer: $N(N - 1)/2$ pairs, each arrow can be in two directions, plus N self-arrows makes N^2 places.

- b. Sparseness p is the number of actual arrows divided by the number of places arrows could have been. What is the sparseness of our example *E. coli* network with $A = 500$ and $N = 400$?

Answer: $p = A/N^2 = 0.002$. Low values of p are typical of biological networks.

- c. What is the difference between sparseness and mean connectivity $\lambda = A/N$?

3.2 *Subgraphs in random networks*: A subgraph is a pattern of nodes and arrows, found inside a network.

- a. Compute the expected number of subgraphs G with n nodes and g arrows, $\langle N_G \rangle$, in an ER network of size N and mean connectivity $\lambda = A/N$.

Solution:

There are approximately N^n ways of choosing n specific nodes (N for the first node, $N - 1$ for the second node, etc., leading to $N(N - 1) \dots (N - n) \sim N^n$ provided we are dealing with subgraphs that are small compared to the network size, $n \ll N$). The probability for finding g arrows between these nodes is p^g for sparse networks, where p is the sparseness defined in Exercise 3.1. Thus, $N_G = a^{-1} N^n p^g = a^{-1} N^{(n-g)} \lambda^g$. The factor a is a symmetry factor equal to the number of permutations of the nodes that leave the graph unchanged. For a feedback loop of three nodes, $a = 3$, and for the FFL $a = 1$.

- b. How many FFLs and three-node feedback loops are there on average in an ER network?

Solution:

Both subgraphs have $n = g$ and thus scale as N^0 , so for large networks their numbers do not depend on network size:

$$\langle N_{FFL} \rangle = \lambda^3, \quad \langle N_{FBL} \rangle = \frac{1}{3} \lambda^3.$$

- c. How many fan-in subgraphs of two nodes regulating a third?

Solution:

$n = 3, g = 2$ and $a = 2$ because the two regulating nodes can be interchanged, so that $\langle N_V \rangle = 1/2 N \lambda^2$. The number scales with the size of the network N .

- d. How many cliques? Cliques are fully connected triads, three nodes with all possible arrows between them.

Solution:

$n = 3, g = 6, a = 6$ (all six permutations of the three nodes are equivalent), so $\langle N_{clique} \rangle = 1/6 N^{-3} \lambda^6$. These subgraphs are very rare in large, sparse random networks.

3.3 Algorithm for detecting network motifs:

- Write a computer program to count FFLs and three-node feedback loops (FBLs) in a given network. The input is a list of pairs of numbers, indicating the origin and target of each arrow. Use as an input the *E. coli* example network in this book website: <https://www.weizmann.ac.il/mcb/UriAlon/e-coli-transcription-network>.
- Write a computer program to generate random ER networks with a given number of nodes and arrows.
- Use (a) and (b) to write a computer program that computes the mean and standard deviation of the FFL and FBL in ER networks, and outputs whether they are network motifs (choose a suitable statistical criterion for their over-abundance), or anti-motifs (under-abundance).

3.4 Hubs in networks: Hubs are nodes connected with many more arrows than average nodes. They are often central regulators in transcription networks.

- Write a computer program to calculate the number of incoming and outgoing arrows to each node in the *E. coli* example network (see Exercise 3.3). Plot the distributions of incoming and outgoing arrows. Which distribution has a longer tail (more probability for values higher than average)?

- b. Which nodes are hubs in terms of outgoing arrows?
 - c. Are there strong hubs in terms of incoming arrows? If not, explain why this might be.
 - d. Do ER networks have hubs? Compute (analytically or by computer simulation) the distribution of number of arrows per node in ER networks, and compare to the real network.
- 3.5 *Degree-preserving random networks:* Each node in a network has an in-degree, the number of incoming arrows, and an out-degree, the number of outgoing arrows. Degree-preserving random networks (DPRNs) have the same number of nodes and arrows as the real network, and also preserve the in-degree and out-degree of each node in the network. Sounds hard to achieve?
- a. The switching algorithm (Maslov and Sneppen, 2002; Milo et al., 2002) to generate DPRN is as follows: Start with the real network. Choose a pair of arrows at random, and switch their target nodes. Thus, arrows $x_1 \rightarrow x_2$ and $x_3 \rightarrow x_4$ are switched to $x_1 \rightarrow x_4$ and $x_3 \rightarrow x_2$. Repeat until many changes/arrows have been done. Explain why this algorithm preserves the degrees of the nodes.
 - b. Write a computer program that implements the switching algorithm. Run it on the example *E. coli* network (Exercise 3.3) to generate 1000 DPRNs.
 - c. Compute the average numbers of FFLs and FBLs in the DPRNs. How does this compare to the average number of these subgraphs in ER networks?
 - d. Is the FFL a network motif compared to DPRN? What about the FBL?
 - e. Explain, using the concept of hubs, why DPRNs can be more realistic network models than ER networks.

Coherent FFL

- 3.6 *The second input:* What is the effect of steps of S_y on the expression dynamics of Z in the C1-FFL with AND logic? Are there delays in Z expression for ON or OFF steps of S_y ? What is the response time of Z for such steps? Assume that S_x is present throughout.
- 3.7 *OR-gate logic:* Analyze the C1-FFL with OR logic at the Z promoter. What is the length of the delay following ON and OFF steps of S_x ? What could be the biological use of such a design?

Solution:

After an ON step of S_x , X becomes active X^* . On a rapid timescale it binds the Z promoter. Since Z is regulated by OR logic, X^* alone can activate transcription without need for Y^* . Therefore, there are no delays following an ON step of S_x beyond the times for transcription and translation.

After an OFF step of S_x , X^* rapidly becomes inactive, X . However, protein Y is still present in the cell, and if S_y is present, Y is active, Y^* . Since the Z input function is an OR gate, Y^* continues to activate transcription of Z even in the absence of X^* . Therefore, Z production persists until Y degrades/dilutes below its activation threshold for Z . The dynamics of Y^* are given by $dY^*/dt = -\alpha Y^*$ (there is no production term because X is inactive following the removal of S_x), so that $Y^* = Y_m^* e^{-\alpha t}$, where Y_m^* is the level of Y^* at time $t = 0$. The OFF delay is given by the time it takes Y^* to reach its activation threshold for Z , K_{YZ} . Solving for this time, $Y^*(T_{\text{OFF}}) = Y_m^* e^{-\alpha t} = K_{YZ}$, yields

$$T_{\text{OFF}} = \frac{1}{\alpha} \log \left(\frac{Y_m^*}{K_{YZ}} \right).$$

The OR gate C1-FFL could be useful in systems that need to be protected from sudden loss of activity of their master regulator X . The OR gate FFL can provide continued production during brief fluctuations in which X activity is lost. This protection works for OFF pulses shorter than T_{OFF} . Note that T_{OFF} can be tuned by evolutionary selection by adjusting the biochemical parameters of protein Y , such as its expression level and its activation threshold.

- 3.8 *A motif within a motif*: The regulator Y in C1-FFLs in transcription networks is often negatively autoregulated. How does this affect the dynamics of the circuit, assuming that it has an AND input function at the Z promoter? How does it affect the delay times? The Y regulator in an OR gate C1-FFL is often positively autoregulated. How does this affect the dynamics of the circuit? How does it affect the delay times?
- 3.9 *The diamond*: The four-node diamond pattern occurs when X regulates Y and Z , and both Y and Z regulate gene W .
- How does the mean number of diamonds scale with network size in random ER networks (see Exercise 3.2)?
 - What are the distinct types of sign combinations of the diamond (where each arrow is either activation $+$ or repression $-$)? How many of these are coherent? (Answer: 10 types, of which 6 are coherent.)
 - Consider a diamond with four activation arrows. Assign activation thresholds to all arrows. Analyze the dynamics of W following a step of S_x , for both AND and OR logic at the W promoter. Are there sign-sensitive delays?
- 3.10 *Type-3*: Solve the dynamics of the coherent type-3 FFL (Figure 3.4) with AND logic at the Z promoter in response to steps of S_x . Here, AND logic means that Z is produced if both X^* and Y^* do not bind the promoter. Are there delays? Compare to the other coherent FFL types.
- 3.11 *All coherent types*: Write a computer program that computes the dynamics of all four coherent FFL types. Use logic input functions, and set all parameters α , β to 1. Use K parameters equal to 0.1.

- a. Use AND logic at the Z promoter, and plot the response to ON and OFF steps in S_x in which S_x changes between 0 and 1. Assume $S_y = 1$ throughout.
 - b. Repeat using OR logic.
 - c. Compare the results to [Table 3.1](#).
- 3.12 *Steady-state response*: Analyze the steady-state output Z of all four coherent FFL types with AND and OR gates as a function of inputs. Each of the two inputs S_x and S_y can have one of two values, either zero or saturating (giving maximal activity). This gives four possible input combinations for each circuit.
- a. Are there circuits which do not respond to one of the inputs? Explain.
 - b. Can the answer to (a) partially explain why some FFL types are more common than others?

Incoherent FFLs

- 3.13 *Speedup for I1-FFL*: Consider an I1-FFL in which production of Z is β_1 if $Y^* < K$. Production goes down to β_2 when $Y^* \geq K$. What is the speedup compared to simple regulation with production rate β_2 ? Assume equal removal rate for the two circuits.

Solution:

After an ON step of input, the I1-FFL starts out with $Z \approx \beta_1 t$ and reaches halfway to its steady state $Z_{st} = \beta_2/\alpha$ at time $T_{1/2}^{I1FFL} = \frac{\beta_2}{2\beta_1\alpha}$. Simple regulation has $T_{1/2}^{simple} = \log(2)/\alpha$. Thus, speedup is

$$T_{1/2}^{simple}/T_{1/2}^{I1FFL} = 2\log(2)\frac{\beta_1}{\beta_2}.$$

The larger the repression of Z by Y , the bigger the speedup.

- 3.14 *Shaping the pulse*: Consider a situation where X in an I1-FFL gradually increases as $X(t) = \beta t$. The input signals S_x and S_y are present throughout.
- a. How does the pulse shape generated by the I1-FFL depend on the thresholds K_{XZ} , K_{XY} and K_{YZ} , and on β , the production rate of protein X ?
 - b. Analyze a set of genes $Z_1, Z_2, \dots, Z_n$, all regulated by the same X and Y in I1-FFLs. Design thresholds such that the genes are turned ON in the rising phase of the pulse in a certain temporal order and turned OFF in the declining phase of the pulse with the same order.
 - c. Design thresholds such that the turn-OFF order is opposite to the turn-ON order. Plot the resulting dynamics.
- 3.15 *Biphasic responses*: This problem analyzes the biphasic feature of incoherent type-1 FFLs for sub-saturating stimuli S_x . Consider an I1FFL, such that the activation threshold of Z by X , K_{ZX} , is smaller than the activation threshold of Y by X , K_{YX} . That

is, Z is activated when $X^* > K_{ZX}$ but it is repressed by Y when $X^* > K_{YX}$. Schematically plot the steady-state concentration of Z as a function of X^* . Note that intermediate values of X^* lead to the highest Z expression.

- 3.16 *The diamond again*: The diamond pattern occurs when X regulates Y and Z , and both Y and Z regulate gene W . Analyze the 10 types of diamond structures (where each arrow is either activation $+$ or repression $-$) with respect to their steady-state responses to the inputs S_x , S_y and S_z . Use an AND input function at the W promoter. Do any diamond types lack responsiveness to any input? To all three inputs?
- 3.17 *Repressilator*: Three repressors are hooked up in a cycle $X \dashv Y \dashv Z$ and $Z \dashv X$. What are the resulting dynamics? Use initial conditions in which X is high and $Y = Z = 0$. Solve graphically using logic input functions. This circuit was constructed in bacteria using three well-studied repressors, one of which was also made to repress the gene for green fluorescent protein (Elowitz and Leibler, 2000). What would the resulting bacteria look like under a microscope that dynamically records green fluorescence?
- 3.18 *Interconnected FFLs*: Consider a coherent type-1 FFL with nodes X , Y and Z , which is linked to another coherent type-1 FFL in which Y activates Y_1 , which activates Z .
 - a. Sketch the dynamics of Z expression in response to steps of the signals S_x, S_y, S_{y1} , in which one of the signals goes ON or OFF in the presence of the other signals. Can the dynamics of the interconnected circuit be understood based on the qualitative behavior of each FFL in isolation?
 - b. Repeat for the case where Y represses Z , so that the X, Y, Z FFL is an incoherent type-1 FFL. Assume that Y_1 binding to the Z promoter can alleviate the repressing effect of Y .
- 3.19 *Rare IFFL types*: Compare the I1-FFL to the much less common I2-FFL, I3-FFL and I4-FFL.
 - a. Show that the I1-FFL and I4-FFL can generate a pulse and can accelerate responses to ON steps.
 - b. Show that the I2-FFL and I3-FFL can generate a pulse and speed responses to OFF steps.
 - c. Show that at steady state, for saturating S_x , the output Z_{st} responds to the Y signal S_y in the I1-FFL, but not in the I4-FFL.
 - d. Explain why the loss of response to S_y might affect the selection of the I4-FFL.
- 3.20 *Incoherent FFL with microRNA*: Bleris (Bleris et al., 2011) constructed an IFFL in which Y is a microRNA that binds the RNA of Z and inhibits it. They varied the number of DNA molecules that encode this construct in the cell, and found that the output Z was insensitive to the number of DNA copies. Explain this result.

BIBLIOGRAPHY

- Amit, I. et al. 2007. 'A module of negative feedback regulators defines growth factor signaling', *Nature Genetics*, 39(4), pp. 503–512. doi: 10.1038/ng1987.
- Atay, O., Doncic, A. and Skotheim, J.M. 2016. 'Switch-like transitions insulate network motifs to modularize biological networks', *Cell Systems*, 3(2), pp. 121–132, doi:10.1016/j.cels.2016.06.010.
- Babu, M.M. et al. 2004. 'Structure and evolution of transcriptional regulatory networks', *Current Opinion in Structural Biology*, 14(3), pp. 283–291. doi: 10.1016/j.sbi.2004.05.004.
- Basu, S. et al. 2004. 'Spatiotemporal control of gene expression with pulse-generating networks', *Proceedings of the National Academy of Sciences*, 101, pp. 6355–6360. doi: 10.1073/pnas.0307571101.
- Basu, S. et al. 2005. 'A synthetic multicellular system for programmed pattern formation', *Nature*, 434(7037), pp. 1130–1134. doi: 10.1038/nature03461.
- Bleris, L. et al. 2011. 'Synthetic incoherent feedforward circuits show adaptation to the amount of their genetic template.', *Molecular Systems Biology*, EMBO Press, 7(1), p. 519. doi: 10.1038/msb.2011.49.
- Bulcha et al. 2019. 'A persistence detector for metabolic network rewiring in an animal', *Cell Reports*, 26, pp. 460–468.e4. doi: 10.1016/j.celrep.2018.12.064.
- Chen, S. et al. 2013. 'Incoherent feed-forward regulatory logic underpinning glucocorticoid receptor action', *Proceedings of the National Academy of Sciences, National Academy of Sciences*, 110(5), pp. 1964–1969. doi: 10.1073/PNAS.1216108110.
- Chinenov, Y. et al. 2014. 'Glucocorticoid receptor coordinates transcription factor-dominated regulatory network in macrophages', *BMC Genomics*, 15, p. 656. doi: 10.1186/1471-2164-15-656.
- Conant, G.C. and Wagner, A. 2003. 'Convergent evolution of gene circuits', *Nature Genetics*, 34(3), pp. 264–266. doi: 10.1038/ng1181.
- Elowitz, M. and Leibler, S. 2000. 'A synthetic oscillatory network of transcriptional regulators', *Nature*, 403(6767), pp. 335–338. doi: 10.1038/35002125.
- Elowitz, M. and Lim, W.A. 2010. 'Build life to understand it', *Nature*, 468(7326), pp. 889–90. doi: 10.1038/468889a.
- Kalir, S., Mangan, S. and Alon, U. 2005 'A coherent feed-forward loop with a SUM input function prolongs flagella expression in *Escherichia coli*', *Molecular Systems Biology*, 1(1:2005.0006). doi: 10.1038/msb4100010.
- Kaplan, S. et al. 2008. 'The incoherent feed-forward loop can generate non-monotonic input functions for genes', *Molecular Systems Biology*, EMBO Press, 4(1), p. 203. doi: 10.1038/msb.2008.43.
- Lee, T.I. et al. 2002. 'Transcriptional regulatory networks in *Saccharomyces cerevisiae*', *Science*, 298(5594), pp. 799–804. doi: 10.1126/science.1075090.
- Ma, H.W. et al. 2004. 'An extended transcriptional regulatory network of *Escherichia coli* and analysis of its hierarchical structure and network motifs', *Nucleic Acids Research*, 32(22), pp. 6643–6649. doi: 10.1093/nar/gkh1009.
- Mangan, S. and Alon, U. 2003. 'Structure and function of the feed-forward loop network motif', *PNAS*, 100(21), pp. 11980–11985. doi: 10.1073/pnas.2133841100.
- Mangan, S., Zaslaver, A. and Alon, U. 2003. 'The coherent feedforward loop serves as a sign-sensitive delay element in transcription networks', *Journal of Molecular Biology*, 334(2), pp. 197–204. doi: 10.1016/j.jmb.2003.09.049.
- Mangan, S. et al. 2006. 'The incoherent feed-forward loop accelerates the response-time of the gal system of *Escherichia coli*', *Journal of Molecular Biology*, 356(5), pp. 1073–1081. doi: 10.1016/j.jmb.2005.12.003.
- Maslov, S. and Sneppen, K. 2002. 'Specificity and stability in topology of protein networks', *Science*, 296(5569), pp. 910–913. doi: 10.1126/science.1065103.
- Milo, R. et al. 2002 'Network motifs: simple building blocks of complex networks', *Science*, 298(5594), pp. 824–827. doi: 10.1126/science.298.5594.824.

- Sasse, S.K. and Gerber, A.N. 2015. 'Feed-forward transcriptional programming by nuclear receptors: regulatory principles and therapeutic implications', *Pharmacology & Therapeutics*, 145, pp. 85–91. doi: 10.1016/j.pharmthera.2014.08.004.
- Sasse, S.K., Mailloux, C.M., Barczak, A.J., Wang, Q., Altonsy, M.O., Jain, M.K., Haldar, S.M. and Gerber, A.N. 2013. 'The glucocorticoid receptor and KLF15 regulate gene expression dynamics and integrate signals through feed-forward circuitry', *Molecular and Cellular Biology*, 33(11), pp. 2104–2115. doi: 10.1128/MCB.01474-12.
- Shen-Orr, S.S. et al. 2002. 'Network motifs in the transcriptional regulation network of *Escherichia coli*', *Nature Genetics*, 43(7), p. 613. doi: 10.1038/ng881.

Temporal Programs and the Global Structure of Transcription Networks

4.1 INTRODUCTION

We have seen that transcription networks contain recurring network motifs that perform specific functions. We examined two of these motifs in detail, negative autoregulation and the feedforward loop (FFL). In this chapter, we complete our survey of motifs in sensory transcriptional networks. We will see that these networks are made of just four families of network motifs. These four motif families account for virtually all of the interactions in the networks. We will discuss how the network motifs fit together to build the global structure of transcription networks.

4.2 THE SINGLE-INPUT MODULE (SIM) NETWORK MOTIF

The network motifs we studied so far all had a defined number of nodes (one node in the autoregulation motif, three nodes in FFLs). We will now look for larger motifs. Each of these larger motifs corresponds to a family of patterns that share a common architectural theme. The first such motif family found in transcription networks is called the **single-input module** (Figure 4.1), or **SIM** for short (Shen-Orr et al., 2002).

In the SIM network motif, a master transcription factor X controls a group of target genes, $Z_1, Z_2, \dots, Z_n$. Each of the target genes in the SIM has only one input; No other transcription factor regulates any of the genes. In addition, the regulation signs (activation/repression) are the same for all genes in the SIM. The last feature of the SIM is that the master transcription factor X is usually autoregulatory.

Single-Input Module (SIM)

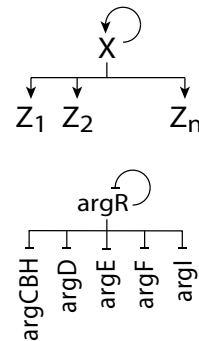


FIGURE 4.1

The most important role of SIMs is to control a group of genes according to the signal sensed by the master transcription factor. The genes in a SIM always have a common biological function. For example, SIMs regulate genes that participate in a specific metabolic pathway (Figure 4.2). These genes encode proteins that work sequentially to assemble a desired molecule atom by atom, in a kind of molecular assembly line.

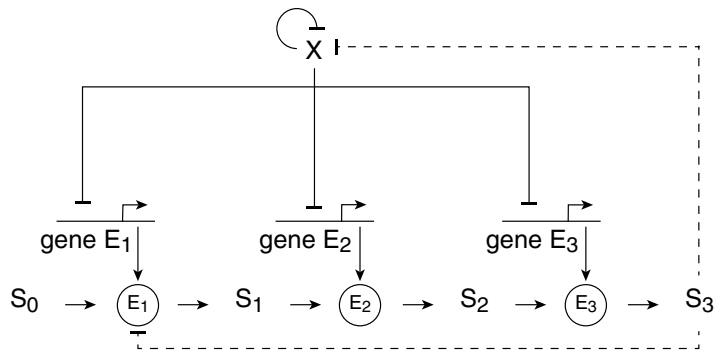


FIGURE 4.2

Metabolic pathways often show additional feedback control that operates on a rapid timescale. The final product of the pathway is often the input signal for the top transcription factor X (Figure 4.2). The final product also often inhibits the first enzyme in the pathway by directly binding it, a regulatory motif called **feedback inhibition**.

Other SIMs control groups of genes that respond to a specific stress (DNA damage, heat shock, etc.). Finally, SIMs can control groups of genes that together make up a protein machine with many subunits (such as a ribosome).

4.3 THE SIM CAN GENERATE TEMPORAL GENE EXPRESSION PROGRAMS

In addition to controlling a gene module in a coordinated fashion, the SIM has a subtler dynamical function. The SIM can generate temporal programs of expression, in which genes are activated in a defined order.

The temporal order is based on different thresholds of X for each of the target genes Z_i (Figure 4.3). When X activity changes gradually with time, it crosses these thresholds, K_i , at different times. X first activates the gene with the lowest threshold. Then it activates the gene with the next lowest threshold, and so on (Figure 4.3). The faster the changes in the activity of X , the more rapidly it crosses the different thresholds, and the smaller the delay between the genes.

When X activity goes down, the genes are affected in reverse order. Hence, the last gene activated is the first one to be deactivated (Figure 4.3). This type of program is called a last-in-first-out (LIFO) order.

Experimentally, temporal order is found in a variety of systems in *E. coli* with SIM architecture. This includes metabolic pathways (Zaslaver et al., 2004) such as the arginine system (Figure 4.4). The genes are sequentially expressed with delays on the

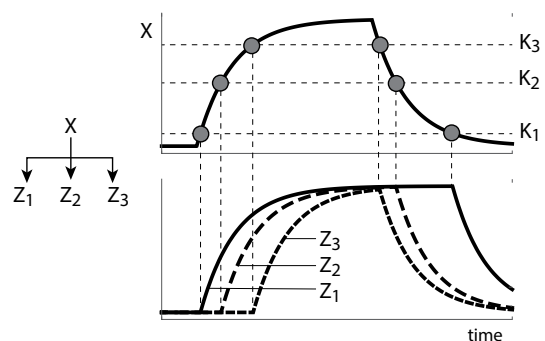


FIGURE 4.3

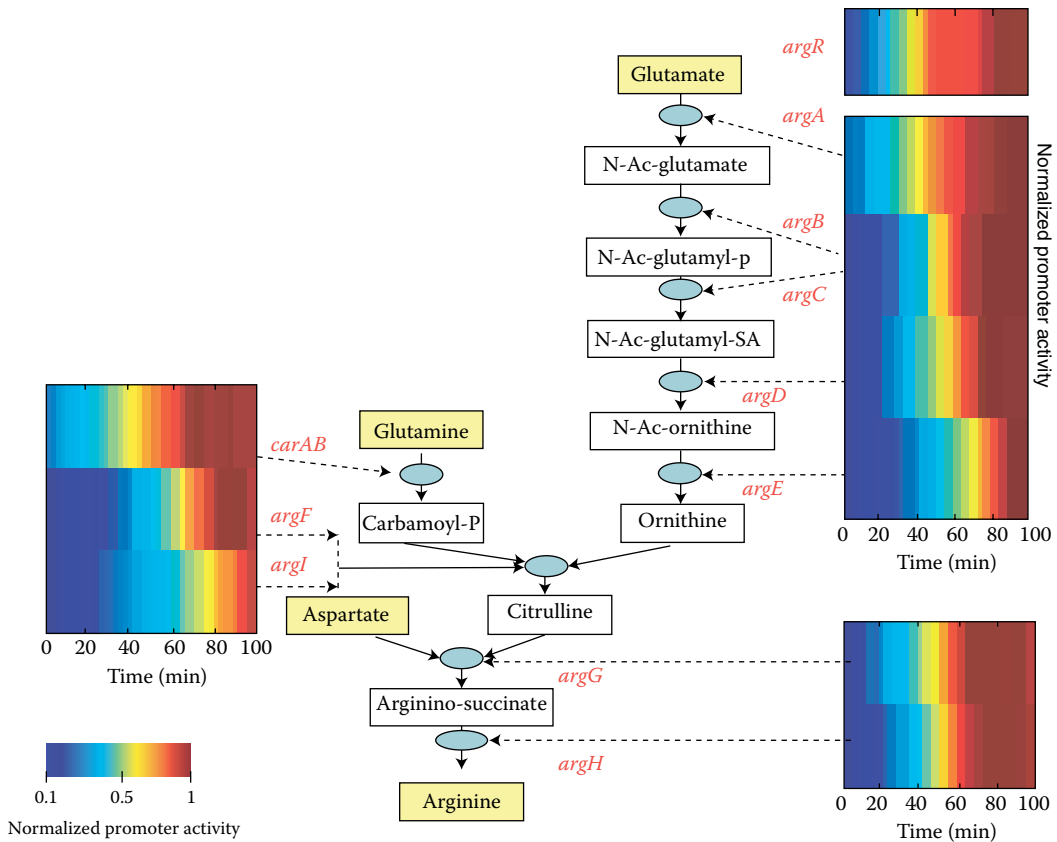


FIGURE 4.4 Adapted from (Zaslaver et al., 2004).

order of 0.1 generation between genes (about 5–10 min). In these pathways, the *temporal order of the genes corresponds to the order of the reactions in the pathway*: the first gene in the pathway is made first, then the second, and so on. This mechanism avoids expression of proteins before they are needed, a principle called **just-in-time production**.

The temporal order generated by a SIM can be varied by mutations that change the thresholds of the genes. For example, mutations in the binding site of X in the promoter of a gene can change the threshold, and accordingly change the order of expression (Kalir and Alon, 2004).

Temporal order is found also in damage repair systems controlled by SIMs. In damage repair systems, turn-ON is usually fast because the damage signal appears sharply, activating the regulator rapidly in order to mobilize repair processes. As damage is repaired, the input signal of the regulator declines and the genes get turned off gradually, reaching 50% of their maximal promoter activity at different times. In the SOS DNA repair system, for example, the genes responsible for the mildest form of repair are turned off first, and those responsible for more severe damage repair are turned off later (Ronen et al., 2002).

Temporal order also characterizes a large number of other global cellular responses. Examples include genes timed throughout the cell cycle in bacteria (Laub et al., 2000;

McAdams and Shapiro, 2003) and yeast (Eisen et al., 1998), genes regulated by different phases of the circadian clock that keeps track of the time of day (Young, 2000; Duffield et al., 2002) and genes in developmental processes (Dubrulle and Pourquié, 2002; Kmita and Duboule, 2003).

In these global well-timed responses, genes are usually regulated by a master regulator and by additional co-regulators responsible for smaller subsystems. Temporal order may be generated by the action of a master coordinating regulator even if the network pattern is not strictly a SIM, as long as all regulators except one have a constant activity during the interval of interest.

SIMs evolved by convergent evolution, just as we saw for the FFL motif. SIMs regulate homologous genes in different organisms, but the master regulators in these SIMs are very different (Ihmels et al., 2005; Tanay, Regev and Shamir, 2005). This means that rather than duplication of an ancestral SIM together with the regulator, evolution converged on the same regulation pattern in the different organisms.

In short, the SIM can generate just-when-needed temporal programs with LIFO order: the activation order of the genes is reversed with respect to the deactivation order (Figure 5.3). However, in many cases, it seems more desirable to have an activation order that is the same as the deactivation order: the first promoter turned on is also the first turned off (first-in-first-out or FIFO order). FIFO order is desirable for assembly processes that require parts in a defined order, some early and some late. In this case, when the process is deactivated, it is better for the early genes to be turned OFF before the late genes, in order to prevent waste from needlessly producing early genes proteins after late ones are OFF. Next, we will describe circuitry that can achieve FIFO order, the multi-output feedforward loop.

4.4 THE MULTI-OUTPUT FEEDFORWARD LOOP

To complete our survey of network motifs in sensory transcription networks, we turn to subgraphs with four nodes. It turns out that we can stop at four, because larger subgraphs found in transcription networks can all be built from motifs with up to four nodes.

There are 199 possible four-node patterns (Figure 4.5). Of these, *only two* are significant motifs in the known sensory transcription networks. Again, the networks show simplicity because they contain only a tiny fraction of the possible types of subgraphs. The two significant network motifs are a **two-output FFL**, which we will discuss now, and an overlapping regulation pattern termed the **bi-fan** (Figure 4.6) considered later in the chapter.

In the two-output FFL, X regulates Y , and both regulate the output genes Z_1 and Z_2 . When looking at larger subgraphs, this generalizes to a **multi-output FFL pattern**, in which a cascade of two regulators jointly controls a set of output genes (Figure 4.7).

One could think of other ways of generalizing the FFL, for example by having multiple inputs X , or multiple middle nodes Y (Figure 4.7). However, neither of these is a network motif (Kashtan et al., 2004).

The 4-node connected subgraphs

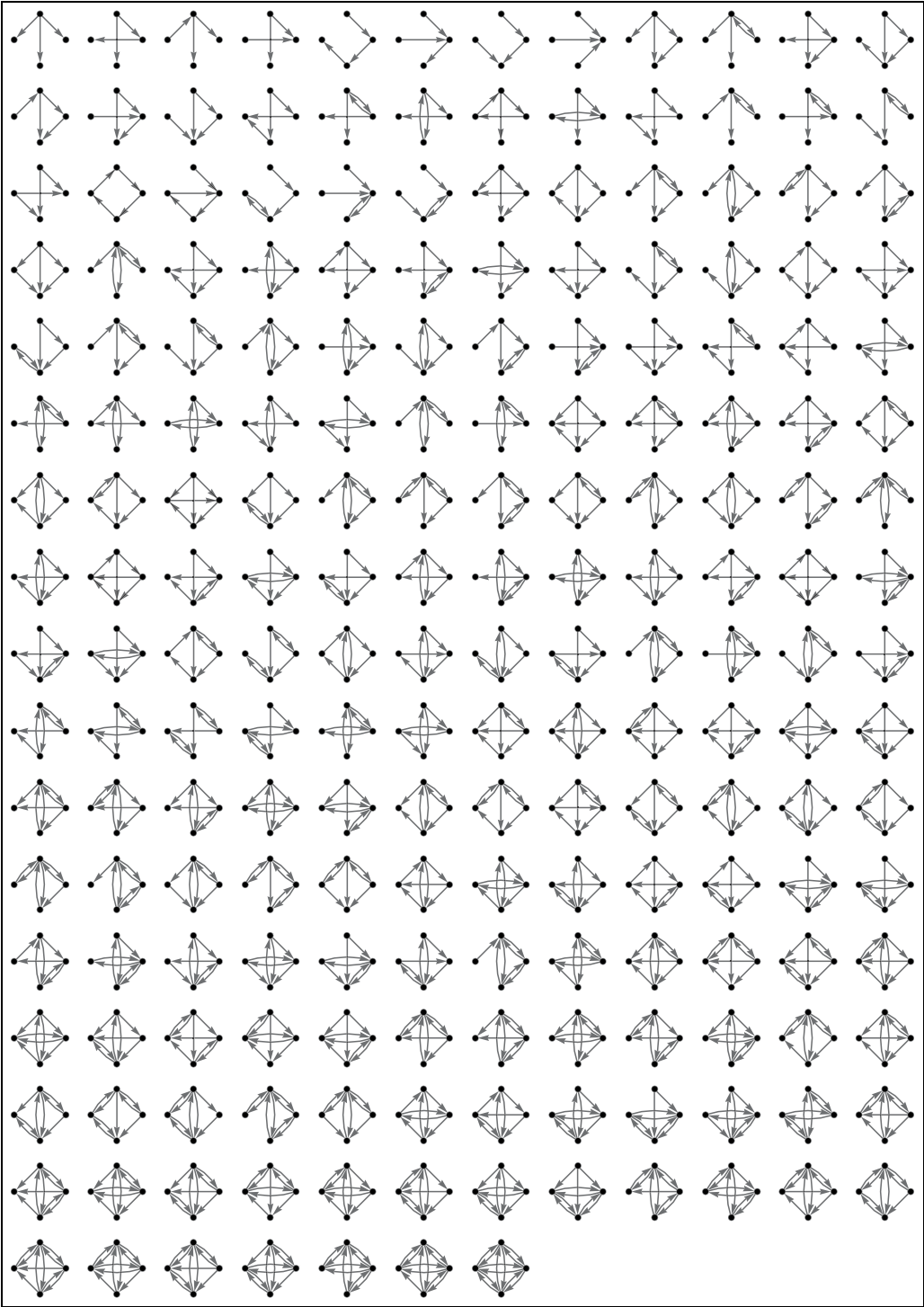


FIGURE 4.5

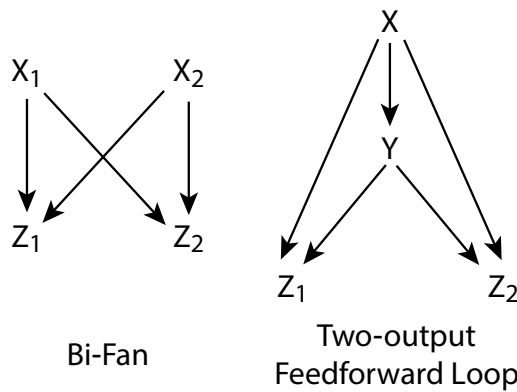


FIGURE 4.6

4.5 THE MULTI-OUTPUT FFL CAN GENERATE FIFO TEMPORAL PROGRAMS

What might be the function of multi-output FFL? To address this question, we will consider a well-characterized case of the multi-output FFL and see that it can generate a FIFO temporal program, in contrast to the LIFO order generated by SIMs.

A multi-output FFL regulates the gene system for the production of flagella, *E. coli*'s outboard motors (Figure 4.8). When *E. coli* is in a comfortable environment with abundant nutrients, it divides happily and does not try to move. When conditions become worse, *E. coli* makes a decision to grow several motors attached to helical flagella (propellers), which allow it to swim. It also generates a navigation system that tells it where to go in search of a better life. We will explore this navigation system called chemotaxis in Chapter 9. For now, let's consider the genes that make the parts of the flagella motor.

The flagella motor is a 50-nm device built of about 30 types of protein (Figure 4.8; Macnab, 2003). The motor is electrical, converting the energy of proton influx to drive rotation at about 100 Hz. The motor rotates the flagellum, which is about 10 times longer than the cell it is attached to (*E. coli* is about 1 micron long). Flagella rotation pushes the cell forward at speeds that can exceed 30 microns/sec.

The motor is put together in stages (Figure 4.8). This is an amazing example of biological self-assembly, like throwing Lego blocks in

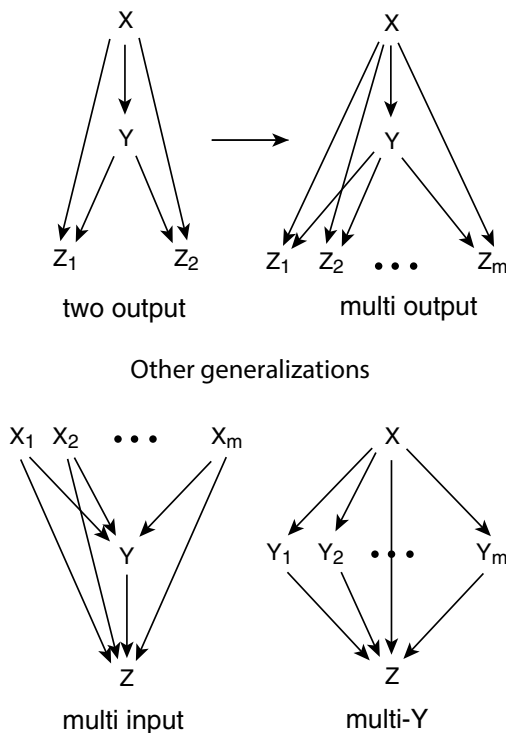


FIGURE 4.7

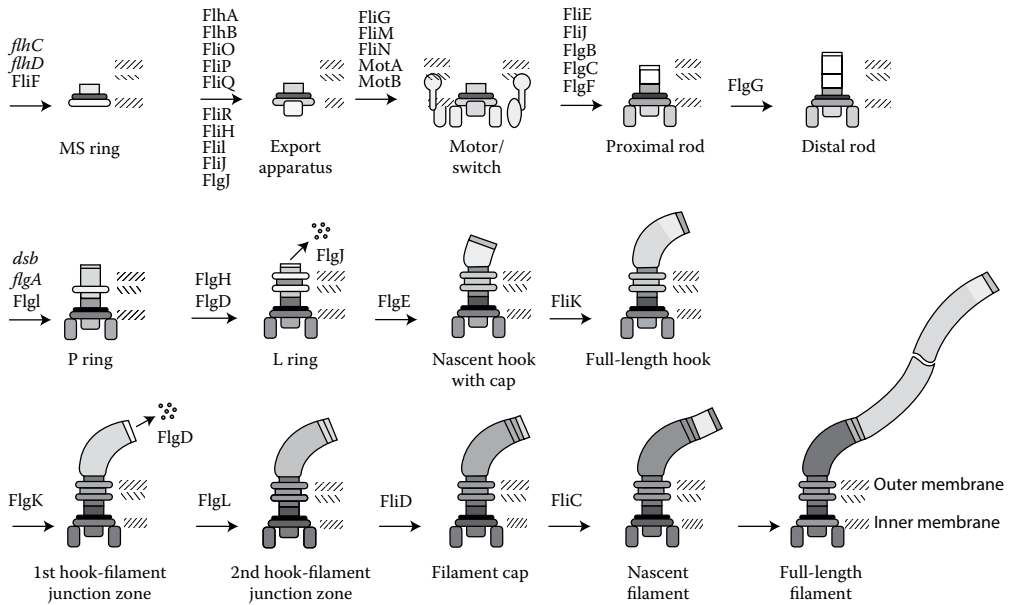


FIGURE 4.8 Adapted from (Macnab, 2003).

the air and watching them assemble into a house. The motor and flagellum have a hollow central tube through which the proteins move to assemble each stage. Thus, each stage of the motor acts as a transport device for the proteins in the next stage.

The proteins that build up the flagella motor are encoded by genes arranged in six operons (an operon is a group of genes transcribed on the same piece of mRNA). The flagella motor operons are regulated by two transcription activators, X and Y called FlhDC and FliA. The master regulator X activates Y , and both jointly activate each of the six operons, $Z_1, Z_2, \dots, Z_6$. This regulatory pattern is a multi-output FFL (Figure 4.9).

In this multi-output FFL, each operon can be activated by X in the absence of Y , and by Y in the absence of X . Thus, the input functions are similar to OR gates (they are actually SUM-gates, the weighted sum of the two active transcription factors).

Experiments by Michael Laub and Lucy Shapiro (Laub et al., 2000) and by Shiraz Kalir et al. (Kalir et al., 2001; Kalir and Alon, 2004), found that the flagella operons show a defined temporal order of expression. When the bacteria sense the proper conditions, they activate the production of protein X . The concentration of X gradually increases, and as a result, the Z genes get turned ON one by one, with about 0.1 cell generations between them. The order in which the operons are turned on matches the order of motor assembly: first a ring in the inner membrane,

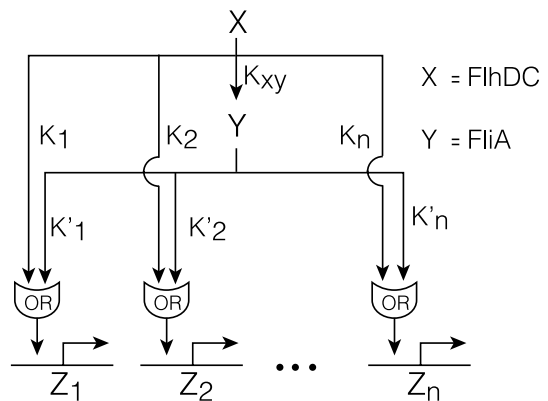


FIGURE 4.9

then a rod, a second ring, and so on. This is the principle of just-in-time production that we discussed in the single-input module (SIM) network motif.

The SIM architecture, however, has a limitation, as mentioned before: the turn-OFF order is reversed with respect to the turn-ON order (last-in-first-

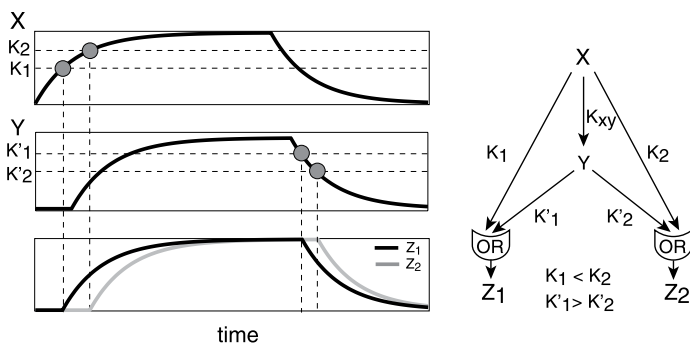


FIGURE 4.10

out, or LIFO order; Figure 4.3). In contrast, the flagella turn-OFF order is the same as the turn-ON order: the first promoter turned on is also the first turned off when flagella are no longer needed. In other words, the genes show a first-in-first-out order.

FIFO order is generated by the multi-output FFL thanks to a hierarchy of activation thresholds (Figure 4.10). In the flagella system, X and Y effectively function in OR gate logic, and, therefore, X alone is sufficient to turn the genes on. Therefore, the turn-ON order is determined by the times when X crosses the activation thresholds $K_1, K_2, \dots, K_n$ (Figure 4.10). If this were all, genes would be turned off in the reverse order once X levels decline, resulting in LIFO order, just like in the SIM. But here Y comes to the rescue. When X decays away, Y is still around for a while. The turn-OFF order in a properly designed OR-gate FFL is, therefore, governed by Y , which has its own thresholds, $K'_1, K'_2, \dots, K'_n$. FIFO order is achieved if the order of the thresholds of Y is reversed compared to that of X . That is, if the X thresholds are $K_1 < K_2$, so that promoter 1 is turned on before 2, the Y thresholds are oppositely ordered $K'_1 > K'_2$, so that promoter 1 is turned off before 2 (Figure 4.10). This opposing hierarchy of thresholds was experimentally found in the flagella system (Kalir and Alon, 2004). The temporal order in this system was rewired by mutations that affected these activation thresholds.

4.5.1 The Multi-Output FFL Also Acts as a Persistence Detector for Each Output

In addition to generating a FIFO temporal order, the multi-output FFL conveys all of the functions of the feedforward loop that we discussed in Chapter 3. In particular, each of the output nodes benefits from the sign-sensitive filter property of the FFL. For example, in the flagella system, the FFL delays the deactivation of the Z genes following the loss of X activity (as described in Section 3.6). It thus filters away brief OFF pulses of X , allowing deactivation only when X activity is gone for a persistent length of time. Such OFF pulses can occur, for example, when the bacterium briefly swims into better conditions.

4.6 SIGNAL INTEGRATION BY BI-FANS AND DENSE-OVERLAPPING REGULONS

In addition to the multi-output FFL, there is a second four-node network motif, the bi-fan. The bi-fan gives rise to a family of motif generalizations, shaped as a layer of inputs with

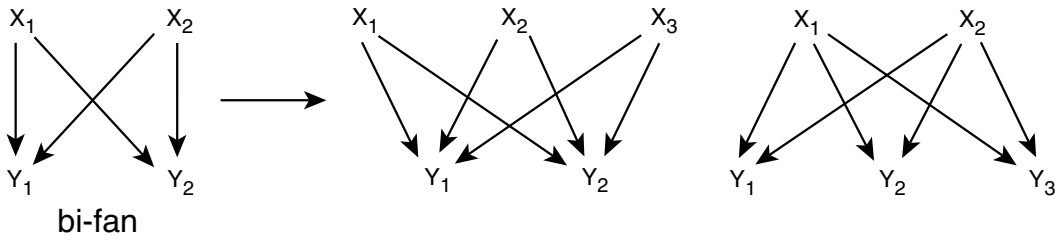


FIGURE 4.11

multiple overlapping connections to a layer of outputs (Figure 4.11). This family of patterns is our last network motif in sensory transcription networks, called **dense-overlapping regulons** (a regulon is the set of genes regulated by a given transcription factor), or **DORs** for short (Figure 4.12; Shen-Orr et al., 2002).

The DOR is a row of input transcription factors that regulate a row of output genes in a densely overlapping way. The DORs are usually not fully wired; that is, not every input regulates every output. However, the wiring is much denser than in the patterns found in randomized networks.

The DOR can be thought of as a combinatorial decision-making device. It functions as an array of gates (input functions) that integrate multiple inputs to compute the regulation of each output gene. To understand the function of the DOR requires knowledge of the multi-dimensional input functions that integrate the inputs at the promoter of each gene. This knowledge is one of the open challenges in systems biology: Although arrows in transcription networks are well-characterized, the input functions of most genes are currently unknown.

Transcription networks, such as those of *E. coli* and yeast, show several large DORs, each controlling tens to hundreds of genes. The genes in each DOR have a shared global function, such as stress response, nutrient metabolism or biosynthesis of key classes of cellular components. Often, a global regulator governs many of the genes, supplemented by numerous regulators that regulate subsets of the genes. For example, in *E. coli*, the global regulator CRP senses starvation and, together with multiple transcription factors that each sense a different sugar, determines which sugar utilization genes are activated in response to the sugars in the environment. The DORs form the backbone of the network's global structure, as we will see next.

Dense overlapping regulons (DOR)

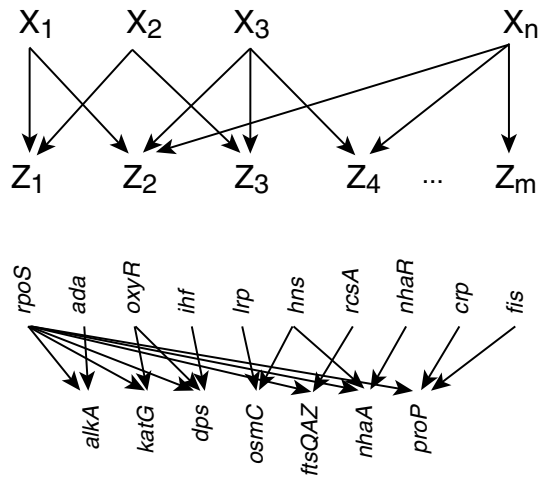


FIGURE 4.12

4.7 NETWORK MOTIFS AND THE GLOBAL STRUCTURE OF SENSORY TRANSCRIPTION NETWORKS

We have described the four main network motif families: autoregulation, feedforward loops, SIMs and DORs. How are these motifs positioned with respect to each other in the network? And how much of the network do they cover?

To answer these questions, it would help to look at an image of the network. It is difficult to draw a complex network in an understandable way (Figure 2.1). However, network motifs can help to produce a slightly less complicated image, based on the following coarse-graining procedure (Itzkovitz et al., 2005). To draw the network, replace each occurrence of a SIM that regulates n genes by a square marked with the number n . Replace each multi-output FFL by a triangle marked with the number of output genes. Replace each DOR with a rectangular box that groups its inputs, outputs and connections. The result is still an intricate picture, but one that can help us to understand the global network structure (see Figure 4.13, which shows part of the *E. coli* transcription network).

The coarse-grained network obtained by this procedure, Figure 4.13, shows that sensory transcription networks such as those of *E. coli* and yeast are made of a single layer of DORs. The DORs do not form cascades – there is no DOR at the output of another DOR. Thus, most of the computation done by the network is done at a layer of promoters within the DORs.

The DORs contain most of the other motifs. The FFLs and SIMs are integrated within the DORs. Negative autoregulation is often integrated within FFLs and also decorates the master regulators of SIMs. Overall, the rather simple way in which the network motifs are integrated makes it possible to understand the dynamics of each motif separately, even when it is embedded within larger patterns.

Virtually all of the genes are covered by these four network motifs in the sensory transcription networks studied so far, including those of bacteria, yeast, worms, fruit flies, mice and humans (Harbison et al., 2004; Penn et al., 2004; Boyer et al., 2005; Odom et al., 2006). Thus, these network motifs represent the major types of patterns that occur in sensory transcription networks. In the next chapter, we will discuss additional network motifs that occur in developmental transcription networks that guide the transformation from a single-celled egg to a multi-cellular organism.

4.8 INTERLOCKED FEEDFORWARD LOOPS IN THE *B. SUBTILIS* SPORULATION NETWORK

What about more complex circuits made of combinations of network motifs? Can we still understand their function? To address this question, we will discuss a network made of interlocking FFLs in the bacterium *Bacillus subtilis*.

When starved, *B. subtilis* stops dividing and turns into a durable spore. The spore contains many proteins that are not found in the growing bacterium. It is a resting cell, almost completely dehydrated, that can survive for a long time in a dormant state. When placed in the right conditions, the spore converts itself again into a growing bacterium.

When *B. subtilis* makes a spore, it must turn on hundreds of genes in a series of temporal waves, to carry out the stages of sporulation. The network that regulates sporulation (Eichenberger et al., 2004) is made of transcription factors arranged in linked coherent and incoherent type-1 FFLs (Figure 4.14).

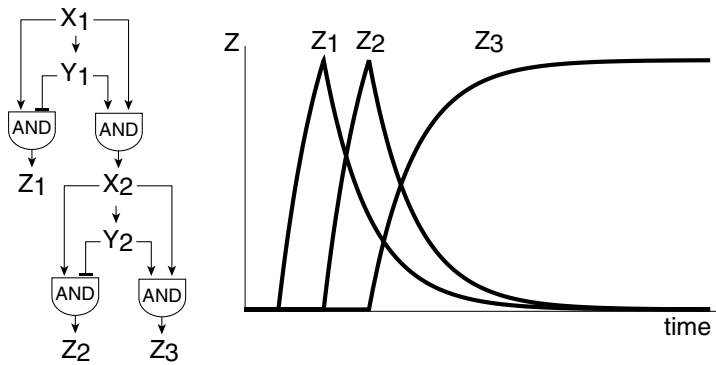


FIGURE 4.14

To initiate sporulation, a

starvation signal activates X_1 . This transcription factor acts in an incoherent type-1 FFL (I1-FFL) with the repressor Y_1 to control a set of output genes Z_1 . The I1-FFL generates a pulse of Z_1 expression.

A second FFL is formed by Y_1 and X_1 , which are both needed to activate X_2 , resulting in a coherent type-1 FFL (C1-FFL) with AND logic. This C1-FFL ensures that X_2 is not activated unless the starvation signal is persistent. Next, X_2 acts in another I1-FFL, where it activates genes Z_2 as well as their repressor Y_2 . This results in a pulse of Z_2 genes, timed at a delay relative to the first pulse. Finally, Y_2 and X_2 together join in an AND-gate C1-FFL to activate genes Z_3 . The result is a three-wave pattern: first a pulse of Z_1 expression, followed by a pulse of Z_2 expression, followed by expression of the late genes Z_3 .

The FFLs in this network are combined in a way that utilizes their delay and pulse-generating features to generate a three-wave temporal program of gene expression. The FFLs are actually multi-output FFLs because the outputs Z_1 , Z_2 and Z_3 each represent large groups of genes. This design can generate finer temporal programs within each group of genes.

We see that the FFLs in this network are linked such that each FFL maintains its autonomy. The dynamics of the network can be easily understood based on the dynamics of each FFL. It is important to note that there are, in principle, many other ways of linking FFLs. Most combinations of linked FFLs do not lend themselves to easy interpretation (can you draw a few of these possible configurations?). This seems to be a general feature of gene regulation networks: motifs are linked in ways that allow easy interpretation based on the dynamics of each motif in isolation.

Such understandability of circuit patterns in terms of simpler sub-circuits could not have evolved just to make life easier for biologists. Understandability is a central feature of engineering. Engineers build complex systems out of simpler subsystems that are well understood. These subsystems are connected so that each subsystem retains its behavior and works reliably. It is an interesting question whether understandability might be a common feature of networks that evolve to function.

In summary, sensory transcription networks across organisms are built of four motif families: autoregulation, FFLs, SIMs and DORs. Almost all of the genes participate in these

motifs. Each network motif carries out a defined dynamical function, such as speeding responses, generating temporal programs, persistence detection or combinatorial decision-making. Most other patterns, such as three-node feedback loops, are conspicuously absent. Three-node feedback loops are often unstable (a common property of negative loops) or bistable (a common property of positive loops), both undesirable properties for sensory networks that need to generate reversible and reliable responses. Indeed, the motifs that are found in sensory transcription networks are extremely stable (Prill, Iglesias and Levchenko, 2005; Angulo, Liu and Slotine, 2015). Sensory transcription networks belong to the family of directed acyclic graphs (DAGs). However, they have even fewer subgraph types than generic DAGs. Hence, the subgraph content of these networks is much simpler than it could have been. They are built of a small set of elementary circuit patterns, the network motifs.

FURTHER READING

(Shen-Orr et al., 2002) “Network motifs in the transcriptional regulation network of *Escherichia coli*.”

Combinations of Network Motifs

(Angulo, Liu and Slotine, 2015) “Network motifs emerge from interconnections that favor stability.”

(Itzkovitz et al., 2005) “Coarse-graining and self-dissimilarity of complex networks.”

Generalizations of Network Motifs

(Berg and Lassig, 2004) “Local graph alignment and motif search in biological networks.”

(Kashtan et al., 2004) “Topological generalizations of network motifs.”

Multi-Output FFLs and Temporal Order

(Kalir and Alon, 2004) “Using a quantitative blueprint to reprogram the dynamics of the flagella gene network.”

Single-Input Module and Temporal Order

(Laub et al., 2000) “Global analysis of the genetic network controlling a bacterial cell cycle.”

(McAdams and Shapiro, 2003) “A bacterial cell-cycle regulatory network operating in space and time.”

(Zaslaver et al., 2004) “Just-in-time transcription program in metabolic pathways.”

EXERCISES

4.1 *Equal timing*: Consider a SIM controlled by regulator X that activates downstream genes Z_i , $i = 1 \dots n$, with thresholds K_i . At time $t = 0$, X begins to be produced at a constant rate β , and removed at rate α . Design thresholds such that the genes are turned on one after the other at equal intervals (use logic input functions).

4.2 *Robust timing*:

- a. For the system of Exercise 4.1, are there biological reasons that favor placing the thresholds K_i much smaller than the maximal level of $X = \beta/\alpha$? Consider the effects of cell–cell variations in the production rate of X .

- b. Would a design in which X is a repressor whose production stops at time $t = 0$ provide more robust temporal programs with respect to cell-cell variations? Explain.
- 4.3 *The multi-output OR C1-FFL*: In a multi-output C1-FFL with OR gate logic at the Z promoters, transcription factor X begins to be produced at a constant rate β at time $t = 0$. At time $t = T$, where T is long enough for all genes to be activated and for Y to rise close to its steady-state level, the production rate β suddenly drops to zero. Calculate the dynamics of the downstream genes Z_i . What are the delays between genes? (Use logic input functions.)
- 4.4 *Generalized diamond*: What are the topological generalizations of the diamond pattern ($X \rightarrow Y_1, X \rightarrow Y_2, Y_1 \rightarrow Z, Y_2 \rightarrow Z$) based on duplication of a single node and all of its edges? How are these different from DORs? What are the topological generalizations of the bi-fan ($X_1 \rightarrow Y_1, X_2 \rightarrow Y_1, X_1 \rightarrow Y_2, X_2 \rightarrow Y_2$)? Many of these five-node generalizations of the diamond and bi-fan are network motifs in the neuronal network of *C. elegans* (Milo et al., 2002).
- 4.5 *SIM with autoregulation*: What is the effect of autoregulation on the master transcription factor X in a SIM? Plot schematically and compare the dynamics of the output genes in a given SIM with positive autoregulation of X , negative autoregulation of X and no autoregulation of X . Discuss when each design might be useful.
- 4.6 *Bi-fan dynamics*: Consider a bi-fan in which activators X_1 and X_2 regulate genes Z_1 and Z_2 . The input signal of X_1 , S_{x1} appears at time $t = 0$ and vanishes at time $t = D$. The input signal of X_2 , S_{x2} , appears at time $t = D/2$ and vanishes at $t = 2D$. Plot the dynamics of the promoter activity of Z_1 and Z_2 , given that their input functions are AND and OR logic, respectively.
- 4.7 *The multi-output AND C1-FFL*: What is the temporal order of turn ON and turn OFF in a multi-output C1-FFL where all genes are regulated by AND gates? Which thresholds determine the ON and OFF orders? Can one obtain FIFO orders?
- 4.8 *Multi-input FFL*: What dynamics might an FFL with two inputs X_1 and X_2 that both regulate Z and its regulator Y , have? Consider (i) the case where all arrows are positive, (ii) the case where all arrows are positive except for Y that inhibits Z and (iii) as in (ii) but with X_1 negatively regulating Y .
- 4.9 *SIM software*: Write a program to identify SIMs in a network. Use it to count SIMs in the *E. coli* network from <https://www.weizmann.ac.il/mcb/UriAlon/e-coli-transcription-network>. What is the largest SIM in the network?
- 4.10 *Feedback inhibition*: What might be the function of feedback inhibition shown in Figure 4.2 by the dashed arrow, in which the pathway product molecule inhibits the first enzyme in the pathway?
- 4.11 *Delays in metabolic pathways*: Consider a bacterial metabolic pathway, like the arginine pathway, in which the substrate for each enzyme is the product of the previous enzyme. Typical enzymes catalyze about 10 reactions per second (Bar-Even et al., 2011). Typical

concentrations needed of each substrate are on the order of 1000s of molecules per bacterial volume. What is the order of magnitude of the delays between genes that would provide proper just-in-time production when the system is activated?

BIBLIOGRAPHY

- Angulo, M.T., Liu, Y.Y. and Slotine, J.J. 2015. 'Network motifs emerge from interconnections that favour stability', *Nature Physics*, 11(10), pp. 848–852. doi: 10.1038/nphys3402.
- Bar-Even, A. et al. 2011. 'The moderately efficient enzyme: evolutionary and physicochemical trends shaping enzyme parameters', *Biochemistry*, 50(21), pp. 4402–4410. doi: 10.1021/bi2002289.
- Berg, J. and Lassig, M. 2004. 'Local graph alignment and motif search in biological networks', *Proceedings of the National Academy of Sciences*, 101(41), pp. 14689–14694. doi: 10.1073/pnas.0305199101.
- Boyer, L.A. et al. 2005. 'Core transcriptional regulatory circuitry in human embryonic stem cells', *Cell*, 122(6), pp. 947–956. doi: 10.1016/j.cell.2005.08.020.
- Dubrule, J. and Pourquié, O. 2002. 'From head to tail: links between the segmentation clock and antero-posterior patterning of the embryo', *Current Opinion in Genetics and Development*, 12(5), pp. 519–523. doi: 10.1016/S0959-437X(02)00335-0.
- Duffield, G.E. et al. 2002. 'Circadian programs of transcriptional activation, signaling, and protein turnover revealed by microarray analysis of mammalian cells', *Current Biology*, 12(7), pp. 551–557. doi: 10.1016/S0960-9822(02)00765-0.
- Eichenberger, P. et al. 2004. 'The program of gene transcription for a single differentiating cell type during sporulation in *Bacillus subtilis*', *PLoS Biology*, 2(10), p. e328. doi: 10.1371/journal.pbio.0020328.
- Eisen, M.B. et al. 1998. 'Cluster analysis and display of genome-wide expression patterns', *Proceedings of the National Academy of Sciences*, 95(25), pp. 14863–14868. doi: 10.1073/pnas.95.25.14863.
- Harbison, C.T. et al. 2004. 'Transcriptional regulatory code of a eukaryotic genome', *Nature*, 431(7004), pp. 99–104. doi: 10.1038/nature02800.
- Ihmels, J. et al. 2005. 'Molecular biology: Rewiring of the yeast transcriptional network through the evolution of motif usage', *Science*, 309(5736), pp. 938–940. doi: 10.1126/science.1113833.
- Itzkovitz, S. et al. 2005. 'Coarse-graining and self-dissimilarity of complex networks', *Physical Review E*, 71(1), p. 72. doi: 10.1103/PhysRevE.71.016127.
- Kalir, S. and Alon, U. 2004. 'Using a quantitative blueprint to reprogram the dynamics of the flagella gene network', *Cell*, 117(6), pp. 713–720. doi: 10.1016/j.cell.2004.05.010.
- Kalir, S. et al. 2001. 'Ordering genes in a flagella pathway by analysis of expression kinetics from living bacteria', *Science*, 292(5524), pp. 2080–2083. doi: 10.1126/science.1058758.
- Kashtan, N. et al. 2004. 'Topological generalizations of network motifs', *Physical Review E*, 70(3), p. 10. doi: 10.1103/PhysRevE.70.031909.
- Kmita, M. and Duboule, D. 2003. 'Organizing axes in time and space; 25 years of colinear tinkering', *Science*, 301(5631), pp. 331–333. doi: 10.1126/science.1085753.
- Laub, M.T. et al. 2000. 'Global analysis of the genetic network controlling a bacterial cell cycle', *Science*. doi: 10.1126/science.290.5499.2144.
- Macnab, R.M. 2003. 'How Bacteria Assemble Flagella', *Annual Review of Microbiology*, 57(1), pp. 77–100. doi: 10.1146/annurev.micro.57.030502.090832.
- McAdams, H.H. and Shapiro, L. 2003. 'A bacterial cell-cycle regulatory network operating in time and space', *Science*, 301(5641), pp. 1874–1877. doi: 10.1126/science.1087694.
- Milo, R. et al. 2002. 'Network motifs: simple building blocks of complex networks', *Science*, 298(5594), pp. 824–827. doi: 10.1126/science.298.5594.824.
- Odom, D.T. et al. 2006. 'Core transcriptional regulatory circuitry in human hepatocytes', *Molecular Systems Biology*, 2, p. 3495. doi: 10.1038/msb4100059.

- Penn, B.H. et al. 2004. 'A MyoD-generated feed-forward circuit temporally patterns gene expression during skeletal muscle differentiation', *Genes and Development*, 18(19), pp. 2348–2353. doi: 10.1101/gad.1234304.
- Prill, R.J., Iglesias, P.A. and Levchenko, A. 2005. 'Dynamic properties of network motifs contribute to biological network organization', *PLoS Biology*, 3(11), p. e343. doi: 10.1371/journal.pbio.0030343.
- Ronen, M. et al. 2002. 'Assigning numbers to the arrows: parameterizing a gene regulation network by using accurate expression kinetics', *Proceedings of the National Academy of Sciences*, 99(16), pp. 10555–10560. doi: 10.1073/pnas.152046799.
- Shen-Orr, S.S. et al. 2002. 'Network motifs in the transcriptional regulation network of *Escherichia coli*', *Nature Genetics*, 31(1), pp. 64–68. doi: 10.1038/ng881.
- Tanay, A., Regev, A. and Shamir, R. 2005. 'Conservation and evolvability in regulatory networks: the evolution of ribosomal regulation in yeast', *Proceedings of the National Academy of Sciences*, 102(20), pp. 7203–7208. doi: 10.1073/pnas.0502521102.
- Young, M.W. 2000. 'Life's 24-hour clock: Molecular control of circadian rhythms in animal cells', *Trends in Biochemical Sciences*, 25(12), pp. 601–606. doi: 10.1016/S0968-0004(00)01695-9.
- Zaslaver, A. et al. 2004. 'Just-in-time transcription program in metabolic pathways', *Nature Genetics*, 36(5), pp. 486–491. doi: 10.1038/ng1348.

Positive Feedback, Bistability and Memory

5.1 NETWORK MOTIFS IN DEVELOPMENTAL TRANSCRIPTION NETWORKS

So far, we've concentrated on transcription networks that are built to sense and respond to external stimuli such as nutrients and stresses. Such **sensory transcription networks** are found in almost all cells.

Multi-cellular organisms also have another type of transcription network, called **developmental transcription networks**. These networks govern the nearly irreversible changes that occur when a cell transforms itself into another type of cell.

An important example is the development of a multi-celled organism such as a human being. We begin as a single-celled egg, which divides into many cells that form the diverse tissues of the body. As the cells divide, they **differentiate** into distinct cell types, in order to form different tissues. To become part of a new tissue, each cell needs to express a specific set of proteins. This specific set of proteins determines whether the cell becomes, say, a nerve cell or a muscle cell. These differentiation processes are governed by developmental transcription networks (Levine and Davidson, 2005).

The developmental transcription networks of well-studied organisms such as fruit flies, worms, sea urchins and humans are composed of several strong network motifs. They display most of the motifs that we found in sensory networks. For example, the feedforward loop (FFL) is a strong network motif (Milo et al., 2002; Penn et al., 2004; Boyer et al., 2005; Odom et al., 2006). The most common FFL types in developmental networks are the coherent type-1 and incoherent type-1 FFLs, just as in sensory networks. Developmental networks also display prominent autoregulation motifs and single-input modules (SIMs).

In addition to these motifs, developmental networks display a few additional network motifs that are not commonly found in sensory transcription networks. We will now describe these network motifs and their functions.

5.1.1 Positive Autoregulation Slows Responses and Can Lead to Bistability

Developmental networks have many more **positive autoregulation** (PAR) loops than sensory networks do (Figure 5.1). In PAR, a protein activates its own transcription.

Positive autoregulation (PAR)

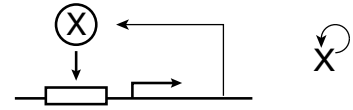


FIGURE 5.1

Positive autoregulation has an opposite effect to that of negative autoregulation: it slows the response time relative to simple regulation (Figure 5.2). The dynamics are initially slow, but as the levels of X build up, it increases its own production and reaches halfway to steady state at a delay relative to simple regulation.

To understand this slowing-down effect, we can repeat the rate analysis that we did for negative autoregulation. In Figure 5.3, we draw the production and removal curves for a gene with PAR. The Equation is $dX/dt = f(x) - \alpha X$. Removal rate is a straight line, αX . Production rate is an increasing input function $f(X)$ appropriate for the auto-activation of X .

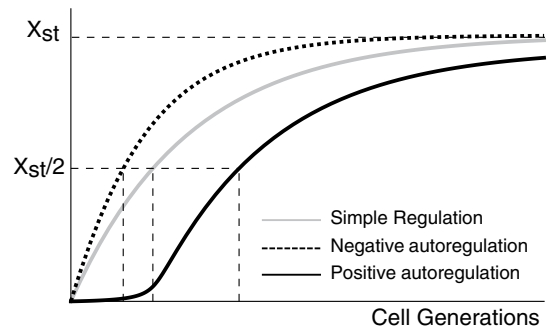


FIGURE 5.2

The two curves intersect at a fixed point, the steady state of the system. The fixed point is stable, because shifting X to either side causes a return to the fixed point. The speed for approaching the fixed point (speed equals the distance between production and removal curves) is smaller in PAR than in simple regulation, which has a flat production curve $f(X) = \beta$ (Figure 5.3). Thus, PAR shows slowdown for any increasing input function $f(X)$.

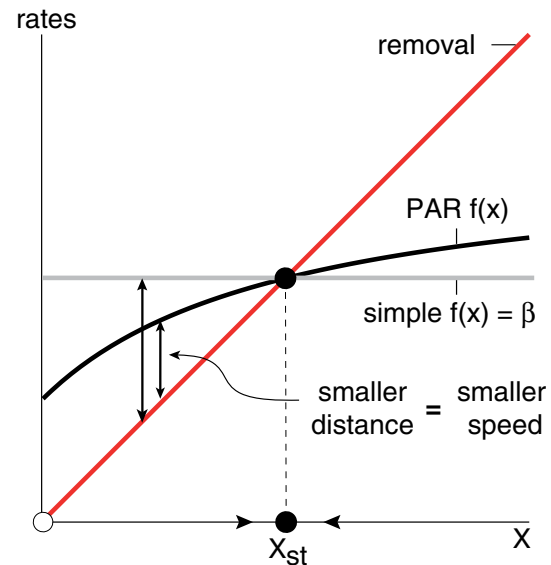


FIGURE 5.3

The slow dynamics provided by positive autoregulation are useful in multi-stage processes that take a relatively long time, such as developmental processes. These processes can benefit from prolonged delays between the production of proteins

responsible for different stages. Slow response times also help filter out rapidly varying noise in input signals, because slow responses integrate over this noise so that it cancels itself out (Hornung and Barkai, 2008).

In addition, positive autoregulation can do something spectacular: **it can make sharp decisions between two states, and then remember the decision for a long time.**

PAR can provide such memory when the autoregulation function $f(X)$ is cooperative (sigmoidally shaped), as in the rate analysis of Figure 5.4. If properly positioned, a sigmoidal $f(X)$ can intersect the removal curve *three* times, generating three fixed points, as opposed to the single crossing point when $f(X)$ is not sigmoidal (Figure 5.3).

There is a high stable fixed point X_{high} , a low stable fixed point X_{low} and an unstable fixed point X_u in the middle. Throughout the book, we will mark stable fixed points with black circles and unstable ones with white circles. You can see that the middle point is unstable because moving X slightly to either side of X_u causes X to flow away from X_u . For example, at values of X a bit higher than X_u , production (black curve) exceeds removal (red curve) and X , therefore, grows and moves further away from X_u .

The system has two stable fixed points, a feature called **bistability**. Which of the two is reached depends on the initial conditions: starting low ($X < X_u$) leads to the low fixed point, X_{low} , and starting high ($X > X_u$) leads to the high steady state, X_{high} . We say that there are two **basins of attraction**, separated by the unstable fixed point.

Any initial conditions inside the lower basin flows to X_{low} and any initial condition in the upper basin flows to X_{high} (Figure 5.5). The circuit has memory of where it started.

Thus, once the gene is sufficiently activated, it is locked into a state of high expression and keeps itself ON. To see how this produces long-term **memory**, suppose there is a transcription factor Y that also activates the X promoter (Figure 5.6). X starts at X_{low} , and then the signal S_Y for Y appears, causing more X to be produced. X crosses the threshold X_u , and reaches a high level. When the signal for Y goes away, X remains high, at its high

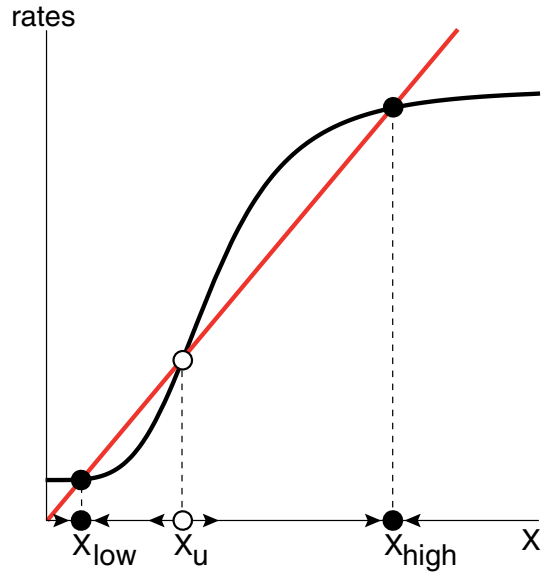


FIGURE 5.4

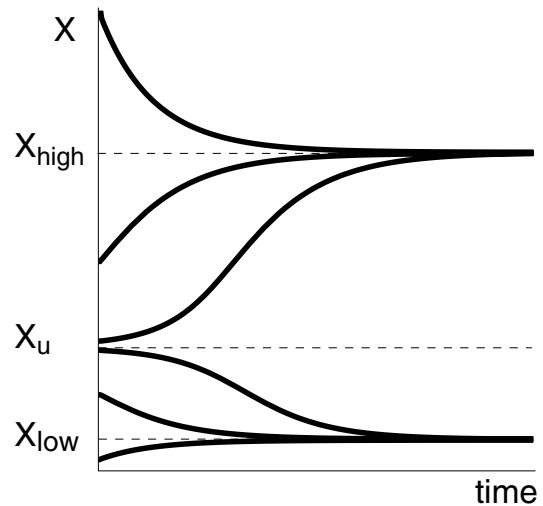


FIGURE 5.5

steady state, X_{high} (Figure 5.7). This is long-term memory because X stays high despite the fact that the input signal S_Y has vanished.

This type of bistable circuit is used in developmental transcription networks to make irreversible decisions that lock a cell into a particular fate (e.g., to determine the type of tissue). For example, when embryonic cells first become muscle cells, the transcription factor myoD activates a positive feedback loop and stays locked ON, activating the muscle-specific genes that make the muscle fibers and proteins needed for force generation. Positive autoregulation makes sure that the cell doesn't forget its muscular identity.

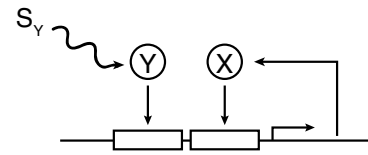


FIGURE 5.6

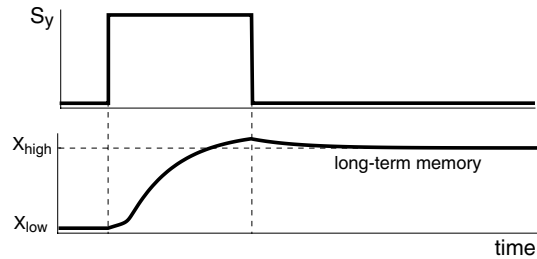


FIGURE 5.7

5.1.2 Two-Node Positive Feedback Loops for Decision-Making

Developmental networks also display a network motif in which two transcription factors regulate each other. This mutual regulation forms a two-node feedback loop. In developmental networks, the regulation signs of the two interactions usually lead to positive feedback loops (Figure 5.8).

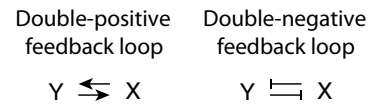


FIGURE 5.8

There are two types of positive feedback loops: double-positive in which the two transcription factors activate each other, and double-negative in which the two transcription factors repress each other.

The **double-positive feedback loop** can have two stable steady states (Thomas and D'Ari, 1990). In one stable state, genes X and Y are both ON, and the two transcription factors enhance each other's production. In the other stable state, X and Y are both OFF. A signal that causes protein X or Y to be produced can lock the system into the ON state. This type of bistable switch is called a **lock-on mechanism** (Davidson et al., 2002) because X and Y are both ON or both OFF. The double-positive feedback loop is most useful when genes regulated by X and genes regulated by Y encode proteins that belong to the same tissue.

We saw that positive autoregulation, a motif with one node, can also lock into a state of high expression. Why, then, do two-node feedback loops appear if one-node loops are sufficient? One reason is that the double-positive feedback loop only locks on after an appreciable delay, and hence can filter out transient input signals.

The **double-negative loop** can also have two stable steady states. In one stable state, X is ON and Y is OFF, because protein X represses Y . The other stable state is the reverse: X is OFF and Y is ON. Thus, unlike the double-positive feedback loop that can express both

X and Y (or neither), the double-negative loop expresses either X or Y . For this reason, this circuit is also called a **toggle switch**. A toggle switch is useful when genes regulated by X belong to different cell fates than the genes regulated by Y .

To understand how two-node circuits like the toggle switch produce two stable states (bistability), there is a useful graphical method, called **nullcline analysis**. This method is like a 2D version of the rate plot method that we used to analyze autoregulation. Nullclines are curves at which one of the proteins in the circuit has zero rate of change (one nullcline is $dX/dt = 0$, and the other is $dY/dt = 0$). The points to watch are the crossing points of the two nullclines, because, at these crossing points, neither protein changes and we have a fixed point. Here is an example.

Solved Example 5.1: Show That a Toggle Switch with Sigmoidal Repression Functions Can Have Bistability

Let's for simplicity assume a symmetric situation where X and Y have the same removal rate α and the same inhibition functions:

$$\frac{dX}{dt} = f(Y) - \alpha X$$

$$\frac{dY}{dt} = f(X) - \alpha Y$$

The inhibition function f is a decreasing sigmoidal function representing repression (e.g., a Hill function with $n > 1$).

To draw the nullcline for X , we ask when $dX/dt = 0$. This nullcline is the solution to $f(X) - \alpha Y = 0$, namely $Y = f(X)/\alpha$. Thus the nullcline is a curve shaped like the repression function. So is the other nullcline, $X = f(Y)/\alpha$. Drawing these two sigmoidal curves on the X - Y plane results in three intersection points (Figure 5.9).

We can add to this plot little arrows showing in which directions X and Y flow. The resulting picture is called the **phase portrait** of the circuit. It shows that there are two stable fixed points: $X_{\text{high}}, Y_{\text{low}}$ and its opposite $X_{\text{low}}, Y_{\text{high}}$. The middle fixed point is unstable. The dashed line, called the **separatrix**, separates the two **basins of attraction**, the regions from which initial conditions flow to the two steady states.

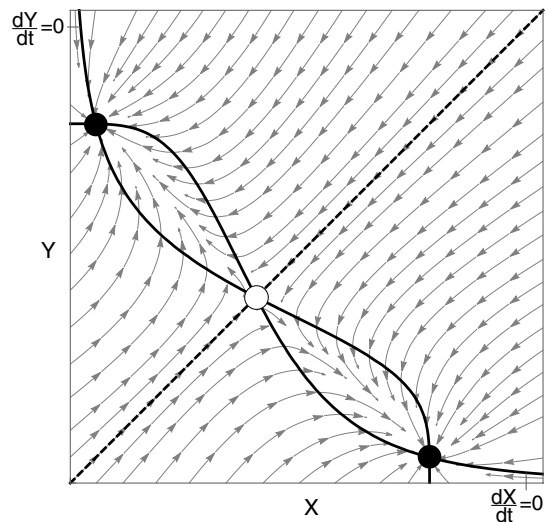


FIGURE 5.9

With nullcline analysis it is easy to see that if both repression functions are not sigmoidal (Hill coefficient $n = 1$), there can only be a single fixed point (Figure 5.10). The non-sigmoidal shapes of the nullclines prevent them from intersecting more than once. The system is mono-stable: X and Y repress each other but both reach an intermediate concentration and stay there.

Often, the transcription factors in the two-node feedback loop also each have positive autoregulation (Figure 5.11). Positive autoregulation loops act to enhance the production of the transcription factor once it is present in sufficient levels. This further stabilizes the ON steady states of the transcription factors. In fact, such PAR can even turn a two-node feedback loop without cooperative regulation into a bistable switch (Exercise 5.3).

A classic example of a toggle switch appears in phage lambda, a virus that infects *E. coli* (Figure 5.12; Ptashne and Gann, 2002). The phage is a protein container that houses a short DNA genome, which the phage can inject into the bacterium. The phage then chooses one of two possible modes of existence. In the lytic mode, the phage makes about 100 new phages which burst the cell open and diffuse away to find new *E. coli* prey (Figure 5.12). In the lysogenic mode, the phage integrates its DNA into the bacterial DNA and sits quiet.

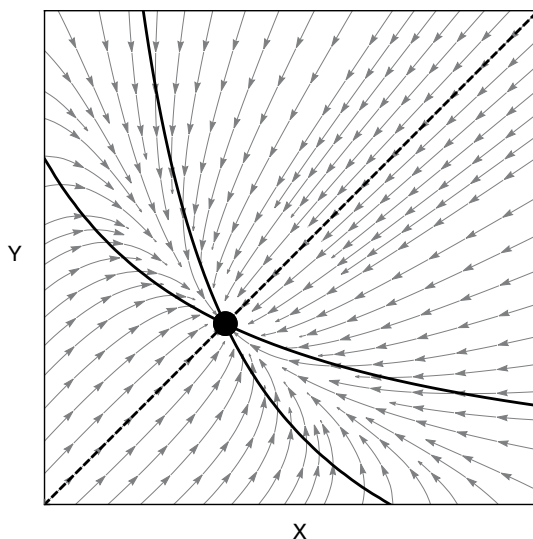


FIGURE 5.10



FIGURE 5.11

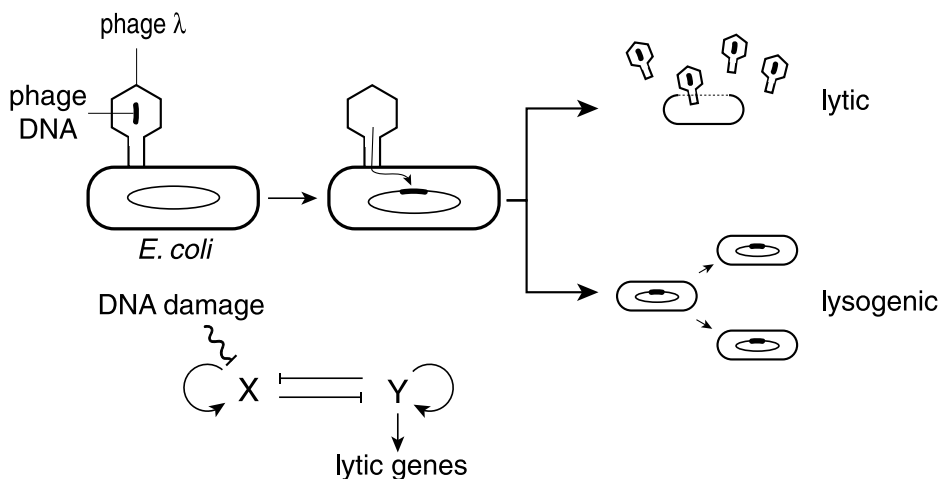


FIGURE 5.12

In both modes, the phage reproduces: in the lytic mode by infecting other cells, and in the lysogenic mode by using the bacterial divisions to reproduce. Thus, lysogeny is preferred when the cell is healthy and can divide, and the lytic mode is preferred when the cells are damaged.

To make the decision between the modes, the phage uses two transcription factors, *X* and *Y* (called C1 and *cro*) that form a toggle switch with positive autoregulation. *Y* is expressed in the lytic mode and activates lytic genes including itself. *X* is the only protein expressed in the lysogenic mode. To switch from lysogeny to lysis, *X* eavesdrops on the cell's damage signals: when DNA is damaged in the cell, a sensor for DNA damage called RecA cleaves protein *X*. As a result, *X* vanishes, *Y* is no longer repressed and the phage switches to lytic mode and kills the cell.

5.1.3 Regulating Feedback and Regulated Feedback

Two-node feedback loops can appear within larger motifs in developmental networks. These networks display two main three-node motifs that contain feedback loops (Milo et al., 2004). The first is a triangle pattern in which the mutually regulating nodes *X* and *Y* both regulate gene *Z* (Figure 5.13), called **regulating feedback**.

The regulating-feedback network motif has 10 possible sign combinations (Figure 5.14). In the simplest case, *X* and *Y*, which activate each other in a double-positive loop, have the same regulation sign on the target gene *Z* (both positive or both negative). In contrast, a double-negative feedback loop will often have opposing regulation signs for *Z* (Figure 5.13). The two sign combinations shown in Figure 5.13 are coherent, in the sense that any two paths between two nodes have the same overall sign.

In addition to the regulating-feedback motif, developmental networks show a network motif in which a two-node feedback loop is regulated by an upstream transcription factor (Figure 5.15). This motif is called **regulated feedback**. Again, several coherent sign combinations are commonly found. For example, the input transcription factor can be an activating regulator that

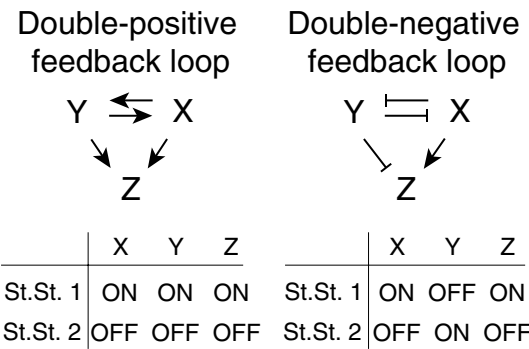


FIGURE 5.13

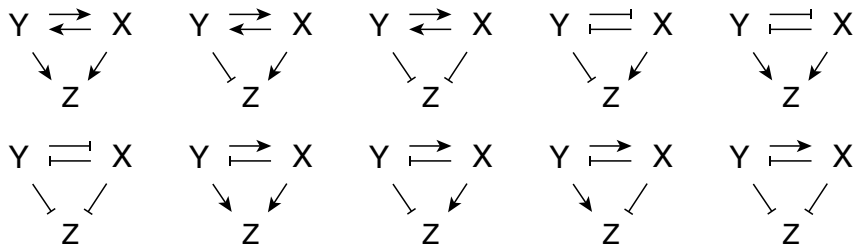


FIGURE 5.14

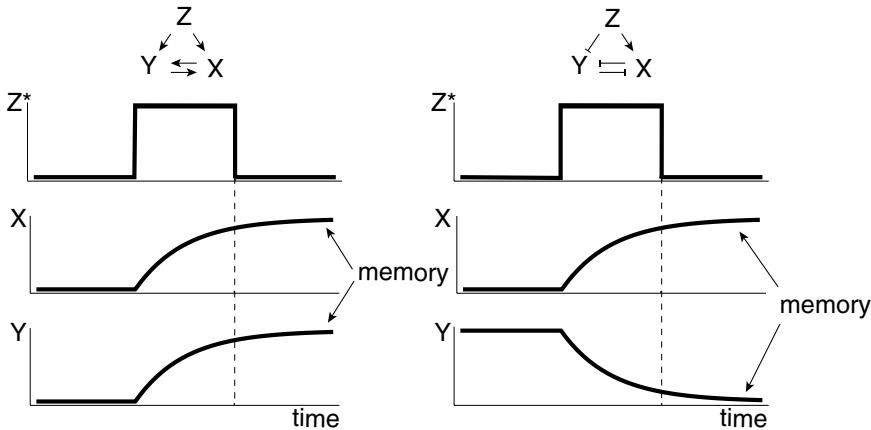


FIGURE 5.15

locks the system ON in the case of a double-positive loop (Davidson et al., 2002). In the case of a double-negative loop, the regulator can have different signs for the two feedback nodes and act to switch the system from one steady state to the other (Gardner, Cantor and Collins, 2000).

As in the case of PAR, the regulated feedback motif can be considered as a memory element: the regulator Z can switch the feedback loop from one state to another, such that the state persists even after Z is deactivated (Figure 5.15). Hence, the circuit can remember whether Z was active in the past. This memory can help cells maintain their fate even after the original developmental signals that determined the fate have vanished (Demongeot, Kaufman and Thomas, 2000; Smolen, Baxter and Byrne, 2000; Xiong and Ferrell, 2003).

5.1.4 Long Transcription Cascades and Developmental Timing

An additional important family of network motifs in developmental networks (that is rare in sensory networks) is long **transcriptional cascades**. Transcriptional cascades are chains of interactions in which transcription factor X regulates Y , which in turn regulates Z , and so on (Figure 5.16).

As we saw in Chapter 1, the response time of each stage in the cascade is governed by the removal rate of the protein at that stage, $T_{1/2} = \log(2)/\alpha$. Long cascades are too slow for sensory transcription networks that need to respond quickly to environmental stresses and nutrients.

Developmental networks, in contrast, work on precisely the scale of one or a few cell generations. This is because cell fates are assigned with each cell division (or several divisions) as the cells divide to form the tissues of the embryo. Hence, the timescale of transcription cascades is well suited to guide developmental processes.

Development often employs **cascades of repressors** (Figure 5.16), whose timing properties are more robust with respect to fluctuations in protein production rates than cascades of activators (Rappaport et al., 2005). For example, neurons in the fly brain develop in a series of well-timed steps coordinated by a cascade of repressors (Averbukh et al., 2018).

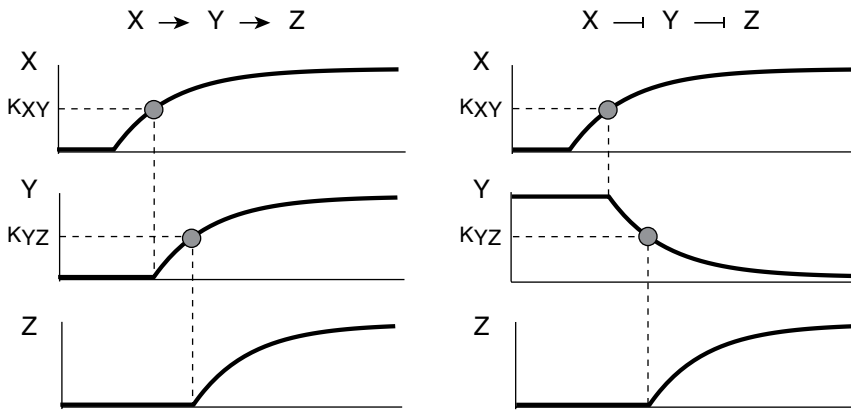


FIGURE 5.16

5.2 NETWORK MOTIFS IN PROTEIN–PROTEIN INTERACTION NETWORKS

So far, we have focused on transcription networks that operate on timescales as slow as the cell's generation time. To elicit rapid responses, the cell also contains much faster information-processing networks, called **protein–protein interaction networks** or **PPI** for short.

PPI networks are fast because they work by interactions between existing proteins. No new proteins need to be made. In PPI, proteins activate and inhibit each other by various means: proteins can bind each other, chemically modify each other or even degrade each other. Their typical timescale is seconds to minutes, compared to hours for transcription networks.

PPI circuits have their own network motifs, with delightfully intricate design. We will discuss their emerging design principles in Part 2 of this book.

5.2.1 Hybrid Network Motifs Include a Two-Node Negative Feedback Loop

For now, it is useful to note that PPI networks and transcription networks operate in an integrated fashion. Many PPI circuits have transcription factors as their output, and proteins in a PPI network are regulated by transcription networks. The integrated network can be described as a network with two colors of edges: one color represents transcription interactions, and a second color represents the much faster protein–protein interactions (Yeger-Lotem et al., 2004). In this section, we will mention some of the network motifs that occur in such two-color networks. Such hybrid network motifs can also be found in networks that integrate more than two levels of interactions (Ptacek et al., 2005; Zhang and Horvath, 2005).

A very common hybrid motif is a feedback loop made of two proteins that interact with each other using two colors of arrows (Figure 5.17). In this motif, protein X is a transcription factor that activates the transcription of gene Y. The protein product Y interacts with X on the protein level (not transcriptionally), often in a negative fashion. This negative regulation can take several forms. In some cases, Y enhances the rate of degradation of protein X. In other cases, Y binds X and inhibits its activity as a transcription factor by preventing its access to the DNA. In metabolic pathways, the final product usually negatively regulates the transcription of

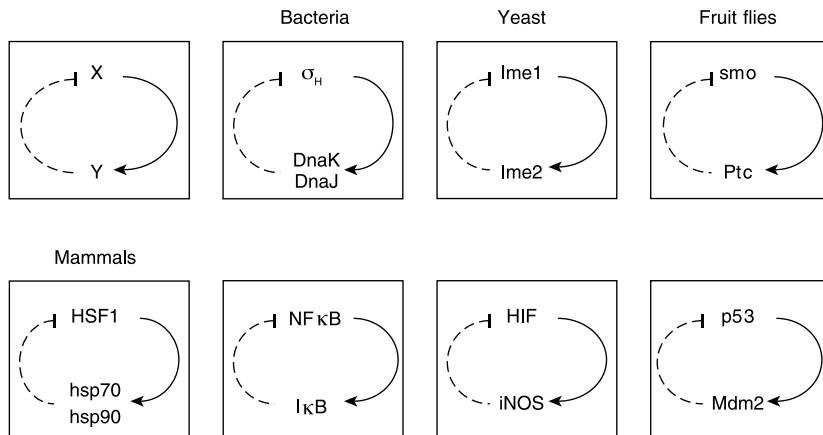


FIGURE 5.17

the pathway enzymes, as well as the activity of the first enzyme in the pathway (feedback inhibition, Figure 4.2). This hybrid motif occurs in most gene systems from bacteria to humans (Lahav et al., 2004).

In the hybrid negative feedback loop, one interaction, transcription, is slow and the other is fast. Note that purely transcriptional two-node negative feedback loops are rare (developmental transcription networks usually display *positive* transcriptional feedback loops, as discussed above). In other words, it is rare for *Y* to repress *X* on the transcription level. What could be the reason that hybrid negative feedbacks are much more common than purely transcriptional ones?

To understand hybrid feedback, we can turn to engineering control theory. Engineers routinely use feedback in which a slow component is regulated by a fast one. The principal use of this type of feedback is to stabilize a system around a desired set point. For example, a heater that takes 15 min to heat a room is controlled in a negative feedback loop by a much faster thermostat (Figure 5.18). The thermostat compares the desired temperature to the actual temperature and adjusts the power accordingly. If the temperature is too high, the power of the heater is reduced so that the room cools down. After some time, the temperature stabilizes around the desired temperature (Figure 5.19, overdamped or damped curves).

One reason for using two timescales (fast thermostat on a slow heater) in this feedback loop is enhanced stability. The rapid response time of the thermostat ensures that the control of the heater is based on the current temperature. Had the thermostat been made of a vat of mercury that takes 15 min to respond to temperature changes, the heater would receive feedback based on the relatively distant past, and the temperature could oscillate (Figure 5.19, undamped oscillations).

By analogy, a negative feedback loop made of two slow transcription interactions is more prone to instability than a feedback loop where a fast interaction controls a slow one. In biology, stability around a set point is called **homeostasis**, and is a central feature of living

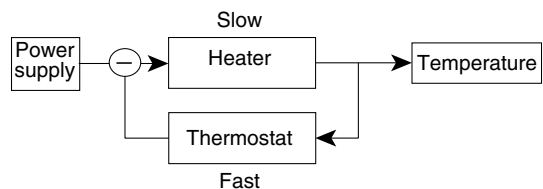


FIGURE 5.18

systems. For an excellent introduction to the mathematical treatment of stability, see Strogatz (2015). We will discuss oscillatory biological circuits in more depth in the next chapter.

5.2.2 Hybrid FFL Motifs Can Provide Transient Memory

Hybrid networks also show FFL motifs in which, for example, X transcriptionally activates Y and Z , which interact at the protein level (Figure 5.20). Such a hybrid FFL appears in the mating pathway of yeast, and was studied experimentally by Andreas Doncic and Jan Skotheim (2013). Here, the input signal S_x is a pheromone made by another yeast cell, and the output is expression of gene Z that causes the cells to stop dividing and prepare to mate. The signal S_x causes transcription factor X to activate the production of Y and Z . S_x also causes Y to chemically modify Z into an active state. Y is a kinase that **phosphorylates** Z by adding a phosphoryl group PO_4^- . Thus, it converts Z into its active form Z_p . Phosphorylation is a common protein–protein interaction that will appear often in the circuits of the coming chapters.

The hybrid FFL provides a persistence-detector function (as in Chapter 3), because the mating signal S_x must be present for enough time in order for protein Y to build up and activate Z . Doncic and Skotheim further showed that the hybrid FFL provides a **transient memory**. It allows yeast to respond more rapidly to a mating signal S_x if that signal was seen in the recent past – because then Y protein is already present, ready to phosphorylate Z , without need to make new Y proteins. Here, “recent past” is determined by the half-life of Y , a few hours. In this way, node Y carries a transient memory of recent signals.

5.2.3 Feedforward Loops Show a Milder Version of the Functions of Feedback Loops

Let’s pause to note a similarity between the functions of feedforward and feedback loops. Both coherent FFLs and positive feedback loops can provide delays and memory. FFLs provide transient memory and positive feedback can provide long-term memory.

Likewise, both incoherent FFLs (IFFLs) and negative feedback loops can speed responses and show pulses. IFFLs show a single pulse and can never on their own show the oscillations that negative feedback loops are capable of, because FFLs are always stable.

Thus, coherent FFLs are similar to positive feedback, and incoherent FFLs to negative feedback, but FFLs show a more stable, reversible or mild version of the dynamics of feedback loops.

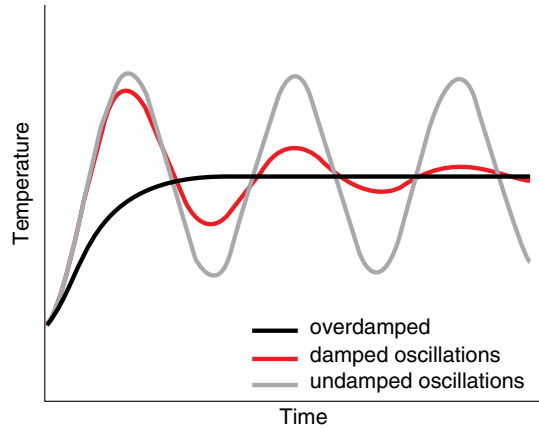


FIGURE 5.19

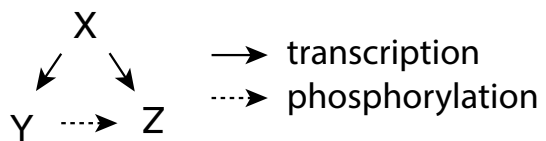


FIGURE 5.20

5.3 NETWORK MOTIFS IN NEURONAL NETWORKS

To end this chapter, let's expand our view beyond regulatory networks in the cell. Many fields of science deal with complex networks of interactions. Sociology studies social networks, in which $X \rightarrow Y$ means, for example, that X chooses Y as a friend (Holland and Leinhardt, 1977; Wasserman and Faust, 1994). Neurobiology studies neuronal networks in which neuron X has synapses to neuron Y . Ecology studies food webs, in which species X eats species Y . Network motifs can be sought in networks from these fields by comparing them to randomized networks (Milo et al., 2002; Milo et al., 2004). This is an example of a concept developed in systems biology that has impacted many other fields of research. One finds that:

1. Most real-world networks are built of a small set of network motifs.
2. Networks from a given field share the same network motifs.
3. Different fields show different network motifs.

Examples can be seen in [Figure 5.21](#). Food webs show cascades, whereas electronic circuits show feedback loops of transistors. In social networks, X might choose Y as a friend, but a

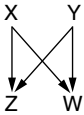
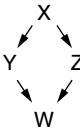
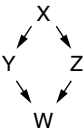
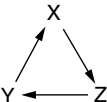
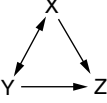
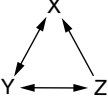
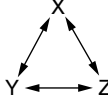
Transcription networks	 Feedforward loop	 Bi-fan	
Neurons	 Feedforward loop	 Bi-fan	 Diamond
Food webs	 Cascade	 Diamond	
Electronic circuits (fractional multipliers)	 Three-node feedback loop	 Bi-fan	 Four-node feedback loop
Social Networks	 Regulating mutual	 Semi-Clique	 Clique

FIGURE 5.21 Adapted from (Milo et al. 2002).

tragedy of life is that *Y* does not always choose *X* back. Perhaps that is one reason that people like to group together into fully connected cliques, a pattern rarely seen in other fields.

It makes sense that networks from different fields display different network motifs, because each field has its own functions and its own constraints.

However, there are intriguing cases in which *unrelated networks share similar network motifs* (and similar anti-motifs, patterns that are rarer than at random). This occurs, for example, when studying the neuronal network of *C. elegans*, a tiny worm composed of about 1000 cells, of which 300 are neurons. The 300 neurons are connected by about 5000 synaptic connections.

This network was fully mapped by White, Brenner and colleagues (White et al., 1986) using electron microscopy. The wiring does not vary much from individual to individual. Strikingly, the neuronal network of *C. elegans* shares many of the motifs found in transcription networks (Milo et al., 2002). For example, the most significant three-node motif in the neuronal network is the FFL. FFLs are also a recurring motif in mammalian neuronal wiring, for example IFFLs in which neuron *X* sends an axon to a distant brain region where it activates both neuron *Z* and its neighboring inhibitory neuron *Y*.

The similarity of network motifs in neuronal and transcription networks is surprising because the networks operate on very different spatial and temporal scales. Neuronal networks are made of micron-sized cells interacting on the sub-second timescale. Transcription networks are made of nanometer-sized molecules interacting over hours. Yet many of the motifs are similar.

Why is the FFL a motif in both neuronal and transcription networks? One point of view is that this is a coincidence. For example, White et al. (1986) raised the possibility that triangular patterns such as the FFL might arise due to the spatial arrangement of the neurons. Neurons that are neighbors tend to connect more often than distant neurons. Such neighborhood effects can produce triangle-shaped patterns, because if *Y* and *Z* are both neighbors of *X*, they are likely to neighbor each other. However, while this effect would produce FFLs, it would also produce three-node feedback loops, in which *X*, *Y* and *Z* are connected in a cycle (Itzkovitz and Alon, 2005). Such feedback loops, however, are rare in this neuronal network: they are in fact anti-motifs. Thus, the origin of FFLs and other motifs in the neuronal network is not solely due to neighborhood effects.

An alternative view is that the similarity in network motifs reflects the fact that both networks evolved toward a similar goal: they perform information-processing on noisy signals using noisy components. Neurons process information between sensory neurons and motor neurons. Transcription networks process information between transcription factors that receive signals and genes that act on the inner or outer environment of the cell. This similarity in function raises the hypothesis that evolution converged on similar network motifs in both networks to perform important signal-processing tasks.

5.3.1 Multi-Input FFLs in Neuronal Networks

Although FFLs are motifs in both transcription and neuronal networks, the way FFLs are joined together to make larger circuits is different.

Instead of the multi-output FFLs found in transcription networks, neurons in *C. elegans* often combine to make **multi-input FFLs**. An example is shown in Figure 5.22: sensory neurons for nose touch and noxious odors both signal to motor neurons that make the worm move backwards. Such a circuit can function as a coincidence detector, generating an output from brief input stimuli that occur in close proximity, even if each stimulus alone is not enough to generate an output (Exercise 5.7).

An interesting multi-input FFL occurs in the neurons for human pain sensation (Figure 5.23). This circuit explains why there are two types of pain. Here is a description adapted from Robert Sapolsky's book, *Behave* (Sapolsky, 2017).

Neuron X_1 sits just below the skin and fires in response to painful stimuli. It excites neuron Z which projects up the spinal cord, sending a pain message to the brain. But neuron X_1 also excites neuron Y which inhibits Z . This is an I1-FFL, a pulse generator. As a result, neuron Z fires for a while and is then silenced. You feel a sharp pain, as if you've been poked by a needle.

The second input neuron is X_2 , located under the skin in the same region as X_1 . X_2 also excites neuron Z , and the message is sent up to the brain. But X_2 *inhibits* Y , forming a type-4 coherent FFL. As do all coherent FFLs, this circuit starts to fire only after a delay and then keeps going. You feel a throbbing, continuous pain, like a burn or abrasion. The delay is strengthened by the fact that action potentials travel down the axon of X_2 much slower than X_1 . So the pain from X_1 is transient and fast, whereas the pain from X_2 is long-lasting and has a slower onset.

The two circuits can interact, and we often intentionally force them to. Suppose you have a continuous, throbbing pain – say, an insect bite. How can you stop the throbbing? Briefly stimulate X_1 , which shuts the system down for a while. And that is precisely what we often

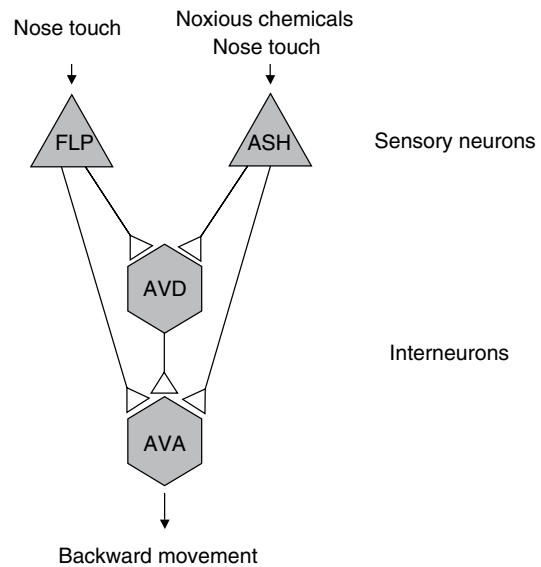


FIGURE 5.22

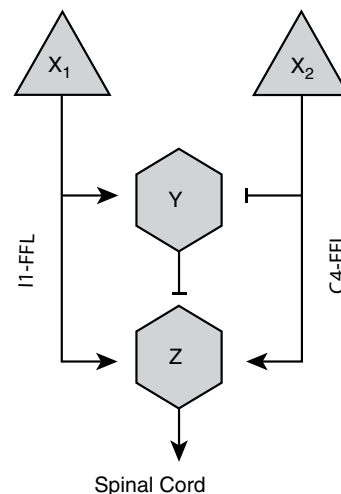


FIGURE 5.23

do in such circumstances, we scratch hard right around the bite to dull the pain, and the slow chronic pain pathway is shut down for up to a few minutes.

Sapolsky writes: “One of the biggest reasons why I love this circuit is that it was first proposed in 1965 by the great neurobiologists Ronald Melzack and Patrick Wall. It was merely proposed as a theoretical model – ‘No one has ever seen this sort of wiring, but we propose that it’s got to look something like this, given how pain works’. And subsequent work showed that’s precisely how this part of the nervous system is wired.”

5.4 REFLECTION

The network representation masks the detailed mechanisms at each node and arrow. But because of this compact level of description, network representations help to highlight the similarity in the circuit patterns in different parts of the network and between different networks. The dynamics of the networks at this level of resolution lend themselves to analysis with simple models. We care only that *X* activates or inhibits *Y*, not precisely how it does so on the biochemical level. This abstraction helps us to define network motifs as functional building blocks of each type of network. These building blocks are often joined together in ways that allow understanding of the network dynamics in terms of the dynamics of each individual motif. Hence, both on the level of local patterns and on the level of combinations of patterns into larger circuits, biological networks appear to display a degree of simplicity.

FURTHER READING

Bistability and Nullclines

(Strogatz, 2015) “Nonlinear dynamics and chaos.”

(Fall et al., 2004) “Computational cell biology.”

Developmental Networks

(Lawrence and Peter, 1995) “The making of a fly: the genetics of animal design.”

(Levine and Davidson, 2005) “Gene regulatory networks for development.”

Hybrid Networks

(Yeger-Lotem et al., 2004) “Network motifs in integrated cellular networks of transcription–regulation and protein–protein interaction.”

Network Motifs in Diverse Networks

(Milo et al., 2002) “Network motifs: simple building blocks of complex networks.”

(Milo et al., 2004) “Superfamilies of designed and evolved networks.”

Neuronal Networks

(Bargmann, 1998) “Neurobiology of the *Caenorhabditis elegans* genome.”

(Hope, 1999) “*C. elegans*: a practical approach.”

(Sapolsky, 2017) “Behave: the biology of humans at our best and worst.”

EXERCISES

5.1 *Alternating stability:*

- Design an autoregulatory circuit $dX/dt = f(X) - \alpha X$ with 5 fixed points. Use a rate plot.
- What is the stability of each point? *Hint:* See if production exceeds removal in the regions between the fixed points, and find out the flow direction of X .
- Explain why no two adjacent fixed points can have the same stability (both unstable or both stable). Thus, adjacent fixed points must have alternating stability.

5.2 *Nullclines for double-positive feedback:* The double positive feedback loop is defined by two increasing input functions, f and g , that describe the mutual repression:

$$\frac{dX}{dt} = f(Y) - \alpha_1 X$$

$$\frac{dY}{dt} = g(X) - \alpha_2 Y$$

- Explain the terms in these equations.
 - Use nullclines to show that a double positive feedback loop can be mono-stable (single stable fixed point) or bistable.
 - Show that bistability requires cooperative (sigmoidal) input functions.
- 5.3 *Positive autoregulation in a double-positive feedback loop:* Consider a two-node positive feedback loop in which each node is also positively autoregulated (PAR). In this circuit, the interactions are not cooperative. Hence, without the PAR, there would be no bistability, and instead only a single steady-state level.
- Write equations for this system. Assume that autoregulation and cross-activation are multiplicative (resembling an AND gate).
 - Draw the nullclines. Show that the circuit can show bistability.
- 5.4 *Tristability:* Toggle switches can show **tristability**, in which there are three stable states: the usual $X_{\text{high}}/Y_{\text{low}}$ and its opposite $X_{\text{low}}/Y_{\text{high}}$, and a third state in which both X and Y are expressed at intermediate levels (Huang et al., 2007). Tristability can occur when the autoregulation of X and Y is strong and cooperative enough. The nullclines have the fancy shape shown in Figure 5.24. There are five crossing points, two unstable and three stable.

Consider a toggle switch in which the inhibition function of X by Y is $f(Y) = 1/(1 + Y/K_1)$ and the autoregulation function is sigmoidal $g(X) = \beta_0 + \beta_1(X/K_2)^2/(1 + (X/K_2)^2)$. The inhibition and autoregulation functions combine multiplicatively (resembling an

AND gate). The interaction functions for Y are the same as for X (symmetric circuit).

- Write the equations for this circuit.
- Write the equations for the nullclines.
- Sketch the nullclines. How many intersection points can they have for different choices of parameters?
- Find a parameter set with tristability. Plot the nullclines in this case.

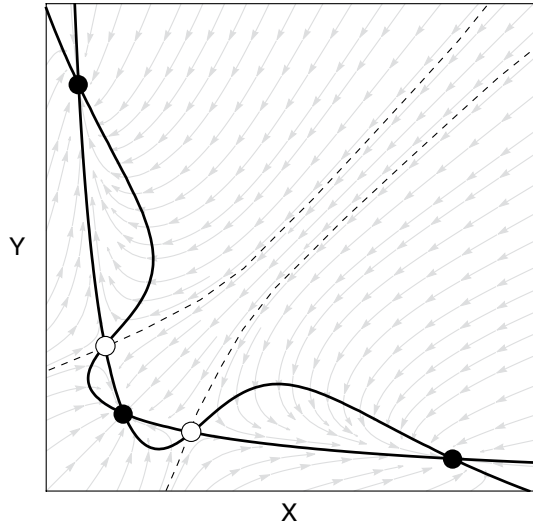


FIGURE 5.24

- Is symmetry essential for tristability? If not, sketch non-symmetric nullclines with tristability.
- Suggest a biological scenario in which tristability might be useful.

5.5 *Phase portraits*: Find an online program that plots phase portraits for sets of two ordinary differential equations. Plot the phase portraits for the systems of Exercises 5.2 and 5.3.

5.6 *Memory in the regulated feedback network motif*: Transcription factor X activates transcription factors Y_1 and Y_2 which mutually activate each other. The input function at the Y_1 and Y_2 promoters is an OR gate (e.g., Y_2 is activated when either X or Y_1 bind the promoter). At time $t = 0$, X begins to be produced from an initial concentration of $X = 0$. Initially, $Y_1 = Y_2 = 0$. All production rates are $\beta = 1$ and removal rates are $\alpha = 1$. All of the activation thresholds are $K = 0.5$. At time $t = 3$, production of X stops.

- Plot the dynamics. What happens after X decays away?
- Consider the same problem, but now Y_1 and Y_2 repress each other and X activates Y_1 and represses Y_2 . At time $t = 0$, X begins to be produced, and the initial levels are $X = 0$, $Y_1 = 0$ and $Y_2 = 1$. At time $t = 3$, X production stops. Plot the dynamics of the system. What happens after X decays away?

5.7 *Multi-input FFL as a coincidence detector*: A multi-input FFL is made of two coherent type-1 FFLs with inputs X_1 and X_2 , SUM logic at Y and $(X_1 \text{ OR } X_2) \text{ AND } Y$ logic at the output Z . Assume all parameters are equal to one. Write equations using logic input functions. The analog of the degradation/dilution rates α for neuronal circuits are the relaxation time of the membrane potential.

- a. Show that a simultaneous brief stimulation of X_1 and X_2 can lead to an output, even if each input stimulated alone would not.
 - b. Show that (a) can work even if the brief input pulses are timed with a small delay.
 - c. Why can this circuit be called a coincidence detector for the two inputs?
- 5.8 *Transient memory in the FFL*: A coherent type-1 FFL with AND logic responds to the signal S_x . The signal is present for a long time, and then vanishes for a time D . The signal then goes ON again.
- a. What is the response time of the output Z as a function of D ?
 - b. Why can we say that this circuit provides short-term memory for S_x ?
- 5.9 *Series of pulses*: Design a circuit made of linked FFLs that can provide a series of a given number n of pulses of expression of gene Z .
- 5.10 *Memory of multiple past inputs*: Design a circuit that provides a locked-ON output Z only if, in the past, signal S_1 was present for longer than time D , and then signal S_2 was present for longer than time D .
- 5.11 *No oscillations in FFLs*: Explain why FFLs can never show sustained oscillations.

BIBLIOGRAPHY

- Averbukh, I. et al. 2018. 'A repressor-decay timer for robust temporal patterning in embryonic *Drosophila* neuroblast lineages', *Elife*, pii: e38631. doi: 10.7554/eLife.38631.
- Bargmann, C.I. 1998. 'Neurobiology of the *Caenorhabditis elegans* genome', *Science*, 282(5396), pp. 2028–2033. doi: 10.1126/science.282.5396.2028.
- Boyer, L.A. et al. 2005. 'Core transcriptional regulatory circuitry in human embryonic stem cells', *Cell*, 122(6), pp. 947–956. doi: 10.1016/j.cell.2005.08.020.
- Davidson, E.H. et al. 2002. 'A genomic regulatory network for development', *Science*, 295(5560), pp. 1669–1678. doi: 10.1126/science.1069883.
- Demongeot, J., Kaufman, M. and Thomas, R. 2000. 'Positive feedback circuits and memory', *Comptes Rendus de l'Académie des Sciences - Series III - Sciences de la Vie*, 323(1), pp. 69–79. doi: 10.1016/S0764-4469(00)00112-8.
- Doncic, A. and Skotheim, J.M. 2013. 'Feedforward regulation ensures stability and rapid reversibility of a cellular state', *Molecular Cell*, 50(6), pp. 856–868. doi: 10.1016/j.molcel.2013.04.014.
- Fall, C.P. et al. eds. 2004. *Computational cell biology*. New York, NY: Springer New York (Interdisciplinary Applied Mathematics). doi: 10.1007/b97701.
- Gardner, T.S., Cantor, C.R. and Collins, J.J. 2000. 'Construction of a genetic toggle switch in *Escherichia coli*', *Nature*, 403(6767), pp. 339–342. doi: 10.1038/35002131.
- Holland, P.W. and Leinhardt, S. 1977. 'A dynamic model for social networks', *The Journal of Mathematical Sociology*, 5(1), pp. 5–20. doi: 10.1080/0022250X.1977.9989862.
- Hope, I.A. 1999. *C. elegans: a practical approach*. Oxford University Press.
- Hornung, G. and Barkai, N. 2008. 'Noise propagation and signaling sensitivity in biological networks: a role for positive feedback', *PLoS Computational Biology*, 4(1), p. e8. doi: 10.1371/journal.pcbi.0040008.

- Huang, S. et al. 2007. 'Bifurcation dynamics in lineage-commitment in bipotent progenitor cells', *Developmental Biology*, 305, pp. 695–713. doi: 10.1016/j.ydbio.2007.02.036.
- Itzkovitz, S. and Alon, U. 2005. 'Subgraphs and network motifs in geometric networks', *Physical Review E*, 71(2), p. 5663. doi: 10.1103/PhysRevE.71.026117.
- Lahav, G. et al. 2004. 'Dynamics of the p53-Mdm2 feedback loop in individual cells', *Nature Genetics*, 36(2), pp. 147–150. doi: 10.1038/ng1293.
- Lawrence, P.A. and Peter, A. 1995. *The making of a fly: the genetics of animal design*, 1st edn. Blackwell Science.
- Levine, M. and Davidson, E.H. 2005. 'Gene regulatory networks for development', *Proceedings of the National Academy of Sciences*, 102(14), pp. 4936–4942. doi: 10.1073/pnas.0408031102.
- Milo, R. et al. 2002. 'Network motifs: simple building blocks of complex networks', *Science*, 298(5594), pp. 824–827. doi: 10.1126/science.298.5594.824.
- Milo, R. et al. 2004. 'Super families of evolved and designed networks', *Science*, 303(5663), pp. 1538–1542. doi: 10.1126/science.1089167.
- Odom, D.T. et al. 2006. 'Core transcriptional regulatory circuitry in human hepatocytes', *Molecular Systems Biology*, 2, p. 3495. doi: 10.1038/msb4100059.
- Penn, B.H. et al. 2004. 'A MyoD-generated feed-forward circuit temporally patterns gene expression during skeletal muscle differentiation', *Genes and Development*, 18(19), pp. 2348–2353. doi: 10.1101/gad.1234304.
- Ptacek, J. et al. 2005. 'Global analysis of protein phosphorylation in yeast', *Nature*, 438(7068), pp. 679–684. doi: 10.1038/nature04187.
- Ptashne, M. and Gann, A. 2002. *Genes & signals*. Cold Spring Harbor Laboratory Press. Available at: https://books.google.co.il/books/about/Genes_Signals.html?id=IvMQRwA3bEAC&redir_esc=y (Accessed: November 25, 2018).
- Rappaport, N., Winter, S., Barkai, N. 2005. 'The ups and downs of biological timers', *Theoretical Biology and Medical Modelling*, 2(1), p. 22.
- Sapolsky, R.M. 2017. *Behave: the biology of humans at our best and worst*. Penguin Press.
- Smolen, P., Baxter, D.A. and Byrne, J.H. 2000. 'Mathematical modeling of gene networks', *Neuron*, 26(3), pp. 567–580. doi: 10.1016/S0896-6273(00)81194-0.
- Strogatz, S.H. 2015. *Nonlinear dynamics and chaos: with applications to physics, biology, chemistry, and engineering*, 2nd edn. Westview Press.
- Thomas, R. and D'Ari, R. 1990. *Biological feedback*. CRC Press, Inc.
- Wasserman, S. and Faust, K. 1994. *Social network analysis: methods and applications*. Cambridge University Press.
- White, J.G. et al. 1986. 'The structure of the nervous system of the nematode *Caenorhabditis elegans*', *Philosophical Transactions of the Royal Society B: Biological Sciences*, 314(1165), pp. 1–340. doi: 10.1098/rstb.1986.0056.
- Xiong, W. and Ferrell, J.E. 2003. 'A positive-feedback-based bistable "memory module" that governs a cell fate decision', *Nature*, 426(6965), pp. 460–465. doi: 10.1038/nature02089.
- Yeger-Lotem, E. et al. 2004. 'Network motifs in integrated cellular networks of transcription-regulation and protein-protein interaction', *Proceedings of the National Academy of Sciences*, 101(16), pp. 5934–5939. doi: 10.1073/pnas.0306752101.
- Zhang, B. and Horvath, S. 2005. 'A general framework for weighted gene co-expression network analysis', *Statistical Applications in Genetics and Molecular Biology*, 4(1). doi: 10.2202/1544-6115.1128.



Taylor & Francis

Taylor & Francis Group

<http://taylorandfrancis.com>

How to Build a Biological Oscillator

Oscillations are thrilling – hearts beat, cells divide every cell cycle, circadian clocks keep time and neurons click in trains of regularly spaced spikes. Biological oscillations attracted theoretical work from pioneers such as Arthur Winfree, Albert Goldbeter and John Tyson, and are still an active area of research. Let's discuss the design principles of biological circuits that oscillate.

6.1 OSCILLATIONS REQUIRE NEGATIVE FEEDBACK AND DELAY

At the heart of an oscillator is a negative feedback loop. Molecule X acts to reduce its own amounts, so that high levels go to low and then back to high, and so on. Negative feedback on its own, however, is not enough. The simplest negative feedback motif, negative autoregulation, does not oscillate but instead monotonically returns to steady state as we saw in [Chapter 1](#) ([Figure 6.1](#)).

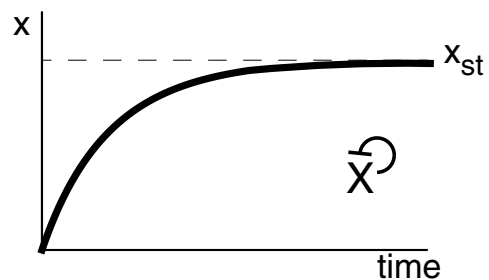


FIGURE 6.1

6.1.1 In Order to Oscillate, You Need to Add a Sizable Delay to the Negative Feedback Loop

Negative feedback plus delay reminds me of the shower we had when I was a child. The water started cold. I would turn on the hot water – acting as a feedback controller. But the hot water took time to arrive, so I would turn the handle too much, and the water would be scalding – Ouch! So I would turn it back strongly to the cold, but because of the delay, I would go too far and the water would be freezing – Arrgh! And so on in a cycle of Ouch! Arrgh! Ouch! Fortunately, in our more modern shower, there is less delay and I can easily tune the desired temperature.

A delay in biological circuits can be achieved by adding components in the negative feedback loop to make longer paths in the circuit. Autoregulation is just a one-step path, a single self-closing arrow. Add another species, Y , and you get a negative feedback loop made of two arrows (Figure 6.2). Here, you start to see a hint of oscillations: you can get damped oscillations with pulses that settle down to steady state (Figure 6.3). Damped oscillations require (1) strong (preferably cooperative) feedback and (2) that the timescales on the two arrows are similar. If the timescales are very different, with one arrow much faster than the other, the fast path is not much of a delay element, and the circuit is overdamped: it acts effectively like autoregulation with no damped oscillations (Figure 6.4).

An easy way to grasp the roles of feedback strength and timescales is to use **linear stability analysis**, as in the solved example “below. Linear stability analysis is nicely introduced in Strogatz (2015). If you don’t care for details right now, skip the example.

Solved Example 6.1: Show That a Two-Component Negative Feedback Loop Shows Damped Oscillations If Timescales Are Similar Enough and Feedback Is Strong Enough

Consider a two-node negative feedback circuit in which X activates Y according to the increasing function $g(X)$ and Y represses X according to the decreasing function $f(Y)$ (Figure 6.5)

$$\frac{dX}{dt} = f(Y) - \alpha_1 X$$

$$\frac{dY}{dt} = g(X) - \alpha_2 Y$$

The α ’s are the removal rates that set the timescales of the two arrows. The dynamics of small perturbations $x(t)$ and $y(t)$ around the fixed point X_{st} , Y_{st} are governed by the linear equations

$$\frac{dx}{dt} = -\beta_1 y - \alpha_1 x \quad \frac{dy}{dt} = \beta_2 x - \alpha_2 y$$

where the feedback parameters β_1 and β_2 are the derivatives of f and g at the fixed point: $-\beta_1 = df/dY$, and $\beta_2 = dg/dX$. Note that $-\beta_1$ is negative because f is a decreasing

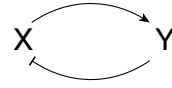


FIGURE 6.2

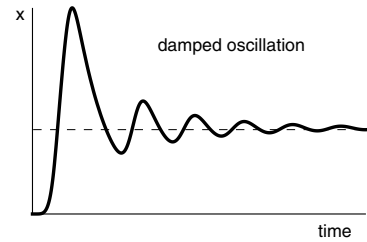


FIGURE 6.3

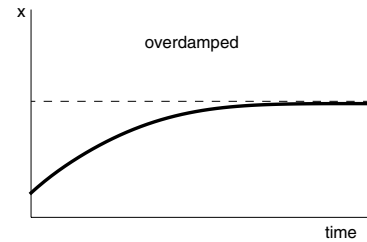


FIGURE 6.4

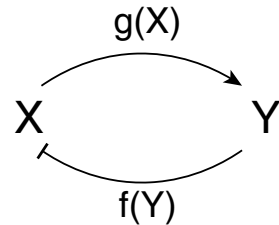


FIGURE 6.5

function (repression), whereas β_2 is positive because g is increasing (activation). Thus, this dynamical system can be described in matrix notation by

$$\frac{d}{dt} \begin{pmatrix} x \\ y \end{pmatrix} = J \begin{pmatrix} x \\ y \end{pmatrix}, \quad J = \begin{bmatrix} -\alpha_1 & -\beta_1 \\ \beta_2 & -\alpha_2 \end{bmatrix}$$

where J is the **Jacobian matrix**. For a general system, $dx/dt = f(x,y)$, $dy/dt = g(x,y)$, the Jacobian is given by the derivatives evaluated at the fixed point,

$$J = \begin{bmatrix} df/dx & df/dy \\ dg/dx & dg/dy \end{bmatrix}$$

The solution of such linear equations is a sum of exponentials of time: $c_1 e^{\lambda_1 t} + c_2 e^{\lambda_2 t}$. The exponents are the two eigenvalues λ_1 and λ_2 of the matrix J . Damped oscillations (Figure 6.3) occur when the eigenvalues have an imaginary part: $\lambda_{1,2} = a \pm ib$. To see this, recall from Euler's formula that $e^{a+ib} = e^a(\cos(b) + i \sin(b))$, so that the solutions are an oscillating wave with frequency determined by the imaginary part $\omega = 2\pi/b$, and an amplitude that decays exponentially to zero if the real part is negative, $a < 0$. The result is known as a **spiral fixed point**, because the system spirals down into a steady-state solution (Figure 6.6).

To find the eigenvalues we solve the characteristic equation of the matrix J , $(-\alpha_1 - \lambda)(-\alpha_2 - \lambda) + \beta_1\beta_2 = 0$ to find $2\lambda_{1,2} = -(\alpha_1 + \alpha_2) \pm \sqrt{(\alpha_1 - \alpha_2)^2 - 4\beta_1\beta_2}$. The eigenvalues in this circuit always have a negative real part, thanks to the negative removal terms, and so the steady state is stable and all initial conditions flow back to it. A useful rule for stability of two-variable systems is that the sum of the diagonal terms τ (the trace of the Jacobian matrix) is negative and the determinant Δ is positive. This is the case here, $\tau = -(\alpha_1 + \alpha_2) < 0$ and $\Delta = \alpha_1\alpha_2 + \beta_1\beta_2 > 0$. In fact, the signs of τ and Δ stem from the sign structure of the Jacobian, $\begin{bmatrix} - & - \\ + & - \end{bmatrix}$, which always leads to stability.

Damped oscillations occur when the eigenvalues have imaginary parts, which happens when the term inside the square root is negative, namely

$$(\alpha_1 - \alpha_2)^2 < 4\beta_1\beta_2$$

The term on the right-hand side is called the **feedback strength**, $\beta_1\beta_2$. When the timescales of the two arms are equal ($\alpha_1 = \alpha_2$), damped oscillations always occur, for any feedback strength.

The larger the mismatch in timescales $|\alpha_1 - \alpha_2|$, the larger the feedback strength $\beta_1\beta_2$ needed for damped oscillations. If feedback is not strong enough compared

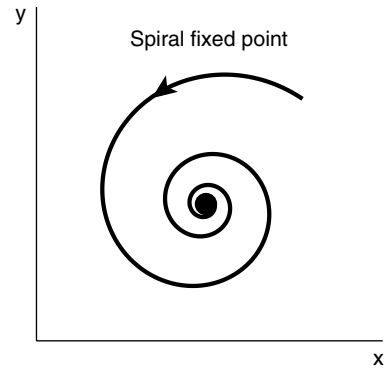


FIGURE 6.6

to the timescale separation, the system is **overdamped** and decays monotonically to the fixed point with no overshoot (Figure 6.4). Thus, strong separation of timescales counteracts the tendency to oscillate.

The feedback strength $\beta_1\beta_2$ is determined by the slopes of the regulation functions g and f at the steady-state point, dg/dX and df/dY . The steeper these regulation functions – for example the higher their Hill coefficient, the greater the feedback strength. ***Cooperativity therefore enhances the tendency to oscillate.***

Cooperativity helped the oscillations in my childhood shower, because the faucet had a very steep curve – very hot or very cold for most of the range, making it harder to home in on the right temperature.

Thus, a two-step negative feedback loop can only show damped oscillations. This observation prompted Galit Lahav, when she was a postdoc in my group, to try to visualize these damped oscillations in living cells. She used a feedback loop that involves an important protein known as the “guardian of the genome,” p53.

p53 is called the guardian of the genome because it governs cell decisions when DNA is damaged. The cell must decide to repair the DNA, or, if it is too damaged, to avoid becoming cancerous by committing programmed cell death or becoming a zombie-like senescent cell that stops dividing. p53 is a transcription factor that regulates genes for repair and for cell death/senescence. That is why p53 is mutated in most cancers, bypassing cell death and allowing cancer cells to proliferate despite damage. p53 forms a negative feedback loop with another protein called mdm2: p53 transcriptionally activates mdm2, and mdm2 leads to the degradation of p53 (Figure 6.7).

Galit Lahav fused the genes for p53 and mdm2 to cyan and yellow fluorescent proteins. That way, she could see in the microscope how cyan and yellow fluorescence varied in individual human cells over time, reporting the changes in the two proteins. Seeing the proteins in individual cells was an advance over the way experiments on p53 had been done before, by averaging over millions of cells, and thus potentially masking dynamic processes.

Galit gave the cells some gamma irradiation to induce DNA damage, and filmed the cells. She even brought a rollaway bed and a sleeping bag to the lab because she had to focus the microscope every 20 minutes over 24 h of filming. (This is heroic. A year later we got a microscope with automated focus.)

To her surprise, Galit did not see damped oscillations, but instead full-fledged oscillations that did not damp out. p53 goes in and out of the nucleus with pulses that have noisy amplitudes and a relatively precise 6 h period (dynamics of four typical cells are shown in Figure 6.8). The second protein in the feedback loop, mdm2, also oscillates, with the opposite phase.

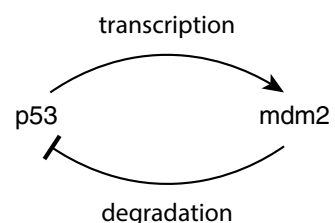


FIGURE 6.7

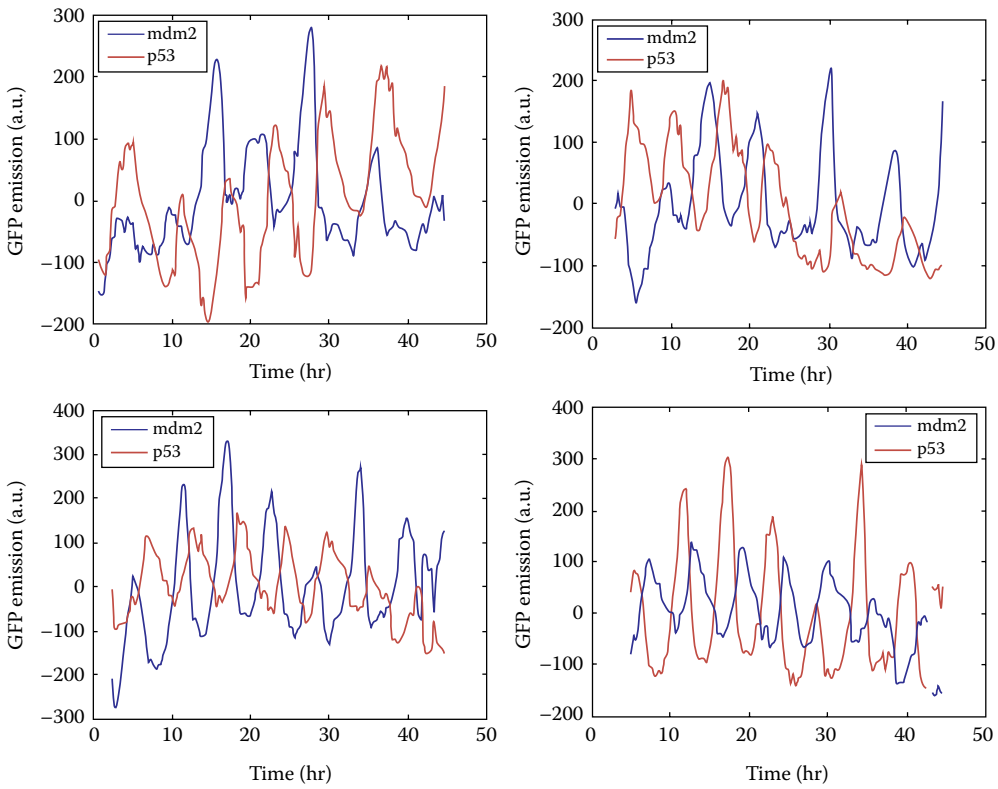


FIGURE 6.8 Adapted from (Geva-Zatorsky et al. 2010).

6.2 NOISE CAN INDUCE OSCILLATIONS IN SYSTEMS THAT HAVE ONLY DAMPED OSCILLATIONS ON PAPER

It took us a while to figure out what was going on. It turns out that circuits that show only damped oscillations on paper, such as a two-component negative feedback loop, can still oscillate indefinitely in the cell. This occurs when noise is strong enough. **Noise kicks the system away from the spiral fixed point and prevents the oscillations from damping out** (Figure 6.9). As a result, the circuit

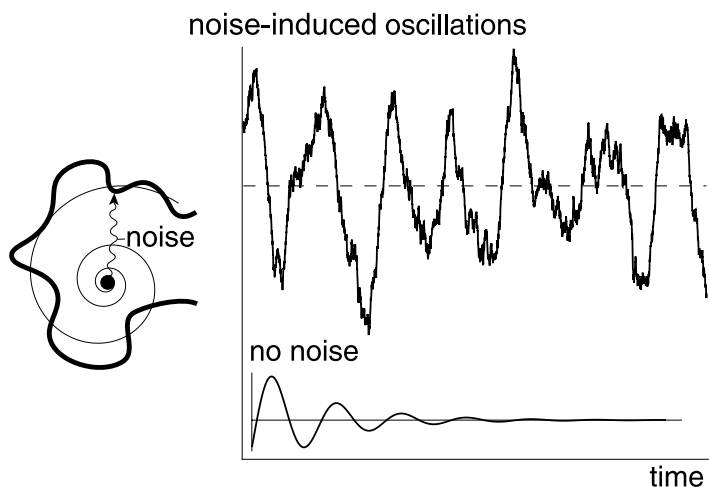


FIGURE 6.9

shows pulses with noisy amplitude but rather precise frequency. The frequency is that of the original (noiseless) damped oscillation, given by the imaginary part of the eigenvalues. Another way to think about this is that the damped oscillator amplifies the part of the noise close to its resonance frequency.

A diagnostic for such **noise-induced oscillations** is that the amplitudes of the pulses are more variable than their frequency (Figure 6.8), and that the amplitude increases with noise strength. Theoretical work on noise-driven oscillations around a spiral fixed point (Li and Lang, 2008) shows that the distribution of the peak amplitudes of the pulses goes as $P(A) \sim Ae^{-A^2/A_0^2}$ where A_0 is the ratio of the noise amplitude and the real part of the spiral-fixed-point eigenvalues. This formula describes the p53 pulses well (Geva-Zatorsky et al., 2010).

In recent years, many other transcription factors have been found to oscillate in and out of the nucleus. For example, the inflammation regulator NF- κ B shows oscillation pulses with noisy amplitude and accurate frequency like p53 (Nelson, 2004). Some transcription factors have multiple isoforms in the same cells (regulators with very similar DNA sequence), with some isoforms showing oscillations and others a graded response to a given signal (e.g., NFAT1 and NFAT4 in immune signaling, Yissachar et al., 2013).

One reason that transcription factors (TFs) may oscillate is to keep exciting their downstream genes that would otherwise show exact adaptation to TF level, as described in Chapter 10. Pulses “wake up” circuits that otherwise adapt.

Other TFs, such as Crz1 in yeast, show trains of pulses of nuclear entry, each lasting about 2 minutes, whose frequency increases with the input signal, while their amplitude does not depend on signal (Cai, Dalal and Elowitz, 2008). A potential utility of TF pulse trains is to coordinate the expression of different genes, because gene expression depends on the fraction of time the TF is in the nucleus rather than its amplitude, as discussed in Exercise 6.4.

6.3 DELAY OSCILLATORS

Full-fledged, undamped oscillations even without noise can appear if we go to feedback loops with three or more steps. In order to oscillate, such loops need to have strong feedback and similar timescales for the different steps.

A three-component negative feedback loop featured in one of the first theoretical models of biological oscillators, by Goodwin. Several decades later, in 2000, a three-component loop helped start the field of synthetic biology, when Michael Elowitz and Stanislas Leibler built a cycle of three repressors, called the **repressilator** (Figure 6.10). They combined the genes and regulatory regions of the three repressors in *E. coli* such that each represses the next repressor in the cycle. One of the repressors also regulated a green fluorescent protein gene (GFP) as a readout. To make sure the circuit parameters supported oscillations, Elowitz and Leibler made the timescales of the components as similar as possible. The repressilator oscillated in *E. coli*, with the GFP readout blinking green, black, green with a period of about 160 minutes. The repressilator was recently updated in a more minimal and precise version (Potvin-Trottier et al., 2016).

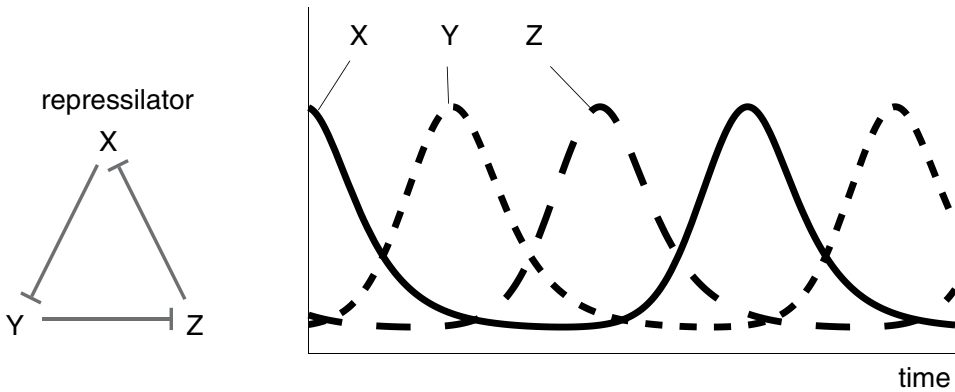


FIGURE 6.10

What happens when you add more than three components in the negative feedback loop? The more components in the cycle, the larger the range of parameters for oscillations, and the weaker the degree of cooperativity required for oscillations. The frequency of the oscillations in such **delay oscillators** is approximately equal to the sum of the half-lives of the components, the overall delay time around the loop. The circadian clock in cells is thought to be a delay oscillator, with a free-running period of about one day.

6.4 MANY BIOLOGICAL OSCILLATORS HAVE A COUPLED POSITIVE AND NEGATIVE FEEDBACK LOOP MOTIF

In sum thus far, negative feedback plus delays and/or noise can provide oscillations. Nonlinearity (cooperativity) and similar timescales for the opposing arms help the feedback loop to oscillate. But when we look at the circuits for the best-studied oscillators in biology, such as heart cells, neurons and cell cycles, we see an additional feature – a *positive feedback loop* is added to the negative feedback loop (Figure 6.11). What is the role of positive feedback?

The positive feedback loop adds a delay as we saw in Chapter 5. Delays help oscillations, increasing the parameter range that provides oscillations. For example, positive feedback can make a two-node negative feedback loop show sustained oscillations even with one arm much faster than the other. Separation of timescales between the interactions in the negative feedback loop is, in fact, a recurring feature of the oscillator motif of Figure 6.11.

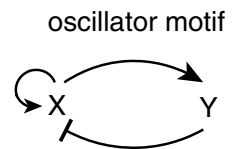


FIGURE 6.11

To see how positive feedback can make a two-component loop oscillate, we can use linear analysis of the fixed point. Without positive feedback, the two eigenvalues have a negative real part and we have a stable fixed point or stable spiral. Positive feedback can make the real part go positive, turning the stable spiral into an unstable spiral (Figure 6.12). The trajectory spirals out.

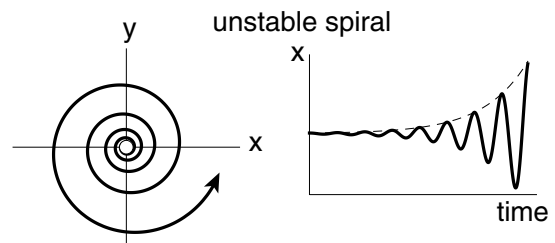


FIGURE 6.12

The trajectory cannot diverge to infinity, however. This is because, once concentrations rise sufficiently, all feedback terms saturate and we are left only with the removal terms $-\alpha_1 X$, $-\alpha_2 Y$ that push concentrations back down. Biochemical circuits have the saving grace that concentrations cannot diverge and cannot go negative.

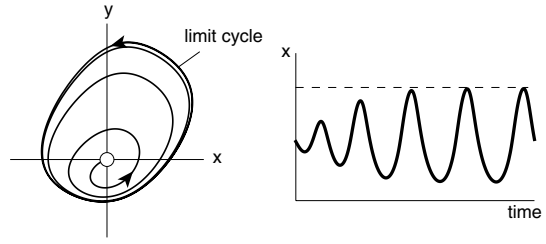


FIGURE 6.13

Thus, trajectories are kept away from the unstable spiral fixed-point and also away from infinity. They must reside somewhere in the middle. A fundamental theorem of two-component dynamical systems, the Poincaré–Bendixson theorem (Strogatz, 2015) shows that such confined trajectories settle into a sustained oscillation called a **limit cycle** (Figure 6.13).

In the next solved example, we provide details of how positive feedback destabilizes the fixed point. Again, feel free to skip this.

Solved Example 6.2: Positive Feedback Can Destabilize the Fixed Point of a Two-Component Negative Loop

The equations are like the two-node feedback loop analyzed above, except that X has positive autoregulation. As a result, its production rate is a function of both X and Y , $f(X, Y)$, which rises with X (autoregulation) but drops with Y (negative feedback). The equations are

$$\frac{dX}{dt} = f(X, Y) - \alpha_1 X$$

$$\frac{dY}{dt} = g(X) - \alpha_2 Y$$

The Jacobian matrix at the fixed point in this case is $\begin{bmatrix} -\alpha_1 + P & -\beta_1 \\ \beta_2 & -\alpha_2 \end{bmatrix}$ where $P = df/dX$

is positive due to the positive autoregulation. The negative feedback arms are $-\beta_1 = df/dY < 0$, $\beta_2 = dg/dX > 0$. The stability of the fixed point is determined by the real parts of the eigenvalues. The sum of the real parts is equal to the trace of the matrix, $\tau = P - \alpha_1 - \alpha_2$. When τ becomes positive, the fixed point becomes unstable because one of the eigenvalues has a positive real part. This occurs when positive autoregulation strength exceeds removal, $P > P_c = \alpha_1 + \alpha_2$. The sign structure of the Jacobian goes from the stable $\begin{bmatrix} - & - \\ + & - \end{bmatrix}$ to $\begin{bmatrix} + & - \\ + & - \end{bmatrix}$. If the positive feedback P is increased gradually, a stable spiral fixed point turns unstable when $P = P_c$ in what is known as a **Hopf bifurcation**.

Positive feedback can also make a more dramatic contribution, as we saw in [Chapter 5](#): bistability. **Bistability is powerful in an oscillator, because it can make the oscillations more decisive and less noisy.** The circuit makes sharp transitions between the two states, going tic-toc between high and low concentrations. The amplitude is well-defined by the difference between the high and low states, and hence frequency can be changed if needed without affecting the amplitude. The role of bistability was worked out nicely in one of biology's most fundamental oscillators, the circuit that drives the cell cycle (Ferrell et al., 2011).

Cell-cycle oscillator circuits are usually complicated. They have dozens of components that act as checkpoints to make sure important steps such as replicating the DNA are completed before cells divide. In some cells, however, the cell-cycle circuit is stripped down to a minimum, offering a good model system for basic understanding. An example is the circuit in charge of the first divisions of the frog egg, which occur every 20 min ([Figure 6.14](#)). The circuit has a negative feedback loop between X (called CDK1) and Y (called APC) and positive autoregulation on X . X can be in two states:

phosphorylated X_p , the active form, and unphosphorylated, X_0 , which is inactive. X_p acts as a kinase that can phosphorylate and thus activate proteins for cell division. Thus X_p is the output of the circuit. X_p also activates Y which degrades X , closing the negative feedback loop. A protein called cyclin is needed for X_p to be active, and is also degraded by Y .

The positive feedback on X is due to the fact that X_p activates enzymes that increase its own phosphorylation. A detail we will return to later is that X_p activates itself in two ways: activating the kinase (called Cdc25) that phosphorylates X_0 to make X_p , and inhibiting the opposing enzyme (called Wee1) that dephosphorylates X_p back to X_0 .

In the frog-egg cell-cycle circuit, the positive feedback loop causes bistability, as experimentally shown by Pomeranig, Sontag and Ferrell (2003). They used frog-egg extracts and added a nondegradable version of cyclin in order to activate X : the more cyclin added, the more X is activated. When you start with zero cyclin and now slowly increase its amounts, X_p starts low and gradually rises ([Figure 6.15](#)). When cyclin reaches a certain threshold level, c_{high} , the autoregulation kicks in and X_p jumps to a very high level, because it induces its own phosphorylation and inhibits its own

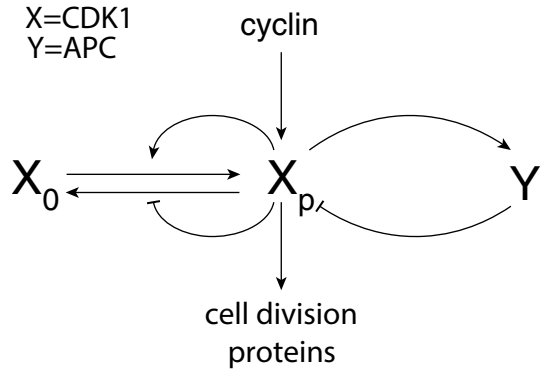


FIGURE 6.14

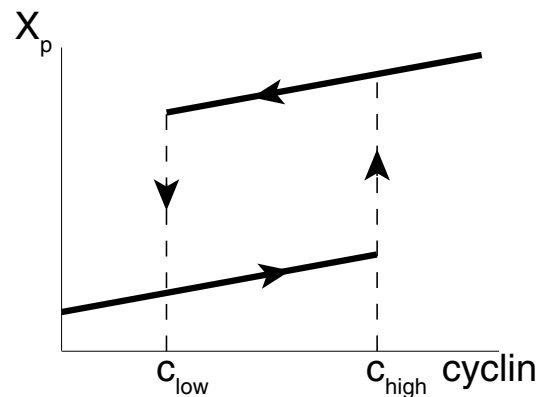


FIGURE 6.15

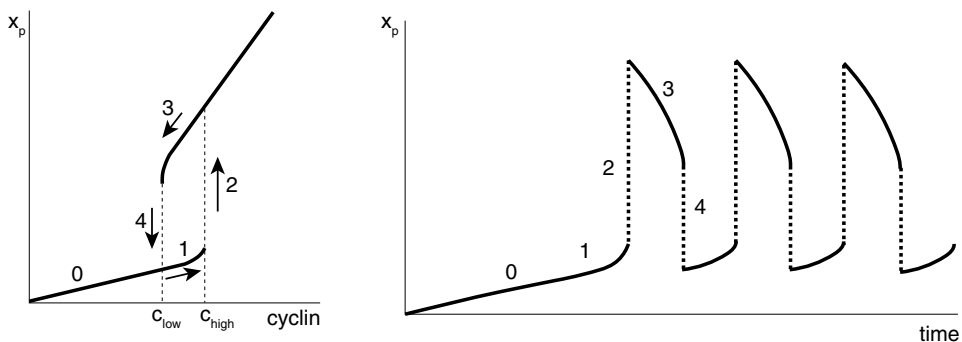


FIGURE 6.16

dephosphorylation. Remarkably, when you start with high cyclin and now reduce its levels, the behavior is different than when you increase cyclin. X_p begins high, and only when cyclin is lowered below a low threshold c_{low} , does autoregulation become weak enough that X_p drops to its original level (Figure 6.15). The existence of two thresholds and a loop-like graph in which the circuit behaves differently when input rises and when input falls is called **hysteresis**. Hysteresis is a common feature of bistable circuits.

Hysteresis is the basis for how this oscillator works (Figure 6.16): at first Y is low, and since cyclin is transcribed at a constant rate, cyclin levels gradually accumulate, and with them X_p . When cyclin reaches c_{high} , X_p shoots up. The clock goes tic. As a result Y rises, degrading cyclin. But thanks to the hysteresis, X_p goes down on the high arm of the hysteresis loop, until cyclin drops below the low threshold c_{low} . Then X_p crashes down and Y drops, resetting the cycle. The clock goes toc. Cyclin is no longer degraded by Y and starts accumulating again, beginning another cycle.

The gap between the two transition points, c_{high} and c_{low} , makes the transition from high X_p to low X_p robust to noise in the dynamics: A simple threshold mechanism would have just one threshold for the up and down transitions, making X_p jitter up and down if cyclin dawdles around the threshold.

The levels of X_p show a slow increase and then an explosive spike (Figure 6.16). These crisp oscillations are characteristic of **relaxation oscillators**. The name alludes to the sudden relaxation of the tension built up as cyclin increases. The oscillation has an asymmetric pulse shape with slow buildup at first, then accelerated rise as the positive feedback kicks in, and a rapid decline. These asymmetric spikey pulses lie in contrast to the more symmetric pulses that are typical of delay oscillators or noise-induced oscillators.

To change the frequency of this relaxation oscillator, all you need to do is change the rate at which cyclin accumulates (its transcription rate; Figure 6.17).

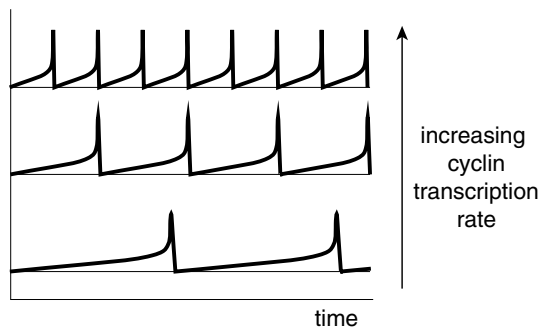


FIGURE 6.17

The amplitude of the spikes remains almost unchanged. Easily tunable frequency is an advantage for a cell-cycle clock, because cell-cycle times range between 20 min in rapid embryonic development to days/weeks in adult tissues. Tunable frequency also occurs in heart cells, as our heart rate changes over a threefold range to meet our need for oxygen. Tunable frequency is harder to achieve in a simple delay oscillator without positive feedback, because changes to frequency are coupled with changes in amplitude (Tsai et al., 2008).

6.5 ROBUST BISTABILITY USING TWO POSITIVE FEEDBACK LOOPS

To complete our analysis of the cell-cycle oscillator, let's look in more detail at the positive feedback loop in the circuit. As mentioned above, X_p increases its own level in two ways, by activating its own production and inhibiting its own removal back to X_0 . This makes two positive feedback loops.

Why two loops and not just one? I love the elegant answer proposed by James Ferrell (2008). The two-loop circuit can make bistability robust to wide variations in parameters.

To see this, we will use the rate plot method. Let's begin with a circuit that has no feedback (Figure 6.18), just

production of X_p from X_0 by phosphorylation by the kinase (Cdc25), and the removal of X_p back to X_0 by dephosphorylation by the phosphatase (Wee1). We will use simplified mass-action kinetics. The rate of removal is a rising line, X_p times the rate of the phosphatase. The rate of production is the rate of

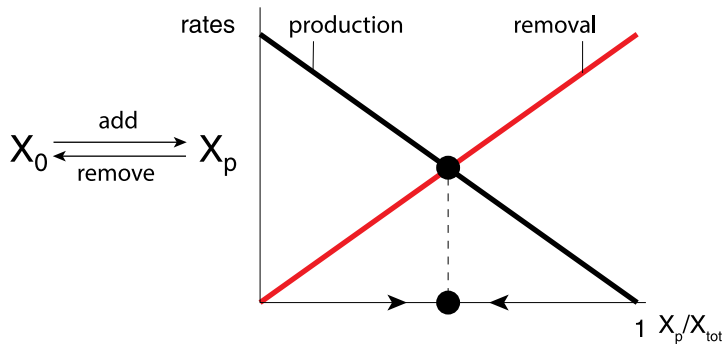


FIGURE 6.18

the kinase times X_0 . This is a decreasing line, that falls to zero when all of X is in the X_p form ($X_p = X_{\text{total}}$), because there is no more X_0 to be phosphorylated. The important part is where the two lines cross. The crossing points are the steady-state points at which production equals removal. There is only one crossing point, making a single stable steady state, with no bistability.

Now let's add one positive loop, in which X_p activates its own production. This loop can show bistability, but this bistability is fragile. It is lost upon slight changes in the removal rate parameter. To see this, notice that feedback makes the production rate curve have a hump shape (Figure 6.19) – it rises with X_p due to the positive feedback (X_p activates its own production), and falls to zero when all of X is phosphorylated ($X_p = X_{\text{tot}}$). If you arrange things just right, you can get bistability with three crossing points (a low and a high steady state, and an intermediate unstable fixed point). However, shifting the removal rate line slightly, due to a change in the number of phosphatases for example, is enough for the three-fixed-point structure to be lost. Bistability is not robust.

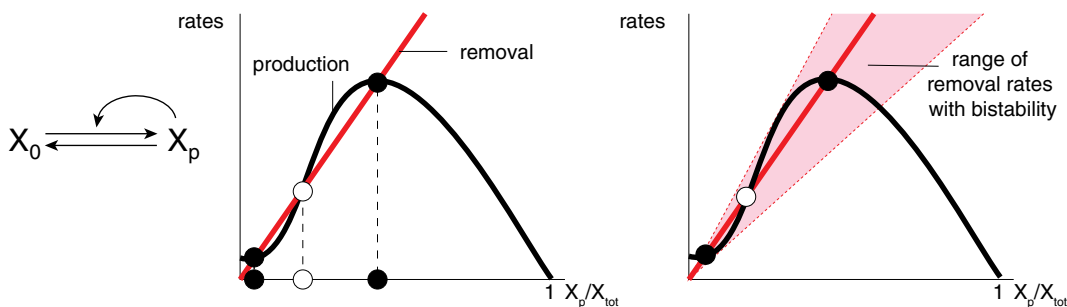


FIGURE 6.19 Adapted from (Ferrell, 2008).

In contrast, the two-positive loop design means that X_p both increases its own production and decreases its own removal. This creates a symmetry in the production and removal curves. Now both are hump shaped (Figure 6.20). As a result, their intersection points are much less sensitive to changes in parameters. The three fixed-point structure survives changes in parameters that are 10-fold larger than the one-loop design, providing robust bistability and hence robust relaxation oscillations. Indeed, the two-positive feedback loop mechanism is conserved in cell-cycle circuits throughout evolution from yeast to humans, highlighting its importance.

In sum, oscillations require negative feedback and are aided by delays and cooperativity. Noise can turn damped oscillations into sustained pulsations with noisy amplitude and relatively precise frequency. Many biological oscillators, such as the cell-cycle clock, use a motif in which a negative feedback loop is coupled to a positive feedback loop, resulting in spike-like pulses with tunable frequency and robust amplitude.

We have now finished Part 1 of this book. Congratulations! Part 1 was devoted to the principle that complex biological networks are built of a small set of network motifs, each with specific dynamical functions. A summary of the motifs we have discussed is shown in Figure 6.21.

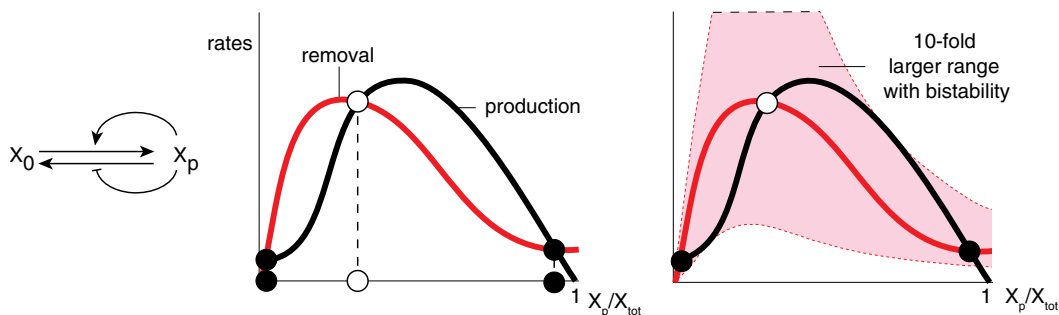


FIGURE 6.20 Adapted from (Ferrell, 2008).

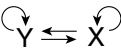
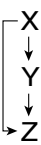
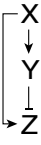
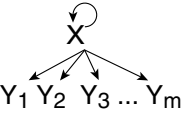
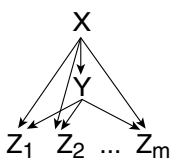
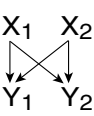
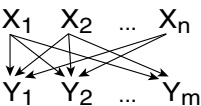
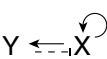
Autoregulation		
Negative autoregulation (NAR)		Speeds response time, reduces cell-cell variability of X concentration
Positive autoregulation (PAR)		Slows response time, possible bistability
Positive feedback loops		
Double-positive (Lock-ON)		Joint bistability X, Y either both ON or OFF
Double-negative (Toggle)		Exclusive bistability X ON, Y OFF or vice versa
Regulated double-positive		Lock-on: X, Y stays ON after input Z turns OFF
Feedforward loops		
Coherent feedforward loop (C1-FFL)		Sign-sensitive delay filters out brief ON input pulses when the Z-input function is AND logic, and OFF pulses when the input function is OR logic.
Incoherent feedforward loop (I1-FFL)		Pulse generation, sign-sensitive, response acceleration, biphasic dose response, fold-change detection.
Single-input module (SIM)		
		Coordinated control, Temporal (LIFO) order of promoter activity
Multi-output feedforward loop (multi-output FFL)		
		Acts as FFL for each input (sign-sensitive delay, etc) FIFO temporal order of promoter activity
Bi-fan		
		Combinatorial logic based on multiple inputs, depends on input function of each gene
Dense overlapping regulons (DOR)		
		
Negative feedback loops		
Oscillator motif		Can generate relaxation oscillations with tunable frequency
Repressilator		Can generate delay oscillations

FIGURE 6.21

FURTHER READING

Biological Oscillators

(Barkai and Leibler, 2000) “Circadian clocks limited by noise.”

(Forger, 2017) “Biological clocks, rhythms, and oscillations: the theory of biological timekeeping.”

(Strogatz, 2015) “Nonlinear dynamics and chaos: with applications to physics, biology, chemistry, and engineering.”

(Tyson, Chen and Novak, 2003) “Sniffers, buzzers, toggles and blinkers: dynamics of regulatory and signaling pathways in the cell.”

(Winfree, 2001) “The geometry of biological time.”

Oscillating Transcription Factors

(Cai, Dalal and Elowitz, 2008) “Frequency-modulated nuclear localization bursts coordinate gene regulation.”

(Geva-Zatorsky et al., 2010) “Fourier analysis and systems identification of the p53 feedback loop Fourier analysis of p53.”

(Lahav et al., 2004) “Dynamics of the p53-Mdm2 feedback loop in individual cells.”

(Nelson et al., 2004) “Oscillations in NF- κ B signaling control the dynamics of gene expression.”

(Purvis and Lahav, 2013) “Encoding and decoding cellular information through signaling dynamics.”

(Yissachar et al., 2013) “Dynamic response diversity of NFAT isoforms in individual living cells.”

Repressilator

(Elowitz and Leibler, 2000) “A synthetic oscillatory network of transcriptional regulators.”

(Potvin-Trottier et al., 2016) “Synchronous long-term oscillations in a synthetic gene circuit.”

The Cell-Cycle Oscillator

(Ferrell, 2008) “Feedback regulation of opposing enzymes generates robust, all-or-none bistable responses.”

(Ferrell, Tsai and Yang, 2011) “Modeling the cell cycle: why do certain circuits oscillate?”

(Novák and Tyson, 2008) “Design principles of biochemical oscillators.”

EXERCISES

- 6.1 *One-node circuits can't oscillate:* Consider a circuit with one variable x , described by the equation $dx/dt = f(x)$, where f is a continuous function. Show that $x(t)$ cannot oscillate.

Solution:

We can draw dx/dt versus x , to make a rate plot. When $f(x)$ is positive, x flows to higher levels, and when $f(x)$ is negative, x flows to lower levels (Figure 6.22). Thus, each value of x has a prescribed direction of motion, either increasing or decreasing but not both. In other words, in the

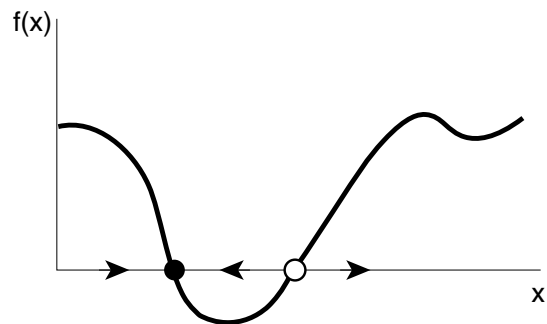


FIGURE 6.22

one-dimensional phase portrait (the x -axis in the figure), each point has an arrow that points to the left or to the right. For oscillations, however, there must be a value x which increases some of the time and decreases at other times. This is not possible in this one-variable circuit.

6.2 *Linear stability analysis:* Analyze the eigenvalues of the following two-component circuits. Find the fixed point. Which circuits have a stable fixed point? Which have a spiral fixed point?

$$\text{i. } \frac{dx}{dt} = \frac{1}{1+y} - x; \quad \frac{dy}{dt} = x - y \quad (\text{P6.1})$$

$$\text{ii. } \frac{dx}{dt} = 1 - xy; \quad \frac{dy}{dt} = x - y \quad (\text{P6.2})$$

$$\text{iii. } \frac{dx}{dt} = \frac{1}{y} - x; \quad \frac{dy}{dt} = \frac{x^4}{1+x^4} - y \quad (\text{P6.3})$$

6.3 *Equations for the oscillator motif:*

$$\frac{dx}{dt} = a + \frac{x^n}{1+x^n} - \alpha_1 xy$$

$$\frac{dy}{dt} = x - \alpha_2 y$$

Numerically solve, and find parameters that show oscillations. Find parameters that do not show oscillations. Plot the phase plot of the system in the two cases.

6.4 *Pulse trains, bang-bang control and robust expression ratios:* In this exercise, we will explore a suggestion by Cai, Dalal and Elowitz (2008) for the utility of a series of transcription factor pulses, whose frequency increases with input signal level. We can call this a frequency-modulation or FM strategy. Let's compare it to an alternative, in which the level of the transcription factor rises with signal, with no pulsing. We call this an amplitude modulation or AM strategy.

- a. Consider two genes controlled by X . Their production rate is $g_1 = \beta_1 X / (K_1 + X)$ and $g_2 = \beta_2 X / (K_2 + X)$.

In an AM strategy, X is proportional to input signal s . Show that the ratio of gene production rates g_1/g_2 changes with signal s . What is the smallest and largest ratio obtained?

- b. Consider the same two genes controlled by X with an FM strategy. Here X is either at level 0 or level X_{high} . The fraction of time that X is high is proportional to the signal s . Show that production of each gene is proportional to signal, but the ratio g_1/g_2 is constant as a function of signal.
- c. When might FM control be advantageous given that it keeps fixed ratios between the expression of genes?

- 6.5 *More on AM and FM strategies*: Consider the definitions in the previous exercise of FM and AM strategies.
- What disadvantage might an FM strategy have at very low signal. (*Hint*: Consider the stochastic variation of the number of pulses in a given time period.)
 - Certain genes in the NFAT system are controlled by two transcription factors, one with an FM strategy and one with an AM strategy, in response to the same input signal. How would you design the promoters in order to gain the advantages of AM at low signal and of FM at high signal (Issaschar et al., 2013)?
- 6.6 *Oscillating transcription factors and downstream protein half-life*: Consider a gene Y regulated by X that oscillates with a frequency ω . The removal rate of the protein Y is α .
- Plot $Y(t)$ for different values of α/ω .
 - At what value of α is the peak expression of Y highest?
 - In the p53 system, response to double-stranded DNA breaks is p53 oscillations, whereas the response to DNA damage caused by UV radiation is p53 concentration that rises gradually with time without oscillation. Consider genes activated by p53. How would genes with different α respond to the two types of damage?
- 6.7 *Repressilator stability analysis*: Consider a feedback loop made of three repressors.
- Write equations for the circuit assuming identical input functions and removal rates.
 - Write the linearized equations near the fixed point.
 - Write the Jacobian matrix.
 - Derive conditions for stability and instability of the fixed point. When does this circuit oscillate?
 - Simulate the equations numerically for different parameters, assuming Hill input functions. Check the results of (d).

BIBLIOGRAPHY

- Barkai, N. and Leibler, S. 2000. 'Biological rhythms: circadian clocks limited by noise', *Nature*, 403(6767), pp. 267–268. doi: 10.1038/35002258.
- Cai, L., Dalal, C.K. and Elowitz, M.B. 2008. 'Frequency-modulated nuclear localization bursts coordinate gene regulation', *Nature*, 455(7212), pp. 485–490. doi: 10.1038/nature07292.
- Elowitz, M.B. and Leibler, S. 2000. 'A synthetic oscillatory network of transcriptional regulators', *Nature*, 403(6767), pp. 335–338. doi: 10.1038/35002125.
- Ferrell, J.E. 2008. 'Feedback regulation of opposing enzymes generates robust, all-or-none bistable responses', *Current Biology*, 18(6), pp. R244–R245. doi: 10.1016/j.cub.2008.02.035.
- Ferrell, J.E., Tsai, T.Y.C. and Yang, Q. 2011. 'Modeling the cell cycle: why do certain circuits oscillate?', *Cell*, 144(6), pp. 874–885. doi: 10.1016/j.cell.2011.03.006.

- Forger, D.B. 2017. *Biological clocks, rhythms, and oscillations: The theory of biological timekeeping*. MIT Press.
- Geva-Zatorsky, N. et al. 2010 'Fourier analysis and systems identification of the p53 feedback loop', *Proceedings of the National Academy of Sciences*, 107(30), pp. 13550–13555. doi: 10.1073/pnas.1001107107.
- Lahav, G. et al. 2004. 'Dynamics of the p53-Mdm2 feedback loop in individual cells', *Nature Genetics*, 36(2), pp. 147–150. doi: 10.1038/ng1293.
- Li, Q. and Lang, X. 2008. 'Internal noise-sustained circadian rhythms in a drosophila model', *Biophysical Journal*, 94(6), pp. 1983–1994. doi: 10.1529/biophysj.107.109611.
- Nelson, D.E. et al. 2004. 'Oscillations in NF- κ B signaling control the dynamics of gene expression', *Science*, 306(5696), pp. 704–708. doi: 10.1126/science.1099962.
- Novák, B. and Tyson, J.J. 2008. 'Design principles of biochemical oscillators', *Nature Reviews Molecular Cell Biology*, 9(2), pp. 981–991. doi: 10.1038/nrm2530.
- Pomerening, J.R., Sontag, E.D. and Ferrell, Jr. J.E. 2003. 'Building a cell cycle oscillator: Hysteresis and bistability in the activation of Cdc2', *Nature Cell Biology*, 5(4), p. 346.
- Potvin-Trottier, L. et al. 2016. 'Synchronous long-term oscillations in a synthetic gene circuit', *Nature*, 538(7626), pp. 514–517. doi: 10.1038/nature19841.
- Purvis, J.E. and Lahav, G. 2013. 'Encoding and decoding cellular information through signaling dynamics', *Cell*, 152(5), pp. 945–956. doi: 10.1016/j.cell.2013.02.005.
- Strogatz, S.H. 2015. *Nonlinear dynamics and chaos: with applications to physics, biology, chemistry, and engineering*, 2nd edn. Westview Press.
- Tsai, T.Y.C. et al. 2008. 'Robust, tunable biological oscillations from interlinked positive and negative feedback loops', *Science*, 321(5885), pp. 126–129. doi: [10.1126/science.1156951](https://doi.org/10.1126/science.1156951).
- Tyson, J.J., Chen, K.C. and Novak, B. 2003. 'Sniffers, buzzers, toggles and blinkers: dynamics of regulatory and signaling pathways in the cell', *Current Opinion in Cell Biology*, 15(2), pp. 221–231. doi: 10.1016/S0955-0674(03)00017-6.
- Winfree, A.T. 2001. *The geometry of biological time: interdisciplinary applied mathematics*. Springer. doi: 10.1007/978-1-4757-3484-3.
- Yissachar, N. et al. 2013. 'Dynamic response diversity of NFAT isoforms in individual living cells', *Molecular Cell*, 49(2), pp. 322–330. doi: 10.1016/j.molcel.2012.11.003.



Taylor & Francis

Taylor & Francis Group

<http://taylorandfrancis.com>

Part 2

Robustness



Taylor & Francis

Taylor & Francis Group

<http://taylorandfrancis.com>

Kinetic Proofreading and Conformational Proofreading

7.1 INTRODUCTION

Welcome to Part 2 of this book, devoted to the principle of robustness. This principle states that *biological systems have special designs that make their essential function work precisely despite naturally occurring noise*. Thus, to define robustness, we will ask for any given system which of its functions are robust with respect to which source of noise or errors. We will explore robustness at the level of proteins, circuits and entire tissues. Robustness plays an important role in systems biology because it explains how biological systems can function in their noisy environment. Robust systems are also fun to study because they have elegant designs.

In this chapter, we will examine robustness to a fundamental source of errors in cells. These errors result from the presence, for each molecule X , of many chemically similar molecules that can confound the specific *recognition* of X by its interaction partners. So far, we took it for granted that a transcription factor or receptor can bind its input signal molecule without mistakes. Now we ask how can this be? How can a biochemical recognition system, such as a receptor or transcription factor, pick out a specific molecule in a sea of other molecules that bind it with only slightly weaker affinity?

We will see that diverse molecular recognition systems in the cell employ the same principle to achieve high precision. This principle is called **kinetic proofreading**. It was proposed independently by John Hopfield (1974) and Jacques Ninio (1975).

To describe kinetic proofreading, we'll begin with the reading of the genetic code during translation. We will then consider kinetic proofreading in the immune system (McKeithan, 1995; Goldstein, Faeder and Hlavacek, 2004). We will also discuss a different proofreading strategy used by proteins that, unlike kinetic proofreading, does not require spending energy. Finally, we will see how minimizing errors can lead to rules for gene regulation.

Kinetic proofreading is a somewhat subtle idea, and so we will use three different approaches to describe it. In the context of translation, we will use kinetic equations

to derive the error rate. In the context of immune recognition, we will use a delay time argument. But first we will tell a story about a recognition problem in a museum.

A museum curator wants to design a room that can help to identify Picasso lovers from among the museum visitors. In this museum, half of the visitors are Picasso lovers and half do not care for Picasso. The curator opens a door in a busy corridor. The door leads to a room with a Picasso painting, allowing visitors to enter the room at random. Picasso lovers that happen to enter the room hover near the picture for, on average, 10 min, whereas others stay in the room for only 1 min. Because of the high affinity of Picasso lovers for the painting, the room becomes enriched with 10 times more Picasso lovers than nonlovers.

The curator wishes to do even better. At a certain moment, the curator locks the door to the room and reveals a second one-way revolving door. The nonlovers in the room leave through the one-way door, and after several minutes, the only ones remaining are Picasso lovers, still hovering around the painting. Enrichment for Picasso lovers is much higher than 10-fold.

If the revolving door was two-way, allowing visitors to enter the room at random, only a 10-fold enrichment for Picasso lovers would again occur. Kinetic proofreading mimics the Picasso room by using nearly irreversible, nonequilibrium reactions as one-way doors.

7.2 KINETIC PROOFREADING OF THE GENETIC CODE CAN REDUCE ERROR RATES

Consider the fundamental biological process of translation. In translation, a ribosome produces a protein by linking amino acids one by one into a chain (Figure 7.1). The type of amino acid added at each step to the elongating chain is determined by the information encoded in an mRNA. Each of the 20 amino acids is encoded by a codon, a series of three letters on the mRNA. The mapping between the 64 codons and the 20 amino acids is called the **genetic code** (Figure 7.2). For example, the codon AGG encodes the amino acid arginine (Arg).

To make the protein, the codon must be read and the corresponding amino acid must be brought into the ribosome. Each amino acid is brought into the ribosome connected to a specific tRNA molecule. That tRNA has a three-letter recognition site that pairs with the codon sequence for that amino acid on the mRNA (Figure 7.1). There is a tRNA for each of the codons that specify amino acids in the genetic code. Translation, therefore, communicates information from mRNA codons to the amino acids in the protein chain. The codons must recognize and bind the correct tRNA.

This molecular recognition process works under thermal noise, and thus has an error rate. The wrong tRNA can attach to the codon, resulting in a translation error where a wrong amino acid is incorporated into the elongating protein. These translation errors occur at a frequency of

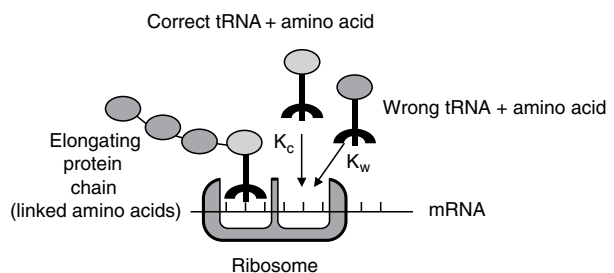


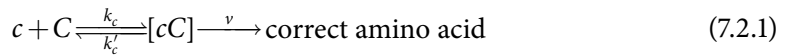
FIGURE 7.1

about 10^{-4} . This means that a typical protein of 100 amino acids has a 1% chance to have one wrong amino acid. A much higher error rate would be disastrous, because it would result in the malfunction and misfolding of an unacceptable fraction of the cell's proteins.

7.2.1 Equilibrium Binding Cannot Explain the Precision of Translation

The simplest model for this recognition process is **equilibrium binding** of tRNAs to the codons. We will now see that simple equilibrium binding cannot explain the observed error rate. This is because equilibrium binding generates error rates that are equal to the ratio of affinities of the correct and incorrect tRNAs. This would result in error rates that are about 100 times higher than the observed error rate.

To analyze equilibrium binding, consider codon C on the mRNA in the ribosome that encodes the amino acid to be added next to the protein chain. We begin with the rate of binding of the correct tRNA, denoted c , to codon C . Codon C binds c with an on-rate k_c . The tRNA unbinds from the codon with off-rate k'_c . When the tRNA is bound, there is a probability ν per unit time that the amino acid attached to the tRNA will be covalently linked to the growing, translated protein chain. In this case, the freed tRNA unbinds from the codon and the ribosome shifts to the next codon in the mRNA. The equilibrium process is hence,



At equilibrium, the concentration of the complex $[cC]$ is given by the balance of the two arrows marked k_c and k'_c (the rate ν is much smaller than k_c and k'_c and can be neglected). Hence, at steady state, collisions of c and C that form the complex $[cC]$ at rate k_c balance the dissociation of the complex $[cC]$, so that $cCk_c = [cC]k'_c$. This results in a concentration of the complex $[cC]$ given by the product of the concentrations of the reactants divided by the dissociation constant K_c :

$$[cC] = cC/K_c \quad (7.2.2)$$

where K_c is equal to the ratio of the off-rate and on-rate for the tRNA binding:

$$K_c = k'_c/k_c \quad (7.2.3)$$

The smaller the dissociation constant, the higher the affinity of the reactants. We can now write the incorporation rate of the correct amino acid, equal to the concentration

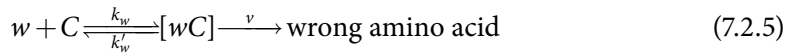
		Second letter					
		U	C	A	G		
First letter	U	UUU Phe UUC UUA Leu UUG	UCU UCC Ser UCA UCG	UAU Tyr UAC UAG STOP UAG STOP	UGU Cys UGC UGA STOP UGG Trp	U C A G	Third letter
	C	CUU CUC Leu CUA CUG	CCU CCC Pro CCA CCG	CAU His CAC CAA Gln CAG	CGU CGC Arg CGA CGG	U C A G	
	A	AUU Ile AUC AUA Met AUG	ACU ACC Thr ACA ACG	AAU Asn AAC AAA Lys AAG	AGU Ser AGC AGA Arg AGG	U C A G	
	G	GUU GUC Val GUA GUG	GCU GCC Ala GCA GCG	GAU Asp GAC GAA Glu GAG	GGU GGC Gly GGA GGG	U C A G	

FIGURE 7.2

of the bound complex times the rate at which the amino acid is linked to the elongating protein chain:

$$R_{\text{correct}} = v[cC] = v c C / K_c \quad (7.2.4)$$

In addition to the correct tRNA, the cells contain different tRNAs that carry the other amino acids. These tRNAs compete for binding to codon C . Let us consider, for simplicity, only one of these other tRNAs, the most dangerous one – the tRNA that carries a different amino acid that has the highest affinity to codon C . It is this incorrect tRNA that has the highest probability to yield false recognition by binding the codon C , leading to incorporation of the wrong amino acid. The concentration of this incorrect tRNA is about equal to the concentration of the correct tRNA (many of the tRNAs have approximately the same concentrations). The wrong tRNA, denoted w , can bind the codon C in the following equilibrium process:



Note that the linkage rate v is the same for the correct and wrong codons because linkage occurs at a distant site on the ribosome, away from the site of codon recognition. The considerations discussed above show that the rate at which the wrong amino acid is linked into the protein is

$$R_{\text{wrong}} = v w C / K_w \quad (7.2.6)$$

Since w is the wrong tRNA, it has a larger dissociation constant for binding C than the correct tRNA, c , that is, $K_w > K_c$, and hence a smaller linking rate $R_{\text{wrong}} < R_{\text{correct}}$.

The resulting **error rate**, F_o , is the ratio of the rates of incorrect and correct amino acid incorporation. The error rate is approximately equal to the ratio of the dissociation constants, since all other concentrations (tRNA concentrations) are about the same for c and w :

$$F_o = R_{\text{wrong}} / R_{\text{correct}} = v w C K_c / v c C K_w \approx K_c / K_w \quad (7.2.7)$$

To repeat the main conclusion, the error rate in equilibrium recognition is determined by the ratio of dissociation constants for the correct and incorrect tRNAs. As occurs for many biological binding events, the on-rates for both w and c are limited by diffusion and are about the same, $k_w = k_c$ (Appendix A). *It is the off-rate, k'_w which distinguishes the correct codon from the incorrect one:* the wrong tRNA unbinds more rapidly than the correct tRNA, $k'_w \gg k'_c$, because of the weaker chemical bonds that hold it in the bound complex. Using Equation 7.2.3, we find an error rate proportional to the off-rate ratio:

$$F_o = R_{\text{wrong}} / R_{\text{correct}} = K_c / K_d \approx k'_c / k'_w \quad (7.2.8)$$

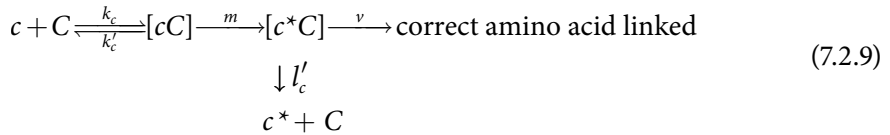
The off-rates are akin to the dissociation rates of museum visitors from the Picasso painting in the Picasso room story.

How does equilibrium recognition compare with the actual error rates? The affinity of codons to correct and incorrect tRNAs was experimentally measured to find an affinity ratio of about $K_c/K_w \sim 1/100$. Hence, there is a large discrepancy between the predicted equilibrium recognition error, $F_o \sim K_c/K_w \sim 1/100$, and the actual translation error rate, $F = 1/10,000$. It therefore seems that equilibrium recognition cannot explain the high fidelity found in this system.¹

7.2.2 Kinetic Proofreading Can Dramatically Reduce the Error Rate

We just saw that equilibrium binding can only provide discrimination that is as good as the ratio of the chemical affinity of the correct and incorrect targets. What mechanism can explain the high fidelity of the translation machinery, which is a 100-fold higher than predicted from equilibrium recognition?

The solution lies in a reaction that occurs in the translation process, which was well-known at the time that Hopfield analyzed the system, but whose function was not understood and was considered a wasteful side reaction. In this reaction, the tRNA, after binding the codon, undergoes a chemical modification. That is, c binds to C and is then converted to c^* . This reaction is virtually irreversible, because it is coupled to the hydrolysis of a GTP molecule.² The modified tRNA, c^* , can either fall off of the codon or donate its amino acid to the elongating protein chain:



The fact that the modified tRNA can fall off seems wasteful because the correct tRNA can be lost. Moreover, to make c^* costs energy: each amino acid incorporated into a protein requires hydrolysis of GTP which is about one ATP's worth of energy. This cost adds up to a large part of the cell's energy balance. However, it is precisely this design that generates high fidelity. The secret is that c^* offers a second discrimination step: the wrong tRNA, once modified, can fall off the codon, but it cannot mount back on. This irreversible reaction acts as the one-way door in the Picasso story.

To compute the error rate in this process, we need to find the concentration of the modified bound complex, $[c^*C]$. This concentration is given by the balance of the two

¹ Why not increase the ratio of the off-rates to improve discrimination? Such an increase may be infeasible due to the chemical structure of codon-anticodon recognition, in which different codons can differ by only a single hydrogen bond. In addition, decreasing the off-rate of the tRNAs would cause them to stick to the codon for a longer time. This would interfere with the need to rapidly bind and discard many different tRNAs in order to find the correct one, and slow down the translation process (Exercise 7.3). Thus, biological recognition may face a trade-off between speed and specificity.

² Near-irreversibility is attained by coupling a reaction to a second reaction that expends free energy. For example, coupling a reaction to ATP hydrolysis can shift it away from equilibrium by factors as large as 10^8 , achieved because the cell continuously expends energy to produce ATP.

processes described by the arrows marked with the rates m and l'_c (since the rate v is much smaller than the other rates), leading to $m[cC] = l'_c[c^*C]$, yielding a steady-state solution:

$$[c^*C] = m[cC]/l'_c \quad (7.2.10)$$

The rate of correct incorporation is the linking rate v times the modified complex concentration (Equation 7.2.9):

$$R_{\text{correct}} = v[c^*C] = vmcC/l'_cK_c \quad (7.2.11)$$

The same applies for the wrong codon w . The conversion of w to w^* occurs at the same rate, m , as the conversion of c to c^* , because the modification process does not discriminate between tRNAs. The rate that the wrong tRNA w^* falls off the codon is, however, much faster than the rate at which c^* falls off. This is because of the weaker chemical affinity of the wrong tRNA to the codon. The off-rate ratio of the correct and incorrect modified tRNAs is the same as the ratio for the unmodified tRNAs, since they are all recognized by the same codon C :

$$\frac{l'_w}{l'_c} = \frac{k'_w}{k'_c} \approx K_w/K_c \quad (7.2.12)$$

Thus, w^* undergoes a second discrimination step, with a significant chance that the wrong tRNA is removed. The rate of wrong amino acid linkage is the same as in Equation 7.2.11, with all parameters for c replaced with the corresponding parameters for w :

$$R_{\text{wrong}} = v[w^*C] = vmwC/l'_wK_w \quad (7.2.13)$$

resulting in an error rate

$$F = R_{\text{wrong}}/R_{\text{correct}} = \left(\frac{K_c}{K_w}\right)\left(\frac{l'_c}{l'_w}\right) = \left(\frac{K_c}{K_w}\right)^2 \quad (7.2.14)$$

Thus, the irreversible reaction step affords a proofreading event that adds a multiplicative factor of K_c/K_w to the error rate. It effectively works like two separate recognition processes, the second working on the output of the first. This results in an error rate that is the square of the equilibrium recognition error rate:

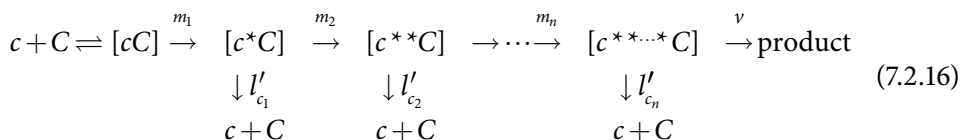
$$F = F_o^2 \quad (7.2.15)$$

It is important to note that had all reactions been reversible and at equilibrium, no improvement would be gained over the simple scheme (Equation 7.2.1). This is due to

detailed balance and is discussed in Exercise 7.2. The equilibrium model with detailed balance is similar to the Picasso room in which the one-way door is changed to a two-way door that allows visitors in and out at random.

Thus, the proofreading step implemented by a modification of the tRNA can reduce the error rate from the equilibrium recognition rate of about $F_0 = 1/100$ to a much lower error rate, $F = F_0^2 = 1/10,000$, similar to the observed error rate.

An even higher level of fidelity can be attained by linking together n irreversible (or nearly irreversible) proofreading processes:



Each irreversible step adds a proofreading factor F_0 , resulting in an overall error rate of

$$F = F_0^{n+1} \quad (7.2.17)$$

7.3 RECOGNITION OF SELF AND NON-SELF BY THE IMMUNE SYSTEM

We have just seen how kinetic proofreading uses a nonequilibrium step to reduce errors in translation. We will now use a slightly different (but equivalent) way to explain kinetic proofreading, based on time delays. For this purpose, we will study kinetic proofreading in the immune system.

The immune system monitors the body for dangerous pathogens. It is made of a vast collection of cells that communicate and interact in many ways. When it detects pathogens, the immune system computes and mobilizes the appropriate responses.

One of the major tools of the immune system is antibodies. Each antibody is a protein designed to bind with high affinity to a specific foreign protein made by pathogens, called the antigen.

One of the important roles of the immune system is to scan the cells of the body for antigens, such as fragments of proteins made by a virus that has infected the cell. The scanning task is carried out by T-cells. Each of the T-cells has receptors that are specific against a foreign protein antigen. To provide information for the T-cells, each cell in the body presents fragments of proteins on the cell surface. The proteins are presented in dedicated protein complexes on the cell surface called MHCs (Figure 7.3).

The goal of the T-cell is to eliminate infected cells. Each T-cell can recognize a specific antigen in the MHC because its receptor can bind that foreign peptide. If the T-cell receptor recognizes its antigen, the foreign protein fragment in the MHC on a cell, it triggers a

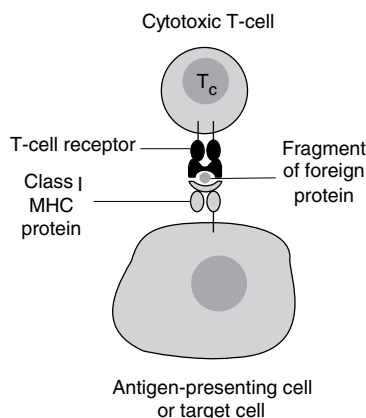


FIGURE 7.3

signal-transduction cascade inside the T-cell. The signaling causes the T-cell to mount an immune response to kill the cell that presented the foreign peptide. This eliminates the infected cell and protects the body from the virus.

In the recognition process, it is essential that the T-cell does not kill cells that present proteins that are normally produced by the healthy body. If such misrecognition occurs, the immune system attacks the cells of the body, potentially leading to an autoimmune disease.

The precision of the recognition of non-self-proteins by T-cells is remarkable. T-cells can recognize minute amounts of a foreign protein antigen in a background of self-proteins, even though the self-proteins have only a slightly lower affinity to the T-cell receptor than the foreign target. The error rate of recognition is less than 10^{-6} , although the affinity of the antigen is often only 10-fold higher than the affinities of the self-proteins.

7.3.1 Equilibrium Binding Cannot Explain the Low Error Rate of Immune Recognition

The receptors on a given T-cell are built to recognize a specific foreign protein, which we will call the correct ligand, c . The correct ligand binds the receptors with high affinity. In addition to c , the receptors are exposed to a variety of self-proteins, which bind the receptor with a weaker affinity. In particular, some of these self-proteins are quite similar to the correct ligand and pose the highest danger for misrecognition, in which the receptors mistake a self-protein for the correct ligand. For clarity, let us treat these wrong ligands as a single entity w , with a lower affinity to the receptor. We will begin by the simplest model for recognition, in which c and w bind the receptor in an equilibrium process. As in the previous section, this yields error rates that are proportional to the ratio of affinities of the incorrect and correct targets. Since the affinities of the correct and incorrect ligands are not very different, equilibrium recognition results in an unacceptably high rate of misrecognition.

The dynamics of binding of the correct ligand c to the receptor R includes two processes. The first process is collisions of c and R at a rate k_{on} to form a complex, $[cR]$, in which the ligand is bound to the receptor. The inverse process is dissociation of the complex, in which the ligand unbinds from the receptor at a rate k_{off} , with $K_c = k_{\text{off}}/k_{\text{on}}$.

When the ligand binds the receptor, it triggers a signal-transduction pathway inside the T-cell, which leads to activation of the T-cell. Once ligand binds the receptor, the signaling pathway is activated with probability ν per unit time. Therefore, the rate of T-cell activation in the presence of a concentration c of correct ligand is

$$A_{\text{correct}} = [cR]\nu = cR\nu/K_c \quad (7.3.1)$$

A similar set of equations describes the binding of the incorrect ligand w to the receptor.

$$A_{\text{wrong}} = [wR]\nu = wR\nu/K_w \quad (7.3.2)$$

Hence, the error rate of the T-cells, defined by the ratio of incorrect to correct activations, is

$$F_o = A_{\text{wrong}}/A_{\text{correct}} = (K_c/K_w)(w/c) \quad (7.3.3)$$

The error rate in this equilibrium recognition process is thus given by the ratio of affinities of the incorrect and correct ligands times the ratio of their concentrations. In the immune system, the incorrect ligands often have only a 10-fold lower affinity than the correct ligand, $K_c/K_w \sim 0.1$. Furthermore, the concentration of incorrect ligand (proteins made by the healthy body) often greatly exceeds the concentration of the correct ligand (pathogen protein). Hence, the equilibrium error rate is $F_o > 0.1$. This is far higher than the observed error rate in T-cell recognition, $F = 10^{-6}$ or lower.

How can we bridge the huge gap between the high rate of equilibrium recognition errors and the observed low error rate in the real system? The next section describes a kinetic proofreading mechanism in the receptors that amplifies small differences in affinity into large differences in the recognition rates.

7.3.2 Kinetic Proofreading Increases Fidelity of T-Cell Recognition

The actual recognition process in T-cell receptors includes several additional steps, which may at first sight appear to be superfluous details. After ligand binding, the receptor undergoes a series of covalent modifications, such as phosphorylation on numerous sites (Figure 7.4). These modifications are energy-consuming and are held away from thermal equilibrium. When modified, the receptor binds several protein partners inside the cell. Activation of the signaling pathway inside the T-cell begins only after all of these modifications and binding events are complete. Kinetic proofreading relies on these extra steps to create a delay τ that allows the system to reduce its error rate. The basic idea is that only ligands that remain bound to the receptors for a long enough time have a chance to activate the T-cell (McKeithan, 1995). Experimentally, the cutoff dwell time for activation is about 3–5 sec.

To understand this, let us examine a binding event of the correct ligand. Once bound, the ligand has a probability per unit time k_{off} to dissociate from the receptor. Hence, the probability that it remains bound for a time longer than t after binding is

$$P(t) = e^{-k_{\text{off}}t} \quad (7.3.4)$$

Signaling in the cell only occurs at a delay τ after the ligand binds the receptor, due to the series of modifications of the receptors needed to activate the signaling pathway. Hence, the probability per ligand binding event that the T-cell is activated is equal to the probability that the ligand is bound for a time longer than τ :

$$A_{\text{correct}} = e^{-k_{\text{off}}\tau} \quad (7.3.5)$$

Similarly, the incorrect ligand has an off-rate k'_{off} .

The off-rate of the incorrect

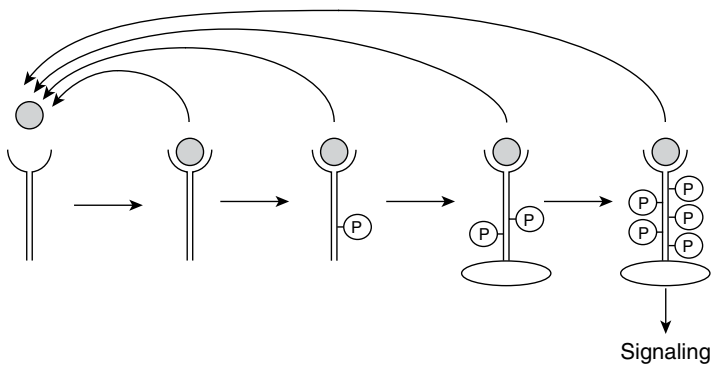


FIGURE 7.4

ligand is larger than that of the correct ligand, because it binds the receptor more weakly. The probability that the incorrect ligand activates the receptor is

$$A_{\text{correct}} = e^{-k'_{\text{off}}\tau} \quad (7.3.6)$$

Hence, the error rate in the delay mechanism is the ratio of these activation rates:

$$F = A_{\text{wrong}}/A_{\text{correct}} = e^{-(k'_{\text{off}} - k_{\text{off}})\tau} \quad (7.3.7)$$

This allows a very small error rate even for moderate differences between the off-rates, provided that the delay is long enough. For example, if the off-rate of the correct ligand is $k_{\text{off}} = 0.5 \text{ sec}^{-1}$ and the incorrect ligand is $k'_{\text{off}} = 5 \text{ sec}^{-1}$, and the delay is $\tau = 3 \text{ sec}$, one finds

$$F \sim e^{-(5 - 0.5) \cdot 3} = e^{-13.5} \sim 10^{-6} \quad (7.3.8)$$

Thus, long delays can enhance fidelity. However, this comes at a cost. The longer the delay, the larger the number of binding events of the correct ligand that unbind before signaling can begin. Thus, increasing the delay can cause a loss of sensitivity. The loss of sensitivity is tolerated because of the greatly improved discrimination between the correct ligand and incorrect but chemically similar ligands.

Kinetic proofreading is a general mechanism that provides specificity due to a delay step, which gives the incorrect ligands a chance to dissociate before recognition is complete. In order for kinetic proofreading to work effectively, the receptors must lose their modifications when the ligand unbinds, before a new ligand molecule can bind. Otherwise, the wrong ligand can bind to receptors that have some of the modifications from a previous binding event, resulting in a higher probability for misrecognition.

Experiments to test kinetic proofreading used a series of ligands with different k_{off} values (reviewed in [Goldstein, Faeder and Hlavacek, 2004]). The experiments were designed so that the fraction of the receptors bound by each ligand was the same. This was achieved by using higher concentrations of ligands with weaker binding (larger k_{off}), or by normalizing the results per binding event. Simple equilibrium recognition predicts a constant probability for triggering signaling per ligand binding event, regardless of the k_{off} of the ligand. In contrast, the experiments show that the probability of activation of the signaling pathway depends inversely on k_{off} . This means that the longer the ligand is bound to the receptor, the higher the probability that it triggers signaling. This is consistent with the kinetic proofreading picture.

Advances by Gregoire Altan-Bonnet, Ronald Germain and Paul François show that specificity can be greatly improved by adding a negative feedback loop to kinetic proofreading. In this negative loop, the modified receptor states inhibit further modification and thus slow down the progress of the kinetic proofreading. Such a negative feedback loop indeed operates in T-cell signaling because the phosphorylated receptors activate a phosphatase that removes the phosphorylation modifications (François and Altan-Bonnet, 2016).

Kinetic proofreading uses modification of the T-cell receptor after ligand binding to create a delay. This process is not unique to T-cell receptors. In fact, these types of modifications occur in practically every receptor in mammalian cells, including receptors that sense hormones, growth factors and other ligands. This raises the possibility that delays and kinetic proofreading are widely employed by receptors to provide robustness against misrecognition of the background of diverse molecules in the organism.

7.4 KINETIC PROOFREADING OCCURS IN DIVERSE PROCESSES IN THE CELL

The hallmark of kinetic proofreading is a nonequilibrium reaction in the recognition process that forms an intermediate state, providing a delay after ligand binding. The system must operate away from equilibrium, so that ligands cannot circumvent the delay by rebinding directly in the modified state. New ligand binding must primarily occur in the unmodified state.

These ingredients are found in diverse recognition processes in the cell. An example is DNA binding by repair proteins (Sancar and Reardon, 2004) and recombination proteins (Tlusty, Bar-Ziv and Libchaber, 2004). In bacterial DNA repair, for example, recognition protein A binds the damaged strands, because it has a higher affinity to damaged DNA than to normal DNA. After binding, protein A undergoes a modification (phosphorylation). When phosphorylated, it recruits additional proteins B and C that nick the DNA on both sides of A and remove the damaged strand, allowing specialized enzymes to fill in the gap and polymerize a fresh segment in place of the damaged strand. The modification step of protein A may help prevent misrecognition of normal DNA as damaged.

An additional example occurs in the coupling of amino acids to their specific tRNAs (Hopfield, 1974; Hopfield et al., 1976). A special enzyme recognizes the tRNA and its specific amino acid and covalently joins them. Covalent joining of the wrong amino acid to the tRNA would lead to the incorporation of the wrong amino acid in the translated protein. Interestingly, the error rate in the tRNA formation process is about 10^{-4} , similar to the translation error rate we examined in Section 7.2.1 due to misrecognition between tRNAs and their codons.³ This low error rate is achieved by an intermediate high-energy state, in which the enzyme that connects the amino acid to the tRNA first binds both reactants, then chemically modifies the tRNA and only then forms the covalent bond between the two. Again, we see the hallmarks of kinetic proofreading.

Intermediate states are found also in the process of protein degradation in eukaryotic cells (Rape, Reddy and Kirschner, 2006). Here, a protein is marked for degradation by means of a specific enzyme that covalently attaches a chain made of a small protein subunit called ubiquitin (Hershko and Ciechanover, 1998). A de-ubiquitinating enzyme can remove the ubiquitin, saving the tagged protein from destruction. The addition of ubiquitin subunits one by one implements a delay, so that there is a chance for the wrong protein

³ It is interesting to consider whether the two error rates are tuned to be similar. It may not make sense to have one error rate much larger than the other, because the larger error would dominate the final errors in proteins. Natural selection often leads to such convergence of costs or error rates (Rosenheim et al., 2010).

to be de-ubiquitinated and not destroyed. This can allow differential degradation rates for proteins that have similar affinities to their ubiquitinating enzyme.

7.5 CONFORMATIONAL PROOFREADING PROVIDES SPECIFICITY WITHOUT CONSUMING ENERGY

Kinetic proofreading depends on energy-burning steps, such as ATP hydrolysis, and it was thought that such nonequilibrium steps were essential for proofreading. It was therefore surprising when a different mechanism for specificity was discovered that works at equilibrium without burning ATP.

The story begins with the way that proteins bind their ligands. In the early days of biochemistry, it was thought that ligands fit the proteins like a rigid match between **lock and key**. In the mid-twentieth century, the lock-and-key paradigm was replaced with the finding that proteins are more flexible, and that they deform when they bind the ligand – a phenomenon called **induced fit** (Figure 7.5).

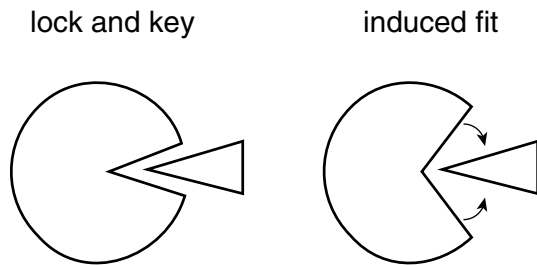


FIGURE 7.5

Induced fit was puzzling because it costs elastic energy to deform the proteins, and thus induced fit reduces the binding of the ligand. But most proteins show induced fit – including antibodies that recognize viruses and ribosomes that bind tRNAs – despite the fact that such processes would be more efficient if the protein and target fitted together like rigid locks and keys.

Yonatan Savir and Tsvi Tlusty (Savir and Tlusty, 2007; Savir and Tlusty, 2010) provided a fresh view of induced fit. They showed that induced fit is a good idea if the goal is not to bind tightly but to avoid binding to the wrong partner. This mechanism is called **conformational proofreading**. In conformational proofreading one adds an energy handicap h to the free energy of binding. The elastic bending energy of induced fit is such a handicap. The handicap gets added to the “lock and key” free energy of binding of the correct and wrong targets, ΔG_c and ΔG_w , which become $\Delta G_c + h$ and $\Delta G_w + h$. Since the binding rate goes as $1/(1 + e^{\Delta G})$, the handicap affects the wrong target much more strongly than the correct target. It provides a spacing between their binding curves (Figure 7.6). There is an optimal bending energy h in which the binding of the correct target is reduced slightly, but the binding of the wrong target is much reduced.

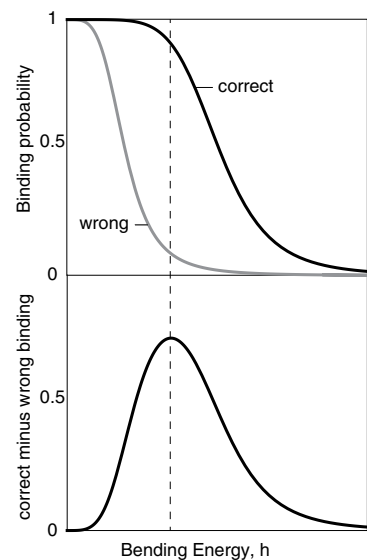


FIGURE 7.6

Conformational proofreading predicts the optimal degree of bending that maximizes specificity. This prediction agrees with the extent of the sizable bend

that ribosomes make when binding tRNA (Savir and Tlusty, 2013). Conformational proofreading also explains how the DNA repair protein RecA can find a needle in a haystack. RecA binds a piece of DNA, D , and searches the entire genome for a perfect match for D . To do so, RecA stretches D by 50%, an extreme distortion of the DNA strand. This creates the optimal handicap to allow RecA to avoid binding numerous look-alike DNA sequences that differ from D by one letter, and to find the perfect match even in huge genomes (Savir and Tlusty, 2010). In ribosomes and RecA, conformational and kinetic proofreading work together to enhance specificity.

7.6 DEMAND RULES FOR GENE REGULATION CAN MINIMIZE ERRORS

An additional principle for reducing errors is found when considering gene regulation by transcription factors. As we saw in [Chapter 1](#), there are two modes of gene control: activation and repression. Both modes of control are equivalent in principle: a gene can be turned ON either by the binding of an activator or by the unbinding of a repressor. One might therefore ask whether the choice of activator or repressor for each gene is arbitrary and depends on historical accident, or if instead there are rules for this choice.

In the 1970s, Michael Savageau provided evidence for a rule, called the **demand rule**. The demand of a gene is the probability that the gene is expressed at the high end of its range in the environment in which the organism evolved. Thus, high demand genes are usually ON, and low demand genes are usually OFF.

Savageau found that in *E. coli*, high demand genes tend to be regulated by activators, and low demand genes by repressors (Savageau, 1974). For example, the sugar lactose is rare in the lower intestinal environment of *E. coli* because human enzymes absorb lactose early in the digestive process. The lactose system is hence in low demand, and is accordingly regulated by a repressor, LacI. Similarly, arabinose is a common sugar in the lower intestine, hence the arabinose system is high demand. It is regulated by the activator AraC.

To understand the demand rule, one can define an underlying principle: **the binding site in the promoter is bound to its regulator most of the time** (Shinar et al., 2006). A low demand gene is normally OFF, and the repressor is thus normally bound; a high demand gene is normally ON, and the activator is thus normally bound, [Figure 7.7](#). In both cases, the regulator binds its site most of the time. Binding of the regulator to its site protects

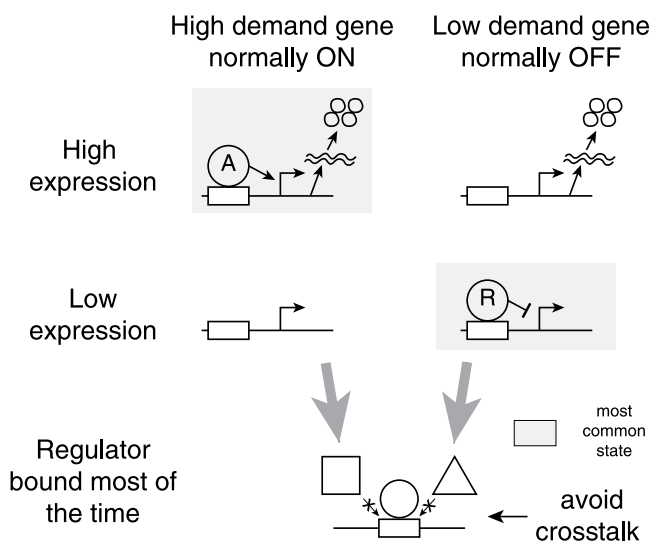


FIGURE 7.7

it from nonspecific binding by the other regulators in the cell, a phenomenon known as **crosstalk** (Sasson et al., 2012). Therefore, the demand rule minimizes crosstalk errors. For example, if a high demand gene went against the rule and was regulated by a repressor, its site would be unbound most of the time and more exposed to errors.

The demand rule can be generalized to the case of genes regulated by multiple transcription factors. For example, a gene with two inputs can be regulated by two activators (AA), by an activator and a repressor (AR) or by two repressors (RR). The demand rule predicts that the choice goes according to the expression state that is most frequent in the organism's environment. Imagine that the

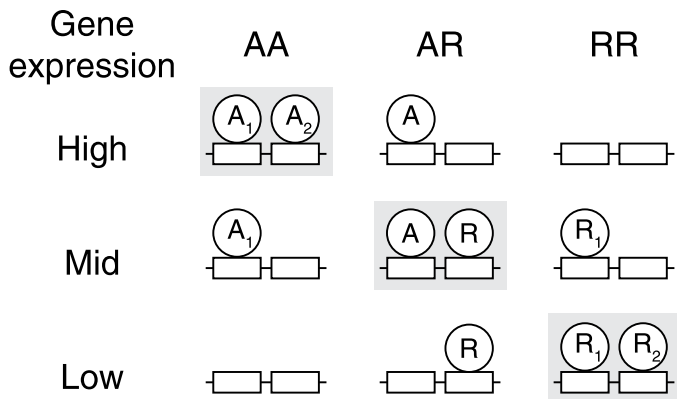


FIGURE 7.8

gene can have low, medium or high expression, Figure 7.8. If high expression is the most common state, the AA design offers the best protection against errors because the two regulators are bound most of the time. If the lowest expression state is most commonly needed, the RR design is best. The AR design is selected when medium expression is most commonly needed. In all of these cases, the selected design ensures that the site is bound most of the time.

In summary, the need for robustness to mis-binding errors leads to specific designs in biological recognition systems. One design principle is kinetic proofreading, that allows precise recognition of a target despite the background noise of other molecules similar to the target. Kinetic proofreading can explain seemingly wasteful side reactions in biological processes that require high specificity. These side reactions contribute to the fidelity of recognition at the expense of energy and delays. Likewise, conformational proofreading can confer specificity at the cost of reduced sensitivity due to specifically crafted deformations and mismatches between the ligand and its binding site. Finally, the need to minimize crosstalk can give rise to demand rules that can explain the choice of mode of control, activation versus repression. These principles can help us to make sense of seemingly arbitrary, wasteful or superfluous features of diverse systems, and explain their specificity in a unified manner.

FURTHER READING

Kinetic Proofreading

(François and Altan-Bonnet, 2016) “The case for absolute ligand discrimination: modeling information processing and decision by immune T cells.”

(Goldstein, Faeder and Hlavacek, 2004) “Mathematical and computational models of immune-receptor signalling.”

(Hopfield, 1974) “Kinetic proofreading: a new mechanism for reducing errors in biosynthetic processes requiring high specificity.”

(McKeithan, 1995) “Kinetic proofreading in T-cell receptor signal transduction.”

Conformational Proofreading

(Savir and Tiusty, 2007) “Conformational proofreading: the impact of conformational changes on the specificity of molecular recognition.”

(Savir and Tiusty, 2010) “RecA-mediated homology search as a nearly optimal signal detection system.”

(Savir and Tiusty, 2013) “The ribosome as an optimal decoder: a lesson in molecular recognition.”

Rules for Gene Regulation

(Sasson et al., 2012) “Mode of regulation and the insulation of bacterial gene expression.”

(Savageau, 1974) “Genetic regulatory mechanisms and the ecological niche of *Escherichia coli*.”

(Savageau, 1977) “Design of molecular control mechanisms and the demand for gene expression.”

(Shinar et al., 2006) “Rules for biological regulation based on error minimization.”

EXERCISES

- 7.1 *At any rate*: Determine the error rate in the proofreading process of Equation 7.2.9. What conditions (inequalities) on the rates allow for effective kinetic proofreading?

Solution:

The rate of change of $[cC]$ is governed by the collisions of c and C with on-rate k , their dissociation with off-rate k'_c and the formation of $[cC^*]$ at rate m :

$$d[cC]/dt = kcC - (m + k'_c)[cC] \quad (\text{P7.1})$$

so that at steady state, defined by $d[cC]/dt = 0$, we have

$$[cC] = kcC/(m + k'_c) \quad (\text{P7.2})$$

Similarly, $[cC^*]$ is produced at rate m , dissociates at rate l'_c and produces a product at rate v :

$$\frac{d[cC^*]}{dt} = m[cC] - (v + l'_c)[cC^*] \quad (\text{P7.3})$$

so that at steady state, using Equation P7.2,

$$[c^*C] = m/(v + l'_c)[cC] = mkcC/(v + l'_c)(m + k'_c) \quad (\text{P7.4})$$

Similar considerations for the wrong ligand w can be made, noting that the on-rate k , the complex formation rate m and the product formation rate v are the same as for c ,

but that the off-rates k'_w and l'_w are larger than the corresponding rates for c due to the weaker affinity of w to C . Thus,

$$[w^*C] = mkwC / (\nu + l'_w)(m + k'_w) \quad (\text{P7.5})$$

The error rate F is the ratio of incorrect and correct production rates:

$$F = \nu[w^*C] / \nu[c^*C] = \frac{w}{c} (\nu + l'_c)(m + k'_c) / (\nu + l'_w)(m + k'_w) \quad (\text{P7.6})$$

When $\nu \ll l'_c, l'_w$ and when $m \ll k'_c, k'_w$, we have the minimal error rate in this process:

$$F = \frac{w}{c} l'_c k'_c / l'_w k'_w \quad (\text{P7.7})$$

Thus, minimal errors require that the complexes $[wC]$ dissociate much faster than the rate of formation of $[w^*C]$, and that $[w^*C]$ dissociate much faster than the rate of product formation. This gives many opportunities for the wrong ligand to fall off of the complex, before an irreversible step takes place.

In processes where the dissociation from the state $[cC]$ and $[c^*C]$ are based on the same molecular site (e.g., the tRNA–codon interaction), we have $l'_c = k'_c$, and the same for w , so that $F = F_0^2$ (assuming equal concentrations of the targets $w = c$) where F_0 is the equilibrium error rate.

7.2 *Detailed balance*: Determine the error rate in a kinetic proofreading scheme in which all transitions have a reverse reaction. The transition from $[cC]$ to $[c^*C]$ occurs at a forward rate m_c and backward rate m'_c , transitions from $[c^*C]$ to $c + C$ occur at forward rate l_c and backward rate l'_c . There are corresponding rate constants for w . Assume that the product formation rate ν is negligible compared to the other rates. Consider the case where all reactions occur at equilibrium. Use the **detailed balance** conditions, where the flux of each reaction is exactly equal to the flux of the reverse reaction, resulting in zero net flux along any cycle (also known in biochemistry as the **thermodynamic box** conditions).

- Show that detailed balance requires that $k_c m_c l'_c = k'_c m'_c l_c$, and the same for w .
- Calculate the resulting error rate F . Explain your results.

7.3 *Optimal tRNA concentrations*: In order to translate a codon, different tRNAs randomly bind the ribosome and unbind if they do not match the codon. This means that, on average, many different tRNAs need to be sampled for each codon until the correct match is found. Still, the ribosome manages to translate several dozen codons per second. We will try to consider the optimal relations between the concentrations of the different tRNAs, which allow the fastest translation process, in a toy model of the ribosome.

Let the concentration of tRNA number j (j goes from 1 to the number of different types of tRNAs in the cell) be c_j . The relative concentration of tRNA number j is

therefore $r_j = c_j/\Sigma c_j$. Suppose that each tRNA spends an average time t_0 bound to the ribosome before it unbinds or is used for translation.

- (a) What is the average time needed to find the correct tRNA for codon j ? Assume that there is no delay between unbinding of a tRNA and the binding of a new tRNA, and neglect the unbinding of the correct tRNA.
- (b) Suppose that the average probability to find codon j in the coding region of genes in the genome is p_j . What is the optimal relative concentration of each tRNA that allows the fastest translation? Use a Lagrange multiplier to make sure that $\Sigma r_j = 1$.

Solution:

- a. When codon j is to be read, the ribosome must bind tRNA _{j} . The probability that a random tRNA is tRNA _{j} is r_j . Thus, on average one must try $1/r_j$ tRNAs before the correct one binds the ribosome. Hence, the average time to find the correct tRNA for codon j is

$$T_j = t_0/r_j$$

- b. The time to translate the average codon is the sum of the times T_j weighted by the codon probabilities in the genome:

$$T = \Sigma T_j p_j = \Sigma p_j t_0/r_j$$

To minimize the translation time, we need to minimize T . Taking the derivative of T with respect to each r_j , we look for the relative concentrations that yield a minimum. To find the minimum we use a Lagrange multiplier L to make sure that $\Sigma r_j = 1$:

$$dT/dr_j = d/dr_j (\Sigma p_j t_0/r_j + L \Sigma r_j) = -t_0 p_j/r_j^2 + L = 0$$

Solving for r_j , and using a value of L such that $\Sigma r_j = 1$, yields an optimal r_j that is proportional to the square root of the codon probability p_j :

$$r_j^{opt} = \sqrt{p_j} / (\Sigma_j \sqrt{p_j})$$

Thus, the rarer the codon, the lower the relative concentration of its tRNA.

7.4 Optimal genetic code for minimizing errors: In this exercise we consider an additional mechanism for reducing translation errors, based on the structure of the genetic code.

- (a) First consider a code based on an alphabet of two letters (0 and 1), and where codons have two letters each. Thus, there are four possible codons ([00], [01],

[10] and [11]). This genetic code encodes two amino acids, A and B (and no stop codons). Each amino acid is assigned two of the four codons.

- a. What are the different possible genetic codes?
- b. Assume that misreading errors occur, such that a codon can be misread as a codon that differs by one letter (e.g., [00] can be misread as [01] or [10], but not as [11]). Which of the possible codes make the fewest translation errors?
- c. Assume that the first letter in the codon is misread at a higher probability than the second letter (e.g., [00] is misread as [10] more often than as [01]). Which of the codes has the lowest translation errors?
- d. Study the real genetic code in [Figure 7.2](#). Compare the grouping of codons that correspond to the same amino acid. How can this ordering help reduce translation errors?
- e. Based on the structure of the genetic code, can you guess which positions in the codon are most prone to misreading errors? Can you see in the code a reflection of the fact that U and C in the third letter of the codon cannot be distinguished by the translation machinery (a phenomenon called “third-base wobble”)?
- f. In the real genetic code, chemically similar amino acids tend to be encoded by codons that differ by only one letter. Discuss how this might reduce the impact of translation errors on the fitness of the organism. In this sense, the genetic code is said to be “one in a million” (Freeland and Hurst, 1998).

7.5 *Optimal position of stop codons:* The genetic code contains three stop codons that instruct translation to stop. One question is why three stop codons? And why these specific codons? In this exercise, we consider an answer based on minimizing the effects of **frameshift errors**: The ribosome can, at a low frequency, skip one letter in the mRNA. As a result, all subsequent codons are one letter off.

- a. Explain why it is beneficial to hit a stop codon as soon as possible in case of a frameshift.
- b. Itzkovitz and Alon (2007) found that the codons for the most common amino acids tend to reach stop codons upon a frameshift much more frequently than rare amino acids. Explain why this can help minimize the effect of frameshift errors.
- c. What might be the trade-off between having many and few stop codons in the genetic code?

7.6 *Optimal handicap in conformational proofreading:* The free energy of binding of the correct and wrong target, ΔG_c and ΔG_w become $\Delta G_c + h$ and $\Delta G_w + h$ when a handicap h is added. Binding goes as $1/(1 + e^{\Delta G})$. Compute the value of h that maximizes the difference in binding of the correct and wrong targets as a function of ΔG_c and ΔG_w .

- 7.7 *Optimal mismatch*: An enzyme has a flexible binding pocket of size x , to bind a rigid target of size y . The most dangerous wrong target is also rigid and has size z which is larger than y . In an optimal conformational proofreading design, would x in the unbound conformation be larger or smaller than y ?
- 7.8 *Demand rules for genes of concordant and opposing functions*: Genes in the same pathway tend to have the same regulation mode (all activation or all repression), as in the SIM network motif of [Chapter 4](#).
- Why does this make sense in light of the demand rules?
 - What would you predict for genes that carry out antagonistic function, such as genes for synthesis of a molecule and genes for its degradation?
- 7.9 *Demand rules for phosphorylation*: Proteins in the cell can be activated by phosphorylation, in which a phosphoryl group is added to a specific site on the protein. We will call such proteins phospho-active. Other proteins are inactivated by a phosphorylation event, and we will call these phospho-inactive. Phosphoryl groups are added by kinases, and removed by phosphatases.
- Develop demand rules to predict when a phospho-active versus a phospho-inactive design is best.
 - How does the answer depend on the relative error rates of kinases and phosphatases?
 - Proteins are often phosphorylated on multiple sites, with some sites activating and the others inhibiting. What rules might explain this regulation?
 - The enzyme that synthesizes glycogen, the liver's sugar-storage molecule, is phospho-inactivated. The enzyme that breaks glycogen down to glucose is phospho-activated. Both phosphorylations are carried out by the same kinase, which in turn is activated by the starvation hormone glucagon. Draw the circuit. Discuss using the demand rules.
- 7.10 *Avoiding crosstalk by efficient coding*: Suppose that transcription factors in the cell recognize a site that is $N = 6$ letters (base-pairs) long. Each factor has a site sequence which it binds most strongly, and it binds with a weaker but still effective strength sites which are different by up to $m = 2$ letters.
- How can you design the binding sites for different transcription factors to avoid crosstalk?
 - What is the maximal number of different factors that can be present in one cell, if no crosstalk is allowed? (Sengupta et al., 2002)
 - In engineering, one can code information using binary strings such that there is a minimal chance for a misreading error to affect the meanings. How is this coding problem related to transcription factor sites?

BIBLIOGRAPHY

- François, P. and Altan-Bonnet, G. 2016. 'The case for absolute ligand discrimination: modeling information processing and decision by immune T cells', *Journal of Statistical Physics*, 162(5), pp. 1130–1152. doi: 10.1007/s10955-015-1444-1.
- Freeland, S.J. and Hurst, L.D. 1998. 'The genetic code is one in a million', *Journal of Molecular Evolution*, 47(3), pp. 238–248.
- Goldstein, B., Faeder, J. R. and Hlavacek, W.S. 2004. 'Mathematical and computational models of immune-receptor signalling', *Nature Reviews Immunology*, 4(6), 2004, pp. 445–456. doi: 10.1038/nri1374.
- Hershko, A. and Ciechanover, A. 1998. 'The ubiquitin system', *Annual Review of Biochemistry*, 67(1), pp. 425–479. doi: 10.1146/annurev.biochem.67.1.425
- Hopfield, J.J. 1974. 'Kinetic proofreading: a new mechanism for reducing errors in biosynthetic processes requiring high specificity', *Proceedings of the National Academy of Sciences*, 71(10), 4135–4139. doi: 10.1073/pnas.71.10.4135.
- Hopfield, J.J. et al. 1976. 'Direct experimental evidence for kinetic proofreading in amino acylation of tRNA^{Ala}', *PNAS*, 73(4), 1164–1168. doi: 10.1073/pnas.73.4.1164
- Itzkovitz, S. and Alon, U. 2007. 'The genetic code is nearly optimal for allowing additional information within protein-coding sequences', *Genome Research*, 17(4), pp. 405–412. doi: 10.1101/gr.5987307
- McKeithan, T.W. 1995. 'Kinetic proofreading in T-cell receptor signal transduction', *Proceedings of the National Academy of Sciences*, 92(11), pp. 5042–5046. doi: 10.1073/pnas.92.11.5042.
- Rape, M., Reddy, S.K. and Kirschner, M.W. 2006. 'The processivity of multiubiquitination by the APC determines the order of substrate degradation', *Cell*, 124(1), pp. 89–103. doi: 10.1016/j.cell.2005.10.032.
- Rosenheim, J.A., Alon, U. and Shinar G. 2010. 'Evolutionary balancing of fitness-limiting factors', *The American Naturalist*, 175(6), pp. 662–674. doi: 10.1086/652468.
- Sancar, A. and Reardon, J.T. 2004. 'Nucleotide excision repair in *E. coli* and man', *Advances in Protein Chemistry*, 69, 43–71. doi: 10.1016/S0065-3233(04)69002-4.
- Sasson, V. et al. 2012. 'Mode of regulation and the insulation of bacterial gene expression', *Molecular Cell*, 46(4), 399–407. doi: 10.1016/j.molcel.2012.04.032.
- Savageau, M.A. 1974. 'Comparison of classical and autogenous systems of regulation in inducible operons', *Nature*, 252(5484), 546–549. doi: 10.1038/252546a0.
- Savageau, M.A. 1974. 'Genetic regulatory mechanisms and the ecological niche of *Escherichia coli*', *Proceedings of the National Academy of Sciences* 71(6), 2453–2455. doi: 10.1073/pnas.71.6.2453.
- Savageau, M.A. 1977. 'Design of molecular control mechanisms and the demand for gene expression', *Proceedings of the National Academy of Sciences*, 74(12), 5647–5651. doi: 10.1073/pnas.74.12.5647.
- Savir, Y. and Tlusty, T. 2007. 'Conformational proofreading: the impact of conformational changes on the specificity of molecular recognition', *PLoS ONE*, 2(5), e468. doi: 10.1371/journal.pone.0000468.
- Savir, Y. and Tlusty, T. 2010. 'RecA-mediated homology search as a nearly optimal signal detection system', *Molecular Cell*, 40(3), 388–e396. doi: 10.1016/j.molcel.2010.10.020.
- Savir, Y. and Tlusty, T. 2013. 'The ribosome as an optimal decoder: a lesson in molecular recognition', *Cell*, 153(2), 471–479. doi: 10.1016/j.cell.2013.03.032.
- Sengupta, A.M., Djordjevic, M. and Shraiman, B.I. 2002. 'Specificity and robustness in transcription control networks', *Proceedings of the National Academy of Sciences*, 99(4), pp. 2072–2077. doi: 10.1073/pnas.022388499.
- Shinar, G. et al. 2006. 'Rules for biological regulation based on error minimization', *Proceedings of the National Academy of Sciences*, 103(11), 3999–4004. doi: 10.1073/pnas.0506610103.
- Tlusty, T., Bar-Ziv, R. and Libchaber, A. 2004. 'High-fidelity DNA sensing by protein binding fluctuations', *Physical Review Letters*, 93, 25, 2004, 413. doi: 10.1103/PhysRevLett.93.258103.

Robust Signaling by Bifunctional Components

The proofreading principles of the previous chapter can help a receptor to bind the correct signal molecule. Receptors can thus reliably measure the external concentration of the signal. We now ask how this information can be robustly transduced into the cell, so that the cell can respond to the signal.

The task of transducing information is carried out by **signal-transduction circuits**. These circuits begin with a receptor, which has one part outside the cell and another inside the cell. When the receptor binds the signal molecule outside the cell, it acts to chemically modify proteins inside the cell, thereby changing their activities. These proteins in turn modify other proteins, and so on. Finally, transcription factors get modified and activate the genes that respond to the signal. These reactions take place on the timescale of seconds to minutes. In Part 1 of the book, we considered these pathways as instantaneous, and regarded them as the signals S_X and S_Y that activate transcription factors. Now we look into these pathways in more detail.

We will ask a robustness question: How can signal-transduction circuits work precisely despite the fact that they are built of proteins whose concentrations vary from cell to cell and over time? Thus, some cells will have more receptors than others, even when the cells are genetically identical and grown in the same conditions. If the average is 1000 receptors per cell, one cell will have 800 and another 1200. The same applies to each of the other proteins in the circuit (see Appendix D). How then can cells precisely transmit information and compute a response that is just right for the input signal level?

8.1 ROBUST INPUT-OUTPUT CURVES

Let's define the robustness we seek. Suppose that a signal-transduction circuit has an input signal S , sensed by the receptor. The output of the circuit is the concentration of activated transcription factor. The output as a function of input, $f(S)$, is the circuit's **input-output curve**. We ask how the input-output curve can be precise despite cell-to-cell variations in

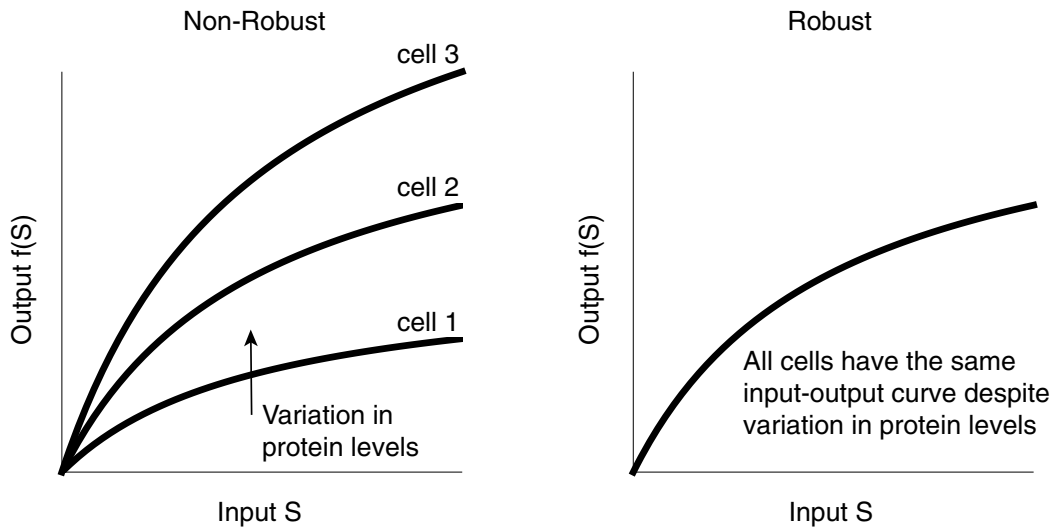


FIGURE 8.1

the concentrations of the proteins that make up the circuit, such as the receptor and the transcription factor.

If the input–output curve depends on the concentrations of the proteins that make up the circuit, we say that it is **non-robust** (Figure 8.1). Since variations in protein concentration are an unavoidable property of biological matter, a non-robust $f(S)$ means that different individual cells will show a different response to the same input signal. The input is inaccurately read by most cells.

In contrast, a **robust input–output curve** is insensitive to (and ideally completely independent of) variations in the concentrations of the proteins that make up the circuit. A robust input–output curve $f(S)$ allows all cells to have the same output to a given input signal (Figure 8.1). Cells accurately perceive their environment.

8.2 SIMPLE SIGNALING CIRCUITS ARE NOT ROBUST

Input–output robustness is difficult to achieve. To illustrate this, let's consider a circuit made of the typical components of signal-transduction pathways (Figure 8.2). The signal S is sensed by a receptor protein X . Signal causes the receptor to change conformation and thus transmit information into the cell. Information is passed to a messenger protein Y in the form of a chemical modification, such as phosphorylation, in which a phosphoryl group is added to Y . To do this, X acts as a kinase, an enzyme that takes phosphoryl from ATP and adds it to protein Y . We say that X phosphorylates Y , at a rate that depends on the input signal.

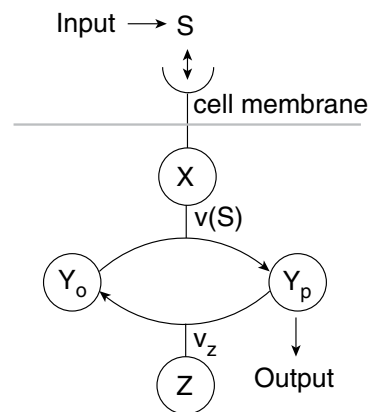


FIGURE 8.2

Phosphorylation causes Y , a transcription factor, to change into its active conformation ($Y^* = Y_p$). The phosphorylated messenger, Y_p , binds promoters and activates output genes. Hence, Y_p is the output of the circuit. To stop the signaling when the signal goes away, Y_p is continually dephosphorylated by a phosphatase protein Z . The phosphatase removes the phosphoryl group from Y_p and returns it to the cytoplasm in the form of inorganic phosphate, denoted P_i .

We will now solve the input–output curve of this circuit, to see that the curve depends on the concentrations of all of the three proteins that make up the circuit, X , Y and Z . We'll see that there is no input–output robustness. The math is simple.

Solved Example 8.1: Show That the Signal-Transduction Pathway in Figure 8.2 Has a Non-Robust Input–Output Curve

Receptor X phosphorylates Y at a rate $v(S)$ that depends on input signal S . The phosphorylation is removed by a phosphatase Z at rate v_z . Let's compute the steady-state input–output curve $Y_p = f(S)$, and see how it depends on the concentrations of the three proteins in the circuit, X , Y and Z .

Y can either be phosphorylated, Y_p , or not, Y_0 , so that the total concentration of Y protein is the sum of these two, $Y_T = Y_0 + Y_p$. This is an example of a **conservation law**. We'll describe the dynamics of Y_p using **mass-action kinetics**: phosphorylation occurs when X and Y_0 collide, at a rate $v(S)XY_0$. Dephosphorylation occurs when Y_p and Z collide, at rate v_zZY_p . It's convenient to use the conservation law to replace Y_0 by $Y_T - Y_p$. At steady state, phosphorylation and dephosphorylation must balance, $dY_p/dt = v(S)X(Y_T - Y_p) - v_zZY_p = 0$. Solving for Y_p , we find that the input–output curve is an increasing function of the signal $v(S)$:

$$f(S) = Y_p = \frac{v(S)XY_T}{v(S)X + v_zZ} \quad (8.2.1)$$

The input–output curve rises with the input signal S , so that the cell receives information about the input signal. This input–output curve, however, depends on the levels of all three proteins in the circuit: the receptor X , total messenger Y_T and phosphatase Z (Figure 8.3). The more X and Y a cell has, or the less Z it has, the higher the input–output curve for a given signal S . Since protein concentrations typically vary by tens of percents, and this variation lasts an entire cell generation time, it will be common to have a twofold difference in output between cells. There is no input–output robustness.

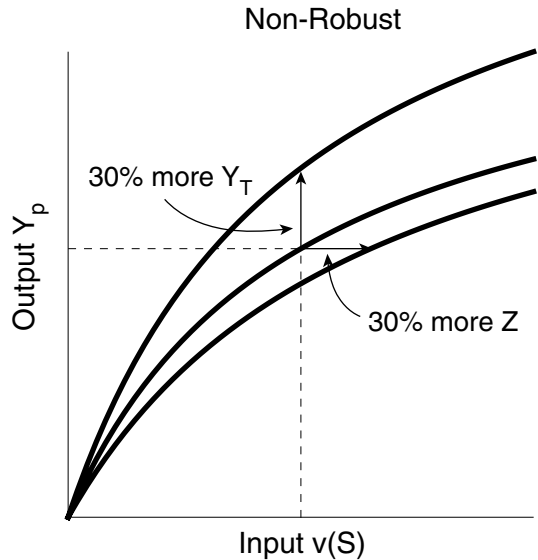


FIGURE 8.3

8.3 BACTERIAL TWO-COMPONENT SYSTEMS CAN ACHIEVE ROBUSTNESS

In order to achieve input–output robustness, the protein levels must somehow cancel out in the expression for the input–output curve. Remarkably, such robustness occurs in bacterial **two-component systems** – a class of thousands of systems. Each two-component system is made of a receptor X that senses specific inputs, and its dedicated messenger Y .

At the heart of the design of two-component systems is a **bifunctional** component: The receptor X catalyzes two opposing reactions: both phosphorylating and dephosphorylating Y (Figure 8.4).

Thus, the opposing kinase and phosphatase activities **are rolled up into the same protein**, instead of being separated on two different proteins.¹ The receptor has another seemingly arbitrary biochemical detail that will turn out to be crucial: X is an **autokinase** and a **phosphotransferase**: it first uses ATP to phosphorylate itself and only then transfers the phosphoryl group to Y (Figure 8.4).

A canonical example of a two-component system is the osmotic response system of *E. coli*, EnvZ/OmpR. Here, the receptor X is EnvZ, and the messenger Y is OmpR. When external osmolarity is high, the receptor X phosphorylates itself and then transfers the phosphoryl group to Y_0 to form Y_p . The output, Y_p , transcriptionally activates osmo-response genes such as transporters and enzymes that act to adjust the cell to the osmotic pressure in its environment. A robust input–output curve is crucial in this system because the response to osmotic pressure had better be accurate, to avoid the cell bursting or imploding.

Tom Silhavy and colleagues discovered that X carries out two antagonistic reactions (Hsing et al., 1998): it not only acts as a kinase that phosphorylates Y ; it is also the phosphatase of Y_p (Figure 8.4). It thus both adds and removes the chemical modification. This bifunctionality, acting as both a kinase and phosphatase, was suggested by Russo and Silhavy (1993) to enable robustness in the circuit. The intuitive reason is that a change in the concentration of the bifunctional protein X changes both phosphorylation and dephosphorylation rates by the same factor, thus canceling out the effect on the steady-state output Y_p .

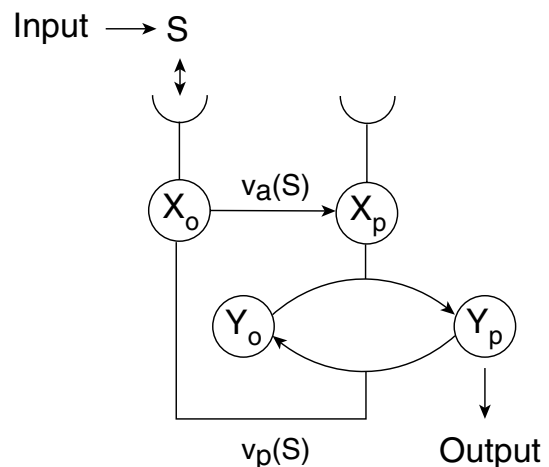


FIGURE 8.4

¹ Note that every enzyme catalyzes a reaction and its reverse, with total flux determined by the concentrations of product and substrate. Bifunctional enzymes do something more specific: they catalyze different reactions, at different catalytic sites. For example, phosphorylation entails breaking down ATP to ADP, whereas dephosphorylation is not the reverse reaction which returns the phosphoryl to ADP, but instead moves the phosphoryl into the cytoplasm as inorganic phosphate.

Robustness in this system was modeled mathematically and demonstrated experimentally by Eric Batchelor and Mark Goulian (2003). Batchelor and Goulian experimentally changed the levels of the proteins X and Y in the circuit, by deleting the gene for X or Y in the DNA, and adding back a copy of the gene on a circular piece of DNA called a plasmid. They fused the added gene to a promoter that is regulated by a chemical inducer that can be added to the liquid media in which the cells grow. The chemical inducer allowed them to control the concentrations of X or Y in the cell. They found that the output is robust despite large changes in the levels of the proteins. For example, the output (the amount of Y_p at a given input level of osmolarity) changed by less than 20% upon 10-fold changes in the amount of total Y protein.

Shinar et al. (2007) extended the theoretical analysis of this system, showing how the special biochemical features of the receptor (autokinase, phosphotransferase and phosphatase) combine to make Y_p levels completely insensitive to variations in the concentrations of all proteins in the circuit – X and Y – and yet responsive to the input signal of the system, S .

It's fun to solve this system, in order to see how this cancellation comes about. The solution also uses a black-box trick that can be generalized to other systems.

Solved Example 8.2: Show That the Bifunctional EnvZ/OmpR Design Has Input–Output Robustness at Steady State with Respect to Fluctuations in the Levels of the Proteins X and Y

One way to solve this example is to write seven mass-action equations for the system of Figure 8.4, and find their fixed point. This algebra is described in Exercise 8.3. An easier way to obtain the input–output curve presents itself when we view the system as a **black box** that breaks down ATP and releases phosphoryl groups back to the cytoplasm (Figure 8.5). Consider the fluxes of phosphoryl into and out of the box. The influx, J_{in} , is due to the receptor that takes phosphoryl from ATP and phosphorylates itself at a rate that depends on the input signal, $v_a(S)$. Thus, $J_{in} = v_a(S)X_0$. The outflux is the rate of dephosphorylation of Y_p by the receptor, $J_{out} = v_p(S)X_0Y_p$, which releases the phosphoryl groups back into the cytoplasm as inorganic phosphate, P_i . At steady state, influx and outflux must balance, $J_{in} = J_{out}$, otherwise the black box would fill up with phosphoryl groups. This means that

$$v_a(S)X_0 = v_p(S)X_0Y_p \quad (8.3.1)$$

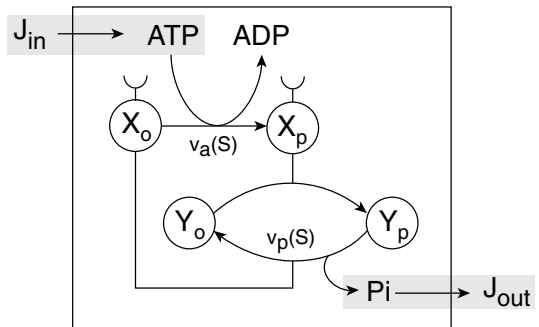


FIGURE 8.5

Notice how X_0 can be elegantly canceled out from both sides of the equation (as long as $X_0 \neq 0$). We obtain a robust input–output curve $f(S) = Y_p$ that depends only on kinetic rate constants:

$$Y_p = v_a(S)/v_p(S) \quad (8.3.2)$$

The output Y_p thus responds to the input signal S , through the rate constants $v_a(S)$ and $v_p(S)$. Importantly, the output Y_p *does not depend on the level of any of the proteins in the system*. The mechanism thus shows a robust input–output relationship.

Input–output robustness is achieved by the coordinated effect of the biochemical details of this system. If the receptor was not bifunctional, and instead dephosphorylation was carried out by a separate phosphatase protein Z , the balance of phosphoryl influx and outflux would require that $X \sim ZY_p$. No cancellation of X is possible. This would result in a steady-state level $Y_p \sim X/Z$ that depends on the intracellular levels of both X and Z . Robustness would be lost. Similarly, the two-step nature of the kinase is also essential for robustness. If the receptor directly transferred a phosphoryl group from ATP to Y_0 without first phosphorylating itself, the influx would depend on the concentration of the complex XY_0 . Balancing influx and outflux gives $XY_0 \sim XY_p$. As a result, the output Y_p would be proportional to Y_0 and would thus depend on the total level of Y , Y_T , abolishing robustness.

In summary, robustness in the present mechanism seems to require the combined effects of several biochemical features. These features occur in virtually all of the thousands of known two-component systems that respond to stress and environmental signals in diverse bacterial species (Capra and Laub, 2012). For example, *E. coli* has about 30 different two-component systems and all but one have this bifunctional design. We will discuss the exceptional circuit, bacterial chemotaxis, in the next chapter.

For years, I thought that input–output robustness is impossible due to considerations of units. The units of the output Y_p are units of concentration. Units of concentration, in any mechanism that I could imagine, come from the concentrations of the proteins in the circuit (as in the simple circuit of [Figure 8.2](#)). So where do the units of concentration come from in the bifunctional mechanism? The answer is the molecular rate constants. The units of concentration come from the ratio of a uni-molecular and a bi-molecular rate constant, v_a/v_p (v_a has units of 1/time, and v_p of 1/time/concentration). These intrinsic molecular rate constants are determined by the structure of the circuit proteins, which is the same in all cells. The rate constants are therefore much more hard-wired (much less variable from cell to cell) than protein concentrations.

In fact, these intrinsic rate constants make the input–output curve even more elegant when the receptor does its two opposite functions according to Michaelis–Menten binding of the signal S (see Appendix A for an explanation of Michaelis–Menten functions). If the kinase rate rises with binding as $v_a \sim S/(K + S)$ and the phosphatase rate decreases with binding as $v_p \sim K/(K + S)$, the output becomes linear in signal, $Y_p = v_a/v_p \sim S/K$. It's nice to have a linear undistorted readout of the input information.

8.3.1 Limits of Robustness

Robustness always has its limits. In this system, robustness is lost if the total level of protein Y , denoted Y_T , falls below the robust Y_p level of Equation 8.3.2 for a given input signal (Figure 8.6). There is not enough Y protein to provide the needed output.

When this happens, analysis of the model (Exercise 8.5) shows that all of the Y molecules become phosphorylated, so that $Y_p = Y_T$. All of X is also phosphorylated $X_p = X_T$. Hence, the system is maxed out and no longer responds to the input signal (mathematically speaking, X_0 is zero and

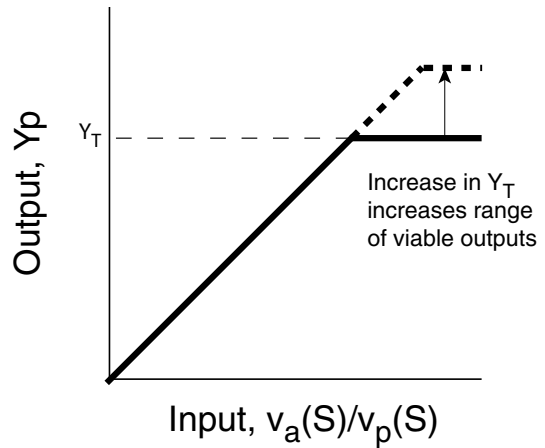


FIGURE 8.6

cannot be canceled out in the black-box equation). We conclude that both robustness and responsiveness to the signal require that total Y levels, Y_T , exceed a certain threshold, given by the maximal desired output level in the expected physiological conditions.

There is also call for caution. When studying models like this, we need to watch out for additional reactions in the cell, perhaps too weak to be experimentally detected, that can potentially ruin robustness. To explore this possibility, we can add to the model every possible reaction arrow and assign to it a small rate ε . For example, we can add spontaneous dephosphorylation of Y_p (that is, $Y_p \rightarrow Y_o + P_i$ without help from X). This spontaneous reaction is known to occur in the EnvZ/OmpR system on a timescale of minutes, which is much slower than the other reactions that take seconds or less (thus, ε is $\sim$ seconds/minutes ~ 0.01). We can also add spontaneous dephosphorylation of X_p , reverse phosphotransfer, effects of ATP and ADP as cofactors, and so on. Some of these possibilities are explored in Exercises 8.7 and 8.8. The upshot is that the effect on robustness of these additional reactions is either nonexistent, or is small (in the sense that the relative shift in the input–output curve due to protein fluctuations is of order ε).

8.3.2 Remarks on the Black-Box Approach

To analyze the robustness mechanism, we considered the system as a black box that breaks down ATP. The black-box approach can be used more generally, to suggest a wider class of systems that show robust input–output relations.

The black-box argument depends only on balancing two reactions, the entry and exit of phosphoryl groups from the box. This leaves us with freedom to add any number of reactions inside the box, as long as a stable steady state is reached and no new entry or exit points are added. For example, we can introduce a cascade of phosphotransfer events (as occurs in some bacterial signaling systems), Figure 8.7, and still maintain the robustness of Y_p .

The black box also points to system characteristics that rule out such robustness. The black box suggests that robustness of the present type cannot generally occur if there is more than one reaction that introduces (or removes) phosphoryl groups into the system. If two different influxes J_{in} and J'_{in} exist, they generally cannot be canceled out with J_{out} (in the sense of Equation

8.3.1 above), leading to a loss of robustness. If such secondary leaks or inputs are small, of order ε , robustness is only lost to order ε .

Not all bacterial signaling systems show the hallmarks of the present mechanism. Important examples include bacterial chemotaxis, which we will study in the next chapter, and sporulation. Apparently, these signaling systems do not need robust input–output relations, but instead rely on cell–cell variation in their output in order to provide a wider range of solutions to a given situation. A robust input–output mechanism should perhaps be expected only in signaling systems in which there is a sufficiently heavy fitness penalty for imprecision.

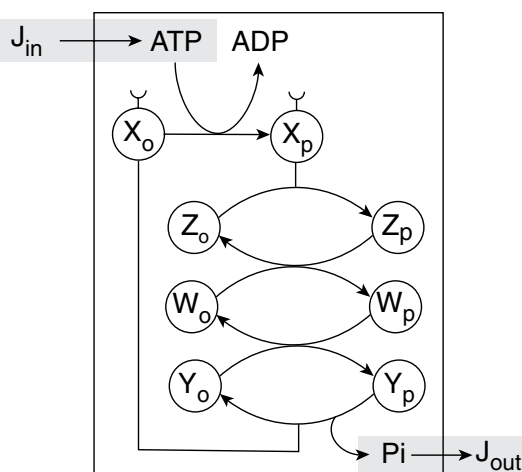


FIGURE 8.7

8.3.3 Bifunctional Components Provide Robustness in Diverse Circuits

The mechanism of robustness by bifunctional enzymes that catalyze antagonistic reactions (also called **paradoxical** enzymes) applies to other systems and organisms (Hart and Alon, 2013). In each case, a bifunctional enzyme is at the core of the mechanism, and additional biochemical features combine to allow robustness. Examples include nitrogen regulation in *E. coli*, explored in Exercise 8.12, in which a paradoxical enzyme modifies and de-modifies a key metabolic enzyme in nitrogen control. Similarly, a paradoxical enzyme in human cells makes and breaks an allosteric regulator of the main nutritional pathway, glycolysis. Paradoxical enzymes also operate in tissue-level circuits. For example, a paradoxical enzyme called Piezo1 helps maintain proper numbers of epithelial cells. It makes epithelial cells both proliferate and die according to pressure signals that indicate if there are too few or too many cells in the tissue (Gudipaty et al., 2017). In the immune system, T-cells secrete a signal molecule called IL-2 that makes them both proliferate and die, helping to maintain a desired concentration of T-cells (Hart et al., 2014). Theoretical analysis of bifunctional enzymes led to a mathematical theorem that can predict which components of a complicated biochemical reaction system might be robust (Shinar and Feinberg, 2010; Karp et al., 2012).

In summary, in biological signaling circuits, bifunctional components can provide robust input–output curves despite unavoidable fluctuations in the levels of the proteins that make up the circuit. The robustness is due to a combination of specific biochemical details, and thus provides a systems-level meaning to biochemical reactions that may otherwise appear arbitrarily complicated.

FURTHER READING

(Batchelor and Goulian, 2003) “Robustness and the cycle of phosphorylation and dephosphorylation in a two-component regulatory system.”

(Hart and Alon, 2013) “The utility of paradoxical components in biological circuits.”

(Shinar et al., 2007) “Input output robustness in simple bacterial signaling systems.”

(Shinar and Feinberg, 2010) “Structural sources of robustness in biochemical reaction networks.”

EXERCISES

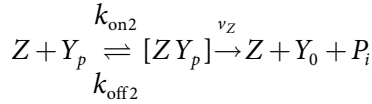
- 8.1 *Mass-action for the non-robust circuit:* Solve the mass-action kinetics of the three-protein signaling circuit of [Figure 8.2](#), taking into account the complexes of the reactants. Show that the input–output curve is not robust.

Solution:

Let’s assume that ATP binds X strongly, a realistic assumption for most signaling systems, so that free X is always bound to ATP. The reactions are



and



Thus the rate of change of the complex $[X \text{ ATP } Y_0]$ is a balance of binding, unbinding and catalysis

$$\frac{d[X \text{ ATP } Y_0]}{dt} = [X \text{ ATP}]Y_0k_{\text{on}1} - [X \text{ ATP } Y_0](\nu(S) + k_{\text{off}1})$$

At steady state, $d/dt = 0$, and hence $[X \text{ ATP } Y_0] = [X \text{ ATP}]Y_0/K_1$, where $K_1 = k_{\text{on}1}/(\nu(S) + k_{\text{off}1})$. Typically, $\nu(S)$ can be neglected because it is much smaller than the off rates in enzymes. The phosphorylation rate is $\nu(S)$ times the complex concentration, $\nu(S)[X \text{ ATP } Y_0] = \nu(S)[X \text{ ATP}]Y_0/K_1$. The dephosphorylation rate, from an analogous calculation, is $\nu_Z Z = Y_p/K_2$ where $K_2 = k_{\text{on}2}/(k_{\text{off}2} + \nu_Z)$. Balancing the two, and using $Y_0 + Y_p = Y_T$, we obtain

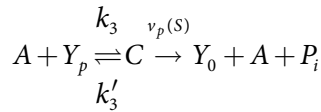
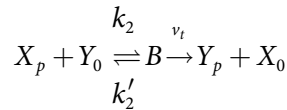
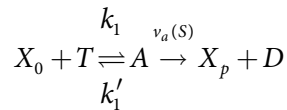
$$Y_p = \frac{[X \text{ ATP}]Y_T\nu(S)}{\nu(S)[X \text{ ATP}] + \frac{\nu_Z K_1 Z}{K_2}}$$

This input–output curve is non-robust because it depends on the concentrations of the proteins in the system.

- 8.2 *Correlated expression can increase robustness*: One way to partially address the non-robustness of the three-protein signaling circuit of Figure 8.2 is to make fluctuations in protein levels correlated, by putting Y and Z on the same operon and hence expressed from the same mRNA molecule. Discuss why this can improve robustness, but not make the circuit absolutely robust.

Two-component mechanism

- 8.3 *The seven mass-action equations for the two-component circuit*: A detailed two-component mechanism includes ATP and ADP and the complexes of the reactants. The reactions are:



where T and D are ATP and ADP, A is the complex $[XT]$, B is $[X_pY_0]$, C is $[AY_p]$ and P_i is inorganic phosphate.

- Write the mass-action kinetic equations and conservation laws.
- Solve for the steady states.
- Show that one steady state describes the case where there is enough Y protein for the desired output, and the other applies when there is not enough Y .

Solution:

- There are two conservation laws, for total X :

$$X_T = X_0 + X_p + A + B + C$$

and total Y :

$$Y_T = Y_0 + Y_p + B + C$$

The seven mass-action equations are:

$$\frac{dX_0}{dt} = -k_1 X_0 T + k_1 A + v_t B$$

$$\frac{dX_p}{dt} = -k_2 X_p Y_0 + v_a(S)A + k'_2 B$$

$$\frac{dY_0}{dt} = k'_2 B + v_p(S)C - k_2 Y_0 X_p$$

$$\frac{dY_p}{dt} = -k_3 A Y_p + k'_3 C + v_t B$$

$$\frac{dA}{dt} = k_1 X_0 T - (v_a(S) + k'_1)A$$

$$\frac{dB}{dt} = k_2 X_p Y_0 - (v_t + k'_2)B$$

$$\frac{dC}{dt} = k_3 A Y_p - (v_p(S) + k'_3)C$$

- b. Solving these at steady state shows two solutions. In solution 1, $X_{p,1} = X_T$, $Y_{p,1} = Y_T$ and all other concentrations are equal to zero. In solution 2, $Y_{p,2} = \frac{v_a(S)}{v_p} (k'_3 + v_p)/k$, which is robust because no protein concentrations appear in it.
- c. Stability analysis (try it yourself) shows that solution 2 is the only stable solution when $Y_{p,2} < Y_T$. If $Y_{p,2} \geq Y_T$, solution 1 is the only stable solution.
- 8.4 *A more precise black-box calculation:* The calculation in the main text did not take into account protein complexes. Repeat the black-box calculation taking into account the complexes.

Solution:

The exit flux is equal to the rate of dephosphorylation, $J_{out} = v_p[XY_p]$. Let's compute the concentration of the complex $[XY_p]$. The complex is formed by the binding of X to Y_p , and lost when the constituents dissociate or when the dephosphorylation reaction takes place: $d[XY_p] = k_3 X Y_p - (k'_3 + v_p)[XY_p]$. Thus, at steady state, the concentration of the complex is proportional to the product of its component concentrations: $[XY_p] = k_3 / (k'_3 + v_p) X Y_p$, which yields $J_{out} = v_p k_3 / (k'_3 + v_p) X Y_p$. At steady state, $J_{in} = J_{out}$, otherwise the black box would fill up with phosphoryl

groups. Therefore, we obtain a robust input–output curve that depends only on kinetic rate constants:

$$f(S) = Y_p = \frac{(k'_3 + v_p)v_a(S)}{k_3v_p} \quad (\text{P8.1})$$

which is the same as the full solution of Exercise 8.3b.

8.5 *Loss of robustness when Y_T is too low:*

- i. Do a black-box calculation in the case that Y_T falls below the value of Y_p expected from the robust solution. Show that the system enters a saturated state in which all of Y and all of X are phosphorylated. Why is this state a non-signaling state?
- ii. What happens when total levels of receptor X becomes very low? Does this place limitations on signaling?

8.6 *Limits to linearity of the output curve:* In the robust mechanism, when $v_a(S) \sim S/(K + S)$ and $v_p(S) \sim S/(K + S)$ both depend on Michaelis–Menten binding of the signal S , the output curve can be linear in S : $Y_p = v_a/v_p \sim S/K$. But every linearity must have its limits. Explain what processes might break linearity.

8.7 *Reverse phosphotransfer from Y_p does not affect robustness:* Add a reverse phosphotransfer reaction to the two-component model, in which $Y_p + X_0 \rightarrow X_p + Y_0$. Use the black-box approach to argue that this additional reaction does not affect robustness or the steady-state output.

8.8 *Spontaneous dephosphorylation leads to small loss of robustness:* In the EnvZ/OmpR circuit, Y_p can be spontaneously dephosphorylated without the action of X . The half-life of Y_p due to this reaction is much longer than the half-life of seconds due to the dephosphorylation catalyzed by X .

- a. Write an equation for Y_p dynamics assuming the two-component mechanism also has a reaction of spontaneous dephosphorylation at rate ε .
- b. Use the black-box approach to calculate the steady-state level of Y_p .
- c. Explain why robustness is only lost to order ε .

8.9 *Energy consumption:* The EnvZ/OmpR system continually uses up ATP, even for constant input signals.

- a. Discuss why constant energy expenditure might be useful in this signaling circuit.
- b. Suppose there are 100 molecules of X per cell that use 100 ATP/second. Estimate the fraction of the bacteria's ATP consumption that goes to running this circuit.

- 8.10 *Dynamics of the robust mechanism*: Suppose that the input signal rises in a step from level S_1 to level S_2 .
- Compute the dynamics of the robust mechanism, $Y_p(t)$. Assume low signals so that most of X and Y are unphosphorylated.
 - What is the response time?
 - Is the response time robust to variations in X ? in Y ?
- 8.11 *Positive autoregulation and robust input-output relations*: In many two-component systems, the output transcription factor Y_p is a transcriptional activator of its own gene and the gene for the receptor X (often both genes are on the same operon). Since the signaling output Y_p is robust to total X and Y levels, what can be the role of this positive autoregulation? (*Hint*: Consider strong input signals.)
- 8.12 *Paradoxical control in E. coli carbon/nitrogen balance*: *E. coli* must balance their uptake of carbon and nitrogen. The key enzyme that assimilates nitrogen (from ammonia) into biomass is the enzyme GS, made of 12 identical subunits (a dodecamer). GS produces the amino acid glutamine, Q . The dilemma is that Q is made from a carbon backbone that is a key metabolite in the tricarboxylic acid (TCA) cycle, alpha-ketoglutarate, denoted K . Making too much Q depletes K and interferes with carbon metabolism; therefore, the Q/K ratio is important and stays nearly constant in a wide range of conditions (Senior, 1975; Brauer et al., 2006). The robustness of the Q/K ratio depends on a bifunctional enzyme AT/AR which both activates and deactivates GS by removing and adding an adenylyl modification. The levels of Q and K affect the rates of these two opposite reactions, v_1 and v_2 . The twist is that AT/AR can bind two GS subunits in the same dodecamer (Figure 8.8), and hence shows a strong **avidity** effect: if it binds one subunit, it is likely to bind both. Thus, a ternary complex T in which the bifunctional enzyme binds two substrates, one modified and the other unmodified, carries out most of the reactions (Figure 8.8). The rates of adenylation and de-adenylation must be equal at steady state: $v_1(Q,K)T = v_2(Q,K)T$.
- How can this design lead to a robust Q/K ratio?
 - Explain the results of an experiment in which GS levels are controlled by expressing it from a plasmid, and the Q/K ratio is measured in wild-type cells and in cells deleted for the gene for AT/AR (Figure 8.8).
 - Propose experiments to test the hypothesis that the bifunctionality of AT/AR is causal for robustness. Use the fact that the two reactions are carried out by different parts of the enzyme AT/AR, and that mutants are available that knock out one or the other function (Hart et al., 2011).

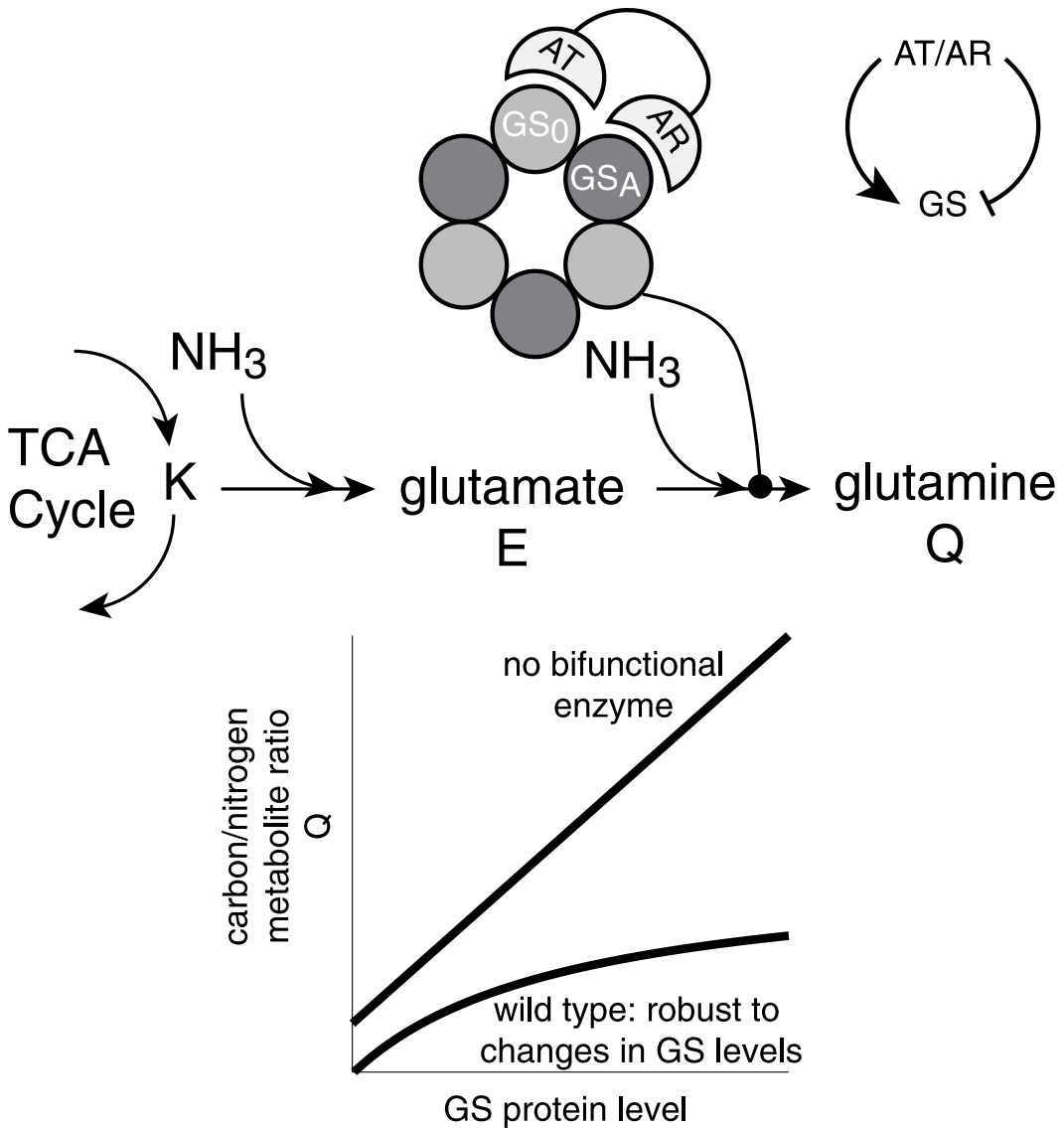


FIGURE 8.8

BIBLIOGRAPHY

- Batchelor, E. and Goulian, M. 2003. 'Robustness and the cycle of phosphorylation and dephosphorylation in a two-component regulatory system', *Proceedings of the National Academy of Sciences*, 100(2), pp. 691–696. doi: 10.1073/pnas.0234782100.
- Brauer, M.J. et al. 2006. 'Conservation of the metabolomic response to starvation across two divergent microbes', *Proceedings of the National Academy of Sciences*, 103(51), pp. 19302–19307. doi: 10.1073/pnas.0609508103.
- Capra, E.J. and Laub, M.T. 2012. 'The evolution of two-component signal transduction systems', *Annual Review of Microbiology*, 66(1), pp. 325–347. doi: 10.1146/annurev-micro-092611-150039.

- Gudipaty, S.A. et al. 2017. 'Mechanical stretch triggers rapid epithelial cell division through Piezo1', *Nature*, 543(7643), pp. 118–121. doi: 10.1038/nature21407.
- Hart, Y. et al. 2011. 'Robust control of nitrogen assimilation by a bifunctional enzyme in *E. coli*', *Molecular Cell*, 41(1), pp. 117–127. doi: 10.1016/j.molcel.2010.12.023.
- Hart, Y. and Alon, U. 2013. 'The utility of paradoxical components in biological circuits', *Molecular Cell*, 49(2), pp. 213–221. doi: 10.1016/j.molcel.2013.01.004.
- Hart, Y. et al. 2014. 'Paradoxical signaling by a secreted molecule leads to homeostasis of cell levels', *Cell*, 158(5), pp. 1022–1032. doi: 10.1016/j.cell.2014.07.033.
- Hsing, W. et al. 1998. 'Mutations that alter the kinase and phosphatase activities of the two-component sensor EnvZ', *Journal of Bacteriology*, 180(17), pp. 4538–4546.
- Karp, R.L. et al. 2012. 'Complex-linear invariants of biochemical networks', *Journal of Theoretical Biology*, 311, pp. 130–138. doi: 10.1016/j.jtbi.2012.07.004.
- Russo, F.D. and Silhavy, T.J. 1993. 'The essential tension: opposed reactions in bacterial two-component regulatory systems', *Trends in Microbiology*, 1(8), pp. 306–310. doi: 10.1016/0966-842X(93)90007-E.
- Senior, P.J. 1975. 'Regulation of nitrogen metabolism in *Escherichia coli* and *Klebsiella aerogenes*: studies with the continuous culture technique', *Journal of Bacteriology*, 11(1), p. 270. doi: 10.1186/s13256-017-1421-1.
- Shinar, G. et al. 2007. 'Input output robustness in simple bacterial signaling systems', *Proceedings of the National Academy of Sciences*, 104(50), pp. 19931–19935. doi: 10.1073/pnas.0706792104.
- Shinar, G. and Feinberg, M. 2010. 'Structural sources of robustness in biochemical reaction networks', *Science*, 327(5971), pp. 1389–1391. doi: 10.1126/science.1183372.



Taylor & Francis

Taylor & Francis Group

<http://taylorandfrancis.com>

Robustness in Bacterial Chemotaxis

9.1 INTRODUCTION

We've seen how bifunctional proteins can allow signaling circuits to robustly sense the precise level of an input signal. But not all signaling circuits need to simply sense the signal level. Some circuits are built to make more sophisticated computations, and to do so robustly. To see this, we will now consider the robustness of a remarkable system called **bacterial chemotaxis**, that allows bacteria to navigate. Bacterial chemotaxis is so well characterized on the level of molecules and behavior that it was a testing ground for important ideas in systems biology, including robustness. We will describe the biology of bacterial chemotaxis, and models and experiments that demonstrate how the computation performed by this protein circuit is made robust to changes in protein levels.

9.2 BACTERIAL CHEMOTAXIS, OR HOW BACTERIA THINK

9.2.1 Chemotaxis Behavior

When a pipette containing nutrients is placed in a plate of swimming *E. coli*, the bacteria are attracted to the mouth of the pipette and form a cloud (Figure 9.1). When a pipette with noxious chemicals is placed in the dish, the bacteria swim away from the pipette. This process, in which bacteria sense and move along gradients of specific chemicals, is called **bacterial chemotaxis**.

Chemicals that attract bacteria are called **attractants**. Chemicals that drive the bacteria away are called **repellents**. *E. coli* can sense a variety of attractants, such as sugars and the amino acids serine and aspartate, and repellents, such as certain metal ions and the amino acid leucine. Most bacterial species show chemotaxis,

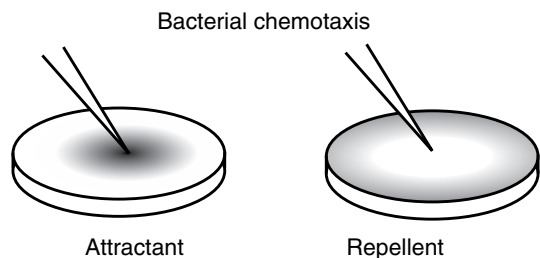


FIGURE 9.1

and some can sense and move toward stimuli such as light (phototaxis) and even magnetic fields (magnetotaxis).

Bacterial chemotaxis achieves remarkable performance despite the physical limitations faced by the bacteria. Bacteria can detect concentration gradients as small as a change of one molecule per cell volume across the cell length. They detect such gradients over background concentrations spanning five orders of magnitude. All this is done while being buffeted by Brownian noise: if the cell tries to swim straight for 10 sec, its orientation is randomized by 90° on average.

How does *E. coli* manage to move up gradients of attractants despite these physical challenges? It is evidently too small to sense the gradient along the length of its own body.¹ The answer was discovered by Howard Berg in the early 1970s: *E. coli* uses a biased-random-walk strategy to sample space and convert spatial gradients to temporal ones. In liquid environments, *E. coli* swims in a pattern that resembles a random walk. The motion is composed of **runs**, in which the cell keeps a rather constant direction, and **tumbles**, in which the bacterium stops and randomly changes direction (Figure 9.2). The runs last about 1 sec on average and the tumbles about 0.1 sec.

To sense gradients, *E. coli* compares the attractant concentration to the concentration in the past. When *E. coli* moves up a gradient of attractant, it detects an increase of attractant. As a result, it reduces the probability of a tumble (it reduces its **tumbling frequency**) and tends to continue going up the gradient. The reverse is true for repellents: if the concentration of repellent increases with time, the cell increases its tumbling frequency, and thus tends to change direction and avoid swimming toward repellents. Thus, chemotaxis senses the temporal derivative of the concentration of attractants and repellents. It follows a simple strategy: If life is getting better, keep going, and if life is getting worse, change direction.

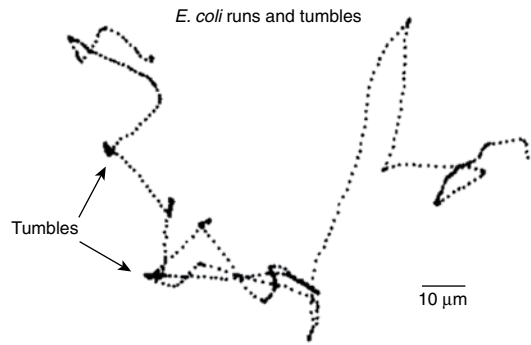


FIGURE 9.2 Adapted from (Berg, 2004).

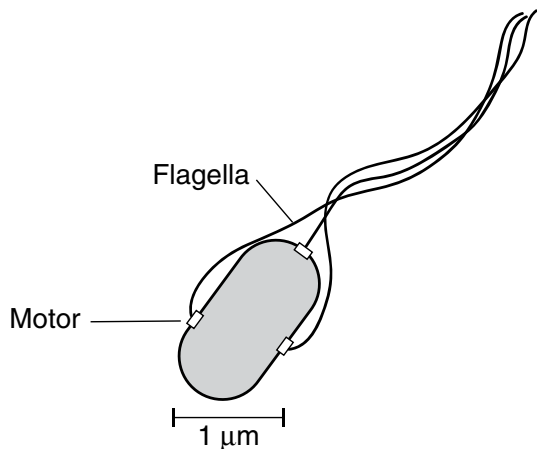


FIGURE 9.3

¹ Noise prohibits a detection system based on differences between two antennae at the two cell ends. To see this, note that *E. coli*, whose length is about 1 micron, can sense gradients as small as 1 molecule per micron in a background of 1000 molecules per cell volume. The Poisson fluctuations of the background signal, $\sqrt{1000} \sim 30$ mask this tiny gradient, unless integrated over prohibitively long times. Animal cells, whose size is about $10 \mu\text{m}$ and whose responses take minutes, can sense spatial gradients directly.

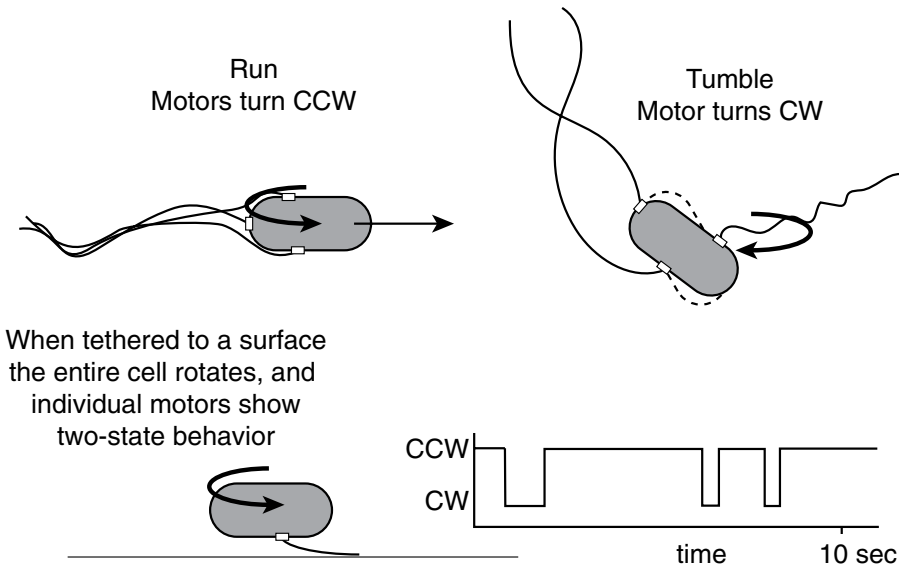


FIGURE 9.4

The runs and tumbles are generated by different states of the motors that rotate the bacterial flagella. Each cell has several flagella motors (Figure 9.3) that can rotate either clockwise (CW) or counterclockwise (CCW). When the motors turn CCW, the flagella rotate together in a bundle that pushes the cell forward. When one of the motors turns CW, its flagellum breaks from the bundle, making the cell tumble and randomize its orientation. When the motor turns CCW again, the bundle is reformed and the cell swims in a new direction (Figure 9.4).

9.2.2 Response and Exact Adaptation

The basic features of chemotaxis can be described by a simple experiment. Bacteria are observed under a microscope swimming in a liquid with no gradients. The cells display runs and tumbles, with an average **steady-state tumbling frequency** f , on the order of $f \sim 1 \text{ sec}^{-1}$. We now add an attractant such as aspartate to the liquid, uniformly in space. The attractant concentration thus increases at once, but no spatial gradients are formed. The cells sense an increase in attractant levels, no matter which direction they are swimming. They think that things are getting better and suppress tumbles: the tumbling frequency of the cells plummets within about 0.1 sec (Figure 9.5).

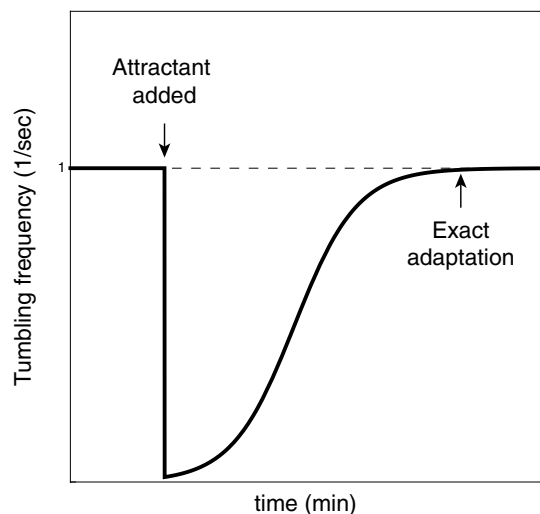


FIGURE 9.5

After a while, however, the cells realize they have been fooled. The tumbling frequency of the cells begins to increase, even though attractant is still present (Figure 9.5). This process, called **sensory adaptation**, is common to many biological sensory systems. For example, when we move from light to dark, our eyes at first cannot see well, but they soon adapt, and we can sense small changes in contrast. Adaptation in bacterial chemotaxis takes several seconds to several minutes, depending on the size of the attractant step.²

Bacterial chemotaxis shows **exact adaptation**: the tumbling frequency in the presence of attractant returns precisely to the same level as before attractant was added. In other words, *the steady-state tumbling frequency is independent of attractant levels*. If more attractant is now added, the cells again show a decrease in tumbling frequency, followed by exact adaptation. Changes in attractant concentration can be sensed as long as attractant levels do not saturate the receptors that detect the attractant.

Exact adaptation poises the sensory system at an activity level where it can respond to multiple steps of the same attractant, as well as to changes in the concentration of other attractants and repellents that can occur at the same time. It prevents the system from straying away from a favorable steady-state tumbling frequency that is required to efficiently scan space by random walk.

9.3 THE CHEMOTAXIS PROTEIN CIRCUIT

We now look inside the *E. coli* cell and describe the protein circuit that performs the response and adaptation computations. The input to this circuit is the attractant concentration, and its output is the probability per unit time that motors turn CW, which determines the cells' tumbling frequency (Figure 9.6). The chemotaxis circuit was worked out using genetics, physiology and biochemistry, starting with J. Adler in the late 1960s, followed by several labs, including those of D. Koshland, S. Parkinson, M. Simon, J. Stock and others.

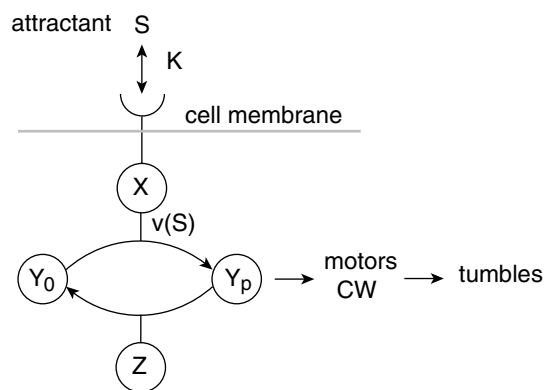


FIGURE 9.6

Attractant and repellent molecules are sensed by specialized receptors. The attractant and repellent molecules bound by a receptor are called its **ligands**. *E. coli* has five types of receptors, each of which can sense several ligands. There are a total of several thousand receptor proteins in each cell. They are localized in a cluster on the membrane, such that ligand binding to one receptor affects the state of neighboring receptors. Thus, a single ligand binding event is amplified, because it can affect more than one receptor (Bray, 2002), increasing the sensitivity of this molecular detection device (Segall, Block and Berg, 1986; Jasuja et al., 1999; Sourjik and Berg, 2004).

² Each individual cell has a fluctuating tumbling frequency signal, so that the tumbling frequency varies from cell to cell and also varies along time for any given cell (Ishihara et al., 1983; Korobkova et al., 2004). The behavior of each cell shows the response and adaptation characteristics within this noise.

Inside the cell, each receptor is bound to a protein kinase called CheA.³ We will consider the receptor and the kinase as a single entity, called X . X transits rapidly between two states, active (denoted X^*) and inactive, on a timescale of microseconds. When X is active, X^* , it phosphorylates a response-regulator protein, CheY which we will denote Y . Phosphorylated Y , denoted Y_p , diffuses through the cell. It can bind the flagellar motor and increase the probability that it switches from CCW to CW rotation. Thus, the higher the concentration of Y_p , the higher the tumbling frequency (Cluzel, Surette and Leibler, 2000).

The phosphorylation of Y_p is removed by the phosphatase CheZ, denoted Z . At steady state, the opposing actions of X^* and Z lead to a steady-state Y_p level and a steady-state tumbling frequency.

Thus, the main pathway in the circuit is phosphorylation of Y by X^* , leading to tumbles (Figure 9.6). We now turn to the mechanism by which attractant and repellent ligands can affect the tumbling frequency.

9.3.1 Attractants Lower the Activity of X

When a ligand S binds receptor X , it changes the probability⁴ that X will assume its active state X^* . The concentration of X in its active state is called the **activity of X** . Attractant ligands reduce the activity X^* , and hence reduce the rate at which X phosphorylates Y . Adding attractant therefore makes levels of Y_p drop, resulting in fewer tumbles. These responses occur within less than 0.1 sec. The response time is mainly limited by the time it takes Y_p to diffuse to the motors that are distributed all around the cell membrane.

The pathway from X to Y to the motor explains the initial response in Figure 9.5, in which attractant reduces tumbling. The reduction in activity X^* due to the binding of attractant S is well described by a Hill function (Figure 9.7):

$$X^* = \frac{X_{\max}}{1 + \left(\frac{S}{K}\right)^n}$$

where $X_{\max}$ is the maximal activity. The half-way-point for reduction of activity is K , the binding constant of the attractant to the receptor. The Hill coefficient n is due to clusters of n receptors that show

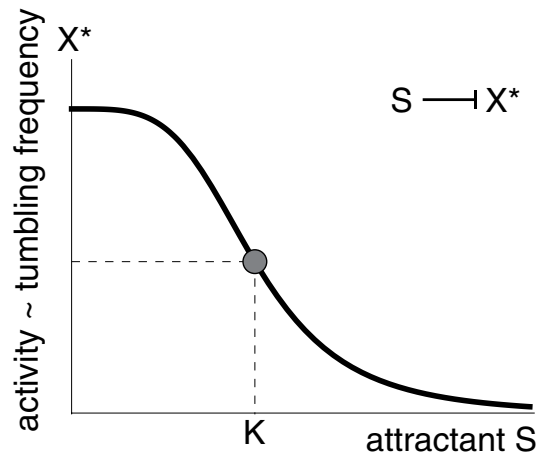


FIGURE 9.7

³ The chemotaxis genes are named with the three-letter prefix *che*, because mutants in these genes are unable to perform chemotaxis.

⁴ Note the strong separation of timescales. The conformation transitions between X and X^* are on a microsecond timescale. Ligands remain bound to the receptor for about 1 msec. Therefore, many transitions occur within a single-ligand binding event. The activity X^* is obtained by averaging over many transitions (Asakura and Honda, 1984; Mello, Shaw and Tu, 2004; Keymer et al., 2006). Phosphorylation–dephosphorylation reactions equilibrate on the 0.1-sec timescale, and methylations occur on the sec-minute timescale.

cooperativity: binding of ligand to one receptor in the cluster changes the activity of neighboring receptors. If this was all, attractant binding would result in reduced X^* and hence sustained low tumbling frequency as long as attractant is around. What causes adaptation?

9.3.2 Adaptation Is Due to Slow Modification of X That Increases Its Activity

The chemotaxis circuit has a second pathway devoted to adaptation. As we saw, binding of attractant reduces the activity of the receptor X . However, each receptor has several biochemical “buttons” that can be pressed to increase its activity and compensate for the effect of the attractant (Figure 9.8). These buttons are **methylation** modifications, in which a methyl group (CH_3) is added to four locations on the receptor. Each receptor can thus have between zero and four methyl modifications. The more methyl modifications, the higher the activity of the receptor.

The methylation buttons work by changing the binding constant K of the receptor to attractants. The more methylated the receptor, the higher is K (Figure 9.9). Each methylation seems to add free energy γ to the free energy of binding, so that K rises exponentially with methylation: $K \sim K_0 e^{\gamma m}$ (Tu, Shimizu and Berg, 2008). The higher K , the less attractant binds, so that there is less inhibition of X activity, X^* . In this way, methylation increases receptor activity.

Methylation of the receptors is catalyzed by a protein called CheR and is removed by a protein called CheB, which we will denote R and B . Methyl groups are continually added and removed by these

two antagonistic proteins, regardless of whether the bacterium senses any ligands (Figure 9.10). This seemingly wasteful cycle has an important function: it allows cells to adapt.

Adaptation is carried out by a negative feedback loop through B . This protein removes methyl groups only from receptors in their active conformation, X^* . Moreover, X^* phosphorylates and thus activates B .

Imagine that attractant is added. Attractant reduces X activity, making B less active. Fewer methyl groups are removed by B . Methyl groups are still added, though, by

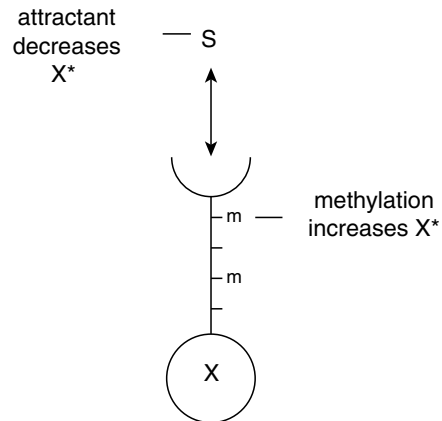


FIGURE 9.8

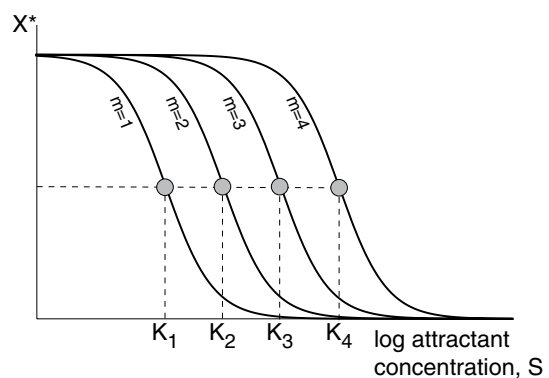


FIGURE 9.9

R at an unchanged rate. Therefore, total methylation increases. Methylation makes the receptor more active, despite the presence of attractant, and tumbling frequency recovers.

Thus, the receptors X first become less active due to attractant binding, and then methylation level gradually increases, restoring X activity. This is a negative feedback loop with a slow arm in which X^* reduces methylation, and a fast arm in which methylation raises X^* (Figure 9.11).

Methylation reactions are indeed much slower – taking seconds to minutes – than the reactions in the main pathway from X to Y_p to the motor that occur on a sub-second timescale. The protein R is present at low amounts in the cell, about 100 copies, and appears to act at saturation (zero-order kinetics). The slow rate of the methylation reactions explains why the recovery phase of the tumbling frequency during adaptation is slower than the initial response.

This feedback circuit is designed so that exact adaptation is achieved. That is, the increased methylation of X *precisely balances* the reduction in activity caused by the attractant. How is this precise balance achieved? Understanding exact adaptation is the goal of the model that we will describe next.

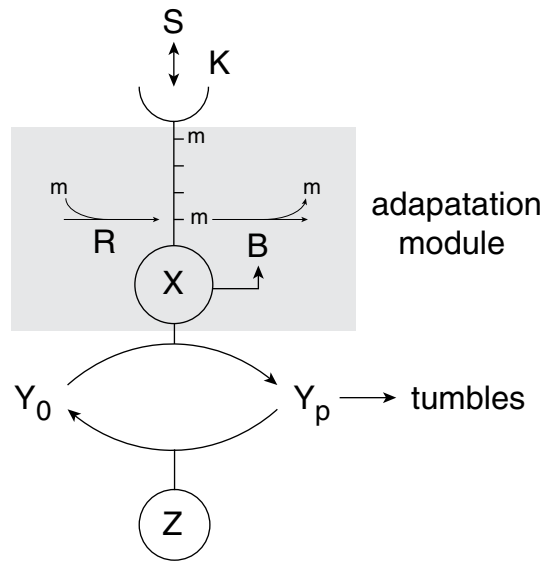


FIGURE 9.10

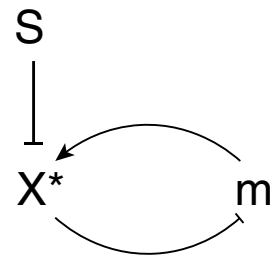


FIGURE 9.11

9.4 THE BARKAI–LEIBLER MODEL OF EXACT ADAPTATION

Early models of chemotaxis used equations to describe the reactions just presented and showed response to attractant and exact adaptation. However, in these models, exact adaptation depended on setting specific values for parameters such as the numbers of R and B proteins per cell. These parameters had to be tuned so that methylation could exactly compensate for the reduction in activity caused by attractant. Changing the protein levels ruined exact adaptation (Figure 9.12). After adding attractant, the cells responded, but then returned to a different basal activity than before the attractant step. We say that exact adaptation in these models is **fine-tuned**. A fine-tuned model is described in solved Exercise 9.4.

A robust mechanism for exact adaptation was proposed by Naama Barkai and Stan Leibler (Barkai and Leibler, 1997). In this mechanism, changing parameters such as R and B protein levels change the steady-state activity. But changing parameters does not

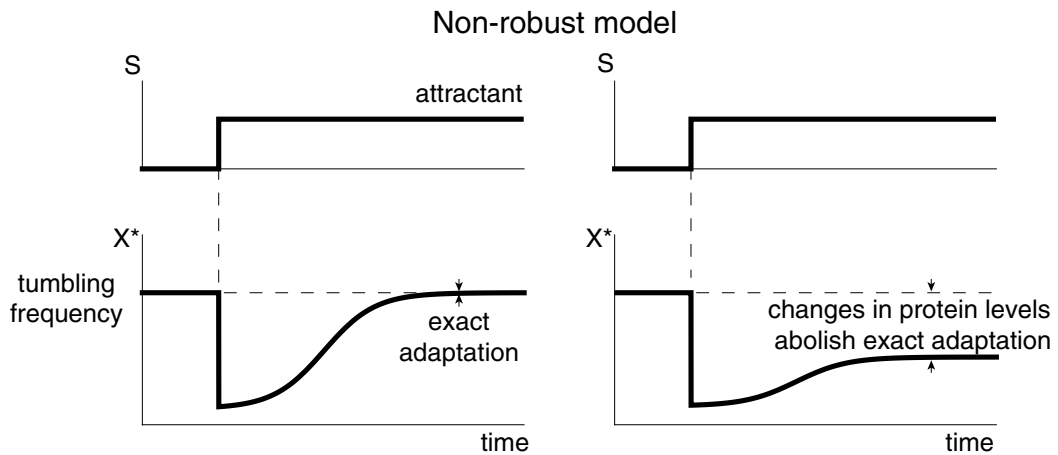


FIGURE 9.12

ruin exact adaptation: after a step of attractant, activity first drops but then returns to the pre-step level (Figure 9.13).

The full model includes several methylation sites and other details, and reproduces many observations on the dynamical behavior of the chemotaxis system (a two-methylation site version is solved in Exercise 9.5). Here, we will analyze a simplified version of the Barkai–Leibler model, aiming to understand how a biochemical circuit can robustly adapt.

The Barkai–Leibler mechanism depends on two molecular features. First, R works at a constant rate independent of its substrate, unmethylated sites on the receptors. This constant rate occurs because R is found at such low numbers and works so slowly that it always has a receptor to which it can add a methyl group (until all methylation sites are modified, a limit in which adaptation breaks down). R thus adds methyl groups at a rate $V_R R$, where R is the number of R proteins.

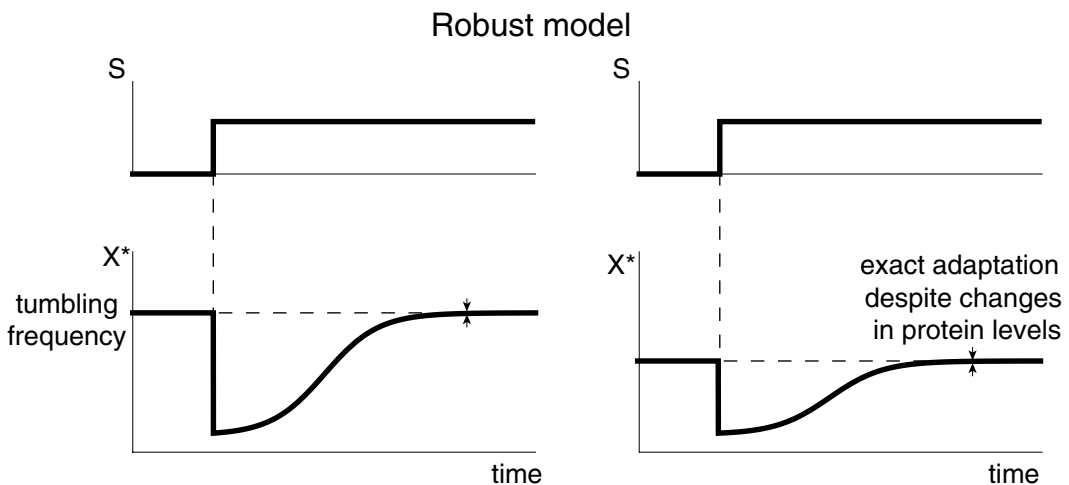


FIGURE 9.13

The second essential feature is that the enzyme B works only on active receptors, X^* . Active receptors have a conformation that exposes the methyl groups and allows B to remove them. Therefore, B removes methyl groups at a rate $V_B B X^*$, where B is the number of B proteins in the cell.

The rate of change of the total number of methyl groups bound to the receptors, m , is given by the difference between the rates of adding methyl by R and removing them by B :

$$\frac{dm}{dt} = V_R R - V_B B X^* \quad (9.4.1)$$

The steady state solution ($dm/dt = 0$) occurs at:

$$X_{st}^* = V_R R / V_B B \quad (9.4.2)$$

Importantly, X_{st}^* *does not depend on attractant concentration*. This means that our system always returns to the same activity level regardless of input signal: we have exact adaptation.

A rate plot of this equation, [Figure 9.14](#), shows that the steady state is stable: if X^* is smaller than X_{st}^* , methylation exceeds demethylation, and as a result X^* rises, stopping when $X^* = X_{st}^*$. Similarly, X^* flows back to steady state if it is too high. The important point is that X_{st}^* does not depend on attractant or repellant levels. Changing parameters like R and B changes the baseline level X_{st}^* . But for a given R and B , activity X^* always returns to its baseline level X_{st}^* .

[Figure 9.13](#) shows the dynamics of this model for two sets of parameters, in which R levels are varied by a factor of 2. It is seen that the steady-state activity changes, but adaptation remains exact, indicating that the model is robust.

Let's review how this mechanism works. Initially the system is at steady state X_{st}^* ([Figure 9.15](#), timepoint [a]). When attractant S is added, it binds the receptors and reduces their activity ([Figure 9.15](#), timepoint [b]). Activity X^* drops below X_{st}^* within 0.1 sec. This causes the abrupt initial drop in tumbling frequency that is observed in the experiments. Adaptation occurs because B only works on the active receptors. The rate of demethylation by B is reduced because of the decrease in active receptors caused by the attractant. R , on the other hand, continues to methylate receptors at a constant rate. Therefore, methylation m gradually increases ([Figure 9.15](#) timepoint [c]). Methylation increases receptor activity. Steady state is reached when the number of active receptors reaches a level that balances the effects of R and B , returning to the steady-state activity level X_{st}^* ([Figure 9.15](#) timepoint [d]). The activity is equal to the pre-attractant activity, despite the presence of attractant. We have exact adaptation.

Exact adaptation occurs for a wide range of variations in any of the parameters of the model, such as V_R , V_B , R and B . In contrast, the value of the steady-state activity to which the cells adapt depends on these parameters. In other words,

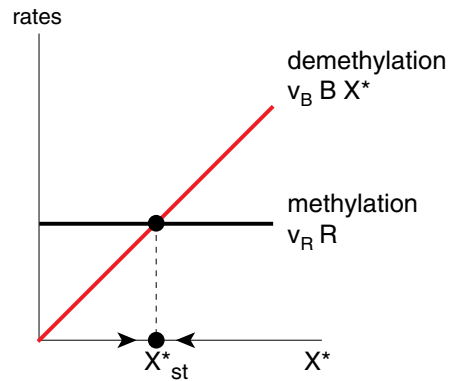


FIGURE 9.14

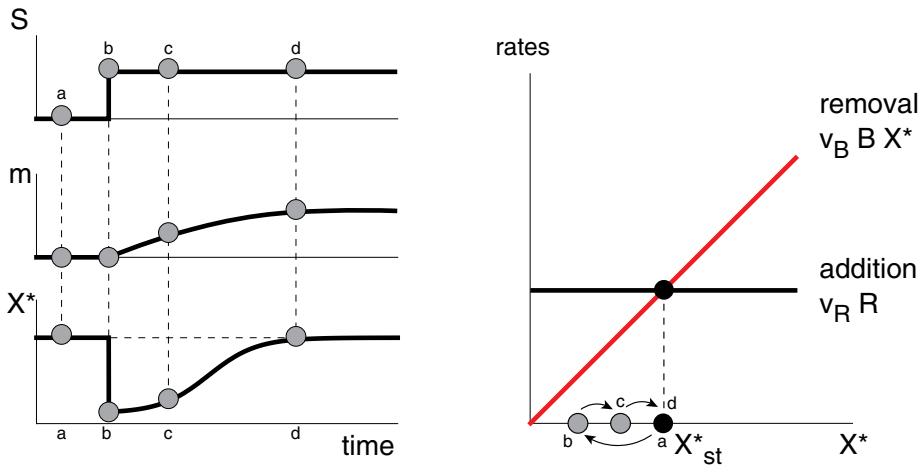


FIGURE 9.15

steady-state activity is a *fine-tuned* feature of this model (Figure 9.13). Exact adaptation, in which the steady state does not depend on ligand levels, is a *robust* feature of the model and does not depend on the precise values of the biochemical parameters.

There are limits to robustness – for example, if receptors become fully methylated, they can no longer compensate for attractants. Indeed, exact adaptation is broken in the case of some attractants such as serine: the serine receptor (Tsr), even when fully methylated, cannot compensate for high concentrations of serine, and there is no exact adaptation at high concentrations of serine.

Robustness of exact adaptation in this model depends on *B* working only on active receptors, and not on receptors that are in their inactive state. This is a specific biochemical detail that is essential for robust adaptation. The assumption that *B* works only on active receptors is not unrealistic, because proteins can be exquisitely specific in discriminating between molecular states. Relaxing this assumption by allowing a small relative rate ε for *B* action on inactive receptors entails a loss of exact adaptation by a factor on the order of ε .

9.4.1 Robust Adaptation and Integral Feedback

At the heart of this mechanism is a feedback loop called **integral feedback** (Yi et al., 2000), which is a central principle in engineering. In integral feedback, there is a slow component (methylation in our case) which integrates an “error” over time, and acts to decrease the error. In chemotaxis, the error is the difference between the activity and the steady-state activity: $error = X_{st}^* - X^*$. The power of integral feedback is that as long as the error is not zero, the integrator keeps accumulating, and the feedback grows until it forces the error to go to zero. There is no choice for the system but to return to X_{st}^* .

The mapping of the Barkai–Leibler model to integral feedback is easiest to see by rewriting the equation for methylation dynamics:

$$\frac{dm}{dt} = V_B B(X_{st}^* - X^*) \quad (9.4.3)$$

Solving this equation by taking an integral over time on both sides shows that methylation integrates over the error:

$$m(t) \sim \int (X_{st}^* - X^*) dt \sim \int \text{error}(t) dt \quad (9.4.4)$$

Because of this integrator effect, the feedback does not stop until the error is zero (until $X^* = X_{st}^*$). Even a small error keeps being integrated over time to lead to a large feedback signal. Linearity is not crucial here: It is enough that methylation changes as a decreasing function of X^* , $dm/dt = g(X^*)$, which crosses zero at X_{st}^* , $g(X_{st}^*) = 0$, to ensure integral feedback and exact adaptation.

Engineers use integral feedback to achieve exact adaptation in many familiar situations. For example, integral feedback ensures that a heater can keep the temperature T of a room at a desired set point T_{st} . In this integral feedback controller, the power to the heater is governed by the integrated error $T - T_{st}$. The power changes slowly and is analogous to methylation (Exercise 9.6).

These equations for ligand binding and methylation capture many experiments on the dynamic response to changing ligands. We will use this model, which is a version of a model presented by Tu, Shimizu and Berg (2008), also in the next chapter. The model becomes simpler when we use the receptor binding constant K as a variable instead of methylation m . To do so, we use the relation $K \sim e^m$ and hence $dK/dt \sim K dm/dt$. We also use as the output the receptor activity normalized by its maximal value $a = X^*/X_{\max}$:

$$\begin{aligned} \frac{dK}{dt} &= cK(a_{st} - a) \\ a &= 1 / (1 + (S/K)^n) \end{aligned} \quad (9.4.5)$$

Parameters that match experimental data are $a_{st} = 0.3$, $c = 1/\text{min}$ and $n = 6$. Figure 9.16 shows how the activity a drops after a step addition of attractant, and shows a pronounced pulse when attractant is removed. Exact adaptation occurs in all cases, as the binding constant K slowly adjusts to the changes in input.

The binding constant K acts like an internal representation of the external signal S . It's as if the receptor adjusts its half-way point K to be sensitive near the new level of attractant. This is like gain-control in a camera, which adjusts its sensitivity to the ambient level of light. Without exact adaptation, the receptors could not be sensitive over many orders of magnitude of

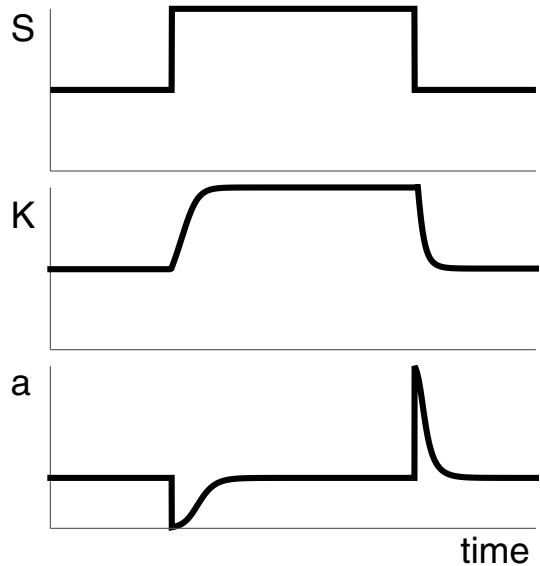


FIGURE 9.16

attractant levels, any more than a camera without gain control could work across orders of magnitude of light.

9.4.2 Experiments Show That Exact Adaptation Is Robust, Whereas Steady-State Activity and Adaptation Times Are Fine-Tuned

An experimental test of robustness employed genetically engineered *E. coli* strains, which allowed controlled changes in the concentration of each of the chemotaxis proteins (Alon et al., 1999). This control was achieved by first deleting the gene for one chemotaxis protein (for example, *R*) from the chromosome, and then introducing into the cell a copy of the gene under control of an inducible promoter (the *lac* promoter). Thus, expression of the protein was controlled by means of an externally added chemical inducer (IPTG). The more inducer added, the higher the *R* concentration in the cells. In this way, *R* levels were varied from 0.5 to 50 times their wild-type levels. The population response of these cells to a saturating step of attractant was monitored using video microscopy on swimming cells. The experiment was carried out with changes in the expression levels of different chemotaxis proteins.

The steady-state tumbling frequency and the adaptation time varied with the levels of the proteins that make up the chemotaxis network (Figure 9.17). For example, steady-state

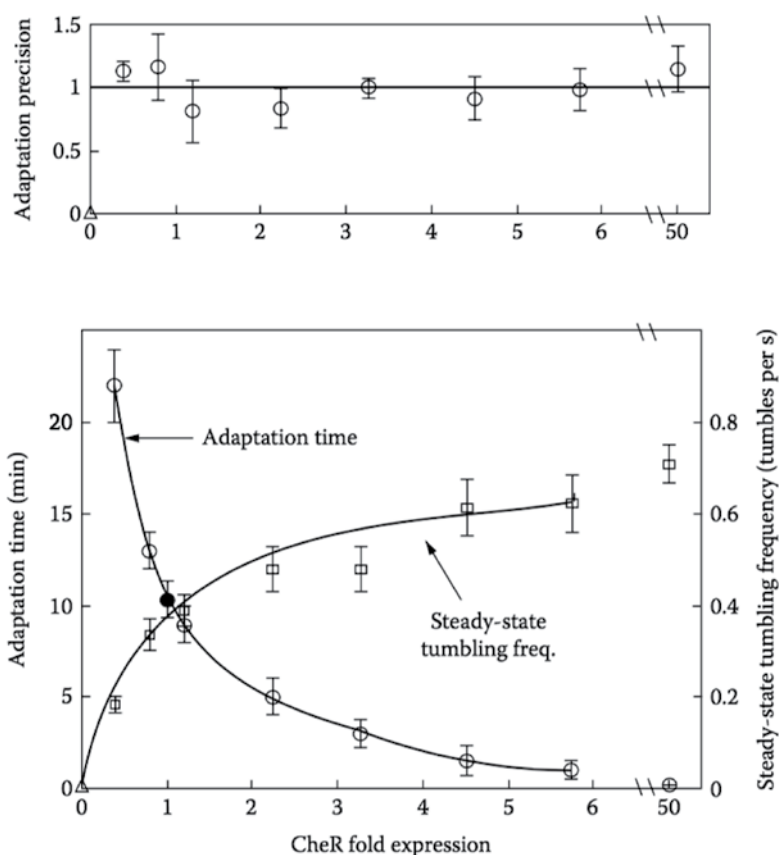


FIGURE 9.17 Adapted from (Alon et al., 1999).

tumbling frequency increased with R levels, whereas adaptation time decreased. Despite these variations, exact adaptation remained robust to within experimental error. These results support the robust model for exact adaptation.

This experiment, which took 3 years, was the way that I transitioned from theoretical physics to experimental biology in my postdoc with Stanislas Leibler at Princeton. I got a lot of help from Mike Surrrette who was a postdoc working on bacterial chemotaxis in the Stock lab next door. My fascination with the robust model was powerful enough to help me make the transformation from theorist to experimentalist.

9.5 INDIVIDUALITY AND ROBUSTNESS IN BACTERIAL CHEMOTAXIS

Spudich and Koshland (1976) observed that genetically identical bacterial cells appear to have an individual character as they perform chemotaxis. Some cells are “nervous” and tumble more frequently than others, whereas other cells are “relaxed” and swim with fewer tumbles than the norm. These individual characteristics of each cell last for tens of minutes. The adaptation time to an attractant stimulus also varies from cell to cell. Interestingly, these two features are correlated: the steady-state tumbling frequency f in a given cell is inversely correlated with its adaptation time, τ , that is, $f \sim 1/\tau$.

The robust model for bacterial chemotaxis can supply an explanation for these chemotactic personalities of *E. coli* cells. The explanation is based on the cell–cell variation in chemotaxis protein levels, and particularly in the least abundant protein in the system, R . Variations in R affect the tumbling frequency f and the adaptation time τ in opposite directions. The Barkai–Leibler model with multiple methylation sites suggests that $f \sim R$ and $\tau \sim 1/R$. Thus, the model predicts that $f \sim 1/\tau$, explaining the observed correlation in these two features (see solved Exercise 9.5).⁵

Despite the cell–cell variability in tumbling frequency, the vast majority of the cells in a population perform chemotaxis and climb gradients of attractants. On the other hand, mutant cells that have wild-type tumbling frequency but cannot adapt precisely (such as certain mutants in both R and B) are severely defective in chemotaxis ability. Evidently, tumbling frequency need not be precisely tuned for successful chemotaxis, whereas exact adaptation is important for most ligands.

In fact, there is an advantage to having a range of tumbling frequencies in a population of bacteria. This is because bacteria cannot know in advance which type of medium they will be moving through (Celani and Vergassola, 2010). In a free liquid, it is optimal to have long runs to sample space broadly (Figure 9.18). But in a liquid dense with obstacles, as occurs in the crowded environments of the soil or

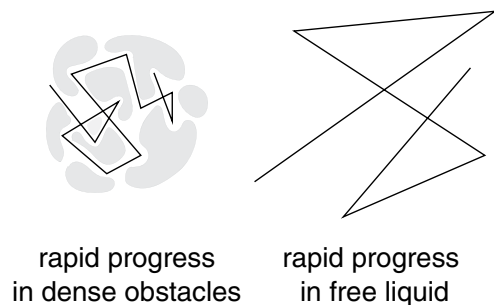


FIGURE 9.18

⁵ Detailed stochastic simulations of this protein circuit were pioneered by D. Bray and colleagues (Shimizu, Aksenov and Bray, 2003).

the intestine, it is optimal to have shorter runs to avoid being stuck against an obstacle for long times. Thus, variation in protein levels can generate a **bet hedging** strategy in which different individuals are suited for different possible future environments. Not all of the eggs are in one basket. Thanks to robustness, no matter what the steady-state tumbling frequency is, every individual will have exact adaptation, and hence be able to work across many orders of magnitude of signal.

In summary, the bacterial chemotaxis circuit has a design such that a key feature – exact adaptation – is robust with respect to variations in protein levels. Other features, such as steady-state activity and adaptation times, are fine-tuned. These latter features show variations within a population due to intrinsic cell–cell variations in protein levels. Because of the robust design, the intrinsic variability in the cell’s protein levels does not abolish exact adaptation.

As a theorist, one can usually write many different models to describe a given biological system, especially if some of the biochemical interactions are not fully characterized. Of these models, only very few will typically be robust with respect to variations in the components. Thus, the robustness principle can help narrow down the range of models that work on paper to the few that can work in the cell. Robust design is an important factor in determining the specific types of circuits that appear in cells.

FURTHER READING

(Alon et al., 1999) “Robustness in bacterial chemotaxis.”

(Barkai and Leibler, 1997) “Robustness in simple biochemical networks.”

(Berg, 2004) “*E. coli* in motion.”

(Berg and Brown, 1972) “Chemotaxis in *Escherichia coli* analyzed by three-dimensional tracking.”

(Berg and Purcell, 1977) “Physics of chemoreception.”

(Celani and Vergassola, 2010) “Bacterial strategies for chemotaxis response.”

(Knox et al., 1986) “A molecular mechanism for sensory adaptation based on ligand-induced receptor modification.”

(Spudich and Koshland, 1976) “Non-genetic individuality: chance in the single cell.”

(Tu, Shimizu and Berg, 2008) “Modeling the chemotactic response of *Escherichia coli* to time-varying stimuli.”

(Yi et al., 2000) “Robust perfect adaptation in bacterial chemotaxis through integral feedback control.”

EXERCISES

- 9.1 *Repellent*: Repellent binding increases receptor activity. Explain how the chemotaxis circuit responds to a step of repellent.
- 9.2 *Reduction in attractant*: Suppose attractant is removed in a step-like manner. What is the response of the cells?
- 9.3 *Sensory adaptation*: Explain how the idea of exact adaptation applies to human senses: vision, hearing and smell. Are there senses which do not show exact adaptation? What might be the reason that some senses do and others do not show exact adaptation?
- 9.4 *Fine-tuned model for exact adaptation*: In this exercise, we will solve a simplified version of a theoretical model of chemotaxis first proposed by Albert Goldbeter,

Lee Segel and colleagues (Knox et al., 1986). This study formed an important basis for later theoretical work on the chemotaxis system. We will see that exact adaptation in this model is fine-tuned.

In the model (Figure 9.19), the receptor complex X can become methylated X_m under the action of R , and demethylated by B . For

simplicity, we ignore the precise number of methyl groups per receptor and group together all methylated receptors into one variable X_m . Only the methylated receptors are active, with activity a_0 per methylated receptor; unmethylated receptors are inactive.

To describe the dynamics of receptor methylation, we need to model the actions of the methylating enzyme R and the demethylating enzyme B . R works at saturation, (i.e., at a rate that is independent of the concentration of its substrate), with rate V_R . In contrast, B works with Michaelis–Menten kinetics (Appendix A.7). Hence, the rate of change of X_m is the difference of the methylation and demethylation rates:

$$\frac{dX_m}{dt} = V_R R - V_B B X_m / (K + X_m) \quad (\text{P9.1})$$

The parameters R and B denote the concentrations of R and B . At steady state, $dX_m/dt = 0$, the dynamics reach a steady-state level of methylated receptor:

$$X_m = K V_R R / (V_B B - V_R R) \quad (\text{P9.2})$$

Recall that the unmethylated receptor has zero activity, whereas X_m has activity a_0 per receptor, resulting in a total steady-state activity of:

$$A_0 = a_0 X_m \quad \text{steady-state activity with no attractant} \quad (\text{P9.3})$$

The activity of the receptors, A_0 , governs the rate at which Y is phosphorylated to create Y_P which generates tumbles. The activity A_0 , therefore, determines the steady-state tumbling frequency, $f = f(A_0)$.

Now saturating attractant is added to the cells, so that all of the receptors bind attractant ligand. The attractant causes receptors to assume their inactive conformation. As a result, the activity per methylated receptor drops to $a_1 \ll a_0$. Therefore, the total activity drops to low values right after attractant is added:

$$A_1 = a_1 X_m \quad (\text{P9.4})$$

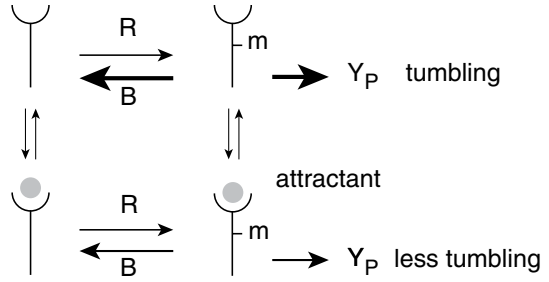


FIGURE 9.19

Gradually, however, the methylation feedback loop kicks in. In this loop, because the receptors bind attractant ligand, the rate of B action is decreased, from V_B to V'_B : demethylation rate is reduced. Thus, this model assumes that the rate of B is a function of attractant concentration. As a result, receptor methylation X_m begins to increase due to continual methylation by R . Receptor methylation at steady state reaches a balance between methylation and demethylation, just as in Equation P9.2, but with the demethylation rate set to its new value, V'_B :

$$X'_m = KV_R R / (V'_B B - V_R R) \quad (\text{P9.5})$$

resulting in a new steady-state activity:

$$A_2 = a_1 X'_m \quad \text{steady-state activity with attractant} \quad (\text{P9.6})$$

Exact adaptation means that the steady-state activity before attractant addition is equal to the steady-state activity in the presence of ligand:

$$A_0 = A_2 \quad \text{exact adaptation} \quad (\text{P9.7})$$

To attain exact adaptation, the increase in receptor methylation must *precisely* balance the decrease in receptor activity caused by the ligand. This results in a relation that must be fulfilled by the parameters of the system, based on equating Equations P9.2 and P9.5:

$$a_0 KV_R R / (V_B B - V_R R) = a_1 KV_R R / (V'_B B - V_R R) \quad (\text{P9.8})$$

Let's play with numbers to get a feel for how exact adaptation works in this model. Suppose that ligand binding causes a 10-fold reduction in receptor activity: activity per receptor before ligand binding is $a_0 = 10$ and after ligand binding is $a_1 = 1$. Let's use $K = 1$, $V_R R = 1$, $V_B B = 2$ (units are not important for the present discussion). These values lead to an activity in the absence of attractant of:

$$A_0 = \frac{a_0 KV_R R}{V_B B - V_R R} = 10 / (2 - 1) = 10 \quad (\text{P9.9})$$

After attractant addition, activity per receptor drops 10-fold to $a_1 = 1$. In order to reach exact adaptation, Equation P9.8 constrains $V'_B B$ to drop to a specific value, namely, $V'_B B = 1.1$, so that the activity adapts to the pre-stimulus level:

$$A_2 = a_1 KV_R R / (V'_B B - V_R R) = 1 / (1.1 - 1) = 10 \quad (\text{P9.10})$$

Exact adaptation in this model depends on a strict relation between the biochemical parameters. What happens if the parameters change? For example, suppose the

concentration of protein R is reduced by a factor of 20%, so that $V_R R$ goes from 1 to 0.8. In this case:

$$A_0 = 10 \cdot 0.8 / (2 - 0.8) = 6.66 \quad (\text{P9.11})$$

and

$$A_2 = 1 \cdot 0.8 / (1.1 - 0.8) = 2.33 \quad (\text{P9.12})$$

We see that exact adaptation is lost, since A_2 is no longer equal to A_0 . In this example, a modest 20% change in the level of a protein (R) caused almost a threefold difference in the steady state activities with and without ligand. Exact adaptation is a **fine-tuned** property in this model.

- 9.5 *Robust model with two methylation sites:* In this exercise we solve the Barkai–Leibler model with two methylation sites. The receptor X can be methylated on two positions, and can thus have zero, one or two methyl groups, denoted X_0 , X_1 and X_2 . The enzyme R works at saturation (zero-order kinetics) to methylate X_0 and X_1 . The demethylating enzyme B works only on the active receptor conformation, removing methyl groups with equal rates from X_1^* and X_2^* . For simplicity, assume that B works with first-order kinetics. The reactions are:

methylation of X_0 $X_0 \rightarrow X_1$ at rate $R V_R X_0 / (X_1 + X_0)$,

the last factor occurs because R is distributed between its substrates X_0 and X_1 :

methylation of X_1 $X_1 \rightarrow X_2$ at rate $R V_R X_1 / (X_1 + X_0)$

demethylation of X_1^* $X_1^* \rightarrow X_0$ at rate $B V_B X_1^*$

demethylation of X_2^* $X_2^* \rightarrow X_1$ at rate $B V_B X_2^*$

- What is the steady-state activity $A = X_1^* + X_2^*$? Does it depend on the concentration of ligand S ? Is there exact adaptation?
- Estimate the adaptation time, the time needed for 50% adaptation after addition of saturating attractant. Note that to adapt to saturating attractant, virtually all of the receptors need to be doubly methylated.
- Explain the finding of Spudich and Koshland (1976) that $A \sim 1/\tau$ using the model, based on cell–cell variations in the concentration of R (Barkai and Leibler, 1997).

Solution:

- The rates of change of the doubly methylated receptor concentration and the non-methylated receptor concentration are:

$$d(X_2 + X_2^*)/dt = R V_R X_1 / (X_1 + X_0) - B V_B X_2^* \quad (\text{P9.13})$$

$$dX_0/dt = -R V_R X_0/(X_0 + X_1) + B V_B X_1^* \quad (\text{P9.14})$$

Subtracting these two equations yields:

$$d(X_2 + X_2^*)/dt - dX_0/dt = R V_R - B V_B (X_1^* + X_2^*) = R V_R - B V_B A \quad (\text{P9.15})$$

The steady-state activity $A = X_1^* + X_2^*$ is, therefore, (setting d/dt terms to zero):

$$A_{st} = R V_R / B V_B \quad (\text{P9.16})$$

This activity does not depend on the ligand concentration. Therefore, the mechanism displays exact adaptation.

- b. In the case of saturating ligand, receptors in all of their forms bind attractant ligand. The attractant reduces the activity of all methylated receptors, and thus at initial times X_1^* is small. In addition, when adaptation is completed, X_1^* is small because the majority of receptors need to be doubly methylated in order to balance the strong inhibitory effect of the saturating attractant. Thus, X_1^* is relatively small throughout most of the dynamics. Since X_1^* is small, the demethylation flux from X_1^* to X_0 is small. Hence, to a good approximation, X_0 dynamics reflect only a reduction due to the action of R , because the term with B in Equation P9.14 is negligible:

$$dX_0/dt \approx -R V_R X_0/(X_0 + X_1) \quad (\text{P9.17})$$

so that X_0 drops with time. At initial times (before attractant addition), let us denote by q the fraction of X_0 among the possible substrates of R , $q = X_0/(X_0 + X_1)$. Thus, the initial slope of the drop in X_0 is $-q R V_R$. The adaptation time to saturating ligand (time to recover to 50% activity) is the time needed to build enough methylated receptors to restore activity, at the expense of most of the unmethylated ones. Thus, it is approximately the time for X_0 to decline to 50% of its initial value. This adaptation time is equal to the number of methylation reactions needed (i.e., methylations equal to 50% of X_0) divided by the rate at which they occur, namely (ignoring the changes in q over this time):

$$\tau \sim 0.5 X_0 / q R V_R \quad (\text{P9.18})$$

Thus, the adaptation time becomes shorter the more R enzymes exist in the cell. This makes sense because the more R enzymes there are, the faster methylation occurs and the faster the adaptation.

Note that the simplified model discussed in the text has a different adaptation time, governed by B and not R . The adaptation time is generally governed by R in

realistic models with more than one methylation site (Barkai and Leibler, 1997). In experiments, the adaptation time is found to decrease with R (Figure 9.17), in agreement with the multi-site models.

- c. We saw above that the adaptation time varies as $\tau \sim 1/R$ (Equation P9.18) and the steady-state activity varies as $A_{st} \sim R$ (Equation P9.16). Thus, if R is the protein with the largest variation between genetically identical cells, one would expect that $A_{st} \sim 1/\tau$, as observed. The protein R is the least abundant chemotaxis signaling protein in *E. coli*, with on the order of 100 copies per cell, whereas there are on the order of several thousand copies of B , Y , Z and X per cell. R , therefore, is the most prone to large relative stochastic variations.
- 9.6 *Integral feedback*: A heater heats a room. The room temperature T increases at a rate proportional to the power of the heater, P , to other sources of heat, S , and decreases due to thermal diffusion to the outside:

$$dT/dt = aP + S - bT \quad (\text{P9.19})$$

An integral feedback controller (a thermostat) is placed in order to keep the room temperature at a desired point T_0 . In this feedback loop, the power to the heater is proportional to the integral over time of the error in temperature, $T - T_0$:

$$P = P_0 - k \int (T - T_0) dt \quad (\text{P9.20})$$

This feedback loop thus reduces the power to the heater if the room temperature is too high, $T > T_0$, and increases the power when the room temperature is too low. Taking the time derivative of the power, we find:

$$dP/dt = -k (T - T_0) \quad (\text{P9.21})$$

- a. Show that the steady-state temperature is T_0 and that this steady state does not depend on any of the system parameters, including the room's thermal coupling to the heater, a , the additional heat sources, S , the room's thermal coupling with the outside, b , or the time constant of the integrator, k . In other words, integral feedback shows robust exact adaptation for the room temperature.
- b. Demonstrate that integral feedback is the *only* solution that shows robust exact adaptation of the room temperature, out of all possible linear control systems. That is, assume a general linear form for the controller:

$$dP/dt = c_1 T + c_2 P + c_3 \quad (\text{P9.22})$$

and show that integral feedback as a structural feature of the system is necessary and sufficient for robust exact adaptation.

- 9.7 *Zero-order ultrasensitivity* (Goldbeter and Koshland, 1981): In this exercise, we will see how two antagonistic enzymes can generate a sharp switch, provided that they work near their zero-order range. A protein X can be in a modified X_1 or unmodified X_0 state. Modification is carried out by enzyme E_1 , and demodification by enzyme E_2 . The rate V_2 of E_2 is constant, whereas the rate V_1 of E_1 is governed by an external signal. Consider V_1 to be the input and X_1 to be the output of this system.
- Assume that E_1 and E_2 work with first-order kinetics. What is the output X_1 as a function of input V_1 .
 - What is the sensitivity of this circuit, defined as the relative change in X_1 per relative change in V_1 , $S(X_1, V_1) = (V_1/X_1)dX_1/dV_1$.
 - Assume now that E_1 and E_2 work with zero-order kinetics. What is X_1 as a function of V_1 ? Note that $X_0 + X_1$ sum up to the total concentration X_{tot} .
 - What is the sensitivity of the zero-order circuit? Explain why this is called “zero-order ultra-sensitivity.”
 - Compare the switching time (time to 50% change in X_1 upon a change in V_1) between the cases of (a) and (c) above.
- 9.8 *Same operon*: CheY, CheZ, CheR and CheB are all on one operon, and are thus all transcribed from one long mRNA.
- Explain why this reduces the relative fluctuations in their protein numbers as opposed to having each transcribed from its own mRNA.
 - How might this reduced noise affect the robustness of exact adaptation, and of the steady-state activity? (Kollmann et al., 2005).
- 9.9 *Linear integral feedback cannot filter out ramps of input*: Suppose that the input signal rises linearly with time, $s(t) = rt$. Show that a linear integral feedback circuit cannot adapt precisely:
- $$\text{Use } \frac{dx}{dt} = k(y_0 - y) \quad \text{and} \quad \frac{dy}{dt} = s(t) - x - y.$$
- Solve the equations.
 - What is the steady-state level of the output y as a function of the slope of the ramp, r ?
 - Do you think there is a circuit that can filter out such input ramps (show exact adaptation to ramps)?

BIBLIOGRAPHY

- Alon, U. et al. 1999. ‘Robustness in bacterial chemotaxis’, *Nature*, 397(6715), pp. 168–171. doi: 10.1038/16483.
- Asakura, S. and Honda, H. 1984. ‘Two-state model for bacterial chemoreceptor proteins. The role of multiple methylation’, *Journal of Molecular Biology*, 176(3), pp. 349–367. doi: 10.1016/0022-2836(84)90494-7.

- Barkai, N. and Leibler, S. 1997. 'Robustness in simple biochemical networks', *Nature.*, 387, pp. 913–917. doi: 10.1038/43199.
- Berg, H.C. 2004. *E. coli in motion*. Springer.
- Berg, H.C. and Brown, D.A. 1972. 'Chemotaxis in *Escherichia coli* analysed by three-dimensional tracking', *Nature*, 239, p. 500. doi: 10.1038/239500a0.
- Berg, H.C. and Purcell, E.M. 1977. 'Physics of chemoreception', *Biophysical Journal*, 20(2), pp. 193–219. doi: 10.1016/S0006-3495(77)85544-6.
- Bray, D. 2002. 'Bacterial chemotaxis and the question of gain', *Proceedings of the National Academy of Sciences*, 99(1), pp. 7–9. doi: 10.1073/pnas.022641699.
- Celani, A. and Vergassola, M. 2010. 'Bacterial strategies for chemotaxis response', *Proceedings of the National Academy of Sciences*, 107(4), pp. 1391–1396. doi: 10.1073/pnas.0909673107.
- Cluzel, P., Surette, M. and Leibler, S. 2000. 'An ultrasensitive bacterial motor revealed by monitoring signaling proteins in single cells', *Science*, 287(5458), pp. 1652–1655. doi: 10.1126/science.287.5458.1652.
- Goldbeter, A. and Koshland, D.E. Jr. 1981. An amplified sensitivity arising from covalent modification in biological systems. *Proc Natl Acad Sci USA*. 78(11), pp. 6840–6844.
- Ishihara, A. et al. 1983. 'Coordination of flagella on filamentous cells of *Escherichia coli*', *Journal of Bacteriology*. doi: 10.1071/BT9700099.
- Jasuja, R. et al. 1999. 'Chemotactic responses of *Escherichia coli* to small jumps of photoreleased L-aspartate', *Biophysical Journal*, 76(3), pp. 1706–1719. doi: 10.1016/S0006-3495(99)77329-7.
- Keymer, J.E. et al. 2006. 'Chemosensing in *Escherichia Coli*: Two Regimes of Two-State Receptors', in *Proceedings of the National Academy of Sciences*, 103(6), pp. 1786–1791. doi: 10.1073/pnas.0507438103.
- Knox, B.E. et al. 1986. 'A molecular mechanism for sensory adaptation based on ligand-induced receptor modification', in *Proceedings of the National Academy of Sciences of the United States of America*. doi: 10.1073/pnas.83.8.2345.
- Kollmann, M., Løvdok, L., Bartholomé, K., Timmer, J. and Sourjik, V. 2005. 'Design principles of a bacterial signalling network'. *Nature*. 438(7067), pp. 504–507.
- Korobkova, E. et al. 2004. 'From molecular noise to behavioural variability in a single bacterium', *Nature*, 428(6982), pp. 574–578. doi: 10.1038/nature02404.
- Mello, B.A., Shaw, L. and Tu, Y. 2004. 'Effects of receptor interaction in bacterial chemotaxis', *Biophysical Journal*, 87(3), pp. 1578–1595. doi: 10.1529/biophysj.104.042739.
- Segall, J.E., Block, S.M. and Berg, H.C. 1986. 'Temporal comparisons in bacterial chemotaxis', in *Proceedings of the National Academy of Sciences*, 83(23), pp. 8987–8991. doi: 10.1073/pnas.83.23.8987.
- Shimizu, T.S., Aksenov, S.V. and Bray, D. 2003. 'A spatially extended stochastic model of the bacterial chemotaxis signalling pathway', *Journal of Molecular Biology*. doi: 10.1016/S0022-2836(03)00437-6.
- Sourjik, V. and Berg, H.C. 2004. 'Functional interactions between receptors in bacterial chemotaxis', *Nature*, 428, p. 437. doi: 10.1038/nature02406.
- Spudich, J.L. and Koshland, D.E. 1976. 'Non-genetic individuality: chance in the single cell', *Nature*, 262, p. 467. doi: 10.1038/262467a0.
- Tu, Y., Shimizu, T.S. and Berg, H.C. 2008. 'Modeling the chemotactic response of *Escherichia coli* to time-varying stimuli', *Proceedings of the National Academy of Sciences*, 105(39), pp. 14855–14860. doi: 10.1073/pnas.0807569105.
- Yi, T.-M. et al. 2000. 'Robust perfect adaptation in bacterial chemotaxis through integral feedback control', in *Proceedings of the National Academy of Sciences*. doi: 10.1073/pnas.97.9.4649.



Taylor & Francis

Taylor & Francis Group

<http://taylorandfrancis.com>

Fold-Change Detection

In this chapter, we continue to explore the sensory systems of cells, building on bacterial chemotaxis. Most types of biological circuits we have looked at so far are sensitive to absolute signal levels. Here, we will study the remarkable ability of certain biological circuits to respond to relative changes in signal, instead of absolute changes.

10.1 UNIVERSAL FEATURES OF SENSORY SYSTEMS

Sensory systems have certain universal features that make them good measurement devices. One such feature is exact adaptation, found in the bacterial chemotaxis system of the previous chapter, and in animal vision, olfaction and hearing. Exact adaptation is the ability to perfectly adjust to the background signal. When we go from sunlight into a dark room lit by a candle, at first we don't see very well but after a while our pupils dilate to let in more light and our eyes adjust.

A second universal feature is the sensing of **relative changes** rather than absolute changes. Suppose that we adapt to a room lit by a candle, and then we add a second candle. We sense a large change in light. But if we add the same candle to a room lit by a chandelier with 50 candles, we barely notice the change. The absolute number of photons added is the same, one candle's worth, but the relative change is very different.

Response to relative changes was described in human senses by Weber in the nineteenth century. For example, Weber studied the sense of weight perception. He let people hold a weight x_0 for a while, and then slowly added small weights to measure the minimal detectable increase $\Delta x_{\min}$, at which people first felt the extra weight. The minimal detectable increase was proportional to the initial weight, $\Delta x_{\min} = kx_0$, where k is Weber's constant. This is called **Weber's law**: the just-noticeable difference is proportional to the background signal. If $k = 0.1$, for example, you can detect 10 g against a background of 100 g, but you only detect 100 g against a background of 1 kilo. The same law applies to detecting light or sound, with a different constant k for each sense.

In all cases, sensing of relative changes is found for an intermediate range of several decades of input signal (typically 2–5 decades). Relative sensing is lost at very weak signals on the brink of detection or very strong signals that saturate the receptors.

Even psychological senses seem to work in terms of relative changes. For example, psychologists measure subjective well-being using questionnaires. Each person has an individual steady-state level of well-being. Positive events such as getting a raise in salary increase well-being for a while, but then well-being adapts back to baseline. The immediate change in well-being seems to depend on relative changes. If you have been earning \$10 a week, a \$10 raise is cause for celebration. If you have been earning \$1000 a week, the same \$10 raise will go almost unnoticed.

Many sensory systems of cells also show these universal features – exact adaptation and sensing of relative changes. For cells, as well as for animals, sensing relative changes is important in order to be robust to noise in the input. To respond correctly, the cell must tell the difference between a true input signal and noise.

Suppose that a cell senses a signal molecule by means of receptors. It suddenly experiences an increase of 10 binding events per second of the signal molecule to the receptors. Is this a true signal or just noise?

The answer depends on the background level of the input signal – or in other words, on the recent context of the signal. If the cell has been sitting for a while in a background of 1 binding event per second, a rise of 10/sec is an eyebrow-raising 10-fold increase, and is likely to be important. If instead the cell has been soaking in 1000 binding events/sec, the same increase of 10/sec is tiny (even smaller than the typical noise of $\sqrt{1000} \sim 30/\text{sec}$), and should be rejected as a fluctuation. Such decisions had best be based on *relative* changes, not absolute changes.

In this chapter, we will ask how relative changes are sensed. We'll begin with bacterial chemotaxis, and go on to ask which kinds of circuits in general can sense relative changes.

10.2 FOLD-CHANGE DETECTION IN BACTERIAL CHEMOTAXIS

The chemotaxis system of *E. coli* can sense relative changes across several orders of magnitude of background signal. This wide dynamic range was discovered by Mesibov, Ordal and Adler (1973). They placed swimming *E. coli* in a dish and let them adapt to a background level S_0 of the attractant alpha-methyl aspartate. Then they placed a pipette with attractant concentration that is 3.2 times higher than S_0 into the dish (Figure 10.1). They repeated this experiment with different levels of S_0 . The number of bacteria that swam into the pipette in an hour was roughly the same across several orders of magnitude of attractant levels (Figure 10.2). It made no difference whether concentrations were in micromolar or millimolar, bacteria could still detect the 3.2-fold higher concentration of attractant in the pipette.

A direct test for relative sensing in bacterial chemotaxis was presented by Milena Lazova, Tom Shimizu and colleagues (Lazova et al., 2011). They used a microfluidic device to present *E. coli* cells with temporal changes in the concentration of the attractant alpha-methyl aspartate, $S(t)$. The signals were all

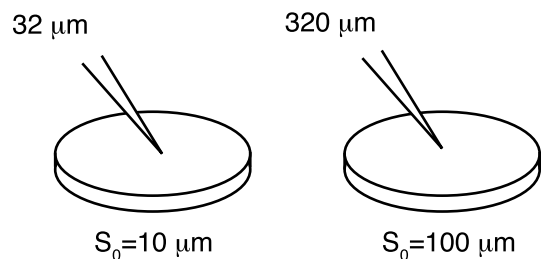


FIGURE 10.1

based on the same pattern: $S(t)$ started at a background level S_0 and then wiggled up and down. Then they multiplied the same signal, including its background, by a factor λ . In this way, they presented the bacteria with a series of signals that had a scale λ that ranged across several orders of magnitude. This experimental design provided input signals with the same fold change over the background, $F(t) = S(t)/S_0$, but very different absolute changes (Figure 10.3).

Lazova and Shimizu measured the chemotaxis output, which we denote $a(t)$, using a fluorescence system developed by Sourjik and Berg, in which the interaction of CheY and flagellar motor is accurately visualized using fluorescence energy transfer (FRET). They found that the output was invariant to the multiplicative constant λ across three decades of background concentration, from about 20 μM to 3 mM of attractant. Bacterial chemotaxis senses relative changes.

10.2.1 Definition of Fold-Change Detection (FCD)

Let's define what we mean by sensing relative changes more precisely. Weber's law states that the just-noticeable-difference in input signal is proportional to the background. Here, we will generalize Weber's law and apply it to the entire dynamic response curve of the system. We want the entire shape of the output curve to depend only on the signal normalized by its background, which is called the **fold change** of the signal. Consider a system with output $a(t)$ that is adapted to a background signal S_0 . Now let the signal change with time, $S(t)$. We define **fold-change detection (FCD)** as a response curve $a(t)$ whose entire shape, including peak amplitude and response time, depends only on the relative change in input $S(t)/S_0$, and not on the absolute change.

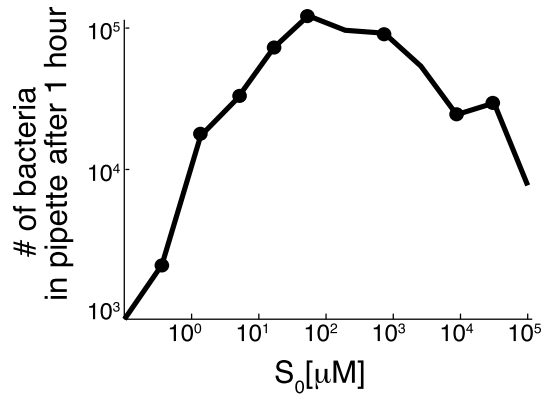


FIGURE 10.2 Adapted from (Mesibov, Ordal and Adler 1973).

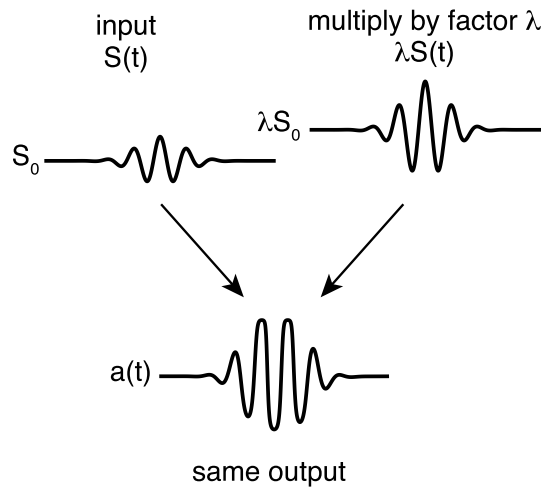


FIGURE 10.3

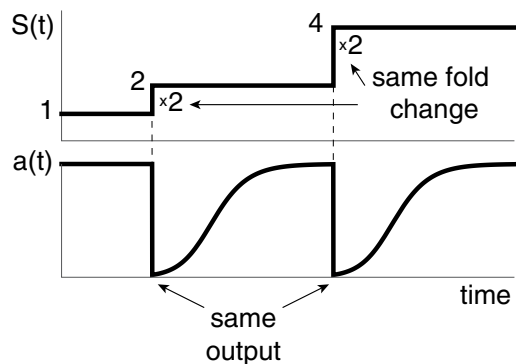


FIGURE 10.4

For example, an input step from level $S = 1$ to level $S = 2$ yields exactly the same response curve as a step from 2 to 4 (Figure 10.4). This is because both steps have the same twofold change, even though the second step is larger in absolute terms. A step from 1 to 3 will yield a larger response, because the fold change is larger. A step from 3 to 9 will have a response identical to that of the step from 1 to 3.

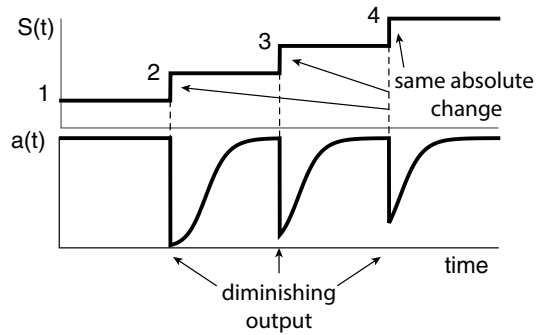


FIGURE 10.5

If we present a system with FCD with a series of steps with the same absolute levels, say from 1 to 2 to 3 to 4, the response will diminish because the fold change gets smaller and smaller (Figure 10.5).

10.2.2 The Chemotaxis Circuit Provides FCD by Means of a Nonlinear Integral-Feedback Loop

Let's understand how bacterial chemotaxis achieves FCD. The intuitive mechanism is that output activity $a(t)$ is a function of attractant signal $S(t)$ divided by its binding constant to the receptors $K(t)$, $a = f(S/K)$. The binding constant K rises proportionally to the background signal thanks to the adaptation system. Thus, if input S is multiplied by a factor λ , so is K , and hence $f(S/K)$ remains unchanged. K is a slowly changing memory that normalizes out the background signal. Let's solve the chemotaxis model to see the origin of FCD.

Solved Example 10.1: The Robust Model for Bacterial Chemotaxis Shows FCD

The equations for the model of bacteria chemotaxis (Equations 9.4.5) show that activity $a(t)$ is a Hill function of the attractant $S(t)$

$$a = \frac{1}{1 + \left(\frac{S}{K}\right)^n} = f\left(\frac{S}{K}\right) \quad (10.2.1)$$

The binding constant $K(t)$ is determined by the methylation reactions, which in turn depend on activity a , providing integral feedback to maintain the steady-state activity a_{st} :

$$\frac{dK}{dt} = cK(a_{st} - a) \quad (10.2.2)$$

Suppose the system adapts to a constant background attractant S_0 . To reach a_{st} , Equation 10.2.2 adjusts K to match the background S_0 so that $f(S_0/K_{st}) = a_{st}$ (Figure 10.6). The solution is

$$K_{st} = S_0 / f^{-1}(a_{st}) \quad (10.2.3)$$

If we multiply S_0 by λ , and let the system adapt, it goes back to the same output a_{st} thanks to exact adaptation. Thus, K must also rise by the same factor

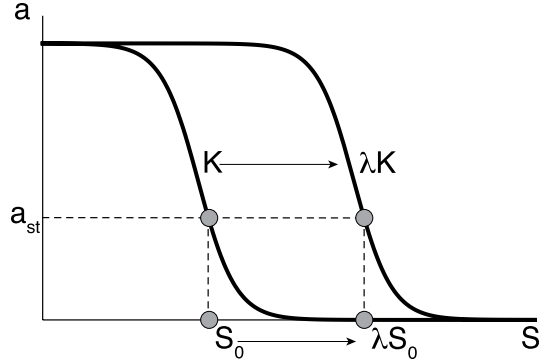


FIGURE 10.6

of λ (Figure 10.6), because this is the only way that the ratio S/K stays at the proper value $f(\lambda S_0 / \lambda K) = f(S_0 / K) = a_{st}$. In other words, at steady state K is proportional to the attractant background.

Now let the input $S(t)$ change with time. To establish FCD, we need to show that if we multiply $S(t)$ by any positive constant λ , we still get the same output dynamics $a(t)$. To test this, we use an important technique that rescales S and K to **dimensionless variables**. Our strategy will be to use these dimensionless variables to obtain dynamic equations that do not depend on input $S(t)$ directly, but instead depend only on the fold change in input. We will also find that the initial conditions do not depend on $S(t)$. Since both the differential equations and their initial conditions depend only on fold change, so does the entire dynamics of the output $a(t)$.

Suppose we begin at steady state with $S = S_0$. We define the **fold change** as $F(t) = S(t)/S_0$, and the scaled binding constant as $\tilde{K}(t) = K(t)/S_0$. First note that at steady state, these variables do not depend on attractant, because $F = 1$ and $\tilde{K} = 1/f^{-1}(a_{st})$, from Equation 10.2.3. The scaled equations can be found by plugging in these new variables into Equations 10.2.1 and 10.2.2:

$$\frac{d\tilde{K}}{dt} = c\tilde{K}(a_{st} - a)$$

$$a = f\left(\frac{F(t)}{\tilde{K}}\right)$$

Note that $S(t)$ does not appear in these equations, only the fold change $F(t)$. In order to prove that the entire dynamics is independent on S , we only need to show that the initial conditions, $\tilde{K}(t=0)$ and $F(t=0)$, also do not depend on S . As mentioned above, at steady state these scaled variables have S -independent values, $F(0) = 1$ and $\tilde{K} = 1/f^{-1}(a_{st})$. Since the differential equations and their initial conditions do not depend on S_0 , the entire dynamics of the variables, including the output $a(t)$, does not depend on S_0 . You can multiply $S(t)$ by any number λ , and because this keeps

fold $F(t)$ unchanged, the output $a(t)$ won't be affected by λ . We conclude that the output $a(t)$ is determined only by the fold change in input, hence FCD.

10.3 FCD AND EXACT ADAPTATION

Bacterial chemotaxis thus has FCD. It also has exact adaptation as we saw in the previous chapter, in which the output returns to a baseline value that is independent of any constant background signal. Is it a coincidence that both of these features, FCD and exact adaptation, appear in the same system? The answer is no: any system with FCD must show exact adaptation. This is because FCD demands that the output remain the same if we multiply the input S_0 by any $\lambda > 0$. Thus, steady-state output a_{st} is independent of the background signal, precisely the definition of exact adaptation.

Does every system with exact adaptation show FCD? The answer again is no. Exact adaptation by itself is not enough to guarantee FCD. In fact, the best-known circuit for exact adaptation in engineering, linear integral feedback, does not show FCD, nor does any other linear circuit. This is because linear equations show output changes that are proportional to absolute (not relative) input changes (Exercise 10.13). The chemotaxis circuit shows FCD by virtue of the nonlinear nature of its integral feedback loop, in which K multiplies the error signal: $dK/dt \sim K(a_{st} - a)$ rather than the linear form $dK/dt \sim a_{st} - a$. That extra K gives a logarithmic flavor to the equation, $d\log(K)/dt \sim a_{st} - a$, needed to reject the input scale λ . The log comes from the fact that $d\log(K)/dt = (1/K) dK/dt$. To emphasize the difference from linear integral feedback, this circuit is termed a nonlinear integral feedback loop (NLIFBL).

FCD is a pretty tough demand on a system – the entire dynamical response must depend only on fold change. Are there any other circuits that show FCD?

10.4 THE INCOHERENT FEEDFORWARD LOOP CAN SHOW FCD

Demanding FCD narrows down the possible circuits to a very few. Intriguingly, among these few is a common network motif. This motif is the incoherent type-1 FFL (I1-FFL), our friend from [Chapter 3](#). It was the first circuit shown to have FCD, by Lea Goentoro and Marc Kirschner et al. (Goentoro et al., 2009).

In the I1-FFL, the input X activates an output gene Z and its repressor Y . In [Chapter 3](#), we modeled the I1-FFL using logic input functions (AND and OR gates). To see its FCD property, we need a more graded regulation.

The I1-FFL can provide FCD when (i) the binding of X to its target promoters is weak (so that Michaelis–Menten terms become approximately linear: $X/(K_x + X) \sim X/K_x$), and (ii) binding of the repressor Y is strong (i.e., when Y exceeds its binding constant K_y to the Z promoter, $Y \gg K_y$, so that the Michaelis–Menten binding term $1/(1 + [Y/K_y])$ becomes, to a good approximation, K_y/Y). In this case, we can write

$$\frac{dY}{dt} = \beta_1 X - \alpha_1 Y \quad (10.4.1)$$

$$\frac{dZ}{dt} = \frac{\beta_2 X}{Y} - \alpha_2 Z \quad (10.4.2)$$

Upon a step of X , output Z first rises, but then Y rises to repress Z production, forming a pulse of output Z that adapts exactly to its previous steady state (Figure 10.7). The repressor Y does not adapt. Instead, it tracks the input level X , like an internal representation of the input. As in chemotaxis, the ratio X/Y in the second equation normalizes out the input scale. In the following solved exercise, we show that these equations have FCD, using dimensionless variables.

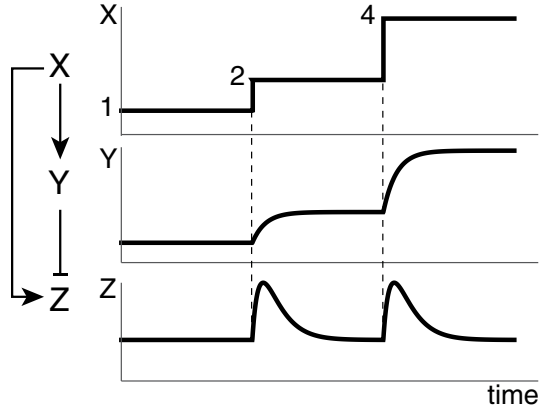


FIGURE 10.7

Solved Example 10.2: The I1-FFL Can Show FCD

Show that Equations 10.4.1 and 10.4.2 have FCD.

Solution:

First, let's see if the circuit has exact adaptation. To test for exact adaptation, we solve the steady-state condition $dY/dt = 0$, $dZ/dt = 0$ for a constant input X_0 . This yields $Y_{st} = \beta_1 X_0 / \alpha_1$, so that the repressor level Y is proportional to the background input. The steady-state output is $Z_{st} = \beta_2 \alpha_1 / \beta_1 \alpha_2$, which does not depend on input level X_0 . Thus, Z shows exact adaptation.

Now let the input signal vary with time, $X(t)$. To test for FCD, let's define new variables, as we did for chemotaxis, by rescaling Y to the steady-state input $\tilde{Y} = Y/X_0$ and define the fold change $F(t) = X(t)/X_0$. With these new variables we get, by dividing Equation 10.4.1 by X_0 , scaled equations that depend only on the fold change $F(t)$:

$$\frac{d\tilde{Y}}{dt} = \beta_1 F - \alpha_1 \tilde{Y}$$

$$\frac{dZ}{dt} = \frac{\beta_2 F}{\tilde{Y}} - \alpha_2 Z$$

Thanks to exact adaptation, the initial conditions are independent of X_0 . The dynamic equations and their initial conditions depend only on fold change F , and thus the output dynamics $Z(t)$ are completely determined by the fold change in input, and hence display FCD.

FCD breaks down in the I1-FFL when Y is too small to ignore the binding coefficient K_Y .

FCD in the I1-FFL circuit occurs for any value of the production and removal rates $\alpha_{1,2}$ and $\beta_{1,2}$ in Equations 10.4.1 and 10.4.2. These parameters affect the shape of the dynamics, by setting the amplitude and response time of the output pulse.

10.5 A GENERAL CONDITION FOR FCD

The two circuits we saw so far, I1-FFL and the nonlinear integral feedback loop (NLIFBL), are the only FCD circuits that have been experimentally characterized in biological systems to date (Figure 10.8, note that node Y in the NLIFBL has autoregulation because K in Equation 10.2.2 multiplies its own production rate). Are there other possible FCD circuits, and if so, how many?

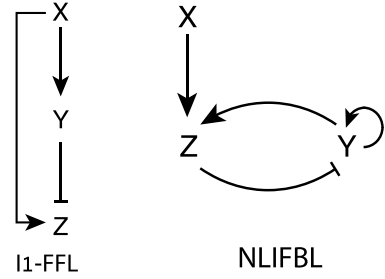


FIGURE 10.8

To address this, Oren Shoval, Eduardo Sontag and colleagues (Shoval et al., 2010) defined a **homogeneity condition** for FCD by which you can check equations for the FCD property. This condition generalizes the dimensionless variable approach we used above. It requires that if the input is multiplied by a constant λ , the system has an internal variable Y that also increases by a factor of λ (or more generally systematically changes with λ). The inner variable Y is used as a memory that divides the output Z , normalizing out λ .

Consider a system with input X , output Z and internal variable Y (Figure 10.9). The dynamics of Y and Z are given by the differential equations

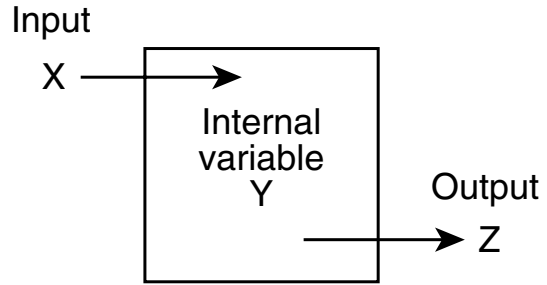


FIGURE 10.9

$$\frac{dY}{dt} = f(X, Y, Z) \quad (10.5.1)$$

$$\frac{dZ}{dt} = g(X, Y, Z) \quad (10.5.2)$$

A sufficient condition for FCD is that the system has a stable steady-state solution, that the output Z shows exact adaptation and that g and f satisfy the following homogeneity conditions for any $\lambda > 0$:

$$f(\lambda X, \lambda Y, Z) = \lambda f(X, Y, Z) \quad (10.5.3)$$

$$g(\lambda X, \lambda Y, Z) = g(X, Y, Z) \quad (10.5.4)$$

If f is linear, the condition is also necessary.

The proof is essentially the same as for the solved examples above. A generalization in which Y can depend more generally on λ is shown in Exercise 10.3.

Both the chemotaxis and the I1-FFL equations above satisfy these homogeneity conditions. For the I1-FFL, for example, $f(x, y, z) = \beta_1 x - \alpha_1 y$, so that $f(\lambda x, \lambda y, z) = \lambda f(x, y, z)$. Similarly, $g(x, y, z) = \beta_2(x/y) - \alpha_2 z$, so that $g(\lambda x, \lambda y, z) = g(x, y, z)$. In the chemotaxis circuit $f(S, K,$

$a) = cK(a_{st} - a)$, so that $f(\lambda S, \lambda K, a) = \lambda f(S, K, a)$. In contrast, linear integral feedback (without the K in front of the parenthesis) does not satisfy the conditions. It shows exact adaptation, but $f(S, K, a) = c(a_{st} - a)$, which fails the homogeneity test.

These conditions highlight the fact that details are important for FCD. A different implementation of the I1-FFL, called a sniffer (Tyson, Chen and Novak, 2003) in which Y inhibits Z not by transcription ($dZ/dt = X/Y - Z$) but by degradation ($dX/dt = X - YZ$) does not show FCD. FCD is not found in the sniffer because response time depends on absolute (and not relative) input changes.

To look for additional types of FCD circuits, Miri Adler, Pablo Szekely, Avi Mayo and colleagues (Adler et al., 2017) used these homogeneity conditions to perform an analytic scan of a class of half-a-million three-node circuits. Only 0.1% of the circuits showed FCD, as opposed to 10% that showed exact adaptation. Due to the enormous number of circuits, this 0.1% meant several hundred FCD circuit topologies. Intriguingly, Adler et al. showed that the two observed designs, I1-FFL and nonlinear integral feedback loop (NLIFBL), are among the handful of circuits that (i) have the minimal number of interaction arrows and (ii) optimally trade-off performance in tasks such as large response amplitude and fast response time. All other minimal FCD circuits do worse on at least one task. We will study such trade-offs in more detail in [Chapter 14](#).

10.6 IDENTIFYING FCD CIRCUITS FROM DYNAMIC MEASUREMENTS

Often, FCD can be observed experimentally using input–output measurements, but the architecture of the underlying circuit is not fully known. Can one use dynamic measurements to tell if an FCD circuit is feedforward (an I1-FFL) or feedback (like the chemotaxis circuit NLIFBL), even if the molecular interactions are not yet known?

The answer is yes, in some cases. I1-FFL and NLIFBL differ, for example, in the way that the amplitude of their output pulse depends on the fold change of an input step. You present the system with a series of input steps of different fold changes, and measure the maximum output in the resulting pulses. A noncooperative I1-FFL has a logarithmic dependence on the fold-change F of an input step, whereas the NLIFBL has a linear or power-law dependence ([Figure 10.10](#)) (Adler, Mayo and Alon, 2014). Interestingly, both circuits have an adaptation time that decreases with F .

Feedback and feedforward can also sometimes be distinguished by their response to pairs of input pulses (Rahi et al., 2017). Another test applies a strong input step: if the output shows damped oscillations (sometimes called ringing), it must contain a feedback system, because purely feedforward circuits never ring.

Let's end this chapter by thinking of the functions that FCD can provide to biological systems.

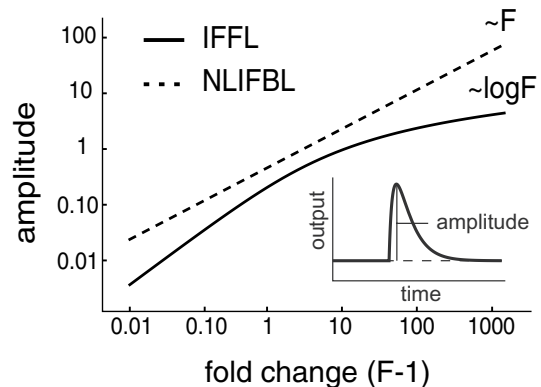


FIGURE 10.10

10.7 FCD PROVIDES ROBUSTNESS TO INPUT NOISE AND ALLOWS SCALE-INVARIANT SEARCHES

One answer was mentioned in the beginning of this chapter, that FCD helps sensory systems distinguish between a true input signal and noise. FCD responds only to changes that are on the same scale as the background, weeding out small fluctuations. A given input surge is ignored or noticed depending on the background signal (Figure 10.11). FCD therefore allows a **wide input dynamic range**, by changing sensitivity according to background level, a feat which in engineering is called gain control.

As we saw, FCD has another role: it makes the response robust to unwanted effects that multiply the input signal by a constant λ , whose value cannot be known

in advance. This solves a crucial problem in cells, as exemplified in an elegant experiment by Susan Gaudet and colleagues. Gaudet studied NF- κ B in mammalian cells, a transcription factor that responds to signals such as tumor-necrosis factor, TNF, by entering the nucleus and activating genes for inflammation and stress response. The readout of nuclear NF- κ B had better be accurate, so that cells can know whether to promote inflammation. Inflammation is a massive response that can fight pathogens, but causes tissue damage and contributes to cancer and other diseases if it occurs too often.

The challenge for precise signaling is that there is a large variation between cells in the total level of NF- κ B protein. One cell might have 10,000 NF- κ B proteins and its neighbor cell might have 30,000. After a given TNF signal, the more NF- κ B a cell has, the more will enter the nucleus to activate genes. Thus, for the same signal, each cell will see a different amount of NF- κ B in the nucleus. In other words, an unknowable factor λ – the basal amount of NF- κ B in each individual cell – multiplies the amount of nuclear TF seen after a given signal. If the response of the downstream genes was absolute and not relative, cells would decide whether to respond based on an arbitrary signal, the cell-to-cell variation in basal NF- κ B.

Gaudet showed that cells resolve this by using an I1-FFL downstream of NF- κ B in order to respond only to fold change in nuclear NF- κ B (Figure 10.12). The role of Y is played by inhibitors such as p50 dimers that compete with NF- κ B for the same site on target genes, and thus inhibit its effects. The I1-FFL allows the cells to get used to the cell-specific level of NF- κ B and to normalize it away. Such a capability to filter out multiplicative protein noise might help explain the prevalence of the I1-FFL in transcription networks from bacteria to humans.

Multiplicative effects with an unknowable factor also occur in human vision. Here, the multiplicative factor is ambient light, L . Light levels can vary by almost ten orders of magnitude

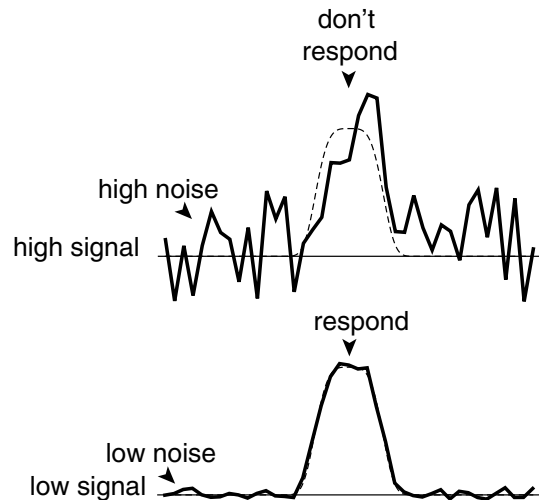


FIGURE 10.11

between midday and a moonless night. Yet we can see effectively over much of this range. To understand the role of ambient light, imagine a visual search for a face in the crowd. We are interested in the contrast field R which carries information about the face. But our eyes see a light input that is the contrast R multiplied by the ambient light $x = RL$. To remove the multiplicative constant L , FCD in the visual system normalizes out the ambient light, and allows us to make an efficient search that is invariant to a wide range of light levels.

Here is an interesting detail: At the level of the retina, there is no FCD because the neuronal output does not show exact adaptation but instead a steady-state level that is logarithmic in light L , providing the brain with information about ambient light. However, the full visual system

processes the input from the retina in various brain regions, and displays exact adaptation. This is shown by experiments that deviously move the visual field to cancel out our rapid eye movements, called saccades. Thus, the subject sees a constant input image. After a few seconds, the visual field seems to turn gray and vision stops working. We see thanks to the changes caused by rapid eye movements.

A similar multiplicative factor occurs in bacterial chemotaxis. Here the goal is to move toward sources of attractants, and the unknowable multiplying factor is the strength of the attractant source. The concentration of attractant diffusing away from a source of strength S_{source} is proportional to S_{source} , due to the linearity of the diffusion (or convection) equation. Specifically, at a distance r from the source, the attractant level is $S(r, t) \sim S_{\text{source}} e^{-r^2/2Dt}$, which is proportional to S_{source} . Thanks to FCD, the navigating bacterium can show run-and-tumble statistics that are invariant to S_{source} (as long as concentrations are within the range for FCD). The upshot is that bacteria can efficiently find the source position, regardless of the source strength. Such a process explains the experiments of Mesibov et al. (1973) with the pipette in the dish (Figure 10.1).

These properties of vision and chemotaxis can be called **scale-invariant search**, and are expected whenever an FCD system controls the movement of an agent in an input field plagued by an unknowable multiplicative factor.

In an imaginary experiment, a person searches for a cheesecake in a dark room using only the sense of smell. The room is in a cheesecake factory and has a certain background level of cheesecake aroma. After some sniffing around, the cake is found. Now do a search for half a cheesecake, but also halve the background level. If olfactory search is FCD, the average search time should be the same.

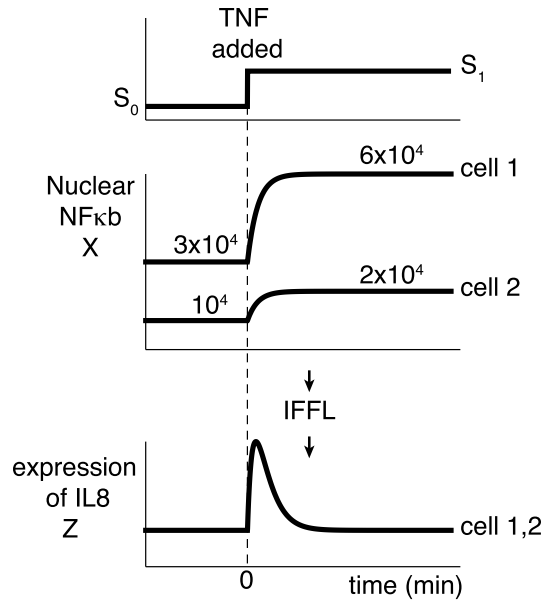


FIGURE 10.12

An intriguing question is whether FCD circuits can also explain scale invariance in psychology, especially since FCD narrows down the range of possible circuits to a very few. Recent evidence suggests that human creative search might have FCD properties (Hart et al., 2018).

Fold-change detection may be an instance where biological circuits evolved to “learn” a scaling symmetry of the physical world: the multiplicative nature of ambient light, protein levels or chemotaxis source strengths. FCD makes the output invariant to the scalar multiplying the input. There are other possible symmetries and invariances to explore. In the next chapter, we will explore an invariance of hormone circuits to certain physiological parameters. Such symmetries and invariances play a fundamental role in physics, and offer a field for further discovery in biology.

FURTHER READING

(Adler and Alon, 2018) “Fold-change detection in biological systems.”

(Adler et al., 2017) “Optimal regulatory circuit topologies for fold-change detection.”

(Adler, Mayo and Alon, 2014) “Logarithmic and power law input–output relations in sensory systems with fold-change detection.”

(Bialek, 2012) “Biophysics: searching for principles.”

(Goentoro et al., 2009) “The incoherent feedforward loop can provide fold-change detection in gene regulation.”

(Lazova et al., 2011) “Response rescaling in bacterial chemotaxis.”

(Shoval et al., 2010) “Fold-change detection and scalar symmetry of sensory input fields.”

EXERCISES

10.1 *Homogeneity conditions*: Which circuits have FCD? The input is x , the internal variable is y and the output is z .

$$\text{i. } \frac{dy}{dt} = x^2 - y; \quad \frac{dz}{dt} = \frac{x}{y} - z \quad (\text{P10.1})$$

$$\text{ii. } \frac{dy}{dt} = (x - y)^2; \quad \frac{dz}{dt} = \frac{x}{y} - z \quad (\text{P10.2})$$

$$\text{iii. } \frac{dy}{dt} = zy(z - z_o); \quad \frac{dz}{dt} = \frac{x}{y} - z \quad (\text{P10.3})$$

$$\text{iv. } \frac{dy}{dt} = x - y; \quad \frac{dz}{dt} = x - yz \quad (\text{P10.4})$$

$$\text{v. } \frac{dy}{dt} = x - y; \quad \frac{dz}{dt} = \left(\frac{x}{y}\right)^n - z^m \quad (\text{P10.5})$$

$$\text{vi. } \frac{dy}{dt} = x(z - z_o); \quad \frac{dz}{dt} = \frac{x}{y} - z \quad (\text{P10.6})$$

$$\text{vii. } \frac{dy}{dt} = x - y; \quad \frac{dz}{dt} = 2x - y - z \quad (\text{P10.7})$$

10.2 Which of the circuits of Exercise 10.1 have exact adaptation?

- 10.3 Show that the sufficient conditions for FCD can be generalized to cases where the inner variable goes as a general function of scale λ : FCD holds if $f(\lambda X, \phi(\lambda, Y)Y, Z) = \partial_Y \phi(\lambda, Y) f(X, Y, Z)$ and $g(\lambda X, \phi(\lambda, Y)Y, Z) = g(X, Y, Z)$. *Hint*: Use the scaling approach.
- 10.4 Use Exercise 10.3 to show that the following model has FCD: $dY/dt = Z - Z_0$, $dZ/dt = \log(X) - Y - Z$.
- 10.5 *I1-FFL as FCD circuit*: Solve this I1-FFL circuit exactly: $dy/dt = x - y$, $dz/dt = x/y - z$. What is the dependence of peak amplitude in response to a step input on the fold change of the step F ?
- 10.6 *Response time*: Plot the response time in the I1-FFL as a function of fold-change F of a step input. Explain why response time decays with F .
- 10.7 *Y removal*: Show that response amplitude to a given step input is larger in the I1-FFL the slower the dynamics of Y , that is the smaller α_1 . Explain this intuitively.
- 10.8 *Response laws*: simulate I1-FFL and NLIFBL for different fold input (use all parameters = 1) for input steps of different fold F . Show that amplitude is approximately logarithmic with F for the I1-FFL and approximately linear for the NLIFBL.
- 10.9 *Decreasing adaptation time with fold*: Simulate the NLIFBL (use all parameters = 1) for steps of different fold F .
- Compute the adaptation time, the time it takes to go down halfway from the peak of the pulse back to steady state. Plot the adaptation time as a function of fold. Explain the decreasing curve intuitively.
 - Compute the peak time of the pulse as a function of F . Explain.
- 10.10 *Response to ramp*: (i) Show that the I1-FFL and NLIFBL circuits with FCD adapt exactly when input goes up linearly with time $X = \gamma t$. (ii) Repeat with an exponential ramp $X = e^{\gamma t}$, and show that both circuits have a steady-state dependent on rate of exponential growth γ .
- 10.11 *Immune FCD* (Sontag, 2017): Pathogens produce proteins (antigens) that are recognized by the immune system. In one important process, T-cells are stimulated by antigen X to differentiate into helper T-cells Z that fight the pathogen and regulatory T-cells that inhibit the helper T-cells. This forms an I1-FFL-type circuit.
- Write equations for this system that show FCD.
 - If pathogens grow exponentially, $X \sim e^{\gamma t}$, how does the immune system respond? What happens if the pathogens grow only linearly with time? Explain how this system can detect exponentially growing threats.
- 10.12 *Simultaneous input signals* (Hart et al., 2013): FCD systems, such as bacterial chemotaxis, often sense two or more signals with the same receptor. For example, for independent binding of two ligands, receptor activity is the product of two

Michaelis–Menten functions: $X \sim (S_1/(K_1 + S_1)) (S_2/(K_2 + S_2))$. Far from saturation, when signal levels are lower than the K 's, one has $X \sim S_1 S_2 / K_1 K_2$. An I1-FFL downstream of this receptor can be written as

$$\frac{dY}{dt} = \beta X - \alpha_1 Y$$

$$\frac{dZ}{dt} = \frac{X}{Y} - \alpha_2 Z$$

- a. Show that a step of fold F_1 in input S_1 , and a simultaneous step F_2 in input S_2 , provide the same response as a step of size $F_1 F_2$ in one input with the other remaining constant.
 - b. What if input 1 rises by twofold and input 2 falls by twofold?
 - c. Solve a and b if binding is a product of two Hill-type functions with Hill coefficients n_1 and n_2 for the two input ligands, in the limit of low signal levels.
 - d. How can FCD help to interpret two simultaneous input signals for the same system? What is the problem with an absolute response system, given that the two signals can have unpredictable background levels?
 - e. The olfactory system can sense complex odors, like that of a rose, which combine tens of different odorants. These odorants arrive at the nose in a correlated fashion as air whiffs from the rose. Use the results of this exercise to explain how the olfactory system might distinguish a rose with great accuracy, despite the fact that the background concentration of each of these odorants can vary widely and independently.
- 10.13 *Linear systems can never show FCD*: Consider the linear system $dx/dt = Ax + B + u$ where u is the input vector, A and B are a constant matrix and vector respectively. Let's define the steady state $x_{st} = -A^{-1}(B + u_{st})$, and the output $y = x - x_{st}$. Show that the system cannot show FCD.

REFERENCES

- Adler, M. et al. 2017. 'Optimal regulatory circuit topologies for fold-change detection', *Cell Systems*, 4(2). doi: 10.1016/j.cels.2016.12.009.
- Adler, M. and Alon, U. 2018. 'Fold-change detection in biological systems', *Current Opinion in Systems Biology*, 8, pp. 81–89. doi: 10.1016/j.coisb.2017.12.005.
- Adler, M., Mayo, A. and Alon, U. 2014. 'Logarithmic and power law input–output relations in sensory systems with fold-change detection', *PLoS Computational Biology*, 10(8). doi: 10.1371/journal.pcbi.1003781.
- Bialek, W. 2012. *Biophysics: Searching for Principles*. Princeton University Press.
- Goentoro, L. and Kirschner, M.W. 2009. 'Evidence that Fold-change, and not absolute level, of β -catenin dictates Wnt signaling', *Molecular Cell*, 36(5), pp. 872–884. doi: 10.2298/CICEQ160829005R.

- Goentoro, L. et al. 2009. 'The incoherent feedforward loop can provide fold-change detection in gene regulation', *Molecular Cell*, 36(5), pp. 894–899. doi: 10.1016/j.molcel.2009.11.018.
- Hart, Y. et al. 2013. 'Comparing apples and oranges: fold-change detection of multiple simultaneous inputs', *PLoS ONE*, 8(3). doi: 10.1371/journal.pone.0057455.
- Hart, Y. et al. 2018. 'Creative exploration as a scale-invariant search on a meaning landscape'. *Nature Communications* 9, pp. 5411. doi: 10.1038/s1467-018-07715-8.
- Lazova, M.D. et al. 2011. 'Response rescaling in bacterial chemotaxis', in *Proceedings of the National Academy of Sciences*. doi: 10.1073/pnas.1108608108.
- Mesibov, R., Ordal, G.W. and Adler, J. 1973. 'The range of attractant concentrations for bacterial chemotaxis and the threshold and size of response over this range: Weber law and related phenomena', *The Journal of General Physiology*. 62(2), pp. 203–23. doi: 10.1085/jgp.62.2.203.
- Rahi, S.J. et al. 2017. 'Oscillatory stimuli differentiate adapting circuit topologies', *Nature Methods*, 14(10), pp. 1010–1016. doi: 10.1038/nmeth.4408.
- Shoval, O. et al. 2010. 'Fold-change detection and scalar symmetry of sensory input fields', *Proceedings of the National Academy of Sciences*, 107(36). doi: 10.1073/pnas.1002352107.
- Sontag, E.D. 2017. 'A dynamic model of immune responses to antigen presentation predicts different regions of tumor or pathogen elimination'. *Cell Syst*. 4(2), pp. 231–241. doi: 10.1016/j.cels.2016.12.003.
- Tyson, J.J., Chen, K.C. and Novak, B. 2003. 'Sniffers, buzzers, toggles and blinkers: dynamics of regulatory and signaling pathways in the cell', *Current Opinion in Cell Biology*, 15(2), pp. 221–231. doi:10.1016/S0955-0674(03)00017-6.



Taylor & Francis

Taylor & Francis Group

<http://taylorandfrancis.com>

Dynamical Compensation and Mutant Resistance in Tissues

We now turn to circuits at a higher level of organization – the level of tissues and organs. Tissues are made of cells that signal to each other. Distant tissues can communicate via hormones that flow in the blood stream. We will see that at the tissue level there are new, fundamental challenges. Tissues must:

- i. Maintain a proper size, despite the fact that cells tend to grow exponentially.
- ii. Signal precisely to other tissues whose parameters are unknown.
- iii. Avoid mutant cells that can grow and take over the tissue.

We will see that new principles arise to allow organs to work robustly, keep the right functional size and resist mutants. In fact, a unifying circuit design can solve all three problems at once.

11.1 THE INSULIN-GLUCOSE FEEDBACK LOOP

As a model system, we will study the insulin control of blood glucose. Glucose is the main sugar used by our cells. When we eat a meal, blood glucose concentration rises. Within a few hours, glucose returns to its baseline concentration of 5 mM (Figure 11.1). This 5 mM baseline is kept constant to within 10% over time and between people.

Tight control over blood glucose is important: if glucose drops too low, the brain doesn't have enough energy and we can pass out and even die. If glucose is too high, it damages blood vessels and other systems over the years, causing the symptoms of diabetes.

Not only is steady-state glucose kept constant, *the entire glucose dynamics $G(t)$ after a meal is tightly controlled*. For example, in a clinical test for diabetes, called the glucose tolerance test, you are asked to drink 75 g of glucose. Then, glucose levels are measured in the blood over the next two hours. Different healthy people show nearly the same glucose dynamics (Figure 11.1). Deviation from the expected dynamics (e.g., more than

11 mM glucose after 2 h) is a criterion to diagnose diabetes (Figure 11.2).

This exquisite control is carried out by a hormone circuit. Glucose is sensed by special cells in the pancreas called **beta cells**. Glucose causes beta cells to secrete the hormone **insulin** that is carried by the blood to all tissues. Insulin is sensed by receptors in the cells of many tissues, and instructs the cells in the muscle, liver and fat to take up glucose from the blood, reducing blood glucose concentration. This closes a negative feedback loop (Figure 11.3) whose timescale is hours. If glucose G is high, insulin level I rises to bring glucose down again.

A classic model for this negative feedback loop, called the minimal model, was developed by Richard Bergman et al. (Bergman et al., 1979), and is used to analyze clinical data. The concentration of blood glucose is increased by a meal input m and is reduced by the action of insulin that promotes removal of glucose from the blood. Thus, the removal rate of glucose rises with insulin:

$$\frac{dG}{dt} = m - s I G \quad (11.1.1)$$

The parameter s , called **insulin sensitivity**, is the effect of a unit of insulin on the removal rate of glucose.

Insulin, in turn, is produced by beta cells, that we denote B , at a rate that increases with glucose, $qf(G)$. Insulin is degraded at a rate γ , with a half-life on the order of 30 min:

$$\frac{dI}{dt} = qBf(G) - \gamma I \quad (11.1.2)$$

Solving this model shows that a meal input causes a rise in glucose, eliciting a rise in insulin, causing glucose to drop back down (Figure 11.1).¹

A fascinating thing about the tight regulation of glucose around 5 mM is that it occurs *despite large differences between people in insulin sensitivity*, s . This parameter can be measured by injecting insulin and noting the reduction in blood glucose. People can vary by a factor of

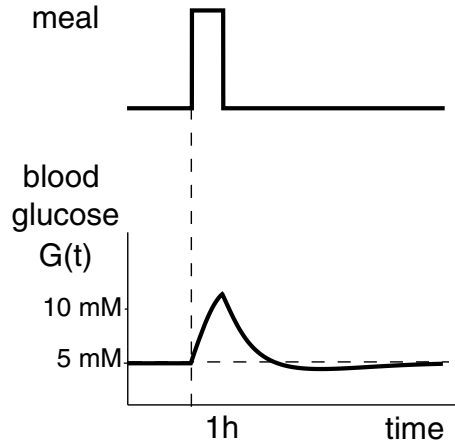


FIGURE 11.1

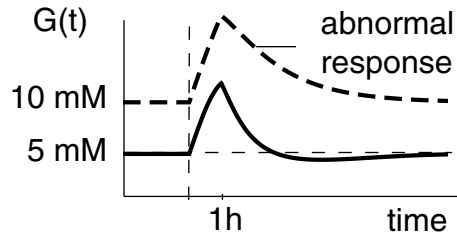


FIGURE 11.2

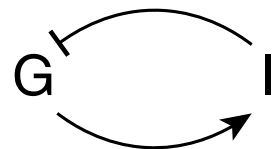


FIGURE 11.3

¹ Many effects are ignored here because they are not crucial to understand the principles in this chapter. This includes production of glucose by the liver, insulin-independent uptake of glucose by the brain, the hormone glucagon which increases liver glucose production when glucose falls below 5 mM, the effects of fat and amino acids in the diet, the delay for insulin to reach peripheral tissues, and so on.

ten in insulin sensitivity, which is affected by exercise, pregnancy, inflammation, stress, obesity, genetics and other factors. Low insulin sensitivity is also called **insulin resistance**.

Insulin sensitivity varies because it is a physiological parameter that controls glucose allocation between bodily systems. For example, exercise increases insulin sensitivity and diverts more glucose to muscle tissues. Infection decreases sensitivity, causing more glucose to stay in the blood to be used by the immune system. Pregnancy decreases mom's insulin sensitivity and hence diverts more glucose for the growth of the fetus – in pathological cases placing the mother at risk for diabetes.

Importantly, despite the large variation in insulin sensitivity, most people do not have diabetes, and show the normal glucose level of 5 mM and the normal glucose dynamics in the glucose test. For example, people with obesity have very low s (high insulin resistance), but most of them have no diabetes, with 5 mM glucose and normal glucose dynamics (Figure 11.4). Our goal is to understand how the system compensates for variations in an important parameter like s .

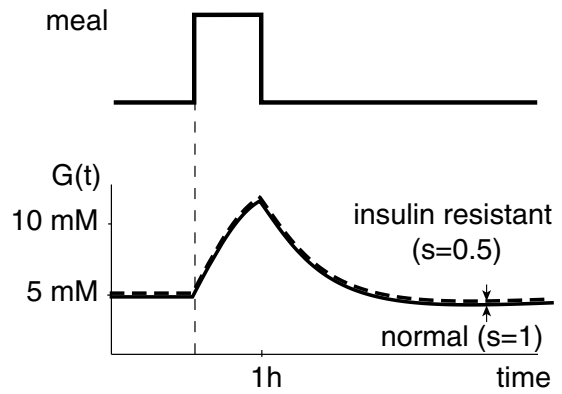


FIGURE 11.4

11.2 THE MINIMAL MODEL IS NOT ROBUST TO CHANGES IN INSULIN SENSITIVITY

So how does the insulin circuit compensate for variations in insulin sensitivity, namely variations in the intrinsic effectiveness of insulin on far-away tissues? The minimal model cannot account for this compensation. It shows a steady-state glucose level and response dynamics that depend on the parameter s . Low levels of s , for example, cause higher steady-state glucose, higher peak responses and longer response times, as can be seen in a numerical solution of the model in Figure 11.5. We next solve the minimal model to analyze this non-robustness.

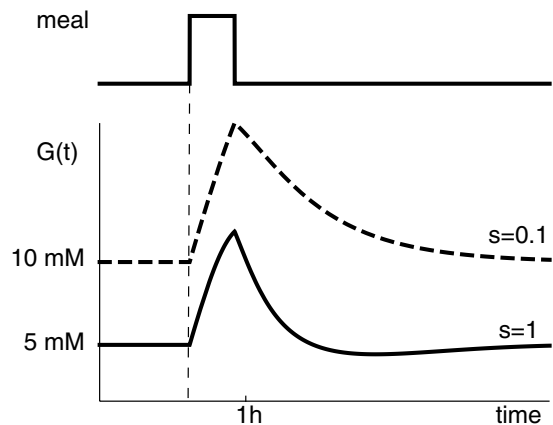


FIGURE 11.5

Solved Example 11.1: Show That Steady-State Glucose Depends on Insulin Sensitivity in the Minimal Model

At steady state, $dG/dt = 0$ and $dI/dt = 0$. Assuming a constant glucose input m_0 (e.g., the basal production of glucose by the liver when we fast overnight), we find from Equation 11.1.1 that $sI_{st}G_{st} = m_0$ and from Equation 11.1.2 that $qBf(G_{st}) = \gamma I_{st}$. Hence, $G_{st} = \gamma m_0 / sqG_{st}f(G_{st})$. Let's use $f(G) = G^2$ as proposed by Topp (Topp et al., 2000). This yields a steady-state glucose level of $G_{st} = (\gamma m_0 / sqB)^{1/3}$, which depends on s . For example, 10-fold reduction in s leads to about a twofold increase in G_{st} , with blood sugar going from 5 mM to a pathological 10 mM. The time it takes glucose to return to baseline would also be longer (Figure 11.5). The minimal model thus shows dynamics whose shape depends on the parameter s . Such dependence on parameters is typical of most models that we can write.

The minimal model cannot explain the robustness of glucose levels to variations in insulin sensitivity. Therefore, robustness must involve additional processes beyond the minimal model's glucose-insulin loop. Indeed, the way that the body compensates for decreased insulin sensitivity is by increasing the number of beta cells in order to increase insulin levels, to exactly match the decrease in s . For example, people with obesity that are insulin resistant have more beta cells than lean individuals. They thus secrete more insulin, compensating for their insulin resistance.

The compensation is seen in a hyperbolic relation that exists between healthy people: an inverse relationship between s and steady-state insulin that keeps the product of the two approximately constant: $sI_{st} = \text{const}$ (Kahn et al., 1993). People thus compensate for low insulin sensitivity with more insulin (Figure 11.6). People with diabetes have values that lie below this hyperbola.

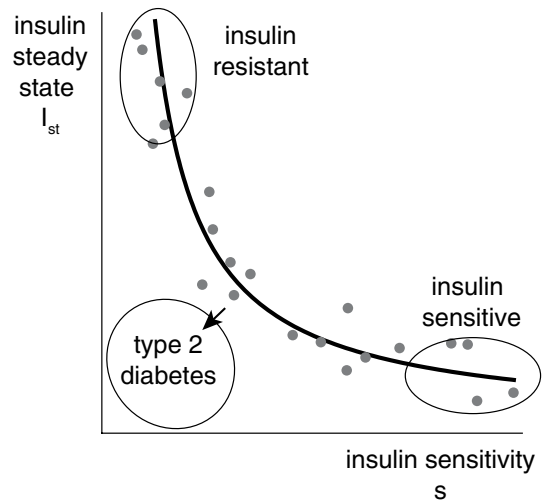


FIGURE 11.6

11.3 A SLOW FEEDBACK LOOP ON BETA-CELL NUMBERS PROVIDES COMPENSATION

To explain how such compensation can come about, we need to expand the minimal model. We need to add equations for the way that beta-cell numbers, B , can change.

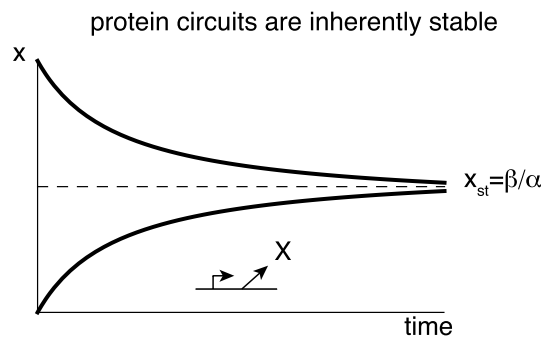


FIGURE 11.7

Here, we enter the realm of the **dynamics of cell populations**. Cell dynamics are quite unlike the dynamics we studied so far for the concentrations of proteins inside cells. For proteins we used equations that, at their core, have production and removal terms, $dx/dt = \beta - \alpha x$, and safely converge to a stable fixed point, $x_{st} = \beta/\alpha$ (Figure 11.7).

Cells, however, live on a knife's edge. Their basic equations contain an inherent instability. The equations describe cell proliferation and removal (cell death; Figure 11.8). Since all cells are made by cells, the proliferation rate is intrinsically autocatalytic, a rate constant times the concentration of cells: proliferation = pB . As a result, the balance between proliferation rate pB and death rate dB leads to exponential growth of cells at rate $\mu = p - d$

$$\frac{dB}{dt} = pB - dB = (p - d)B = \mu B \quad (11.3.1)$$

If proliferation exceeds death, growth rate μ is positive and cell numbers rise exponentially, $B \sim e^{\mu t}$ (Figure 11.9). If death exceeds proliferation, μ is negative, and cell numbers exponentially decay to zero.

Such an explosion in cells numbers occurs in cancer, and a decay in cell numbers occurs in degenerative diseases. This is the problem of **tissue size control**.

To keep cell numbers constant, we need additional feedback control, because we need to balance proliferation and death in order to reach zero growth rate, $\mu = 0$. Moreover, the feedback loop must keep the tissue at a good functional size. Hence, the feedback mechanism must somehow register the biological activity of the cells and accordingly control their growth rate.

Such feedback control occurs for beta cells, as described by Brian Topp and Dianne Finegood (Topp et al., 2000). The feedback signal is blood glucose: glucose controls the cells' proliferation and death rates, so that $\mu = \mu(G)$. The death rate of beta cells

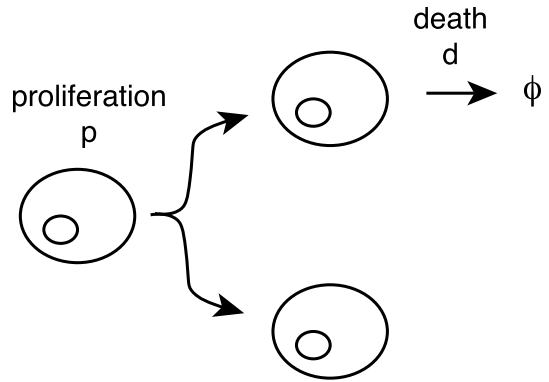


FIGURE 11.8

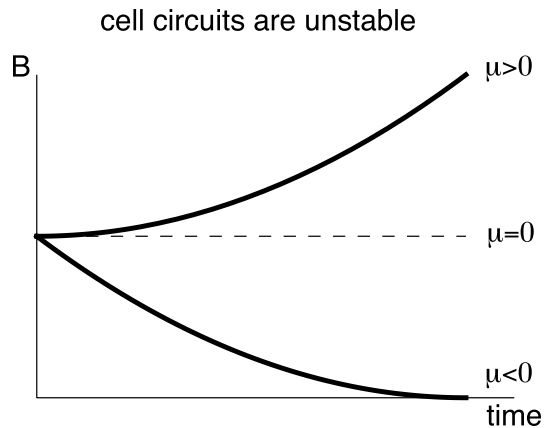


FIGURE 11.9

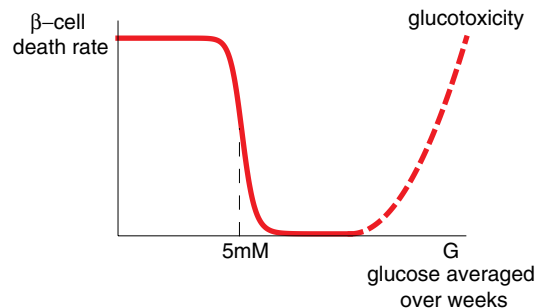


FIGURE 11.10

is high at low glucose, and falls sharply around 5 mM glucose (Figure 11.10). Death rate rises again at high glucose, a phenomenon called **glucotoxicity**, which we will return to soon. For now, let's focus on the region around 5 mM. Proliferation rises with glucose, so that the curves describing the rates for proliferation and death cross near $G_0 = 5$ mM (Figure 11.11). Therefore, $G_0 = 5$ mM is the fixed point that we seek with zero growth rate, $\mu(G_0) = 0$ (Figure 11.12).

Our revised model, the BIG model (Beta-cell-Insulin-Glucose model, Figure 11.13), includes a new equation for the beta cells B

$$\frac{dG}{dt} = m - sIG \quad (11.3.2)$$

$$\frac{dI}{dt} = qBf(G) - \gamma I \quad (11.3.3)$$

$$\frac{dB}{dt} = B\mu(G) \quad \mu(G_0) = 0 \quad (11.3.4) \quad \text{FIGURE 11.12}$$

The point $G_0 = 5$ mM is a stable fixed point for both beta-cells and blood glucose. We can analyze this using the rate plot, Figure 11.11. Note that the horizontal axis is G , unlike the rate plots in earlier chapters. If glucose is above 5 mM, beta cells have proliferation $>$ death, they increase in number, leading to more insulin, pushing glucose back down towards 5 mM. If glucose is too low, beta cells die more than they divide, leading to less insulin, pushing glucose levels back up.

This feedback loop operates on the timescale of weeks, which is the proliferation rate of beta cells. It is much slower than the insulin-glucose feedback that operates over minutes to hours. This **slow feedback loop** keeps beta cells at a proper functional steady-state number and keeps glucose, averaged over weeks, at 5 mM. The principle is, in essence, the same as integral feedback in chemotaxis: the only way to reach steady state in Equation 11.3.4 is when $G = G_0$.

The steep drop of the death curve at G_0 is important for the precision of the fixed point. Due to the steepness of the death curve, variations in proliferation rate do not shift the 5 mM fixed point by much (Figure 11.11). The steep death curve is generated by the cooperativity of key enzymes that sense glucose inside beta cells (Karin et al., 2016).

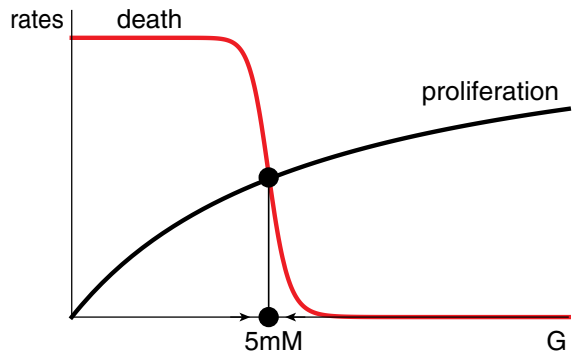


FIGURE 11.11

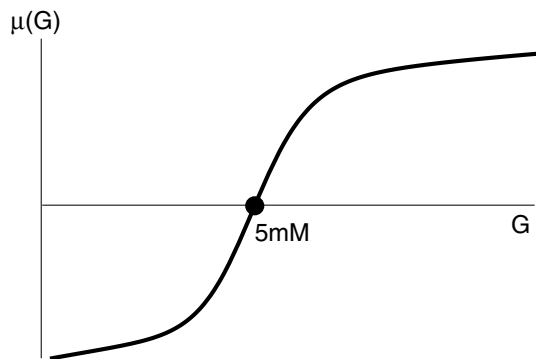


FIGURE 11.12

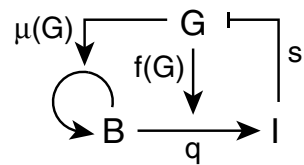


FIGURE 11.13

11.4 DYNAMICAL COMPENSATION ALLOWS THE CIRCUIT TO BUFFER PARAMETER VARIATIONS

The slow feedback on beta cells can thus maintain the 5 mM glucose steady state despite variations in insulin sensitivity, s . Remarkably, this feedback model can also resolve the mystery of how glucose *dynamics* on the scale of hours are invariant to changes in insulin sensitivity. I mean that the BIG model shows how, in the glucose tolerance test, the response to an input m of 75 g glucose yields the same output $G(t)$, including the same amplitude and response time, for widely different values of the insulin sensitivity parameter s . This independence of the entire dynamical curve on a parameter such as s is very unusual, because changing a key parameter in most models changes their dynamics.

This ability of a model to compensate for variation in a parameter was defined by Omer Karin et al. (Karin et al., 2016) as **dynamical compensation** (DC): Starting from steady state, the output dynamics in response to an input is invariant with respect to the value of a parameter. To avoid trivial cases, the parameter must matter to the dynamics, when you start away from steady state. To establish DC in our model requires rescaling of the variables in the equations, as done in the next solved example.

Solved Example 11.2: Show That the BIG Model Has Dynamical Compensation (DC)

To establish DC, we need to show that starting at steady state, glucose output $G(t)$ in response to a given meal input $m(t)$ is the same regardless of the value of s . To do so, we will derive scaled equations that do not depend on s . To get rid of s in the equations, we rescale insulin to $\tilde{I} = sI$, and beta cells to $\tilde{B} = sB$. Hence, s vanishes from the glucose equation

$$\frac{dG}{dt} = m - \tilde{I}G \quad (11.4.1)$$

Multiplying the insulin and beta-cell equations (Equations 11.3.3 and 11.3.4) by s leads to scaled equations with no s

$$\frac{d\tilde{I}}{dt} = q\tilde{B}f(G) - \gamma\tilde{I} \quad (11.4.2)$$

$$\frac{d\tilde{B}}{dt} = \tilde{B}\mu(G) \quad \text{with} \quad \mu(G_0) = 0 \quad (11.4.3)$$

Now that none of the equations depends on s , we only need to show that the initial conditions of these scaled equations also do not depend on s . If both the equations and initial conditions are independent of s , so are the entire dynamics. There are three initial condition values that we need to check, for G , $\tilde{I}$ and $\tilde{B}$. First, $G(t=0) = G_{st}$ is independent of s because $G_{st} = G_0$ is the only way for $\tilde{B}$ to be at steady state in Equation 11.4.3. Therefore, from Equation 11.4.1, $\tilde{I}_{st} = m_0/G_0$ is independent of s , which we can

use in Equation 11.4.2 to find that $\tilde{B}_{st} = \gamma \tilde{I}_{st} / qf(G_0)$ is also independent of s . Because the dynamic equations and initial conditions do not depend on s , the output $G(t)$ for any input $m(t)$ is invariant to s , and we have DC.

Although $G(t)$ is independent of s , insulin and beta cells do depend on it, as we can see by returning to original variables $B = \tilde{B}/s$ and $I = \tilde{I}/s$. The lower s , the higher the steady-state insulin. In fact, the product of insulin and insulin sensitivity is constant, $sI_{st} = m_0/G_0 = \text{const.}$, which explains the hyperbolic relation of Figure 11.6. Also, $sB_{st} = \text{const.}$, as beta cell mass rises to precisely compensate decreases in s .

Similar considerations show that the model has DC with respect to the parameter q , the rate of insulin secretion per beta cell, and also to the total blood volume (Exercise 11.8). There is no DC, however, to the insulin removal rate parameter, γ .

Let's see how dynamical compensation works. Suppose that insulin sensitivity drops by a factor of two, representing insulin resistance (Figure 11.14). As a result, insulin is less effective and glucose levels rise. Due to the death curve, beta cells die less, and their numbers rise over weeks (Figure 11.14 upper panels show the dynamics on the scale of weeks). More beta cells means that more insulin is secreted, and average glucose gradually returns to baseline. In the new steady state, there are twice the number of beta cells and there is as much insulin. Glucose returns to its 5 mM baseline.

Let's now zoom in to the timescale of hours (Figure 11.14, lower panel). The response of glucose to a meal, long after the drop in s (time-point 3), is exactly the same as the response to a meal before the change in s (time-point 1). The insulin response, however, is two times higher. Glucose dynamics in response to a meal are abnormal only during the transient period of weeks in which beta-cell numbers have not yet reached their new, compensatory, steady state (time-point 2).

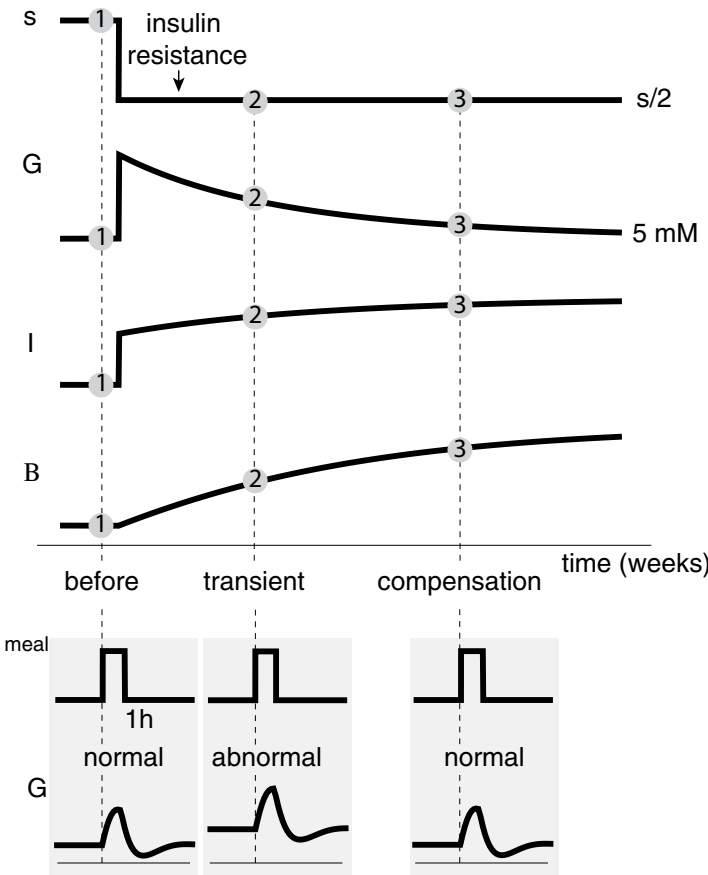


FIGURE 11.14

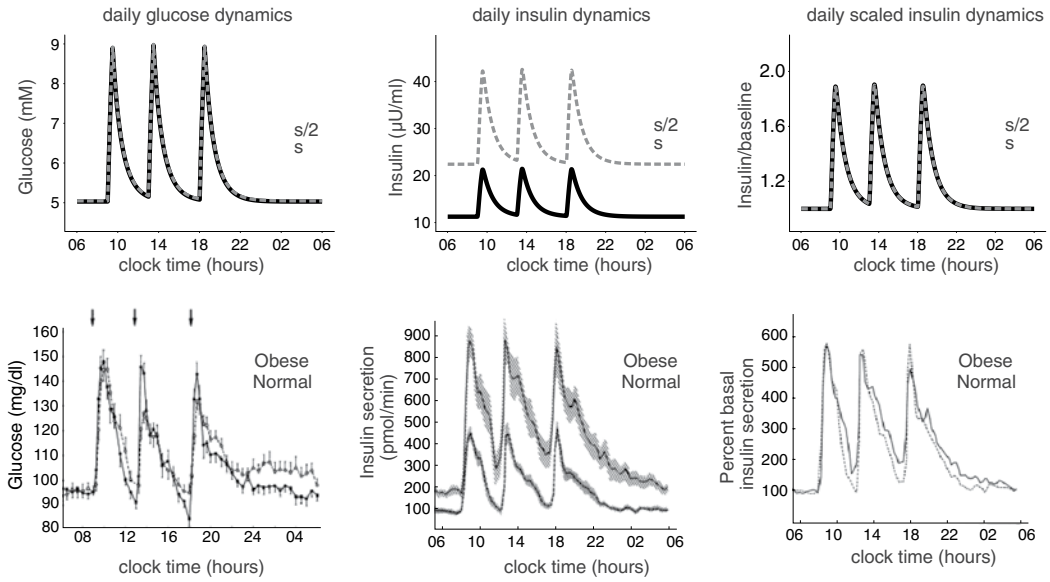


FIGURE 11.15 Adapted from (Karin et al., 2016) and (Polonsky et al., 1988).

The DC model thus predicts that people with different s should show the same glucose meal dynamics, but have insulin dynamics that scale with s . This is indeed seen in measurements that follow non-diabetic people with and without insulin resistance over a day with three meals (Figure 11.15, lower panels). Insulin levels are higher in people with insulin resistance, but when normalized by the fasting insulin baseline, there is almost no difference between the two groups (Figure 11.15). The model (upper panels in Figure 11.15) captures these observations.

The DC property arises from the structure of the equations: the parameter s cancels out due to the linearity of the dB/dt equation with B , which is a natural consequence of cells arising from cells. s also cancels out due to the linearity in B of the insulin secretion term $qBf(G)$, a natural outcome of the fact that beta-cells secrete insulin.

These basic features needed for DC exist in most hormone systems, in which glands secrete hormones that work on distant tissues. For example, free blood calcium concentration is regulated tightly around 1 mM by a hormone called PTH, secreted by the parathyroid gland (Figure 11.16). The circuit has a negative feedback loop similar to insulin-glucose, but with inverted signs: PTH causes increase of calcium, and calcium inhibits PTH secretion. The slow feedback loop occurs because parathyroid cell proliferation is regulated by calcium.

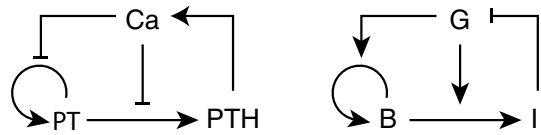


FIGURE 11.16

Other hormone systems and even neuronal systems have similar circuits (Figure 11.17), in which the size of the gland or organ expands and contracts to buffer variation in effectivity

parameters. Moreover, as embryos and children grow, these slow feedback loops can help each gland grow precisely at a rate that keeps important variables such as glucose and calcium at their desired steady-state level (see Exercise 11.8).

The feedback mechanism seems so robust. What about diseases such as diabetes? How and why do things break down? We will see that some forms of diabetes may be due to a dynamic instability that is built into the feedback loop.

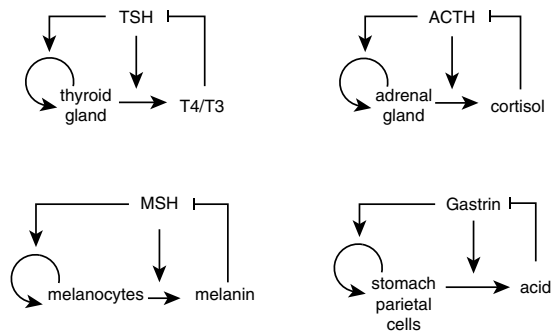


FIGURE 11.17

11.5 TYPE 2 DIABETES IS LINKED WITH INSTABILITY DUE TO A U-SHAPED DEATH CURVE

Type 2 diabetes occurs when production of insulin does not meet the demand, and glucose levels go too high. It is linked with the phenomenon of glucotoxicity that we mentioned briefly above: at very high glucose levels, beta-cell death rate rises (by death here we include all processes that remove beta cell function such as beta-cell exhaustion, de-differentiation and senescence), and eventually patients are not able to make enough insulin.

Glucotoxicity is dangerous because it adds an unstable fixed point, the point at which proliferation rate crosses death rate a second time (white circles in Figure 11.18). As long as glucose fluctuations do not exceed the unstable point, glucose safely returns to the stable 5 mM point. However, if glucose (averaged over weeks) crosses the unstable fixed point, death rate exceeds proliferation rate. Beta cells die, there is less insulin and hence glucose rises even more. This is a vicious cycle, in which glucose disables or kills the cells that control it.

This rate plot can explain several risk factors for type 2 diabetes. The first risk factor is a diet high in fat and sugars. Such a diet makes it more likely that glucose fluctuates to high levels, crossing into the unstable region. A lean diet can move the system back into the stable region. The second risk factor is ageing. With age, proliferation rate of cells drops in all tissues, including beta cells. This means that the unstable fixed point moves to lower levels of G (Figure 11.19), making it

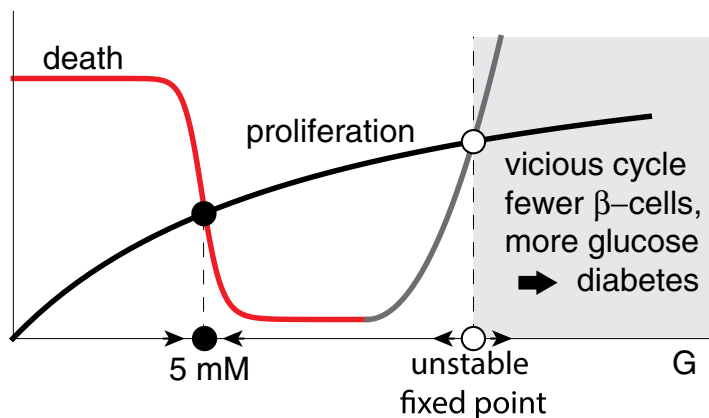


FIGURE 11.18

easier to cross into the unstable region. Note that the stable fixed point also creeps up to slightly higher levels. Indeed, with age, the glucose set point mildly increases in healthy people.

A final risk factor is genetics. It appears that the glucotoxicity curve is different between people. A shifted glucotoxicity curve can make the unstable fixed point come closer to 5 mM (Figure 11.20).

Why does glucotoxicity occur? Much is known about *how* it occurs (which is different from *why* it occurs), because research has focused on this disease-related phenomenon. Glucotoxicity is regulated by the same processes that govern beta-

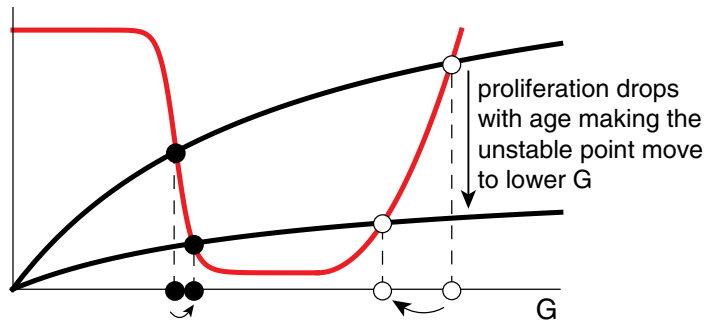


FIGURE 11.19

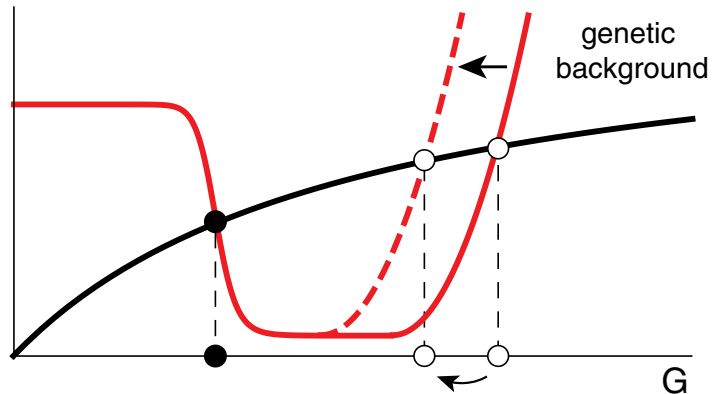


FIGURE 11.20

cell proliferation and insulin secretion. It is enhanced by reactive oxygen species (ROS) generated by the accelerated glycolysis in beta cells presented with high glucose. ROS cause extensive cell damage, and beta-cell death. The sensitivity of beta cells to ROS does not seem to be an accidental mistake by evolution. Beta cells seem designed to die at high glucose – they are among the cells most sensitive to ROS, lacking protective mechanisms found in other cells types. Thus, it is intriguing to find a functional explanation for glucotoxicity.

11.6 TISSUE-LEVEL FEEDBACK LOOPS ARE FRAGILE TO INVASION BY MUTANTS THAT MISREAD THE SIGNAL

Omer Karin et al. (Karin and Alon, 2017) provide an explanation for glucotoxicity by considering a fundamental fragility of tissue-level feedback circuits. This fragility is to **takeover by mutant cells** that misread the input signal. Mutant cells arise when dividing cells make errors in DNA replication, leading to mutations. Rarely but surely, given the huge number of cell divisions in a lifetime², a mutation will arise that affects the way that the cell reads the input signal.

² A gram of tissue has about 10^9 cells. If they divide 1/month, there are about 10^{10} divisions in a year. Mutation rate is 10^{-9} /base-pair/division, so there will be about 10 cells expressing each possible point mutation. Depending on the tissue, cells are renewed on average every few days (intestinal epithelium), weeks-months (skin, lungs, blood cells) or never (most neurons).

Let's examine such a mutation in beta cells. Beta cells sense glucose by breaking it down in a process called glycolysis, leading to ATP production, which activates insulin release through a cascade of events. The first step in glycolysis is phosphorylation of glucose by the enzyme glucokinase. Most cell types express a glucokinase variant with a halfway-binding constant to glucose of $K = 40 \mu\text{M}$, but beta cells express a special variant with $K = 8 \text{ mM}$ – perfect as a sensor for the 5 mM range. Mutations that affect the K of glucokinase, reducing it, say, by a factor of five, cause the mutant cell to sense five times too much glucose. The mutant beta cells do glycolysis as if there was much more glucose around. It's as if the mutant distorts the glucose axis in the rate plots by a factor 5, “thinking” that glucose G is actually $5G$.

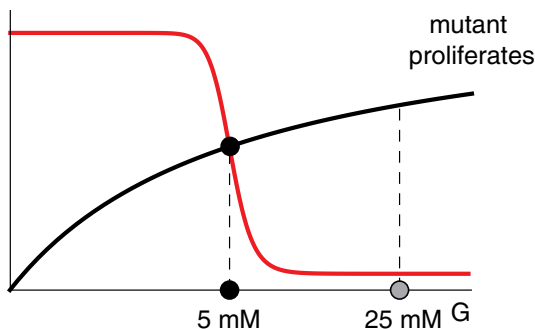


FIGURE 11.21

If our feedback design did not include glucotoxicity, such a mutant that interprets 5 mM glucose as 25 mM would have higher proliferation rate (black curve) than death rate (red curve). It would think “Oh, we need more insulin!” and proliferate (Figure 11.21). The mutant cell therefore has a growth advantage over other beta cells, which sense 5 mM correctly. The mutant will multiply exponentially and eventually take over. This is dangerous because when the mutant takes over, it pushes glucose down to a set-point level that it thinks is 5 mM, but in reality is 1 mM – causing lethally low glucose.

11.7 BIPHASIC (U-SHAPED) RESPONSE CURVES CAN PROTECT AGAINST MUTANT TAKEOVER

To resist such mutants, we must give them a growth disadvantage. This is what glucotoxicity does. The mutant cell misreads glucose as very high, has a death curve that exceeds the proliferation curve and kills itself (Figure 11.22). Mutants are removed.

The downside of this strategy is that it creates the unstable fixed point, with its vicious cycle. There is thus a **trade-off** between resisting mutants and resisting disease.

In our evolutionary past, lifestyle and nutrition was probably such that average glucose rarely stayed very high, and thus the unstable fixed point was rarely crossed. Modern lifestyle makes it more likely for glucose to exceed the unstable point, exposing a fragility to disease.

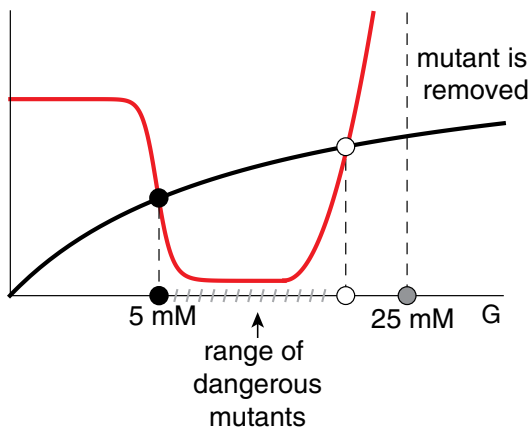


FIGURE 11.22

The glucotoxicity strategy eliminates mutants that strongly misread glucose. However, this strategy is still vulnerable to certain mutants of smaller effect: for example, mutants that misread 5 mM glucose as a slightly higher level that lies between the two fixed points (hatched region in [Figure 11.22](#)). Such mutants have a growth advantage, because they are too weak to be killed by glucotoxicity, but still have higher proliferation rate than removal rate.

Luckily, such intermediate-effect mutants are rarer than mutants that strongly activate or deactivate signaling. Designs that can help against intermediate mutants are found in beta cells: beta cells are arranged in the pancreas in isolated clusters of ~ 1000 cells called islets of Langerhans, so that a mutant can take over just one islet and not the entire tissue. Slow growth rates for beta-cells also help keep such mutants in check. Karin and Alon (2017) estimate that a small fraction of the islets are taken over by mutants in a lifetime.

This mutant-resistance mechanism can be generalized: to resist mutant takeover of a tissue-level feedback loop, the feedback signal must be toxic at both low and high levels. Such U-shaped phenomena are known as **biphasic responses**, and occur across physiology. Examples include neurotoxicity, in which both under-excited and over-excited neurons die, and immune-cell toxicity at very low and very high antigen levels. These toxicity phenomena are linked with diseases, for example Alzheimer's and Parkinson's in the case of neurons.

11.8 SUMMARY

Tissues have robustness constraints beyond those of protein circuits inside cells. First, tissues have a fundamental instability due to exponential cell growth dynamics. They require feedback to maintain steady state and a proper size. Such feedback loops use a signal related to the tissue function, to make both organ size and function stay at a proper stable fixed point. This fixed point is maintained as the cells constantly turn over on the scale of days to months.

Tissue-level circuits, such as hormone circuits, are also challenged by the fact that they need to operate on distant target tissues. These target tissues have variation in their interaction parameters, such as insulin resistance. Hormone circuits can show robustness to such parameters by means of dynamical compensation (DC), which arises due to an invariance built into the structure of the equations. In dynamical compensation, tissue size grows and shrinks in order to precisely buffer the variation in parameters.

Tissue-level feedback loops need to be protected from another consequence of cell growth – the unavoidable production of mutants that misread the signal and can take over the tissue. This constraint leads to a third principle: biphasic responses found across physiological systems, in which the signal is toxic at both high and low levels. Biphasic responses can protect against mutants by giving them a growth disadvantage. This comes at the cost of fragility to dynamic instability and disease. Additional principles of tissue-level circuits no doubt await to be discovered.

FURTHER READING

A General Resource for Models in Physiology

(Keener and Sneyd, 2008) “Mathematical physiology II: systems physiology.”

Dynamical Compensation

(Karin et al., 2016) “Dynamical compensation in physiological circuits.”

History of the Minimal Model

(Bergman, 2005) “Minimal model: perspective from 2005.”

Resistance to Mis-Sensing Mutants

(Karin and Alon, 2017) “Biphasic response as a mechanism against mutant takeover in tissue homeostasis circuits.”

The BIG Model

(Topp et al., 2000) “A model of β -cell mass, insulin, and glucose kinetics: pathways to diabetes.”

EXERCISES

11.1 *The minimal model – linear analysis:*

- a. Perform a linear stability analysis of the minimal model, Equations 11.1.1 and 11.1.2.
- b. Show that the response time to small changes in input m depends on insulin sensitivity, s .

11.2 *The minimal model – additional insulin compartments:* Some models of the insulin-glucose loop include terms for the diffusion-like movement of insulin between physiological compartments.

- a. Interpret this extension of the minimal model

$$\frac{dG}{dt} = m - s I_2 G \quad (\text{P11.1})$$

$$\frac{dI_1}{dt} = qBf(G) - aI_1 - \gamma_1 I_1 \quad (\text{P11.2})$$

$$\frac{dI_2}{dt} = aI_1 - \gamma_2 I_2 \quad (\text{P11.3})$$

What is the meaning of the parameter a ?

- b. Solve these equations numerically, and compare to the minimal model of Equations 11.1.1 and 11.1.2. How are glucose dynamics affected by the equations for the additional compartment?

- 11.3 *Liver production of glucose:* The liver produces glucose at a rate that decreases with insulin. The effect of a unit of insulin on glucose production rate is proportional to a liver insulin sensitivity parameter s_L .
- Write the BIG model with an additional term describing this effect.
 - Does the steady-state glucose level change when this term is added?
 - Suppose that s and s_L change in the same proportions, as they do in healthy people. Show that the model has DC with respect to these parameters.
 - What happens when only one of the two parameters s and s_L changes? What is the impact on the glucose tolerance test?
- 11.4 *Brain uptake of glucose:* The brain takes up glucose at an insulin-independent rate.
- Write a BIG model with a term describing this effect.
 - Does the steady-state glucose level change?
 - Is there dynamical compensation (DC) in this model?
- 11.5 *Linear analysis of the fixed points in the BIG model:*
- Construct the Jacobian of the BIG model (Equations 11.3.1 through 11.3.3).
 - Assume all positive values in the Jacobian are 1, and all negative values are -1 . Write the characteristic equation for the three eigenvalues.
 - Show that the three eigenvalues have negative real parts at the low fixed point $d\mu/dG < 0$, and that one has a positive real part at the high fixed point $d\mu/dG > 0$. Interpret this in terms of the stability of the fixed points.
- 11.6 *Compensation time in the BIG model:* Compute the compensation time in the BIG model, the time for beta cells to reach halfway to their new level upon a step change in s . Assume that $f(G) = G^{n_1}$ and $\mu(G) = \mu_0((G/G_0)^{n_2} - 1)$. Use a quasi-steady-state approximation for the fast variables G and I .
- Plot the response time as a function of n_1 and n_2 .
 - Explain why a steep death curve (the observed curve has $n_2 \sim 8$) speeds compensation time.
 - Compute the response time of glucose to a meal. Assume that beta cell number B is unchanging over the timescale of hours. Show that large n_1 (beta cell stimulation by glucose) reduces response time to a meal.
 - Explain why the model shows a trade-off between fast compensation time and rapid glucose response to a meal.
- 11.7 *Dynamical compensation (DC) for insulin production per beta cell, q :* Show that Equations 11.3.1 through 11.3.3 provide glucose with DC with respect to variations in the parameter q . How does this DC work?

- 11.8 *Dynamical compensation for blood volume*: Blood volume changes during growth of children, pregnancy, exercise and other situations.
- Show that changes in blood glucose act effectively as changes in the parameter q . *Hint*: Each cell secretes a certain number of molecules of insulin per unit time, which is distributed throughout the body's blood volume.
 - Conclude that the BIG model has DC to this variation, as opposed to the minimal model.
 - Explain why such DC can be crucial during childhood growth. Explain data that show that beta cell numbers are proportional to body weight over growth.
- 11.9 *Equations for calcium control*: Write equations for the PTH system of [Figure 11.16](#).
- 11.10 *Calcium compensation time*: Repeat Exercise 11.6 for the PTH system.
- 11.11 *Calcium mutant resistance*: Draw the analog of [Figure 11.22](#) for the PTH system. What kind of biphasic behavior is required to weed out mis-sensing mutants?
- 11.12 *General conditions for DC*: Show that DC with respect to variation in a parameter s occurs in a general class of models with input m , output G and hormone I secreted by cells X :

$$\frac{dG}{dt} = f(m, G, sI) \quad (\text{P11.4})$$

$$\frac{dI}{dt} = g(G, X, I) \quad (\text{P11.5})$$

$$\frac{dX}{dt} = Xh(G) \quad (\text{P11.6})$$

provided the following conditions apply: (i) for all s the system is stable, and there exists a unique solution $sI = I^*$ for $f(0, G^*, sI) = 0$ and a unique solution $sX = X^*$ for $g(G, sX, I^*) = 0$, (ii) homogeneity condition: $g(G, sX, sI) = sg(G, X, I)$ (Karin et al., 2016).

- 11.13 *Biphasic mechanism – Position of the unstable point*: The biphasic mechanism has two fragilities: (i) dynamical instability – if glucose rises above the unstable point, diabetes can set in, (ii) mutation vulnerability – mutant cells can take over that mis-sense 5 mM glucose as slightly higher levels that range between 5 mM and the unstable fixed point.
- Suppose that natural selection can select for changes that cause glucotoxicity (the rise in beta-cell death rate) to occur at lower or higher levels of glucose. How would that affect the position of the unstable fixed point?
 - What situations (e.g., nutrition) will cause pressure to evolve higher or lower values for the unstable fixed point?

- c. A desert rodent (*Psammomys obesus*) quickly gets obesity and diabetes in the lab when fed on a normal mouse diet. Provide an explanation based on the biphasic mechanism.
- 11.14 *Type 1 diabetes*: Type 1 diabetes is an autoimmune disease in which the immune system attacks and removes beta cells. Plot a version of Figure 11.17 for this disease. When the immune removal rate is strong enough, show that $B = 0$ is the only steady-state solution.

BIBLIOGRAPHY

- Bergman, R.N. 2005. 'Minimal model: perspective from 2005', *Hormone Research*, 64, pp. 8–15. doi: 10.1159/000089312.
- Bergman, R.N. et al. 1979. 'Quantitative estimation of insulin sensitivity.' *The American Journal of Physiology*. 236(6), pp. E667-E677. doi: 10.1172/JCI112886.
- Kahn, S.E. et al. 1993. 'Quantification of the relationship between insulin sensitivity and β -cell function in human subjects: evidence for a hyperbolic function', *Diabetes*, 42(11), pp. 1663–1672. doi: 10.2337/diabetes.42.11.1663.
- Karin, O. et al. 2016. 'Dynamical compensation in physiological circuits', *Molecular Systems Biology*, 12, p. 886. doi: 10.15252/msb.20167216.
- Karin, O. and Alon, U. 2017. 'Biphasic response as a mechanism against mutant takeover in tissue homeostasis circuits', *Molecular Systems Biology*, 13, p. 933. doi: 10.15252/msb.20177599.
- Keener, J. and Sneyd, J. 2008. *Interdisciplinary Applied Mathematical Physiology II: Systems Physiology*, Springer, 2nd edition. Available at: <https://link.springer.com/content/pdf/10.1007/978-0-387-79388-7.pdf>.
- Polonsky, K.S. et al. 1988. 'Twenty-four-hour profiles and pulsatile patterns of insulin secretion in normal and obese subjects', *Journal of Clinical Laboratory Analysis* 8, pp. 442–448. doi: 10.1172/JCI113339.
- Topp, B. et al. 2000. 'A model of β -cell mass, insulin, and glucose kinetics: pathways to diabetes', *Journal of Theoretical Biology*. 206(4) pp. 605–619. doi: 10.1006/jtbi.2000.2150.



Taylor & Francis

Taylor & Francis Group

<http://taylorandfrancis.com>

Robust Spatial Patterning in Development

Continuing with principles at the level of tissues, we now ask how tissues arise during the remarkable process of embryonic development. In embryonic development, a single cell, the egg, becomes a multicellular organism. The egg divides many times to form the cells of the body. All of these cells have the same genome. If they all expressed the same proteins, the adult would be a shapeless mass of identical cells.

During development, therefore, the progeny of the egg cell must express different genes in a spatially organized manner to become the various tissues of the organism. In order to form the intricate patterns of the tissues, many gene expression decisions need to occur at the right place and time. In this chapter, we will consider how these spatial patterns can be formed precisely.

To make a spatial pattern requires positional information, so that each cell knows where it is positioned. This information is often carried by gradients of signaling molecules (usually proteins) called **morphogens**. In the simplest cases, **morphogen gradients** are formed in a process that involves diffusion. Morphogen is produced at a certain source position and diffuses into the region that is to be patterned, called the **field**. A concentration profile is formed, in which the concentration of the morphogen is high near the source and decays with distance from the source. The cells in the field are initially all identical and can sense the morphogen by means of receptors on the cell surface. Morphogen binds the receptors, which in turn activate signaling pathways in the cell that lead to expression of a set of genes. Which genes are expressed depends on the concentration of morphogen. The fate of a cell therefore depends on the morphogen concentration at the cell's position.

The prototypical model for morphogen patterning is called the **French flag model** (Figure 12.1; L. Wolpert, 1969; Lewis, Wolpert and Szathmary, 2002). The morphogen concentration $M(x)$ decays with distance from its source at $x = 0$. Cells that sense an M concentration greater than a threshold value T_1 assume fate A. Cells that sense an M lower than T_1 but higher than a second threshold, T_2 , assume fate B. Fate C is assumed by cells

that sense low morphogen levels, $M < T_2$. The result is a three-region pattern (Figure 12.1). Morphogens often lead to patterns with more than three different fates.

Figure 12.1 depicts a one-dimensional tissue, but real tissues are three-dimensional. Patterning in three dimensions is often broken down into one-dimensional problems in which each axis of the tissue is patterned by a specific morphogen.

Complex spatial patterns are not formed all at once. Rather, patterning is a sequential process. Once an initial coarse pattern is formed, cells in each region can secrete new morphogens to generate finer sub-patterns.

Some patterns require the intersection of two or more morphogen gradients. In this way, an intricate spatial arrangement of tissues is formed. The sequential regulation of genes during these patterning processes is carried out by the developmental transcription networks that we discussed in Chapter 5. Additional processes (which we will not discuss), including cell movement, contact, mechanical forces and adhesion, further shape tissues in animals.

Patterning by morphogen gradients is achieved by diffusing molecules sensed by biochemical circuitry, raising the question of the sensitivity of the patterns to variations in biochemical parameters. A range of experiments has shown that patterning in development is robust with respect to a broad variety of genetic and environmental perturbations (Waddington, 1959; Von Dassow, Meir, Munro and Odell, 2000; Eldar, Shilo and Barkai, 2004). The most variable biochemical parameter in many systems is, as we have seen, the production rates of proteins. Experiments show that changing the rate of morphogen production often leads to very little change in the sizes and positions of the regions formed. For example, a classic experimental approach shows that in many systems the patterning is virtually unchanged upon a twofold reduction in morphogen production, generated by mutating the morphogen gene on one of the two sister chromosomes.

In this chapter, we will consider mechanisms that can generate precise long-range patterns that are robust to such perturbations, following the work of Naama Barkai and her colleagues (Eldar et al., 2002; Eldar, Rosin, Shilo and Barkai, 2003; Eldar, Shilo and Barkai, 2004). We will see that the most generic patterning mechanisms are not robust. Requiring robustness leads to special and rather elegant biochemical mechanisms.

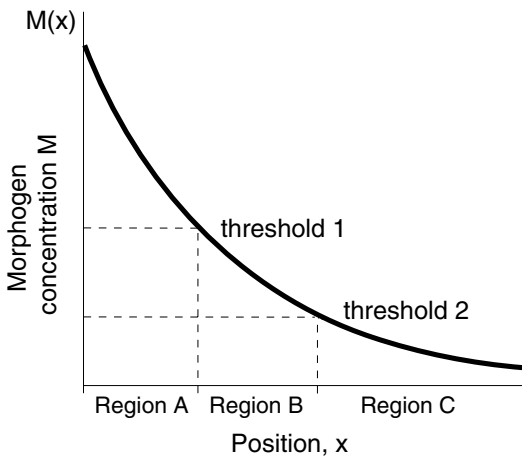


FIGURE 12.1

12.1 THE FRENCH FLAG MODEL IS NOT ROBUST

Let's begin with the simplest mechanism, in which morphogen is produced at a source located at $x = 0$ and diffuses into a field of identical cells. The morphogen is degraded at rate α . We will see that the combination of diffusion and degradation leads to an exponentially decaying spatial morphogen profile, which varies strongly if the source strength is varied.

The concentration of morphogen M in our model is governed by a one-dimensional diffusion–degradation equation. In this equation, the diffusion term, $D\partial^2 M/\partial x^2$, seeks to smooth out spatial variations in morphogen concentrations. The larger the diffusion constant D , the stronger the smoothing effect. The degradation of morphogen is described by the usual linear term $-\alpha M$, resulting in an equation that relates the rate of change of M to its diffusion and degradation:

$$\partial M/\partial t = D\partial^2 M/\partial x^2 - \alpha M \quad (12.1.1)$$

To solve this diffusion–degradation equation in a given region, we need to consider the values of M at the boundaries of the region, called the boundary conditions. The boundary conditions are a steady concentration of morphogen at its source at $x = 0$, namely $M(x = 0) = M_0$, and zero boundary conditions far into the field, $M(\infty) = 0$, because far into the field all morphogen molecules have been degraded.

At steady state, $\partial M/\partial t = 0$, Equation 12.1.1 becomes a linear ordinary differential equation:

$$D \frac{d^2 M}{dx^2} - \alpha M = 0$$

whose solution is an exponential decay that results from a balance of the diffusion and degradation processes:

$$M(x) = M_0 e^{-x/\lambda} \quad (12.1.2)$$

Thus, the morphogen level is highest at the source at $x = 0$, and decays with distance into the field. The decay is characterized by a decay length λ :

$$\lambda = \sqrt{D/\alpha} \quad (12.1.3)$$

The decay length λ is the typical distance that a morphogen molecule travels into the field before it is degraded. The larger the diffusion constant D and the smaller the degradation rate α , the larger is this distance. The decay is dramatic: at distances of 3λ and 10λ from the source, the morphogen concentration drops to about 5% and $5 \cdot 10^{-5}$ of its initial value. Therefore, λ is the typical size of the regions that can be patterned with such a gradient.

The fate of each of the cells in the field is determined by the concentration of M at the cell's position: the cell fate changes when M crosses threshold T . Therefore, a boundary between two regions occurs when M is equal to T . The position of this boundary, x_0 , is given by $M(x_0) = T$, or, using Equation 12.1.2,

$$x_0 = \lambda \log(M_0/T) \quad (12.1.4)$$

Let's now ask about the robustness of the pattern to variations in the morphogen production rate. What happens if the production rate of the morphogen source is perturbed,

so that the concentration of morphogen at the source M_0 is replaced by M'_0 ? Equation 12.1.4 shows that the position of the boundary shifts to $x'_0 = \lambda \log(M'_0/T)$. The difference between the original and the shifted boundary is

$$\delta = x'_0 - x_0 = \lambda \log(M'_0/M_0) \quad (12.1.5)$$

Thus, a twofold reduction in M_0 leads to a shift of the position of the boundary to the left by about $-\lambda \log(1/2) \sim 0.7\lambda$, a large shift that is on the order of the size of the entire pattern (Figure 12.2). Region A in Figure 12.1 would be almost completely lost.

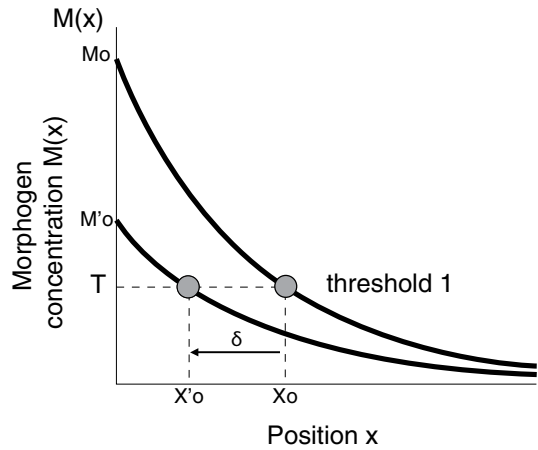


FIGURE 12.2

Hence, this type of mechanism does not seem to explain the robustness observed in developmental patterning. To increase robustness, we must seek a mechanism that reduces the shift δ that occurs upon changes in parameters such as morphogen production.

12.2 INCREASED ROBUSTNESS BY SELF-ENHANCED MORPHOGEN DEGRADATION

The simple diffusion and degradation process described above generates an exponential morphogen gradient that is not robust to the morphogen level at its source M_0 . To generate a more robust mechanism, let's try a more general diffusion–degradation process with a *nonlinear* degradation rate $F(M)$:

$$\partial M / \partial t = D \partial^2 M / \partial x^2 - F(M) \quad (12.2.1)$$

The boundary conditions are as before, a constant source concentration, $M(x=0) = M_0$, and decay to zero far into the field, $M(\infty) = 0$. This diffusion process has a general property that will soon be seen to be important for robustness: the shift δ in the morphogen profile upon a change in M_0 is *uniform in space* – it does not depend on position x . That is, all regions are shifted by the same distance upon a change in M_0 .

This uniform shift certainly occurs in the exponential morphogen profile of the previous section. The shift in boundary position δ described by Equation 12.1.5 does not depend on x . Thus, if several regions are patterned by this morphogen, as in Figure 12.1, all boundaries will be shifted by the same distance δ if morphogen production is perturbed.

More generally, spatially uniform shifts are found with any degradation function $F(M)$ in Equation 12.2.1. This property is due to the fact that the cells in the field are initially identical (unpatterned), and that the field is large (zero morphogen at infinity). This means that Equation 12.2.1 governing the morphogen has **translational symmetry**: the diffusion–degradation equations are invariant to a coordinate change $x \rightarrow x + \delta$. Such shifts only produce changes in the boundary value at $x = 0$, that is, in M_0 , as illustrated

in Figure 12.3. The spatial shift that corresponds to a reduction of M_0 to M'_0 is given by the position δ at which the original profile equals M'_0 , $M(\delta) = M'_0$. The solution of Equation 12.2.1 with boundary condition M'_0 is identical to the solution with M_0 shifted to the left by δ .

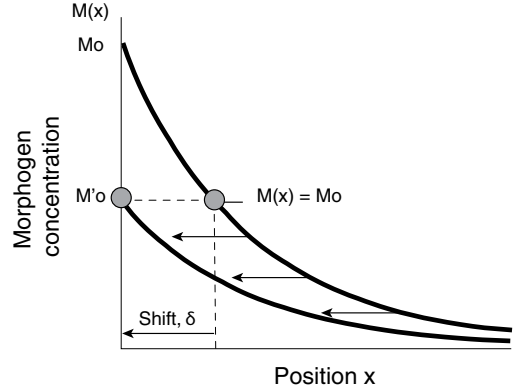


FIGURE 12.3

Our goal is to increase robustness, that is, to make the shift δ as small as possible upon a change of M_0 to M'_0 . To make the shift as small as possible, one must make the decay rate near $x = 0$ as large as possible, so that M'_0 is reached with only a tiny shift. This could be

done with an exponential profile only by decreasing the decay length λ . However, decreased λ comes at an unacceptable cost: the range of the morphogen, and hence the size of the patterns it can generate, is greatly reduced.

Thus, we seek a profile with both long range and high robustness. Such a profile should have two features:

1. Rapid decay near $x = 0$ to provide robustness to variations in M_0 .
2. Slow decay at large x to provide long range to M .

A simple solution would be to make M degrade faster near the source $x = 0$ and slower far from the source. However, we cannot make the degradation of M explicitly depend on position x (i.e., we cannot set $\alpha = \alpha(x)$ in Equation 12.1.1), because the cells in the field are initially identical. A spatial dependence of the parameters would require positional information that is not available without pre-patterning the field.

Our only recourse is nonlinear, self-enhanced degradation: a feedback mechanism that makes the *degradation rate of M increase with the concentration of M* . A simple model for self-enhanced degradation employs a degradation rate that increases polynomially with M , for example,

$$\partial M / \partial t = D \partial^2 M / \partial x^2 - \alpha M^2 \quad (12.2.2)$$

This equation describes a nonlinear degradation rate that is large when M concentration is high, and small when M concentration is low. Note that the parameter α in Equation 12.2.2 has units of $1/(\text{time concentration})$. Such nonlinear degradation can be achieved by several mechanisms described below.

At steady state, $\partial M / \partial t = 0$, the morphogen profile that solves Equation 12.2.2 is not exponential, but rather a power law:

$$M = A(x + \epsilon)^{-2}, \quad \epsilon = \left(\frac{\alpha M_0}{6D} \right)^{-\frac{1}{2}}, \quad A = 6D/\alpha \quad (12.2.3)$$

This power-law morphogen profile can have a very long range compared to exponential profiles.

To obtain long-range patterns, it is sufficient to make the source M_0 very large, so that the parameter $\epsilon \sim 1/\sqrt{M_0}$ in Equation 12.2.3 is much smaller than the pattern size. In this limit of large source, the morphogen profile in the field does not depend on M_0 at all:

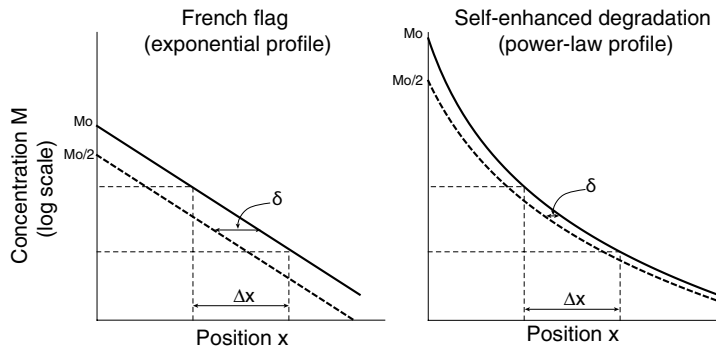


FIGURE 12.4

$$M \sim A/x^2 \quad (12.2.4)$$

so that shifts are negligible even upon large perturbations in M_0 . Patterning is very robust to variations in M_0 , as long as M_0 does not become too small. This is illustrated in Figure 12.4 that compares a French flag exponential gradient and a power-law gradient that generate a region of length Δx . The power-law profile shows a much smaller shift δ upon a twofold reduction in M_0 .

The power-law profile is not robust to changes in the parameter $A \sim D/\alpha$, the ratio of the diffusion and degradation rates. However, parameters such as diffusion constants and specific degradation rates usually vary much less than production rates of proteins such as the morphogen.

In summary, self-enhanced degradation allows a steady-state morphogen profile with a non-uniform decay rate. The profile decays rapidly near the source, providing robustness to changes in morphogen production. It decays slowly far from the source, allowing long-range patterning.

12.3 NETWORK MOTIFS THAT PROVIDE DEGRADATION FEEDBACK FOR ROBUST PATTERNING

We've seen that robust long-range patterning can be achieved using feedback in which the morphogen enhances its own degradation rate. Morphogens in the developmental processes of many species participate in certain network motifs that can provide this self-enhanced degradation. The robustness gained by self-enhanced degradation might explain why these regulatory patterns are so common.

The morphogen M is usually sensed by a receptor R on the surface of the cells in the field. When M binds R , it activates a signal transduction pathway that leads to changes in gene

expression. Two types of feedback loops are found throughout diverse developmental processes (Figure 12.5).

The first motif is a feedback loop in which the receptor R enhances the degradation of M . An example is the morphogen $M = \text{Hedgehog}$ and its

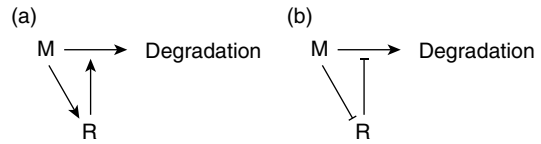


FIGURE 12.5

receptor $R = \text{Patched}$, which participate in patterning the fruit fly and many other organisms (Li et al., 2018). Morphogen binding to R triggers signaling that leads to an increase in the expression of R . Degradation of M is caused by uptake of the morphogen bound to the receptor and its breakdown within the cell, in a process called endocytosis. Thus, M enhances R production and R enhances the rate of M endocytosis and degradation (Figure 12.5a), forming a self-enhancing degradation loop.

The second type of feedback occurs when R inhibits M degradation (Figure 12.5b). A well-studied example in fruit flies is the morphogen $M = \text{Wingless}$ and its receptor $R = \text{Frizzled}$. Binding of M to R triggers signaling that represses the expression of R . R in turn inhibits the degradation of M by binding to and inhibiting a protein that degrades M (an extracellular protease) or by repressing the expression of the protease.

In both of these feedback loops, M increases its own degradation rate, promoting robust long-range patterning.

Next, we discuss a different and more subtle feedback mechanism that can lead to robust patterning. Our goal is to demonstrate how the robustness principle can help select the correct biological mechanism from among many plausible alternatives.

12.4 THE ROBUSTNESS PRINCIPLE CAN DISTINGUISH BETWEEN MECHANISMS OF FRUIT FLY PATTERNING

We end this chapter by considering a specific example of patterning in more detail (Eldar et al., 2002). We begin with describing the biochemical interactions in a small network of three proteins that participate in patterning one of the spatial axes in the early embryo of the fruit fly *Drosophila*. These biochemical interactions can, in principle, give rise to a large family of possible patterning mechanisms. Of these mechanisms, only a tiny fraction is robust with respect to variations in all three protein levels. Thus, the robustness principle helps to home in on a non-generic mechanism, making biochemical predictions that turned out to be correct.

The development of the fruit fly *Drosophila* begins with a series of rapid nuclear divisions. We consider the embryo after 2.5 h of development. At this stage, it includes about 5000 cells, which form a cylindrical layer about 500 μm across. The embryo has two axes: head–tail (called the anterior–posterior axis) and front–back (called the ventral–dorsal axis).

We will consider the patterning of the **dorsal region** (DR). Our story begins with a coarse pattern established by an earlier morphogen, which sets up three regions of cells along the circumference of the embryo (Figure 12.6a). The DR is about 50 cells wide. The goal

of our patterning process is to subdivide this region into several sub-regions using a gradient of the morphogen M .

The cells in the DR have receptors that activate a signaling pathway when M is present at sufficiently high levels. Proper patterning of the DR occurs when the activity of this signaling pathway is high at the middle of the DR and low at its boundaries (Figure 12.6b), that is, when active morphogen M is found mainly near the midline of the region.

The molecular network that achieves this patterning is made of M and two additional proteins. The first is an inhibitor I that binds M to form a complex $C = [MI]$, preventing M from signaling to the cells. The final protein in the network is a protease P that cleaves the inhibitor I . Note that P is able to cleave I when it is bound to M , liberating M from the complex. The morphogen M is not degraded in this system. The three proteins M , I and P , called *scw*, *sog* and *tld*, diffuse within a thin fluid layer outside of the cells. M is produced everywhere in the embryo, whereas I is produced only in the regions adjacent to the DR , and P is found uniformly throughout the DR .

The simplest mechanism for patterning by this system is based on a gradient of inhibitor I , set up by diffusion of I into the DR and its degradation by P (Figure 12.7). The concentration of I is highest at the two boundaries of the DR , where it is produced, and lowest at the midline of the DR . Since the inhibitor I binds and inhibits M , the activity of M (the concentration of free M) is highest at the midline of the DR , and the desired pattern is achieved. In this model, the steady-state concentration of total M (bound and free) is uniform, but its activity profile (free M) is peaked at the midline.

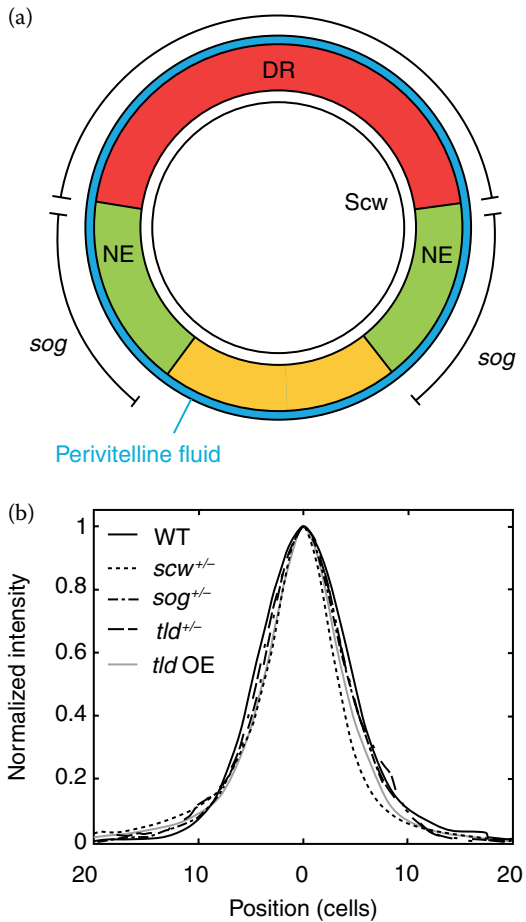


FIGURE 12.6 Adapted from (Eldar et al. 2002).

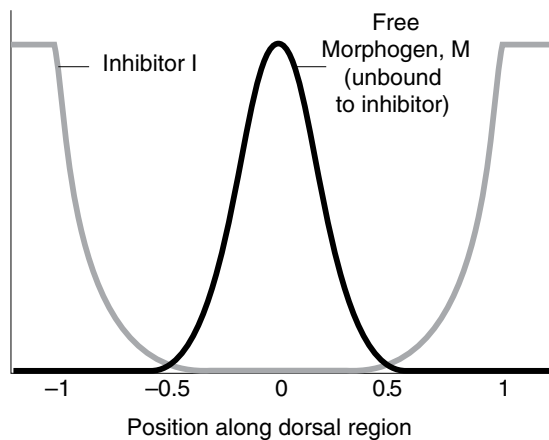


FIGURE 12.7

Unfortunately, this simple mechanism is not robust to changes in the expression of M , I or P . Changes of twofold in any of the three proteins lead to significant changes in the morphogen profile and the resulting patterns (Figure 12.8a). In contrast, experiments show that the profile of free morphogen

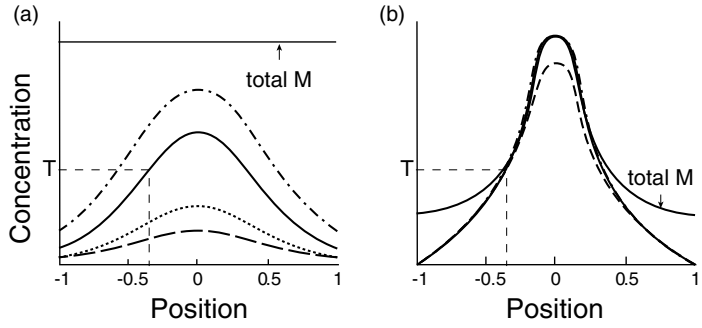


FIGURE 12.8

is robust to changes in the levels of any of the proteins in the system. Figure 12.6b shows that free morphogen is insensitive to deleting one copy of the M , I or P genes.

To make this mechanism robust, we might propose self-enhanced degradation of M , as in the previous section. However, we cannot directly apply the nonlinear degradation mechanism of the previous section, because in this system, M is not appreciably degraded.

To understand how a robust mechanism can be formed with these molecules, let us consider the general equations that govern their behavior. We will learn the approach of **numerical screening** of parameters, to identify a surprising class of robust mechanisms.

The free inhibitor I diffuses and is degraded by P at a rate α_I . Since P is known to be uniformly distributed throughout the DR , the degradation rate of I is spatially uniform and proceeds at a rate $\alpha_I PI$. Free inhibitor is further consumed when it binds free M to form a tightly bound complex, at rate k :

$$\partial I / \partial t = D_I \partial^2 I / \partial x^2 - kIM - \alpha_I PI \quad (12.4.1)$$

The complex $C = [IM]$ is formed at rate kIM and degraded by P at rate α_C :

$$\partial C / \partial t = D_C \partial^2 C / \partial x^2 + kIM - \alpha_C PC \quad (12.4.2)$$

The free morphogen M diffuses, binds inhibitor I at rate k and is liberated when the complex C is degraded:

$$\partial M / \partial t = D_M \partial^2 M / \partial x^2 - kIM + \alpha_C PC \quad (12.4.3)$$

These nonlinear equations are too tough to solve analytically. Eldar and Barkai therefore studied these equations numerically (Eldar et al., 2002). The profiles of M , I and C were found for a given set of parameters (diffusion constants, degradation rates and k). The shift in the free morphogen profile was determined upon a twofold change in the production rate of each of the three proteins M , I and P . This was repeated for different sets of parameters,

scanning four orders of magnitude of change in each parameter – a **numerical screen**. The vast majority of the parameter combinations gave non-robust solutions (97% of the solutions were non-robust according to the robustness threshold used).

The non-robust solutions typically showed exponentially decaying profiles of M activity (Figure 12.8a). The profiles varied strongly when production rates were perturbed, as shown by the different lines in Figure 12.8a. The amount of total M (free and bound to I) was uniform in space.

However, about 0.5% of the parameter sets showed a very different behavior. The profile was robust to changes in any of the protein production rates (Figure 12.8b). The morphogen activity profile was non-exponential and instead had power-law tails. In addition, the distribution of total morphogen was not spatially uniform. Morphogen was concentrated near the midline of the region (Figure 12.8b).

Inspection of the parameter values that provided the robust solutions showed that they all belonged to the same limiting class, in which certain parameters were much smaller than others. In particular, robustness was found when free M could not diffuse; only M within a complex C could diffuse (so that the diffusion constant of the complex is much larger than the diffusion constant of the free morphogen, $D_C \gg D_M$). Thus, the inhibitor is bifunctional: it not only inhibits M , it also acts as a shuttle for M by allowing it to diffuse. Because of this, this model is called the **shuttling mechanism**. Furthermore, in the robust model, free I is not degraded by the protease P . In fact, P can only degrade I within the complex C ($\alpha_C \gg \alpha_I$).

The robust shuttling mechanism is well-described by the following set of steady-state equations, setting time derivatives to zero. They are simpler than the full equations because they have two parameters equal to zero ($D_M = 0$, $\alpha_I = 0$):

$$D_I \partial^2 I / \partial x^2 - kIM = 0 \quad (12.4.4)$$

$$D_C \partial^2 C / \partial x^2 + kIM - \alpha_C PC = 0 \quad (12.4.5)$$

$$-kIM + \alpha_C PC = 0 \quad (12.4.6)$$

Remarkably, these nonlinear equations can be solved analytically. Summing Equations 12.4.5 and 12.4.6 shows that the complex C obeys a simple equation:

$$D_C \partial^2 C / \partial x^2 = 0 \quad (12.4.7)$$

The general solution of this equation is $C(x) = ax + C_0$, but due to the symmetry of the problem in which the left and right sides of the DR are equivalent, the only solution is a spatially uniform concentration of the complex:

$$C(x) = \text{const} = C_0 \quad (12.4.8)$$

Using this in Equation 12.4.6, we find that the product of free I and M is spatially uniform, because P is spatially uniform:

$$kIM = \alpha_c PC_0 \quad (12.4.9)$$

and therefore, Equation 12.4.4 can be written explicitly for M , using the relation between I and M from Equation 12.4.9, to find a simple equation for $1/M$:

$$\partial^2 M^{-1} / \partial x^2 = k/D_I \quad (12.4.10)$$

whose solution is a function peaked near $x = 0$:

$$M(x) = A / (x^2 + \varepsilon^2) \quad A = 2D_I/k \quad (12.4.11)$$

The only dependence of the morphogen profile on the total levels of M , M_{tot} , is through the parameter ε :

$$\varepsilon^2 \sim \pi A/M_{tot} \quad (12.4.12)$$

The parameter ε can be made very small by making the total amount of morphogen M_{tot} sufficiently large. In this case, the morphogen profile effectively becomes a power law that is not dependent on any of the parameters of the model (except $A = 2D_I/k$ which is a hard-wired ratio of diffusion and kinetic constants),

$$M(x) \sim \frac{A}{x^2}; \quad \text{far from midline, } x \gg \varepsilon \quad (12.4.13)$$

In particular, the free $M(x)$ profile away from the midline does not depend on the total level of M or I . The profile also does not depend on the level of the protease P or its rate of

TABLE 12.1 Principles for Robustness of Biological Functions in the Face of Different Sources of Noise

Principle	Function	Noise	Chapter
Kinetic and conformational proofreading, Demand rules	Recognition of correct target	Mis-binding of wrong target	7
Paradoxical components	Precise readout of signal	Variations in protein Levels	8
Integral feedback	Exact adaptation	Variations in protein Levels	9
Fold-change detection	Sensing across many orders of magnitude	Noise in input signal	10
Dynamic compensation	Hormone signalling and tissue size control	Variations in physiological parameters	11
Biphasic response curves	Resistance to mutant takeover	Mutants that mis-sense the signal	11
Self-enhanced degradation, Shuttling	Patterning of tissues	Variations in morphogen production rates	12

action, since these parameters do not appear in this solution at all. In summary, the free morphogen profile is robust to the levels of all proteins in the system and can generate long-range patterns due to its power-law decay.

How does the shuttling mechanism work? Morphogen M cannot move unless it is shuttled into the DR by complexing with the inhibitor I . Once the complex is degraded, the morphogen is deposited and cannot move until it binds a new molecule of I . Since there are more molecules of I near its source at the boundaries of the DR , morphogen is effectively pushed into the DR and accumulates where concentration of I is lowest, at the midline. Free inhibitor that wanders into the middle region finds so much M that it complexes and is, therefore, rapidly degraded by P . Hence, it is difficult for the inhibitor to penetrate the midline region to shuttle M away.

This is a subtle but robust way to achieve an M profile that is sharply peaked at the midline and decays more slowly deep in the field. These properties are precisely the requirements for long-range robust patterning that we discussed in Section 12.2. But unlike Section 2.2, this is done without M degradation. Interestingly, both mechanisms lead to long-ranged power-law profiles.

The shuttling mechanism requires two important biochemical details, as mentioned above. The first is that inhibitor I is degraded only when complexed to M , and not when free. The second is that M cannot diffuse unless bound to I . Both of these properties have been demonstrated experimentally, the latter following the theoretical prediction (Eldar et al., 2002).

We can now celebrate the end of Part 2 of this book, devoted to the principle of robustness. We went from proteins to circuits to tissues, finding a range of principles as summarized in Table 12.1. Robustness can help distinguish between different mechanisms and point to unexpected designs. Only a small fraction of the designs that generate a given function can do so robustly. Therefore, the principle of robustness can help us arrive at biologically plausible mechanisms. Furthermore, the robust circuits often show a pleasing elegance.

FURTHER READING

(Berg, 1993) “Random walks in biology.”

(Eldar et al., 2002) “Robustness of the BMP morphogen gradient in *Drosophila* embryonic patterning.”

(Eldar et al., 2003) “Self-enhanced ligand degradation underlies robustness of morphogen gradients.”

(Eldar et al., 2004) “Elucidating mechanisms underlying robustness of morphogen gradients.”

(Kirschner and Gerhart, 2005) “The plausibility of life.”

(Lawrence, 1995) “The first coordinates.” in “The making of a fly: the genetics of animal design.”

(Li et al., 2018) “Morphogen gradient reconstitution reveals Hedgehog pathway design principles.”

(Slack, 1991) “From egg to embryo.”

(Turing, 1952) “The chemical basis for morphogenesis.”

(Wolpert, 1969) “Positional information and the spatial pattern of cellular differentiation.”

EXERCISES

12.1 *Diffusion from both sides:* A morphogen is produced at both boundaries of a region of cells that ranges from $x = 0$ to $x = L$. The morphogen diffuses into the region

and is degraded at rate α . What is the steady-state concentration of the morphogen as a function of position? Assume that the concentration at the boundaries is $M(0) = M(L) = M_0$. Under what conditions is the concentration of morphogen at the center of the region very small compared to M_0 ?

Hint: The morphogen concentration obeys the diffusion–degradation equation at steady state:

$$D\partial^2 M/\partial x^2 - \alpha M = 0 \quad (\text{P12.1})$$

Find a solution that satisfies the diffusion–degradation equation and the boundary conditions.

- 12.2 *Diffusion with degradation at boundary:* A morphogen is produced at $x = 0$ and enters a region of cells where it is not degraded. The morphogen is, however, strongly degraded at the other end of the region, at $x = L$, such that every molecule of M that reaches $x = L$ is immediately degraded. The boundary conditions are thus $M(0) = M_0$ and $M(L) = 0$.

- What is the steady-state concentration profile $M(x)$?
- Is patterning by this mechanism robust to changes of the concentration at the source, $M(0) = M_0$?

Hint: The morphogen obeys a simple equation at steady state: $D\partial^2 M/\partial x^2 = 0$. Try solutions of the form $M(x) = Ax + B$, and find A and B that satisfy the boundary conditions.

Next, find the position where $M(x)$ equals a threshold T , and find the changes in this position upon a change of M_0 .

- 12.3 *Diffusion with reflecting boundary:* A morphogen is produced at $x = 0$ and enters a region of cells where it is not degraded. The morphogen is reflected from the membrane at the other end of the region, at $x = L$. This means the boundary conditions are $M(0) = M_0$ and $\partial M/\partial x = 0$ at $x = L$.

- Explain the boundary conditions.
- What is the steady-state concentration profile $M(x)$?

- 12.4 *Diffusion inside the bacterium in the chemotaxis system:* In the chemotaxis system of *E. coli*, the protein CheY is phosphorylated at the receptor cluster localized at one end of the cell, to form Yp . The phosphatase Z is localized to the same cluster, where it dephosphorylates Yp .

- What is the steady-state spatial profile of Yp ? Assume reflecting boundary conditions at the far end of the cell, $x = L$. Use Exercise 12.3.
- What would be the steady-state profile if Z was distributed evenly throughout the cell, leading to a uniform removal rate α ?

- c. The flagellar motors which Y_p regulates are arranged at random locations in the cell's membrane. Their rotation is very sensitive to the local concentration of Y_p . What might be the advantage of Z localized at one end of the cell as opposed to uniformly in the cell?

12.5 *Polynomial self-enhanced degradation.* Find the steady-state concentration profile of a morphogen produced at $x = 0$, which diffuses into a field of cells with nonlinear self-enhanced degradation described by

$$\partial M / \partial t = D \partial^2 M / \partial x^2 - \alpha M^n \quad (\text{P12.2})$$

When is patterning with this profile robust to the level of M at the boundary, M_0 ?

Hint: Try a solution of the form $M(x) = a(x + b)^m$ and find the parameters a and b in terms of D , M_0 and α .

- 12.6 *Robust timing:* A signaling protein X inhibits pathway Y . At time $t = 0$, X production stops and its concentration decays due to degradation. The pathway Y is activated when X levels drop below a threshold T . The time at which Y is activated is t_Y . Our goal is to make t_Y as robust as possible to the initial level of X , $X(t = 0) = X_0$.
- Compare the robustness of t_Y in two mechanisms, linear degradation and self-enhanced degradation of X (note that in this problem, all concentrations are spatially uniform). Which mechanism is more robust to fluctuations in X_0 ? Explain.
 - Explain why a robust timing mechanism requires a rapid decay of X at times close to $t = 0$.
- 12.7 *Activator accumulation vs. repressor decay (harder problem):* Compare the robustness of t_Y in Exercise 12.6 to an alternative system, in which X is an activator that begins to be produced at $t = 0$, activating Y when it exceeds threshold T . Consider both linear or nonlinear degradation of X . Is the accumulating activator mechanism more or less robust to the production rate of X than the decaying repressor mechanism?
- 12.8 *Flux boundary condition:* Morphogen M is produced at $x = 0$ and diffuses into a large field of cells where it is degraded at rate α . Solve for the steady-state profile, using a boundary condition of constant flux J at $x = 0$, $J = D \partial M / \partial x$. Compare with the solution discussed in the text, which used a constant concentration of M at $x = 0$, M_0 .
- 12.9 *Salt-and-pepper pattern using Notch-Delta lateral inhibition:* Consider a lattice of cells, like a chess board. Each cell expresses a receptor X (Notch) and can also express its ligand Y (Delta) which is displayed on the membrane. When cell A expresses Y , it binds the receptors X on its neighboring cell B. Binding of X to Y activates the receptor, which turns on signalling that inhibits production of Y in cell B. Thus, a cell that produces Y inhibits Y production in its neighbors. The cells start out with random amounts of Y .
- Explain how this circuit can produce salt-and-pepper patterns of cells, in which there are two types of cell fates arranged like black and white squares on a chess board.

- b. Write a computer program that can simulate this system. How long does it take patterns to form? Do the patterns have defects?
 - c. What possible mechanisms can reduce the risk of defects? (Sprinzak et al., 2010; Lubensky, Pennington, Shraiman and Baker, 2011; Glass, Jin and Riedel-Kruse, 2016)
- 12.10 *Turing patterns*: Alan Turing in a foundational paper in 1952 entitled *The Chemical Basis of Morphogenesis* described a way in which patterns in nature such as stripes and spots can arise naturally out of a homogeneous, uniform state. The idea is based on two chemical reactions that interact with each other. X is autocatalytic and produces itself and Y , whereas Y removes X . Furthermore, X diffuses more slowly than Y .
- a. Explain intuitively why such a mechanism can form patterns in which X is high in certain regions and low in others.
 - b. Write equations for this process (choose reasonable functional forms) and find conditions for linear stability.
- 12.11 *Spontaneous symmetry breaking using LEGI*: In order to move in a certain direction, cells sometimes need to choose a point on their membrane. Here is a mechanism to spontaneously choose such a point. When autocatalytic X is localized to the cell membrane and Y which inhibits X diffuses in the cytoplasm, a “local excitation, global inhibition” (LEGI) mechanism can help the cell spontaneously choose one point on its membrane with high X . Read about LEGI and explain this symmetry-breaking process (Levchenko and Iglesias, 2002; Altschuler, Angenent, Wang and Wu, 2008; Xiong, Huang, Iglesias and Devreotes, 2010).
- 12.12 *Scaling of pattern proportions using expansion-repression*: Scaling is the remarkable feature of development in which proportions of the developing body plan are maintained with precision despite variations in total body size. Large and small individuals are all perfectly proportioned.
- a. Show that the French flag and shuttling models of patterning by morphogen gradients discussed in this chapter do not support scaling.
 - b. Show that scaling arises naturally in a feedback strategy in which the range of the morphogen gradient increases (e.g., its diffusion constant grows) with the abundance of a diffusible molecule X , whose production, in turn, is repressed by morphogen signalling.
 - c. Show that this “expansion–repression” mechanism is analogous to an integral-feedback controller ([Chapter 9](#)) (Ben-Zvi, Shilo, Fainsod and Barkai, 2008).

BIBLIOGRAPHY

- Altschuler, S.J., Angenent, S.B., Wang, Y. and Wu, L.F. 2008. ‘On the spontaneous emergence of cell polarity’, *Nature*. <https://doi.org/10.1038/nature07119>.
- Ben-Zvi, D., Shilo, B.Z., Fainsod, A. and Barkai, N. 2008. ‘Scaling of the BMP activation gradient in *Xenopus* embryos’, *Nature*, 453(7199), pp. 1205–1211. <https://doi.org/10.1038/nature07059>.

- Berg, H.C. 1993. *Random walks in biology*. Princeton University Press. Retrieved from [https://books.google.co.il/books?id=G989DwAAQBAJ&dq=Berg,+H.C.+\(1993\).+Random+Walks+in+Biology.+Princeton+University+Press.&lr=&source=gbs_navlinks_s](https://books.google.co.il/books?id=G989DwAAQBAJ&dq=Berg,+H.C.+(1993).+Random+Walks+in+Biology.+Princeton+University+Press.&lr=&source=gbs_navlinks_s).
- Eldar, A., Dorfman, R., Weiss, D., Ashe, H., Shilo, B.Z. and Barkai, N. 2002. 'Robustness of the BMP morphogen gradient in *Drosophila* embryonic patterning', *Nature*, 419(6904), pp. 304–308. <https://doi.org/10.1038/nature01061>.
- Eldar, A., Rosin, D., Shilo, B.Z. and Barkai, N. 2003. 'Self-enhanced ligand degradation underlies robustness of morphogen gradients', *Developmental Cell*, 5, p. 635. [https://doi.org/10.1016/S1534-5807\(03\)00292-2](https://doi.org/10.1016/S1534-5807(03)00292-2).
- Eldar, A., Shilo, B.Z. and Barkai, N. 2004. 'Elucidating mechanisms underlying robustness of morphogen gradients', *Current Opinion in Genetics and Development*, 14(4), pp. 435–439. <https://doi.org/10.1016/j.gde.2004.06.009>.
- Glass, D.S., Jin, X. and Riedel-Kruse, I.H. 2016. 'Signaling delays preclude defects in lateral inhibition patterning', *Physical Review Letters*, 116(12), 128102. <https://doi.org/10.1103/PhysRevLett.116.128102>.
- Kirschner, M.W. and Gerhart, J.C. 2005. *The Plausibility of Life: Resolving Darwin's Dilemma*. <https://doi.org/10.1353/pbm.2007.0022>.
- Lawrence, P.A. and Peter A. 1995. *The Making of a Fly: The genetics of animal design* (1st ed.). Blackwell Science.
- Levchenko, A. and Iglesias, P.A. 2002. 'Models of eukaryotic gradient sensing: application to chemotaxis of amoebae and neutrophils', *Biophysical Journal*, 82(1), pp. 50–63. [https://doi.org/10.1016/S0006-3495\(02\)75373-3](https://doi.org/10.1016/S0006-3495(02)75373-3).
- Li, P., Markson, J.S., Wang, S., Chen, S., Vachharajani, V. and Elowitz, M.B. 2018. 'Morphogen gradient reconstitution reveals Hedgehog pathway design principles', *Science*, 360(6388), pp. 543–548. <https://doi.org/10.1126/science.aao0645>.
- Lubensky, D.K., Pennington, M.W., Shraiman, B.I. and Baker, N.E. 2011. 'A dynamical model of ommatidial crystal formation', in *Proceedings of the National Academy of Sciences*, 108(27), pp. 11145–11150. <https://doi.org/10.1073/pnas.1015302108>.
- Slack, J.M.W. 1991. *From egg to embryo: regional specification in early development*. Cambridge University Press. Retrieved from https://books.google.co.il/books?id=0_O1FbE6IDkC&dq=From+Egg+to+Embryo&lr=&source=gbs_navlinks_s.
- Sprinzak, D. et al. 2010. 'Cis-interactions between Notch and Delta generate mutually exclusive signalling states', *Nature*, 465(7294), 86–90. <https://doi.org/10.1038/nature08959>.
- Turing, A.M. 1952. *The chemical basis of morphogenesis*. Philosophical Transactions of the Royal Society of London. Series B, Biological Sciences (Vol. 237). Retrieved from <http://www.jstor.org/about/terms.html>.
- Von Dassow, G., Meir, E., Munro, E.M. and Odell, G.M. 2000. 'The segment polarity network is a robust developmental module', *Nature*, 406(6792), pp. 188–192. <https://doi.org/10.1038/35018085>.
- Waddington, C.H. 1959. 'Canalization of development and genetic assimilation of acquired characters', *Nature*, 183(4676), pp. 1654–1655. <https://doi.org/10.1038/1831654a0>.
- Wolpert, L. 1969. 'Positional information and the spatial pattern of cellular differentiation', *Journal of Theoretical Biology*, 25, p. 1. [https://doi.org/10.1016/S0022-5193\(69\)80016-0](https://doi.org/10.1016/S0022-5193(69)80016-0).
- Wolpert, L. and Szathmáry, E. 2002. 'Multicellularity: evolution and the egg', *Nature*, 420(6917), 745–745. <https://doi.org/10.1038/420745a>.
- Xiong, Y., Huang, C.-H., Iglesias, P.A. and Devreotes, P.N. 2010. 'Cells navigate with a local-excitation, global-inhibition-biased excitable network', in *Proceedings of the National Academy of Sciences*, 107(40), 17079–17086. <https://doi.org/10.1073/pnas.1011271107>.

Part 3

Optimality



Taylor & Francis

Taylor & Francis Group

<http://taylorandfrancis.com>

Optimal Gene Circuit Design

13.1 INTRODUCTION

Welcome to Part 3 of this book, devoted to optimality in biological systems. Optimality theory is based on the ability of natural selection to maximize fitness, and hence to converge on particular circuit designs and parameter values in a given environment. Thus, optimality theory aims to predict and understand which circuit and parameters will arise in each situation.

There is extensive evidence that biological circuits are optimized, at least to a certain extent. For example, most mutations and other changes to the cells' networks cause a decrease in performance. Furthermore, we've seen that evolution converges again and again to the same network motifs, presumably due to their functional benefits.

To understand evolutionary optimization more precisely, one needs to define a fitness function that is to be maximized. One difficulty in optimization theories is that we may not know the fitness function in the real world. For example, we currently do not know the fitness functions of cells in complex organisms. Such cells live within a society of other cells, the different tissues of the body, in which they play diverse roles. Fitness functions might not even be well-defined in some cases; disciplines such as psychology and economics deal with processes that do not appear to optimize a single fitness function, but only "satisfice" (Simon, 1997) in the sense of fulfilling several conflicting and incomparable constraints. We will discuss such multi-objective optimality in the next chapter.

We consider optimality as an idealized assumption that is a good starting point for generating testable hypotheses. The goal is to understand the constraints under which a circuit has evolved, and to predict which circuit will evolve in a given environment. You can end up concluding that a certain circuit in the cell is a historical accident or a vaguely good enough solution, but it is a mistake to start by assuming this in advance.

This chapter will, therefore, treat systems in which one can describe the fundamental forces at play during natural selection. For additional examples, see the work on optimality in metabolic networks in books by Savageau, Heinrich and Schuster, and Palsson, and optimality in animals by McNeil Alexander (see Further Reading at the end of this chapter).

Our first question is: What sets the expression level of a protein? Why are some proteins produced at a few copies per cell, others at thousands and yet others at tens or hundreds of thousands?

13.2 OPTIMAL EXPRESSION LEVEL OF A PROTEIN UNDER CONSTANT CONDITIONS

To address this question, we begin with a situation in which fitness can be precisely defined: bacteria that grow in a constant environment that is continually replenished. In this case, the fitness F is the growth rate of the cells. The number of cells, N , grows exponentially with time at rate F until they get too dense:

$$N(t) = N(0)e^{Ft} \quad (13.2.1)$$

Now, if two bacterial species with different values of F compete for growth and utilize the same resources, the one with higher F will outgrow the other and inherit the test tube. Thus, natural selection will tend to maximize F over time. This type of evolutionary process was elegantly described by G.F. Gause in “The Struggle for Existence” (Gause, 1934).

Fitness can help us address our question: What determines the level of expression of a protein? To be specific, we will consider a well-studied gene system, the *lac* system of *Escherichia coli*, which was mentioned in previous chapters. The *lac* system encodes proteins such as LacZ, which breaks down the sugar lactose for use as an energy and carbon source, and LacY, which transports lactose into the cell. When fully induced, *E. coli* makes about $Z = 60,000$ copies of the LacZ protein per cell. Why not 50,000 or 70,000?

Note that we ask “*Why* the cell makes 60,000 copies?” and not “*How* the cell makes 60,000 copies?” “How” questions relate to mechanisms such as the regulatory system, the promoter sequence, and so on, which are well-characterized in the *lac* system. “Why” questions aim to place the system within a wider theory, in this case optimality theory.

Optimality theory predicts that the protein expression level that is selected maximizes the **fitness function**. The fitness function in this case is growth rate F as a function of the number of copies of the protein expressed in the cell, $F(Z)$ (Figure 13.1).

In principle, $F(Z)$ can have local maxima and minima (Figure 13.2). A journey on this fitness function can get stuck on the local maxima or blocked by impassable valleys. Random mutations can cause noise along this journey, as can sampling noise when population sizes are small, an effect called **genetic drift**. Therefore, it is unclear a priori whether evolution can reach the global maximum and if so, how long this might take.

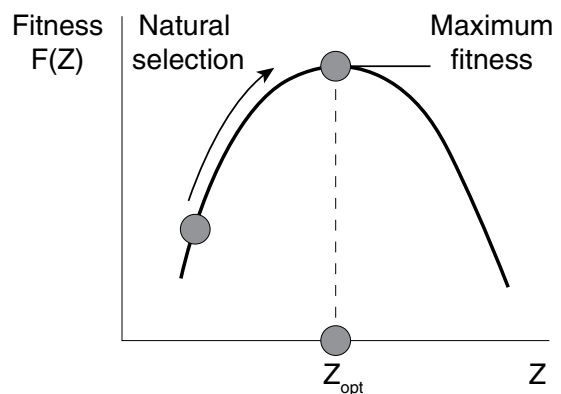


FIGURE 13.1

To answer such questions requires an experiment. We will consider an environment in which conditions are constant. In the case of LacZ, this means an environment with a constant concentration of the sugar lactose. The fitness is composed of two terms: the **cost** of the protein LacZ and the **benefit** it provides to the cells, both in units of growth rate, such that $F = \text{benefit} - \text{cost}$. Erez Dekel, when he was a postdoc with me, designed an experiment that measured benefit and cost in order to test the predictions of optimality theory in the *lac* system (Dekel and Alon, 2005).

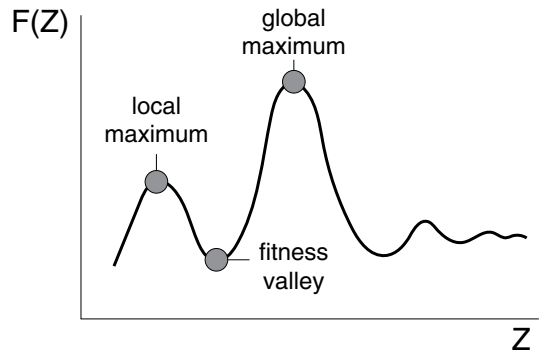


FIGURE 13.2

13.2.1 Cost of the LacZ Protein

To experimentally measure the cost function, Erez Dekel used a classic experimental tool, the inducer IPTG, a chemical that mimics the structure of lactose. IPTG binds to the *lac* repressor and causes expression of the *lac* proteins. Since IPTG cannot be metabolized by the cells, it confers no benefit on its own, and is hence called a **gratuitous inducer**.

To measure the cost of the *lac* system, IPTG was used to induce the *lac* system to various levels in the absence of lactose. The cells grew on another carbon source, glycerol. Expression of LacZ reduced the growth rate of the cells (Figure 13.3). The cost $c(Z)$ is defined as the relative reduction in the growth rate when Z units of the protein are made. Similar costs were measured using mutants with altered expression of the *lac* system, instead of IPTG (triangles in Figure 13.3).

The cost rises linearly with Z at first, so that each unit of protein reduces growth by a fixed amount. At higher protein levels, the cost starts to accelerate. The more proteins produced, the larger the cost of each additional protein. The reason is that production and maintenance of the *lac* proteins not only requires the use of the cells' resources, but also reduces the resources available to other useful proteins.

To describe this cost function, we can assume that the growth rate of the cell depends on an internal resource R (such as the number of free ribosomes in the cell

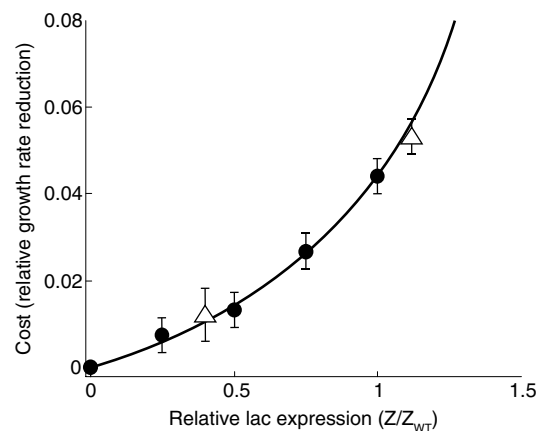


FIGURE 13.3 Adapted from (Dekel and Alon, 2005).

or the cell's energy balance¹). The growth rate is typically a saturating function of resources such as R , following a Michaelis–Menten function:

$$F \sim \frac{R}{K + R} \quad (13.2.2)$$

The production of protein Z , and its maintenance, places a burden on the cells. This burden can be described as a reduction in the internal resource R , so that each unit of protein Z reduces the resource by a small amount, $R \rightarrow R - \varepsilon Z$. Plugging this into Equation 13.2.2 shows, after a few lines of algebra, that the cost begins to diverge as Z approaches a cutoff level M , in which so much Z is produced that R is depleted (see solved Exercise 13.4):

$$c(Z) = \frac{c_0 Z}{1 - Z/M} \quad (13.2.3)$$

When only a few copies of the protein are made, the cost goes as $c(Z) \sim c_0 Z$. The cost of a single protein is about 10^{-6} , which makes sense because there are about 10^6 proteins in the bacterial cell. The cost increases more steeply when Z becomes comparable to the upper limit of expression, M , when it begins to seriously interfere with essential functions of the cell. Proteins cannot come too close to the point $Z = M$, where the cost function diverges. In this experiment, M is about twice the number of Z copies expressed by the cell under saturating lactose, $Z_{WT} = 60,000/\text{cell}$. For other proteins, M can be much larger, and cost functions are often linear over a wider range than in the case of the *lac* system.

The experimental measurements of the cost function agree reasonably with Equation 13.2.3 (Figure 13.3). They show that the relative reduction in growth rate due to the fully induced *lac* system is about 4.5%. Note that this cost of a few percent makes sense, because the fully induced *lac* proteins make up a few percent of the total number of proteins in the cell.

13.2.2 The Benefit of the LacZ Protein

We now turn to the benefit, defined as the relative increase in growth rate due to the action of the protein. In the case of LacZ, the benefit is proportional to the rate at which LacZ breaks down its substrate, lactose. The rate of LacZ is well-described by standard Michaelis–Menten kinetics (see Appendix A). Hence, LacZ breaks down lactose at a rate that is proportional to the number of copies of the protein, Z , times a saturating function of the concentration of lactose, L :

$$b(Z, L) = \frac{b_0 Z L}{K + L} \quad (13.2.4)$$

¹ The cost in this experiment is due primarily to the action of the transporter LacY. When LacY imports a lactose molecule, it exports a proton. This reduces the membrane potential, and thus a good candidate for R is the cell's proton motive force (Eames and Kortemme, 2012).

where K is the Michaelis constant² and b_0 is the maximal growth rate advantage per LacZ protein at saturating lactose.

To measure benefit, growth was measured in the presence of different levels of lactose, by keeping the system maximally induced by means of IPTG. The growth medium included glycerol so that cells could grow even without lactose.

The observed benefit function rose with lactose levels (Figure 13.4), and was well-described by Equation 13.2.4 (black curve in the figure). The relative increase in growth rate due to the fully induced level of LacZ with saturating amounts of lactose is about 17% under the conditions of the experiment.

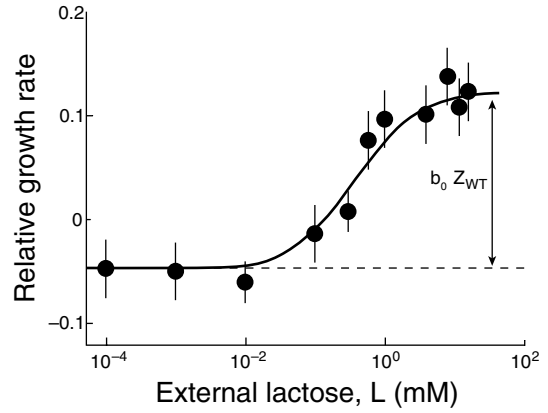


FIGURE 13.4 Adapted from (Dekel and Alon, 2005).

13.2.3 Fitness Function and the Optimal Expression Level

Now that we have the cost and benefit functions, we can calculate the fitness function, equal to the difference between benefit and cost. The fitness function is the growth rate of cells that produce Z copies of LacZ in an environment with a lactose concentration of L :

$$\begin{aligned} F(Z, L) &= \text{benefit} - \text{cost} \\ &= \frac{b_0 Z L}{K + L} - \frac{c_0 Z}{1 - Z/M} \end{aligned} \quad (13.2.5)$$

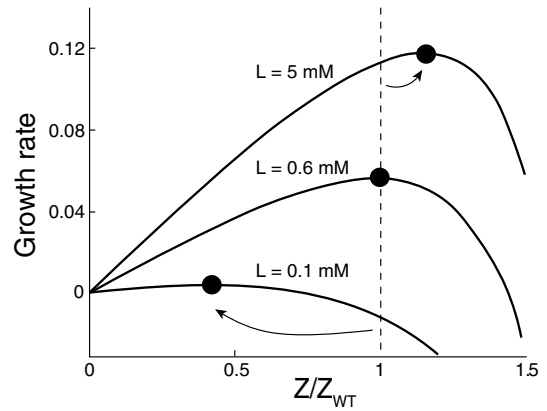


FIGURE 13.5

The fitness function displays a maximum, an expression level Z that maximizes growth rate, as shown in Figure 13.5. The position of this maximum, namely the optimal protein level Z_{opt} , depends on lactose level, L . The optimal protein level Z_{opt} can be found by the point at which the derivative of the fitness function with respect to Z equals zero:

$$\frac{dF}{dZ} = 0 \quad (13.2.6)$$

² The Michaelis constant in this case is determined by the transporter LacY, $K = 0.4$ mM. This is because the influx rate of lactose is limiting under most conditions.

Solving this equation shows that the optimal expression level increases with lactose L :

$$Z_{opt} = M \left(1 - \sqrt{\frac{c_0(K+L)}{b_0L}} \right) \quad (13.2.7)$$

Z_{opt} rises with L because lactose increases the benefit per LacZ protein, and hence increases the selection pressure to produce more proteins. The fully induced wild-type expression level, $Z_{WT} = 60,000/\text{cell}$ is predicted to be optimal when $L \sim 0.6$ mM under these experimental conditions, as shown in Figure 13.5. Growth at higher lactose levels is predicted to be maximized by protein levels that exceed the wild-type level. Conversely, low levels of lactose are predicted to have lower optimal expression levels (Figure 13.5).

When there is no lactose in the environment, the optimal level is $Z_{opt} = 0$, because the *lac* proteins confer only costs and no benefits. In fact, zero expression is optimal as long as lactose L is lower than a threshold L_c , because costs exceed benefits. The threshold L_c can be found by asking when Z_{opt} in Equation 13.2.7 becomes equal to zero:

$$Z_{opt} = 0 \quad \text{when} \quad L < L_c = K \left(\frac{b_0}{c_0} - 1 \right)^{-1} \quad (13.2.8)$$

Under the conditions of the experiments described above, the threshold level of lactose needed for selection of the gene system is $L_c \sim 0.06$ mM. If lactose environments with $L < L_c$ persist for many generations, the organism will tend to lose the gene encoding LacZ. The loss of unused genes is a well-known phenomenon; for example, bacteria grown in a chemostat³ on glucose medium with no lactose lose the *lac* genes within a few days (Hartl and Dykhuizen, 1984).

13.2.4 Cells Reach Optimal LacZ Levels in a Few Hundred Generations in Laboratory Evolution Experiments

To test the predictions of this cost–benefit analysis, Erez Dekel performed a laboratory evolution experiment. The evolution experiment used the technique of **serial dilution**. *E. coli* cells were grown in tubes with a specified level of lactose. Every day, 1/100 of the cells from each tube were passed to a tube with fresh medium (Figure 13.6). The cells grew in the tube until they reached stationary phase. The next morning, 1/100 of the cells were again passed to a fresh tube, and so on. Thus, every day, the cells grew

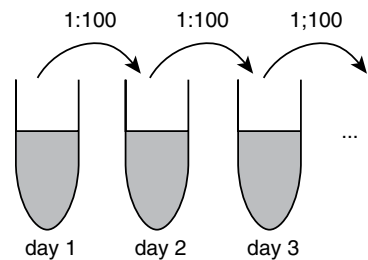


FIGURE 13.6

³ A chemostat is a device that keeps bacteria growing at a constant growth rate, by supplying a constant flow of fresh medium into a mixed aerated chamber, from which medium with cells is removed at the same rate. Cell generation time locks onto the time for exchange of half of the medium in the chamber (Novick and Weiner, 1957; Balagaddé et al., 2005; Ronen and Botstein, 2006).

100-fold, corresponding to $\log_2(100) = 6.6$ generations. Richard Lenski has been evolving *E. coli* for several decades with serial dilution, reaching tens of thousands of generations (Good et al., 2017).

Erez ran the experiment for several months, with seven tubes in parallel, each with a different lactose level, $L = 0, 0.1, 0.2, 0.5, 1, 2$ and 5 mM. IPTG was added to the tubes, to make sure that *lac* system is fully induced, and glycerol was present as a backup carbon source. The concentration of the LacZ protein was monitored over time.

The cells heritably changed their LacZ expression level within several hundred generations (Figure 13.7). Strikingly, the LacZ protein level reached the predicted optimal level in each tube to a good approximation (Figure 13.8).

Cells growing with no lactose lost their *lac* expression altogether (Figure 13.7). These cells could no longer grow on lactose as the sole nutrient. Cells growing on 0.1 and 0.2 mM lactose reached levels of expression lower than wild-type. Cells growing with 0.5 mM lactose kept close to the wild-type expression level of 60,000 per cell, as predicted.

Cells growing with more than 0.5 mM lactose reached higher levels of expression.

Cells evolving at the highest lactose level, 5 mM, reached the predicted expression of about 20% more LacZ. Then at about 400 generations, they showed an unexpected jump to even higher levels (Figure 13.7). This is a great thing about evolution experiments- they often surprise you. Further experiments showed that these cells gained a mutation that increased M , the upper limit in the cost function.

Analysis of the evolutionary dynamics indicated that the cells reached their optimal, adapted levels in each case by means of a mutation that changed the LacZ protein level. For each lactose concentration, there are on the order of 100 possible mutations that can reach the desired optimal expression level. At zero lactose, there are on the order of 1000 mutations that lose expression altogether. Many of these mutants arise in parallel in each tube and outgrow the wild-type cells, eventually taking over the tube (Good et al., 2017).

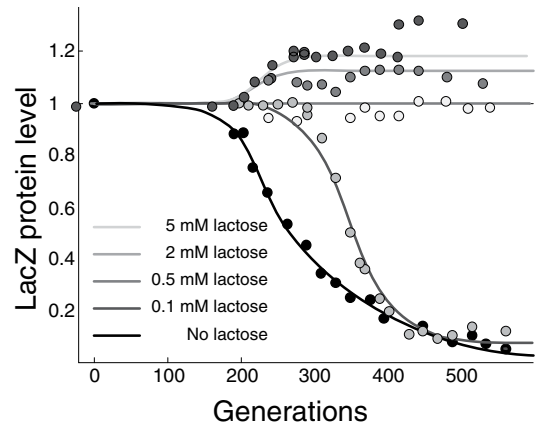


FIGURE 13.7 Adapted from (Dekel and Alon, 2005).

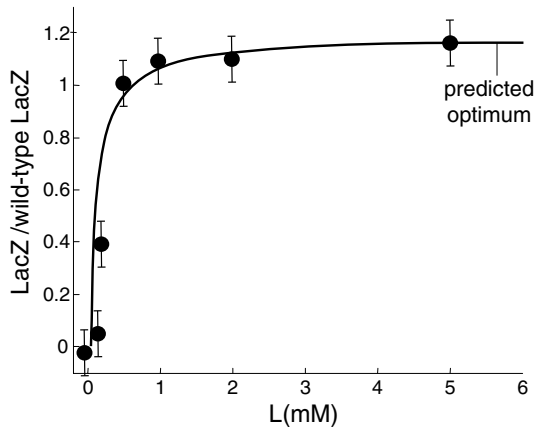


FIGURE 13.8 Adapted from (Dekel and Alon, 2005).

Takeover takes on the order of hundreds of generations for the selection pressures in this experiment, which are on the order of a few percent.

In summary, proteins like LacZ have costs and benefits, which can be used to calculate a fitness function. The fitness function can predict the optimal protein level in a given environment. Cells evolve to this optimal value in evolutionary experiments in several hundred generations. This gives us a sense of the speed and precision in which biological networks can adjust parameters such as protein expression levels.

So far we considered a constant environment. What happens when conditions change with time? We will next treat the principal way that cells deal with changes: gene regulation.

13.3 TO REGULATE OR NOT TO REGULATE? OPTIMAL REGULATION IN CHANGING ENVIRONMENTS

In this section, we ask why some genes are regulated while other genes are expressed continually without regulation. When does it pay to regulate a gene?

Consider an environment that changes over time. Suppose that our gene product Z provides benefit to the cells only in environmental condition C_z . For example, a sugar metabolism enzyme Z is beneficial only when the sugar is available in the environment. The environment displays the condition C_z with probability p , and other conditions, in which Z is superfluous, with probability $1 - p$. The probability p is called the **demand** for Z (as we've seen in [Chapter 7](#)).

To analyze the optimal strategy, let's compare three organisms with different designs for Z regulation. In organism one, protein Z is not regulated and is produced at a constant rate under all conditions. This is known as **constitutive expression**.

This organism constantly produces Z , with a cost c , but gains its benefit b only a fraction p of the time, when Z is in demand, so that it has a fitness

$$F_1 = pb - c \quad (13.3.1)$$

The second organism uses a dedicated regulatory system to produce Z only under the proper conditions. This organism thus saves unneeded production and pays the cost, c , only a fraction p of the time. However, it bears the cost of the regulatory system that can read the environment, calculate and implement the required changes in Z production. It pays this cost of regulation, r , all of the time:

$$F_2 = pb - pc - r \quad (13.3.2)$$

Finally, the third organism lacks the system altogether. It has no benefit or cost, and has a baseline fitness that we will define as zero:

$$F_3 = 0 \quad (13.3.3)$$

Regulation will be selected when organism two has the highest fitness, $F_2 > F_1, F_3$. This leads to the following inequalities:

$$p < 1 - \frac{r}{c}, \quad p > \frac{r}{b - c} \quad (13.3.4)$$

Similarly, the unregulated design in which Z is constitutively expressed will be selected when

$$p > \frac{c}{b}, \quad p > 1 - \frac{r}{c} \quad (13.3.5)$$

These inequalities link a property of the environment, the fraction of time p that condition C_z occurs, to the cost and benefit parameters of protein Z and its regulatory system.

The range of environments in which each of the three designs is optimal is shown in the **phase diagram** of Figure 13.9. Regulation is selected at an intermediate range of demand, p . High demand tends to favor systems that are continually expressed. Constitutive expression of Z is always optimal when $p = 1$, because if Z is always needed, regulation becomes superfluous. When $p = 0$, the protein is never needed and the optimal mechanism is to never express it. The gene is eventually lost. Thus, in constant environments ($p = 0$ or $p = 1$), there is no regulation.

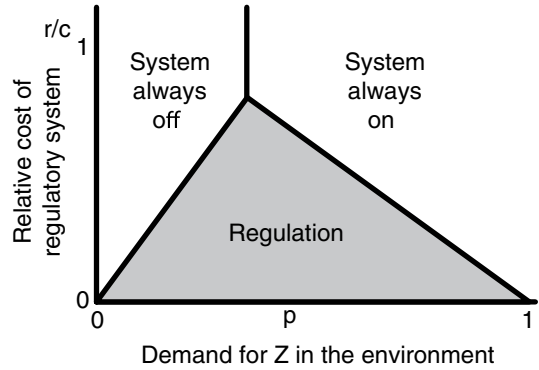


FIGURE 13.9

There exist organisms in nature whose environment is quite constant. For example, a close cousin of *E. coli*, a bacterium called *Buchnera*, lives in symbiosis inside insects called aphids. The aphids supply *Buchnera* with nutrients and stable conditions. In such constant environments, every protein has either $p = 1$ or $p = 0$. These organisms indeed lose virtually all of their regulation systems, such as transcription factors. They also lose most of their genes (keeping only about 600 out of *E. coli*'s 4500). They hold this small set of genes continually expressed. This agrees with the behavior shown in Figure 13.9, on the lines $p = 1$ and $p = 0$.

At the other extreme are bacteria that live in changing and challenging environments such as the soil. These organisms have comparatively large genomes and many regulation systems.⁴ They probably have $0 < p < 1$ for most genes, so that regulation is selected as shown in Figure 13.9.

This analysis assumes that periods C_z in which Z is in demand are long compared to a cell generation, so that we can ignore transients in which Z levels rise or fall when the gene is turned on or off. When environments change rapidly enough, a fourth strategy can be optimal – stochastic gene expression in which a fraction q of the cells express Z constitutively and the rest do not express Z . The cells gamble. This strategy is called **bet hedging**, because if Z is in demand, the cells that happen to produce it win, and if Z is not in demand, the other

⁴ The number of transcription factors tends to increase with the number of genes in the genome as N^a , where N is the number of genes and $a \sim 2$ in bacteria and $a \sim 1.3$ in eukaryotes (Huynen and Van Nimwegen, 1998; Yang et al., 2003). Thus, increasing the number of genes seems to require increasingly elaborate regulation mechanisms with more transcription factors per gene (Maslov et al., 2009).

cells win. The optimal fraction of Z-expressing cells, q , rises with the fraction of time that Z is in demand, p (Kussell and Leibler, 2005).

In summary, regulation makes sense if the environment is sufficiently variable.

As a final example, let's turn to cost–benefit analysis of a gene circuit.

13.4 ENVIRONMENTAL SELECTION OF THE FEEDFORWARD LOOP NETWORK MOTIF

We will now ask under which environmental conditions a particular circuit might be selected. For this purpose, we'll examine a common network motif, the coherent feedforward loop (FFL).

As we saw in [Chapter 3](#), the FFL can perform a basic dynamical function: it can filter out brief input pulses, and respond only to persistent stimuli. Although the FFL is widespread in transcription networks, not every gene is included in a FFL. In *E. coli*, for example, about 40% of the genes regulated by two inputs are regulated by a FFL, whereas the remaining 60% are regulated by a simple two-input design which is not a FFL (the two circuits are shown in [Figure 13.10](#)). It is, therefore, interesting to ask why the FFL is selected in some systems and not in others.

To answer this question, we will do a cost–benefit analysis for the FFL in a given dynamically fluctuating environment (Dekel, Mangan and Alon, 2005). Here, the term environment means the time-dependent profiles of the input signals. We will find conditions that the environment must satisfy in order for the FFL to be selected over a simple-regulation circuit. We will also determine the optimal values of the delay of the FFL circuit as a function of the environment. The full calculations are given in solved Exercises 13.5 through 13.9.

Suppose that the system is presented with a pulse of input S_x of duration D . The impact of this pulse on the cell's growth is given by the integrated fitness over the pulse duration, $\phi(D) = \int_0^D F(t)dt$. This integrated fitness shows that *brief pulses of input signals have a detrimental effect on growth* ([Figure 13.11](#)): they lead to a reduction in growth rate. The reason for this reduction is that when the input pulses are shorter than

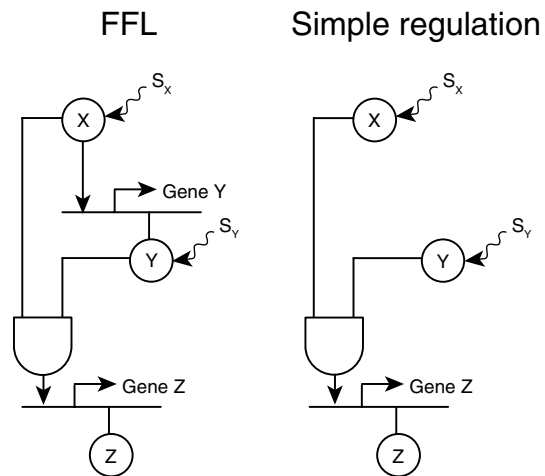


FIGURE 13.10

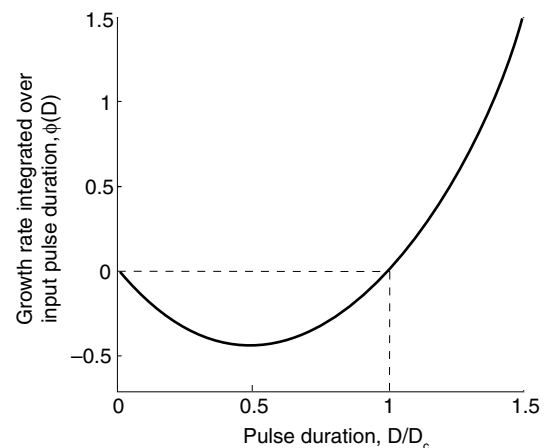


FIGURE 13.11

a critical pulse duration, D_c , protein Z does not have time to build up to levels in which the accumulated benefit exceeds the costs. Very brief pulses thus bear the cost of producing Z without giving it a chance to do any good.

Because fitness is reduced by brief input pulses, a circuit that can avoid responding to brief pulses, but still respond to persistent pulses, can be advantageous. The coherent FFL can perform precisely this type of filtering task. In the coherent type-1 FFL with an AND input function, Z is expressed only at a delay after the signals appear. Therefore, only pulses of input signals longer than the delay time of the FFL will lead to Z expression.

The delay in the FFL, T_{ON} , results from the time it takes for transcription factor Y to accumulate and cross its activation threshold for gene Z . Recall that this delay time is determined by the biochemical parameters of protein Y , namely its degradation rate, maximal level and activation threshold for Z (Chapter 3). The delay can, therefore, be tuned by natural selection to best fit the environment.

The delay in the FFL filters out pulses that are shorter than T_{ON} (Figure 13.12, right panel). This avoids the reduction in growth caused by short pulses. However, the delay also has a disadvantage, because during long pulses, Z is produced only at a delay and misses some of the potential benefit of the pulse (Figure 13.12). This means that in some situations the FFL does more harm than good.

To assess whether the FFL confers a net advantage to the cells relative to simple regulation requires analysis of the full distribution of pulses in the environment, where the probability of pulse of duration D is $P(D)$.

Let us assume for simplicity that the pulses are far apart, so that the system starts each pulse from zero initial Z levels (and Y levels in the FFL). In this case, the average fitness,

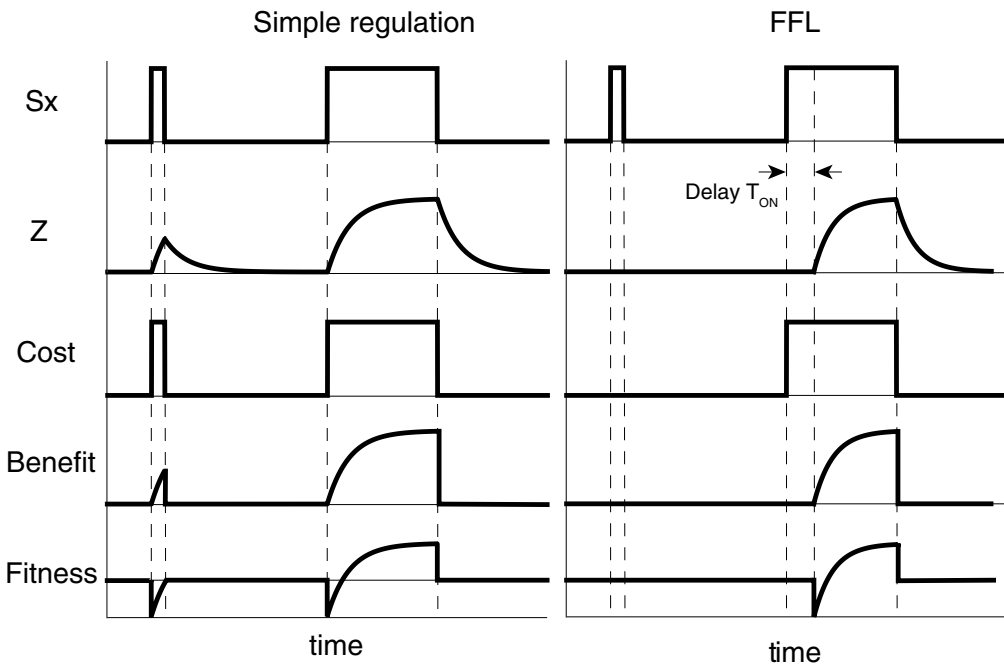


FIGURE 13.12

averaged over many input pulses, can be found by integrating the fitness per pulse over the pulse distribution, $F = \int P(D) \phi(D) dD$. The design with higher average fitness has a selective advantage.

These considerations map out when each circuit has a selective advantage in terms of the environment in which they evolve. This is expressed as relations between certain integrals of the pulse distribution. Exercises 13.7 and 13.8 show that these relations can be solved exactly for certain distributions.

These solutions indicate that the FFL is selected in some environments and not in others. For example, the FFL is never selected over simple regulation in environments with an exponential pulse distribution, $P(D) \sim e^{-D/D_0}$. This distribution is memoryless, and there is no way to predict how long a pulse will last based on its current duration – the FFL is useless. On the other hand, the FFL can be selected in environments with a bimodal pulse distribution, which has a probability p for beneficial long pulses and probability $1 - p$ for short pulses that reduce fitness. The optimal delay for a FFL in such an environment is equal to the duration of the short pulses. This delay filters out the non-beneficial pulses, with minimal negative impact on fitness during long pulses.

The regions in which each type of circuit is selected can be displayed in a phase diagram, [Figure 13.13](#). This diagram shows that the FFL is more fit than simple regulation when deleterious brief pulses are common and the benefit-to-cost ratio is not too high. Simple regulation is superior when brief pulses are rare. When costs exceed benefits, neither circuit is selected. Exercise 13.10 applies this to the case of two sugar systems in *E. coli*.

I hope that this analysis gives a taste for the possibility of using cost–benefit analysis to study the selection of gene circuits and their optimal parameters.

13.5 INVERSE ECOLOGY

We currently have more information about the structure of biological circuits than about the precise environment and ecology in which they evolved. One can imagine an inverse problem – “inverse ecology” – deducing information about the environment based on the observed circuits. This is based on the idea that optimal circuits contain, in a sense, an internal model of the environment. For example, the optimal delay time of the FFL contains information about the distributions of input pulses. Thus, an intriguing goal is to use optimality considerations to connect the molecular details of mechanisms and the environment in which they were selected.

We will continue with these ideas in the next chapter.

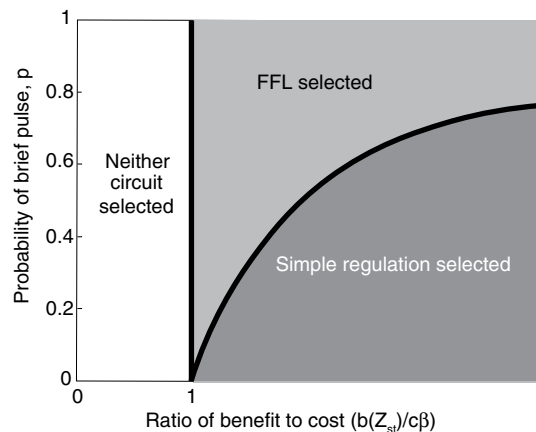


FIGURE 13.13

FURTHER READING

Debates over Optimality

A critique of optimality approaches by Gould and Lewontin calls them “just-so stories” (Gould and Lewontin, 1979). A humorous response by David Queller notes that unlike just-so stories, well-constructed optimality approaches are scientific hypotheses that generate ideas for specific new experiments (Queller, 1995).

Francois Jacob argued that biological systems must be contingent and imperfect because evolution works like a tinkerer, cobbling together existing parts and modifying them (Jacob, 1977). A response (Alon, 2003) notes that one should avoid confusing product with process: while the process of evolution resembles tinkering with available parts, the products of evolution can show design principles similar to engineering, including reuse of a small set of circuit motifs, robustness, optimality and modularity.

Evolution and Optimality

An excellent review is (Parker and Maynard Smith, 1990) “Optimality theory in evolutionary biology.”

(Elena and Lenski, 2003) “Evolution experiments with microorganisms: the dynamics and genetic bases of adaptation.”

(Gause, 1934) “The struggle for existence.”

(Good et al., 2017) “The dynamics of molecular evolution over 60,000 generations.”

Optimality and Evolution in the *lac* System

(Dekel and Alon, 2005) “Optimality and evolutionary tuning of the expression level of a protein.”

(Eames and Kortemme, 2012) “Cost-benefit tradeoffs in engineered *lac* operons.”

(Hartl and Dykhuizen, 1984) “The population genetics of *Escherichia coli*.”

(Poelwijk, De Vos and Tans, 2011) “Tradeoffs and optimality in the evolution of gene regulation.”

(Towbin et al., 2017) “Optimality and sub-optimality in a bacterial growth law.”

Optimality Principles in Metabolism and Animals

(Alexander, 1996) “Optima for animals.”

(Heinrich and Schuster, 1996) “The regulation of cellular systems.”

(Ibarra, Edwards and Palsson, 2002) “*Escherichia coli* K-12 undergoes adaptive evolution to achieve in silico predicted optimal growth.”

(Meléndez-Hevia and Isidoro, 1985) “The game of the pentose phosphate cycle.”

Selection of the FFL Network Motif

(Dekel, Mangan and Alon, 2005) “Environmental selection of the feed-forward loop circuit in gene-regulation networks.”

EXERCISES

- 13.1 *Limiting substrate*: Protein X is an enzyme that acts on a substrate whose concentration is L . Calculate the fitness function $F(X, L)$ assuming linear cost, $c = c_0 X$ and a benefit $b(X, L) = B_0 L X / (X + K)$, appropriate for cases where the substrate, rather than the enzyme X , is limiting. Calculate the optimal enzyme level X_{opt} as a function of L and K .

13.2 *Minimal substrate for selection:* For Exercise 13.1, what is the minimal substrate level L_c required for maintenance of the gene for X by the organism? When is the gene lost? Explain.

13.3 *Optimal expression of a subunit:*

- Multiple units of protein X act together in a multi-unit complex. The benefit is a Hill function, $b(X) = b_0 X^n / (K^n + X^n)$, and the cost function is linear in X . Estimate the optimal protein level. Explain.
- Protein X brings benefit to the cell only when its concentration exceeds X_0 , so that $b(X) = \theta(X > X_0)$, where θ is the step function. What is the optimal expression level of X ?

13.4 *Cost function:*

- Derive the cost function in Equation 13.2.3, based on a limiting resource R , such that the growth rate is equal to $F = F_0 R / (K + R)$. Each unit of protein Z reduces R by a small amount ε .
- In bacterial cells, the resource R often increases as the growth rate decreases (Korem et al., 2018). For example, the fraction of free ribosomes increases as growth rate slows, because at high growth rates the ribosomes are mostly engaged in making new ribosomes. This effect can be added to the model to find similar cost functions at the low to intermediate expression levels of Z relevant to the experiments described in this chapter. Assume that $R = m/F$, where F is the growth rate and m is a parameter. Derive the cost function in this case.

Solution for (a):

The burden of Z production can be described as a reduction in the internal resource R , such that each unit of protein Z reduces the resource by a small amount ε , so that R goes to $R - \varepsilon Z$. Hence, the cost, defined as the relative reduction in growth rate, is as in Equation 13.2.3:

$$c(Z) = \frac{F(0) - F(Z)}{F(0)} = c_0 Z / \left(1 - \frac{Z}{M}\right)$$

where the initial reduction per subunit of Z is $c_0 = K\varepsilon / (K + R)$ and the parameter $M = (K + R)/\varepsilon$. Note that the cost can never diverge, because when Z depletes all of the resource R , that is, when $Z = R/\varepsilon$, one finds $F(Z) = 0$ and the cost is equal to $c = 1$.

13.5 *Brief input pulses have a negative effect on growth:* Exercises 13.5 through 13.9 analyze the selection conditions of the FFL and simple-regulation circuits. Consider a simple gene regulation mechanism with two-input transcription factors X and Y that regulate gene Z (i.e., regulation without the third edge $X \rightarrow Y$ in the FFL). The two inputs are

both needed for Z expression, so that this may be described as a simple-regulation circuit with an AND input function (Figure 13.10). In this design, Z is produced at a constant rate β in the presence of both signals S_X and S_Y , and production rate is otherwise zero. The benefit of Z occurs only while the input S_X is present. Show that pulses of the signal S_X that are very short lead to a reduction in fitness. Only pulses that are long enough lead to a net growth advantage.

Solution:

We'll employ cost-benefit analysis to describe the effects of Z on the growth rate of the cells. We assume that Z production entails a reduction in growth rate $C = -c\beta$, where β is the rate of production of Z and c is the reduction in growth rate per Z molecule produced. On the other hand, the action of protein Z conveys a growth benefit, $b(Z)$. The overall effect of Z on the growth rate is a sum of the cost and benefit terms:

$$F = -c\beta + b(Z) \quad (\text{P13.1})$$

For simplicity, we neglect the nonlinear cost effects described in Section 13.2. Also, note that typically, the costs for the production of the transcription factors X and Y are negligible compared to the production cost of enzyme Z , since they are typically produced in far fewer copies per cell than enzymes (Nguyen et al., 1989; Ghaemmamghami et al., 2003). If Y costs are not negligible, the advantage of FFL over simple regulation increases, because the FFL prevents unneeded Y production.

Now consider a pulse of input signals, in which S_X is present at saturating levels for a pulse of duration D in the presence of S_Y . The growth rate of cells with simple regulation, integrated over time D , is given by:

$$\phi(D) = \int_0^D F(t) dt = -c\beta D + \int_0^D b(Z) dt \quad (\text{P13.2})$$

When the pulse begins, protein Z begins to be produced at rate β , and removed at rate α . The dynamics of Z concentration are given by the dynamical equation we discussed in Chapter 1:

$$\frac{dZ}{dt} = \beta - \alpha Z$$

resulting in the familiar exponential approach to steady state

$$Z_{st} = \frac{\beta}{\alpha} \quad (\text{P13.3})$$

$$Z(t) = Z_{st}(1 - e^{-\alpha t}) \quad (\text{P13.4})$$

For long pulses ($D\alpha \gg 1$), the protein concentration Z is saturated at its steady-state value $Z = Z_{st}$. Protein Z has a net positive effect on cell growth:

$$\phi(D) \approx -c\beta D + b(Z_{st})D > 0 \quad (\text{P13.5})$$

provided that the benefit of Z exceeds its production costs:

$$b(Z_{st}) > c\beta \quad (\text{P13.6})$$

Short pulses, however, can have a deleterious effect on growth. To see this, consider short pulses such that $D\alpha \ll 1$. During the short pulse, the concentration of Z rises approximately linearly with time (as we saw in Equation 1.4.7), with a slope equal to the production rate

$$Z(t) \sim \beta t \quad (\text{P13.7})$$

Since Z cannot reach high levels during the short induction pulse, we can use a series expansion of the benefit function $b(Z) \approx b'Z$, where $b' = db/dZ$ at $Z = 0$. Using this in Equation P13.2, we find that the integrated growth rate is a quadratic function of the duration of the pulse, D (plotted in [Figure 13.11](#)):

$$\phi(D) = \int_0^D (-c\beta + b'\beta t) dt = -c\beta D + \frac{b'\beta D^2}{2} \quad (\text{P13.8})$$

Importantly, the expression of Z causes a reduction in growth ($\phi(D) < 0$) for pulses shorter than a critical pulse duration, D_c , found by solving $\phi(D_c) = 0$:

$$D_c = 2c/b' \quad (\text{P13.9})$$

Pulses with $D = D_c$ are at the break-even point: cost exactly equals the benefit. Only pulses longer than D_c give a net benefit to the cells. Thus, simple regulation leads to growth reduction in environments that have mainly brief pulses, even though Z confers a net advantage for sufficiently long input pulses ([Figure 13.12](#)).

- 13.6 *Conditions for selection of FFL over simple regulation:* Exercise 13.5 showed that expression of Z in response to brief input pulses reduces fitness. Hence, a circuit that filters out brief pulses, and allows responses only to persistent pulses, can be advantageous. The coherent FFL can perform this filtering task. In the coherent FFL, Z is expressed at a delay T_{ON} after the signals appear. Thus, only pulses of input signals

longer than T_{ON} lead to Z expression. However, the filtering of short pulses has a disadvantage, because during long pulses, Z is produced only at a delay and misses some of the potential benefit of the pulse (Figure 13.12). To assess whether the FFL confers a net advantage to the cells, relative to simple regulation, requires analysis of the distribution of pulses in the environment. The environment of the cell can be characterized by the probability distribution of the duration of input pulses, $P(D)$. Assume that the pulses are far apart, so that the system starts each pulse from zero initial Z levels (and Y levels in the case of the FFL). In this case, the overall fitness, averaged over many cell generations, can be found by integrating the fitness per pulse over the pulse distribution. Find conditions for the selection of the FFL over simple regulation.

Solution:

For simple-regulation circuits, the integrated fitness is an integral over all possible pulses, times the fitness per pulse $\phi(D)$:

$$\Phi_{\text{simple}} = \int_0^{\infty} P(D)\phi(D)dD \quad (\text{P13.10})$$

For FFL circuits, production starts after a delay T_{ON} . Pulses shorter than T_{ON} result in no Z production and hence $\phi(D < T_{ON}) = 0$. Long pulses begin to be utilized only after the delay T_{ON} , so that their duration is effectively $D - T_{ON}$ (Figure 13.12), resulting in a contribution in the integral only from pulses longer than T_{ON} :

$$\Phi_{\text{FFL}} = \int_{T_{ON}}^{\infty} P(D)\phi(D - T_{ON})dD \quad (\text{P13.11})$$

Note that the simple-regulation case is equivalent to a FFL with $T_{ON} = 0$. The resulting condition for selection of FFL over simple regulation is when its averaged fitness exceeds that of simple circuits and is positive:

$$\Phi_{\text{FFL}} > \Phi_{\text{simple}} \quad \text{and} \quad \Phi_{\text{FFL}} > 0 \quad (\text{P13.12})$$

Simple regulation is selected when its integrated fitness exceeds that of the FFL

$$\Phi_{\text{simple}} > \Phi_{\text{FFL}} \quad \text{and} \quad \Phi_{\text{simple}} > 0 \quad (\text{P13.13})$$

Neither circuit is selected otherwise ($\Phi_{\text{FFL}} < 0$, $\Phi_{\text{simple}} < 0$). For the purpose of this comparison, the FFL is chosen to have the optimal value for T_{ON} that maximizes Φ_{FFL} , because natural selection can tune this parameter to best adapt to the environment.

- 13.7 *The FFL is not selected in the case of exponential pulse distributions:* Analyze the average fitness of the FFL and simple regulation in an environment in which pulses have a constant probability per unit time to end. Such environments have an exponential pulse distribution:

$$P(D) = D_0^{-1} e^{-D/D_0} \quad (\text{P13.14})$$

Solution:

Using Equations P13.10 and P13.11, we find that

$$\begin{aligned} \Phi_{\text{FFL}} &= \int_{T_{\text{ON}}}^{\infty} P(D) \phi(D - T_{\text{ON}}) dD = e^{-\frac{T_{\text{ON}}}{D_0}} \int_0^{\infty} D_0^{-1} e^{-\frac{D}{D_0}} \phi(D) dD \\ &= e^{-\frac{T_{\text{ON}}}{D_0}} \Phi_{\text{simple}} < \Phi_{\text{simple}} \end{aligned} \quad (\text{P13.15})$$

Thus, the FFL is never selected since $\Phi_{\text{FFL}} < \Phi_{\text{simple}}$. An intuitive reason is related to the fact that exponential distributions are memoryless. Knowledge that a pulse has lasted for time t does not help us to predict how long it will continue to last. The FFL, which effectively reduces the pulse duration by a delay T_{ON} , confers no advantage relative to simple regulation.

- 13.8 *The FFL can be selected in bimodal environments with long and short pulses:* Consider an environment that has two kinds of pulses. A pulse can have either a brief duration, $D_1 \ll D_c$, with probability p , or a long duration, $D_2 \gg 1/\alpha$, with probability $1 - p$. Analyze the conditions for selection of FFL and simple regulation as a function of p and the benefit-to-cost ratio of protein Z (Figure 13.13).

Solution:

The brief pulses D_1 are detrimental, since they are shorter than the critical pulse width at which Z reaches the break-even point, D_c (Figure 12.11). In contrast, the long pulses D_2 are beneficial and have a benefit of approximately (applying Equation P13.5)

$$\phi(D_2) = -c\beta D_2 + b(Z_{st})D_2 > 0 \quad (\text{P13.16})$$

In this case, it is easy to calculate the optimal delay in the FFL: the optimal delay is $T_{\text{ON}} = D_1$, because this delay blocks the short pulses precisely; a longer delay would only reduce the benefit of the long pulses. The condition for selection of FFL over simple regulation, found by solving Equations P13.10 and P13.11, yields

$$\Phi_{\text{simple}} = (1 - p)(b(Z_{st}) - c\beta)D_2 - pc\beta D_1$$

$$\Phi_{\text{FFL}} = (1 - p)(b(Z_{st}) - c\beta)(D_2 - D_1)$$

This shows that the FFL is more fit when the probability of short pulses exceeds a factor related to the ratio of cost to benefit of Z ,

$$p > 1 - c\beta / b(Z_{st}) \quad (\text{P13.17})$$

The phase diagram for selection is shown in [Figure 13.13](#). When the ratio of benefit to cost, $b(Z_{st})/c\beta$, is small, neither circuit is selected (costs outweigh benefits). At large relative benefits, the FFL is selected if brief pulses are common enough – that is, if p is large enough (Equation P13.17). If brief pulses are rare, simple regulation is selected. At a given p , the higher the benefit-to-cost ratio, the more likely the selection of simple-regulation circuits.

- 13.9 *FFL selection in E. coli sugar systems*: In this exercise, we will apply, in a qualitative way, the results of Exercise 13.8 to the case of two sugar systems in *E. coli*. Why is the FFL selected in the arabinose utilization system (*ara* system discussed in [Chapter 3](#)), whereas simple regulation is selected in the lactose system (*lac* system)?

Solution:

Both *ara* and *lac* systems share the same transcription activator, $X = \text{CRP}$, whose input signal is $S_X = cAMP$, a signaling molecule produced by the cell upon glucose starvation. Thus, both systems have the same S_X pulse distribution. However, the systems differ in their benefit-to-cost ratio, $b(Z_{st})/c\beta$. The benefit per lactose molecule, which LacZ splits into glucose + galactose, is greater than the benefit per arabinose molecule (approximately 70 ATPs per lactose molecule vs. approximately 30 ATPs per arabinose molecule). In addition to its smaller benefit, the cost of the *ara* system may be larger than the cost of the *lac* system, because there are at least seven highly expressed Ara proteins (the metabolic enzymes AraB, AraA and AraD, and the pumps AraE and AraFGH), compared to only three highly expressed *lac* proteins (LacZ, LacY and LacA). Thus, the parameter $b(Z_{st})/c\beta$ for the *ara* system may be more to the left in [Figure 13.13](#) relative to the *lac* system, favoring selection of FFL in the former. The delay in the FFL can be tuned by natural selection. As mentioned in [Chapter 3](#), the delay in the *ara* system appears to be on the timescale of the deleterious short pulses in the environment.

- 13.10 *Cascades vs. FFLs*: Repeat the calculations of Exercises 13.6 and 13.7 for a cascade $X \rightarrow Y \rightarrow Z$. Show that cascades are never more optimal than FFLs for environments with pulses of input signals. Explain this result.
- 13.11 *Population dynamics and mutational targets*: Cells grow exponentially at rate F_0 , from an initial number $N(0)$.
- A mutant arises with probability p per division, and grows at a faster rate F_1 . What is the ratio of mutant to wild-type cells as a function of time?
 - The experiment of Dekel and Alon (2005) used 10 mL of growth medium to grow the bacteria. Cells grow exponentially until they stop growing (reach

stationary phase) at a density of about 10^9 cells/mL. How many of the above-mentioned mutants are present in the initial tube in the experiment when it reaches stationary phase?

- c. The **mutational target** is the effective number of point mutations that confer a particular feature. The probability for a point mutation in the genome (a change of one DNA letter) is about $p_0 = 10^{-9}$ per cell division. Thus, if the effective probability for a given mutant phenotype p is K times more than p_0 , we say the mutational target is K . Plot the mutant cell number dynamics for $K = 100$ and $K = 1000$ for the conditions of the experiment, with $F_1/F_0 = 1.05$ signifying a 5% growth advantage. Which values of K correspond more closely to the data for $L = 0$ and $L = 0.2$ in [Figure 13.7](#)? Explain.
- 13.12 *Flagella phases*: The bacterium *Salmonella typhimurium* has a special design for a gene that makes the flagellum. The DNA for this gene can be flipped by special enzymes that cut and paste the gene in a reversed orientation. The flagellar protein produced by the flipped gene has a different structure but can also make a flagellum. The flipping occurs stochastically, so that only some cells have a flipped gene. This means that a population of cells can have two different kinds of flagella.
- a. Explain why this strategy can be useful for *Salmonella*, which is a pathogen that needs to evade the immune system that can produce antibodies against flagella proteins.
 - b. Suppose that *Salmonella* spends a fraction p of the time exposed to the immune system. The immune system in a given infected host recognizes only one of the two phases, but it is not possible to know which. The cost of the reversing enzyme is r and occurs all of the time. When is the gene-reversing strategy selectable based on cost–benefit analysis?
- 13.13 *Randomizing forces*: Discuss which factors can prevent natural selection from reaching the fitness peak. Discuss the roles of population size, mutation rate, the shape of the fitness function, and physical and chemical constraints.
- 13.14 *The cost of noise*:
- a. A protein Z has a fitness function $F(Z)$. Cells express Z at its optimal level plus noise N which has a mean of zero and standard deviation of σ . Estimate the mean reduction in fitness due to this noise. *Hint*: Use a Taylor series expansion of $F(Z)$ to second order around the maximum.
 - b. For the *lac* proteins, use the fitness function in the main text to compute the cost of the noise. Use a 10% noise amplitude, namely $\sigma = 0.1Z_{WT}$.
- 13.15 *Paradox of the plankton*: If two organisms in a test tube share the same resource, only one can survive (the one with higher growth rate F), a principle known as **competitive exclusion**. In contrast, a cup of sea water contains hundreds of different coexisting species of plankton that compete over a much smaller number of resources. How might one resolve this “paradox of the plankton?”

BIBLIOGRAPHY

- Alexander, R.M. 1996. *Optima for animals*. Princeton University Press.
- Alon, U. 2003. 'Biological networks: the tinkerer as an engineer', *Science*, 301(5641), pp. 1866–1867. doi: 10.1126/science.1089072.
- Balagaddé, F.K. et al. 2005. 'Microbiology: long-term monitoring of bacteria undergoing programmed population control in a microchemostat', *Science*, 309(5731), pp. 137–140. doi: 10.1126/science.1109173.
- Dekel, E. and Alon, U. 2005. 'Optimality and evolutionary tuning of the expression level of a protein', *Nature*, 436(7050), pp. 588–592. doi: 10.1038/nature03842.
- Dekel, E., Mangan, S. and Alon, U. 2005. 'Environmental selection of the feed-forward loop circuit in gene-regulation networks', *Physical Biology*. IOP Publishing, 2(2), pp. 81–88. doi: 10.1088/1478-3975/2/2/001.
- Eames, M. and Kortemme, T. 2012. 'Cost-benefit tradeoffs in engineered lac operons', *Science*, 336(6083), pp. 911–915. doi: 10.1126/science.1219083.
- Elena, S.F. and Lenski, R.E. 2003. 'Evolution experiments with microorganisms: the dynamics and genetic bases of adaptation', *Nature Reviews Genetics*. Nature Publishing Group, 4(6), pp. 457–469. doi: 10.1038/nrg1088.
- Gause, G. F. 1934. *The struggle for existence*. Dover Publications.
- Ghaemmaghami, S. et al. 2003. 'Global analysis of protein expression in yeast', *Nature*. doi: 10.1038/nature02046.
- Good, B.H. et al. 2017. 'The dynamics of molecular evolution over 60,000 generations', *Nature*, 425(6959), pp. 737–741. doi: 10.1038/nature24287.
- Gould, S.J. and Lewontin, R.C. 1979. 'The spandrels of San Marco and the Panglossian paradigm: a critique of the adaptationist programme', *Proceedings of the Royal Society of London - Biological Sciences*, 205(1161), pp. 581–598. doi: 10.1098/rspb.1979.0086.
- Hartl, D.L. and Dykhuizen, D.E. 1984. The population genetics of *Escherichia coli*. Available at: www.annualreviews.org.
- Heinrich, R. and Schuster, S. 1996. *The regulation of cellular systems*. Springer, US.
- Huynen, M.A. and Van Nimwegen, E. 1998. 'The frequency distribution of gene family sizes in complete genomes', *Molecular Biology and Evolution*, 15(5), pp. 583–589. doi: 10.1093/oxfordjournals.molbev.a025959.
- Ibarra, R.U., Edwards, J.S. and Palsson, B.O. 2002. '*Escherichia coli* K-12 undergoes adaptive evolution to achieve in silico predicted optimal growth', *Nature*, 420(6912), pp. 186–189. doi: 10.1038/nature01149.
- Jacob, F. 1977. 'Evolution and tinkering', *Science*, 196(4295), pp. 1161–1166. doi: 10.1126/science.860134.
- Kussell, E. and Leibler, S. 2005. 'Phenotypic diversity, population growth, and information in fluctuating environments', *Science*, 309(5743), pp. 2075–2078. doi: 10.1126/science.1114383.
- Korem Kohanim, Y. et al. 2018. 'A bacterial growth law out of steady state.' *Cell Reports* 23(10), pp. 2891–2900. doi: 10.1016/j.celrep.2018.05.007.
- Maslov, S. et al. 2009. 'Toolbox model of evolution of prokaryotic metabolic networks and their regulation', *Proceedings of the National Academy of Sciences*, 106(24), pp. 9743–9748. doi: 10.1073/pnas.0903206106.
- Meléndez-Hevia, E. and Isidoro, A. 1985. 'The game of the pentose phosphate cycle', *Journal of Theoretical Biology*. doi: 10.1016/S0022-5193(85)80220-4.
- Nguyen, T.N.M. et al. 1989. 'Effects of carriage and expression of the Tn10 tetracycline-resistance operon on the fitness of *Escherichia coli* K12', *Molecular Biology and Evolution*, 6(3), pp. 213–225. doi: 10.1093/oxfordjournals.molbev.a040545.

- Novick, A. and Weiner, M. 1957. 'Enzyme induction as an all-or-none phenomenon', *Proceedings of the National Academy of Sciences*, 43(7), pp. 553–566. doi: 10.1073/pnas.43.7.553.
- Parker, G.A. and Maynard Smith, J. 1990. 'Optimality theory in evolutionary biology', *Nature*, 348(6296), pp. 27–33. doi: 10.1038/348027a0.
- Poelwijk, F.J., De Vos, M.G.J. and Tans, S.J. 2011. 'Tradeoffs and optimality in the evolution of gene regulation', *Cell*, 146(3), pp. 462–470. doi: 10.1016/j.cell.2011.06.035.
- Queller, D.C. 1995. 'The Spaniels of St. Marx and the Panglossian Paradox: A critique of a rhetorical programme', *The Quarterly Review of Biology*. University of Chicago Press, 70(4), pp. 485–489. doi: 10.1086/419174.
- Ronen, M. and Botstein, D. 2006. 'Transcriptional response of steady-state yeast cultures to transient perturbations in carbon source', *Proceedings of the National Academy of Sciences*, 103(2), pp. 389–394. doi: 10.1073/pnas.0509978103.
- Simon, H.A. 1997. *The sciences of the artificial* (3rd Ed.), Computers & Mathematics with Applications. doi: 10.1016/S0898-1221(97)82941-0.
- Towbin, B.D. et al. 2017. 'Optimality and sub-optimality in a bacterial growth law', *Nature Communications*. Nature Publishing Group, 8, p. 14123. doi: 10.1038/ncomms14123.
- Yang, E. et al. 2003. 'Decay rates of human mRNAs: correlation with functional characteristics and sequence attributes', *Genome Research*, 13(8), pp. 1863–1872. doi: 10.1101/gr.1272403.

Multi-Objective Optimality in Biology

14.1 INTRODUCTION

So far, we considered evolution toward a single objective, such as maximizing the growth rate of bacteria. A single objective is appropriate for carefully controlled experiments. In nature, however, biological systems usually have multiple objectives. Bacteria, for example, need to grow quickly and also need to survive stresses. We will call such biological objectives ‘tasks’. Multiple tasks lead to a fundamental **trade-off**: no design can be optimal at all tasks at once. There is no animal that can fly like an eagle, swim like a dolphin and run like a cheetah.

In this chapter, we will ask how evolution can optimize in the presence of multiple tasks. This is the art of optimal trade-offs. We will see that multiple tasks lead to simple geometrical patterns in biological data. These patterns can help us to understand the evolutionary trade-offs at play.

14.2 THE FITNESS LANDSCAPE PICTURE FOR A SINGLE TASK

Let’s start with the classical framework for evolutionary theory, the fitness landscape picture. Consider the evolution of a bird’s beak. The **genotype**, DNA, leads to the **phenotype**, the shape of the beak, which leads to fitness, by eating seeds. The better the beak is at eating seeds, the more the bird will have viable chicks and grand-chicks on average, and the higher its fitness. It will pass its genes to the next generation. Thus

$$\text{genotype} \rightarrow \text{phenotype} \rightarrow \text{fitness} \quad (14.2.1)$$

Now suppose we take a ruler and measure the beak length, width, depth, curvature, and so on. These are called beak **traits**, T_i . Each beak phenotype can be represented as a point in a space whose axes are the traits, called **trait space**. We can (in principle) plot the fitness

of each beak in trait space, $F(\vec{T})$, where $\vec{T}$ is the vector of traits, resulting in a **fitness landscape**.

The fitness landscape is a multi-dimensional version of the fitness function from the previous chapter. The fitness landscape can have hills and valleys. Figure 14.1 shows the contours of a fitness landscape as a function of two traits shaped as a circular hill. Natural selection will tend to converge to the summit of the fitness landscape, to the phenotype that maximizes fitness. Phenotypes will perhaps form a cloud around the peak due to randomizing forces.

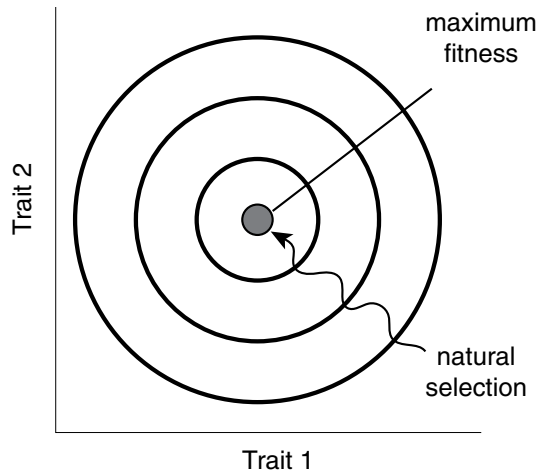
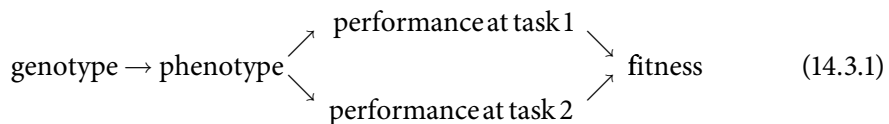


FIGURE 14.1

14.3 MULTIPLE TASKS ARE CHARACTERIZED BY PERFORMANCE FUNCTIONS

But what if the beak needs to do two different tasks that both contribute to fitness: to crack seeds and to pick pollen from flowers? You can't be optimal at two tasks with one beak. Cracking seeds require a beak shaped like a pair of pliers, whereas picking pollen requires a beak shaped like a pair of pincers (Figure 14.2). In this case, we need to modify the genotype $\rightarrow$ phenotype $\rightarrow$ fitness picture, and add in the notion of **performances** at the two tasks (Arnold, 1983). The genotype determines the traits of the phenotype, $\vec{T}$, which determine performance at task 1 (cracking seeds) $P_1(\vec{T})$ and performance at task 2 (picking pollen) $P_2(\vec{T})$. Fitness is a function of these two performance functions: $F = F(P_1(\vec{T}), P_2(\vec{T}))$



Notably, fitness is an **increasing** function of the two performances ($dF/dP_i > 0$): make a beak better at both tasks and fitness is sure to increase.

The precise form of the fitness function depends on the **niche**. In some niches one task is more important than the other, so that $F(P_1, P_2)$ gives more weight to that particular task. In other niches the other task is more important. The precise shape



FIGURE 14.2

of the fitness function F in each niche is usually not known. But, as we will see below, conclusions can be reached that do not depend on knowing the form of F .

14.4 PARETO OPTIMALITY IN PERFORMANCE SPACE

So which beak shapes will evolve under the trade-off between these two tasks? Engineers routinely need to solve this type of problem. They use an approach called **Pareto optimality**. Suppose that you want to design a car. The design specifications require performance at two tasks, say acceleration (time from 0 to 100 km/h) and fuel economy (say km/liter). You take all possible designs, and plot them according to their performance at the two tasks. In this plot, whose axes are the two performances, each design is a point in **performance space**.

Now consider a design B . If there exists a design A that is better than B at both tasks (has higher performance at both tasks), we erase design B (Figure 14.3). We say that B is dominated by A . Erasing in this way all points which are dominated by another point, we remain with the **Pareto front** (Figure 14.4). It is the set of designs that cannot be simultaneously improved at both tasks. This front is what engineers care about.

Which design from the front you choose is based on the market niche of the car: a family car requires better economy at expense of acceleration, and a sports car requires better acceleration at the expense of economy (Figure 14.5).

This approach can be used to analyze biological circuits. For example, Adler et al. (2017) analyzed the performance space of circuits that have fold-change detection (FCD), as mentioned in Chapter 10. They chose performance functions such as large response amplitude and fast response time, and considered circuits with a minimal number of interaction arrows. The Pareto front included only a handful of circuits. Many other minimal circuits did worse on at least one task, and were therefore not on the front. Among the few circuits on the Pareto front were the two circuits observed in biological systems, the I1-FFL and the nonlinear integral feedback loop discussed in Chapter 10. Thus, the Pareto front offers a way to understand why only a few circuit designs are found again and again in different systems, and most other circuits are not found.

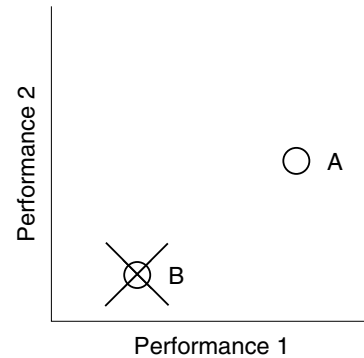


FIGURE 14.3

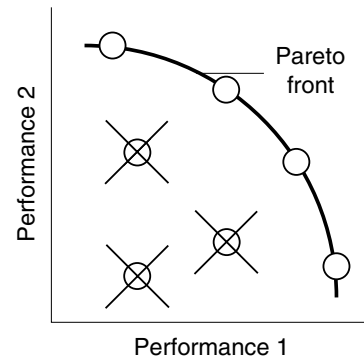


FIGURE 14.4

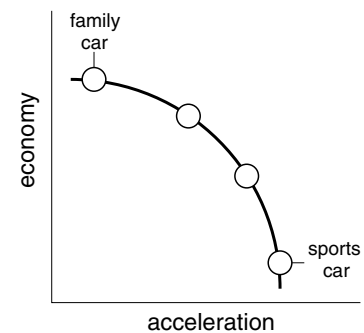


FIGURE 14.5

The Pareto optimality idea can also be used, in a lighthearted way, to help us understand how to choose scientific problems (Alon, 2009). The axes are how feasible and interesting the problem is.

14.5 PARETO OPTIMALITY IN TRAIT SPACE LEADS TO SIMPLE PATTERNS

This standard use of Pareto optimality requires us to define in advance what the tasks are. However, in many cases in biology *we don't know what the tasks are in advance*. We can make an educated guess, but we can't be sure. Thus, we can not directly use performance space to do Pareto optimality, because we don't know what tasks to compare. Even if we did, we cannot evaluate the performance of each phenotype at each task.

Remarkably, we can still make progress, using an approach called **Pareto task inference**, or **ParTI** (Shoval et al., 2012, Hart et al., 2015). We simply plot the data in *trait space*, using all the traits that we can measure. The axes are the traits, and each phenotype is a point in this space. For example, each beak is a point in a space of traits such as beak width, depth, curvature, and so on.

We will now see that evolution under several tasks makes the data show particular geometric shapes. These shapes can help us discern the number of tasks, and even what the tasks might be. Thus, we solve the inverse problem of Pareto optimality, by inferring the tasks from the data. For example, when there are two tasks at play, the data will fall on a line segment (or sometimes on a slightly curved segment as discussed below). The two ends of the segment give us clues about what the tasks are.

To see where this line-segment geometry comes from, let's imagine that each of the two tasks has a performance function, $P_1(\vec{T})$ and $P_2(\vec{T})$. The contours of these performance functions are plotted in trait space in Figure 14.6. The peak of each performance function is a special phenotype, called the **archetype**. The archetype is the phenotype (trait combination) that is best at the task. If there was only that one task, evolution would converge to the archetype. Archetype 1 is the best beak for seeds, and archetype 2 is the best beak for pollen. Performance drops with distance from the archetype.

We want to find the beak shape that maximizes fitness, $F(P_1, P_2)$, where F can be any increasing function. The surprise

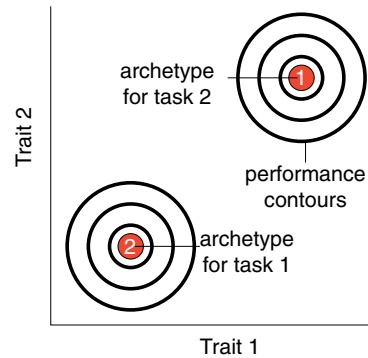


FIGURE 14.6

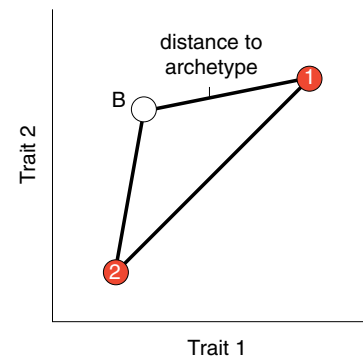


FIGURE 14.7

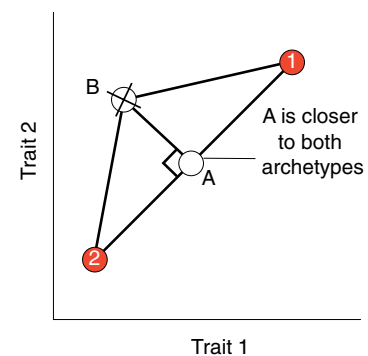


FIGURE 14.8

is that, no matter what F is, one can prove that, under certain assumptions, the optimal solution must fall on the *line segment that connects the two archetypes*. The assumption of the theorem is that the performance functions drop with a metric distance from the archetype (for example, with Euclidean distance in Figure 14.6). But for some reason, as we will see when we look at data, the theorem seems to work well even in cases where it has no right to.

To understand why phenotypes fall on the line segment between the two archetypes, consider a phenotype B that is *not* on the line segment (Figure 14.7). The performance of B in each task is determined by its distances to the archetypes. There is a phenotype A on the line segment connecting the archetypes which is closer to both archetypes (by the triangle inequality) as shown in Figure 14.8. Phenotype A , therefore, has better performance at both tasks, and, therefore, higher fitness than B . In an evolutionary race, A would win, and we can, therefore, erase B .

Now, there was nothing special about point B , so we can erase all of the points and remain with the line segment between the two archetypes (Figure 14.9). This is the set of phenotypes that cannot be improved at both tasks at once – the **Pareto front** (plotted in trait space, not in performance space).

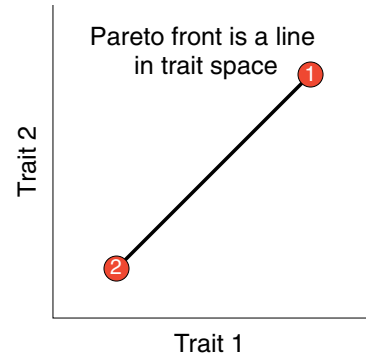


FIGURE 14.9

14.6 TWO TASKS LEAD TO A LINE SEGMENT, THREE TASKS TO A TRIANGLE, FOUR TO A TETRAHEDRON

Thus, a trade-off between two tasks predicts phenotypes on a line segment in trait space. Suppose we measure many beak traits, say 100 traits, making a 100-dimensional trait space. The beaks will still fall on the line segment between the two archetypes in this 100-dimensional trait space (Figure 14.10a). Measuring any two of these traits will still show a line, because the projection of the line on any plane is a line (thin line in Figure 14.10a). Thus, it is not too important which traits you measure, as long as they have to do with the same tasks.

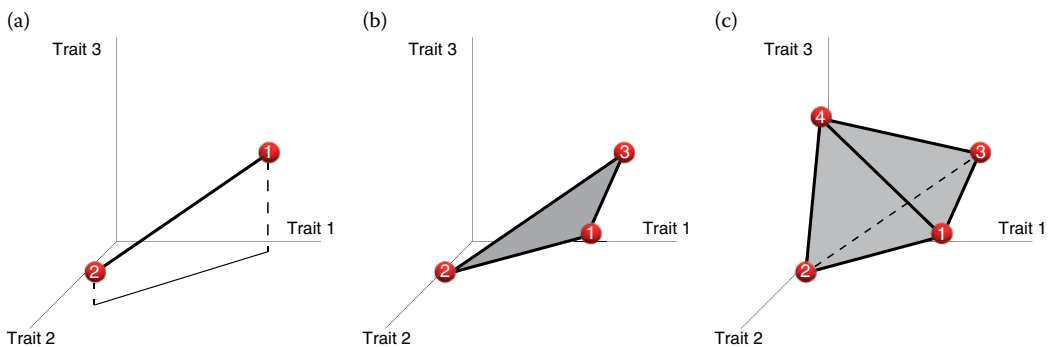


FIGURE 14.10

If there are three tasks, we expect the optimal phenotypes to fall inside a triangle, whose three vertices are the three archetypes (Figure 14.10b). If there are four tasks, the phenotypes will fall inside a tetrahedron (Figure 14.10c). In the case of very many traits, we can use dimensionality reduction methods such as principal component analysis (PCA) to visualize these shapes.

In general, a trade-off between k tasks will result in a Pareto front shaped as a **polytope** with k vertices (a polytope is the generalization of a polygon or polyhedron to any dimension). Each vertex is an archetype for one of the tasks. A proof is given in Solved Exercise 14.1.

The key idea is that fitness is not just any function of traits $F(\vec{T})$, it is an increasing function of k performance functions of the traits $F(P_1(\vec{T}), P_2(\vec{T}), \dots, P_k(\vec{T}))$. The maxima of these performance functions define k points in trait space, which is a polytope. The maximum of F needs to be close to these k points, and hence inside the polytope.

If you make a nonlinear transformation of the traits (e.g., measure T^2 instead of T), the polytopes will be deformed (Figure 14.11). Deformed shapes can also result from other situations, such as a non-metric decline of performance functions (Exercise 14.6). Even if the shapes are deformed, they still have sharp corners at the archetypes.

The neat use of this approach is to *discover what the tasks are directly from biological data*. The sharp corners (vertices) of the polytopes can help us infer the tasks: The phenotypes closest to a vertex should be **specialists** at something, and that something gives clues to what the task might be (Figure 14.12). Phenotypes near the center of the polytope should be **generalists**. This is the ParTI approach of inferring the tasks from the geometric shape of the data in trait space (Hart et al., 2015).

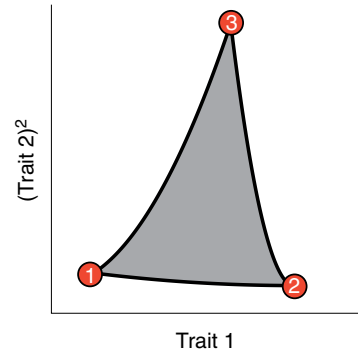


FIGURE 14.11

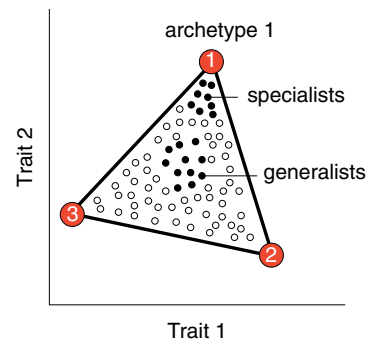


FIGURE 14.12

14.7 TRADE-OFFS IN MORPHOLOGY

Let's see how this works in practice by looking at some data. We'll start with animal morphology, and then move to proteins and gene expression. Morphology is a field that measures the shapes of organisms, and morphology books are full of lines called **allometric relationships**. For example, the molar teeth of rodents (the three big teeth at the back of the mouth called $M1$, $M2$ and $M3$) vary in shape between rodent species. One can plot each species in a trait space of the relative tooth areas, the ratios $M2/M1$ and $M3/M1$.¹ These are dimensionless traits that normalize out the total size. In this trait space, the rodent species fall on a line (Figure 14.13). Most tooth configurations are not found, and thus most of the trait space is empty.

¹ Tooth areas, the traits favored by morphologists, give straight lines. If we plotted tooth length or volume instead of area, the line would be curved.

Each rodent species is represented by a point on the line. The position on the line depends on what the rodent eats. Plant eaters (herbivores) are found at one end, meat eaters (faunivores) at the other end and omnivore generalists in the middle. This suggests a plant-eating archetype with equally sized molars (flat molars with area ratios of 1:1:1), and a meat-eating archetype with spiky molars with area ratios of 2:1:0. The line provides a rule in which the area of the middle molar is the average of its two neighbors. This rule applies also to dinosaur teeth, allowing fossil hunters to infer how much meat versus plants a dinosaur ate.

Kavanagh et al. (2007) also perturbed the development of rodent teeth, by adding morphogens or by blocking morphogen diffusion. The perturbations changed the teeth proportions, but most of the the new proportions were still close to the line. This finding is related to the robustness of the developmental pathways, and to their ability to generate useful shapes even under perturbations, a feature called canalization (Chapter 12). Some perturbations, however, led to phenotypes far from the line, showing that the empty trait space is not impossible, and can be reached.

Morphological data also shows triangles. An example is found in the classic study of Darwin's finches by Peter and Rosemary Grant (Grant, 1986). The Grants lived on a tiny island in the Galapagos and observed finch evolution over decades. They measured five traits for each finch – including mass, bone size and beak shape. This 5D data falls on a plane (the first two principal components explain over 90% of the variation). On this plane, the finches fall within a triangle (Figure 14.14). Their diet reveals three tasks: near the three vertices are species which are specialists at eating large seeds, small seeds and pollen/insects from cactus plants. Species in the middle of the triangle do a combination of these tasks.

A triangle is seen also when each data point is an ant from the same nest (Figure 14.15). E.O. Wilson measured the size of leaf-cutter ants versus the relative size of the gland which makes the pheromone for the ant trail (Wilson, 1980). He also recorded in the behavior of each ant. There are three tasks: staying in the nest and nursing, soldiering and foraging. Ants fill a continuum inside the triangle defined by these three archetypes.

You might ask what is the functional role of the ants in the middle of the triangle? Why not make three clusters of

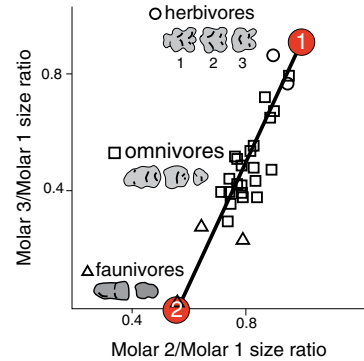


FIGURE 14.13 Adapted from (Kavanagh, Evans and Jernvall, 2007).

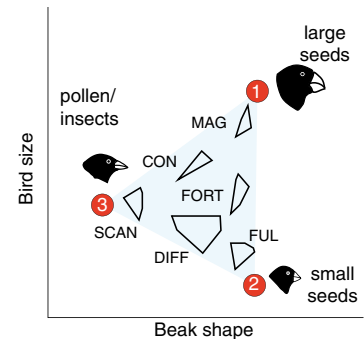


FIGURE 14.14 Adapted from (Shoval et al., 2012).

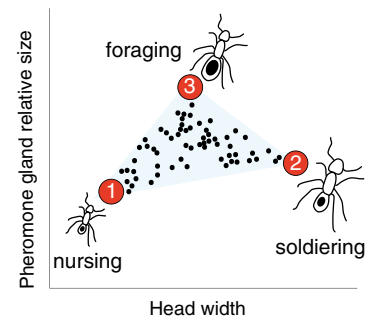


FIGURE 14.15 Adapted from (Shoval et al., 2012).

specialists – optimal nursers, foragers and soldiers, without the generalist ants in the middle which are suboptimal at all tasks? Ant researchers believe that one reason for generalists is dynamic flexibility. Suppose the nest is attacked – there is no time to make more soldiers. Instead, generalist ants can be recruited to supply the needed tasks quickly. We will use this as a metaphor soon for division of labor between cells in an organ.

14.8 ARCHETYPES CAN LAST OVER GEOLOGICAL TIMESCALES

We can also ask whether the archetype positions in trait space move over long evolutionary timescales. A model system for this question is ammonites, marine creatures with detailed morphological data covering 350 million years of evolution. The detailed data was collected in part because ammonite fossils are used to date rocks.

Ammonite shells can be described in an elegant trait space with two parameters, as proposed by paleontologist David Raup (Raup, 1967; Figure 14.16). In this trait space, the outer shell is a logarithmic spiral, whose radius grows with each whorl by a factor W , the whorl expansion rate. The inner shell is also a logarithmic spiral, with a constant ratio between the inner and outer shell radii, denoted D .

In this W - D trait space, ammonite shapes fill out a triangle (Figure 14.17). There is empty trait space, without ammonites, at large D and W . This empty trait space includes shells shaped like French horns, which are found in other clades, but not ammonites. The three archetypes at the corners of the triangle match the shell shapes that are optimal for three tasks: economy (maximal internal volume per shell material), swimming (lowest drag) and predator avoidance (rapid growth of shell diameter) (Tendler, Mayo and Alon, 2015).

There were three mass extinctions in which ammonites were wiped out except for a few surviving genera. For example, the blue dots in Figure 14.17 mark the two surviving genera after the Permian/Triassic extinction 252 million years ago. Remarkably, in about 10 million years after each extinction, ammonites diversified to refill essentially the same triangle. This suggests that tasks and archetypes did not move much in this case.

Only with the last extinction that wiped out the dinosaurs, 65 million years ago, this triangle-filling trick didn't work, perhaps



FIGURE 14.16 Adapted from (Kennedy, 1989).

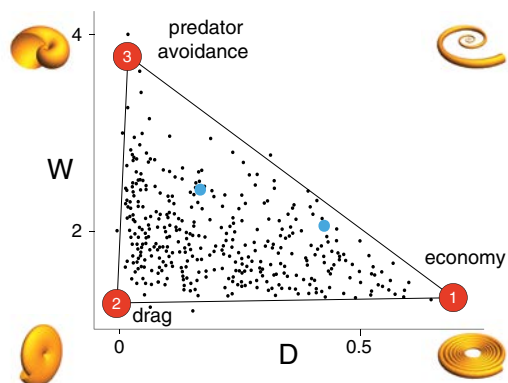


FIGURE 14.17 Adapted from (Tendler, Mayo and Alon, 2015).

due to competition with mammals. There is only one surviving genus in the ammonite lineage, called *Nautilus*.

If archetypes remain relatively fixed, there remains the question of how radically new tasks can appear. How did novelties like vision and flight evolve, given that such tasks require complex organs such as eyes and wings. Organisms must somehow move out of an existing polytope toward a new archetype (e.g., flight performance).

Current thinking is that adaptation to a novel task arises by reuse of parts that have already evolved for a different task. One example is the evolution of wings from body appendages that served as thermal regulation devices. These appendages had selection pressure to grow in order to better radiate heat. When the appendages were large enough, they allowed the organism to glide, sparking selection pressure for aerodynamic gliding properties. Finally, the gliding appendages allowed rudimentary flight, and selection pressure worked to improve their performance as wings. This picture is called **stepping-stone evolution**, because each new task is a stepping stone to the next.

14.9 TRADE-OFFS FOR PROTEINS

Let's turn now from animals to proteins. A protein can also have multiple tasks. For example, Rubisco, one of the most abundant proteins in plants, is tasked with capturing CO_2 from the air and adding it to a sugar molecule that can be used to build biomass. All of the carbon in our bodies comes from Rubisco that made the plant biomass that is the basis for our food.

Rubisco can be characterized by a trait space with four kinetic parameters. Two of these parameters are the catalytic speed k_{cat} and affinity K_m for CO_2 . The other two are the catalytic speed and affinity, k'_{cat} and K'_m , for the main competitor of CO_2 , oxygen O_2 . Capturing O_2 instead of CO_2 is a mistake that requires energy to correct.

To study trade-offs in Rubisco, Yonatan Savir and Tsvi Tlusty compiled these four kinetic traits from different photosynthetic organisms (Savir et al., 2010). They found that the Rubiscos fall approximately on a line in the 4D trait space. Figure 14.18 shows the data in the space of three traits, k_{cat} , K_m and the specificity $S = k_{cat} K'_m / k'_{cat} K_m$, together with the projections of the data on the three planes. At one end of the line segment are the fastest Rubiscos, which occur in organisms like corn, known as C_4 plants, that can concentrate CO_2 relative to the atmospheric concentration. Since these plants reach a high CO_2 concentration inside their leaves, they do not need to worry about oxygen. At the other end are the slowest Rubiscos, which bind CO_2 most strongly. These occur in organisms that do not concentrate CO_2 and face competition from O_2 . Thus, this protein seems to evolve under a speed-specificity trade-off.

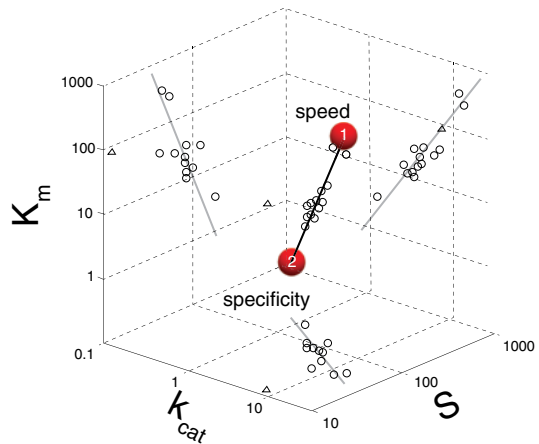


FIGURE 14.18 Adapted from (Savir et al., 2010).

14.10 TRADE-OFFS IN GENE EXPRESSION

The ParTI approach can also be applied to gene expression. At first glance, gene expression in a cell might seem very different from beaks or proteins. Cells can rapidly change gene expression according to their needs, whereas if you are born with a beak of a certain shape, you are stuck with it. Still, gene expression also faces trade-offs.

Consider a brief time period, say a second, in which the cell can make say 1000 proteins. You can't make proteins to optimize rapid growth and at the same time make proteins to optimize stress resistance. Growth and stress require very different sets of proteins. The cell needs to choose which protein portfolio to express based on its expectation of the future. Thus, the cell faces trade-offs between tasks and hence it makes sense to look for polytopes in gene expression data.

Indeed, gene expression of the top 200 promoters in *E. coli*, which make up 90% of the total promoter activity, falls on a line segment (Figure 14.19). Here, trait space is a space of gene expression, in which each axis is the fraction of the total promoter activity in the cell devoted to promoter i , with $i = 1 \dots 200$. At one end of the line segment is the growth archetype, in which gene expression is focused on making ribosomes and machinery for biomass production. At the other end is the survival archetype in which cells express stress-response genes (and a small number of ribosomes in order to restart growth when things improve).

When placed in a test tube with nutrient, *E. coli* starts out close to the growth archetype, and grows exponentially until it begins to deplete the nutrient and pollute its environment. It gradually slides down the line to the survival archetype until conditions are so bad that growth stops. *E. coli* follows approximately the same line for different nutrients and conditions.

Thus, all that *E. coli* needs to do in a new condition is decide about its position on the line segment. This means that it needs to choose a number θ between zero and one, with the growth archetype at $\theta = 0$ and stress archetype at $\theta = 1$. To choose this number, *E. coli* uses a simple mechanism to put its gene expression on a line. This line-making mechanism is based on competition between two **sigma factors**, proteins that bind RNA polymerase (RNAP) and allow it to bind sites in promoters (Figure 14.20). One factor, σ_{70} , binds sites in the promoters of growth genes and the other, σ_s , binds promoters for stress-response genes.

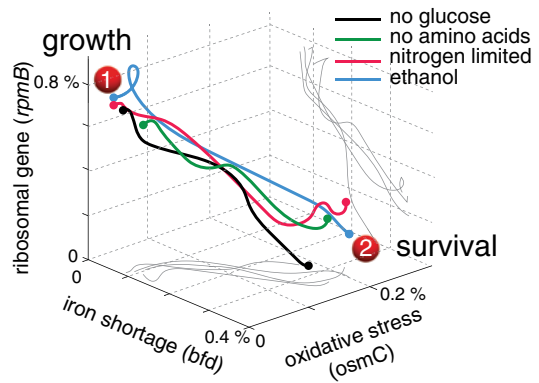


FIGURE 14.19 Adapted from (Shoval et al., 2012).

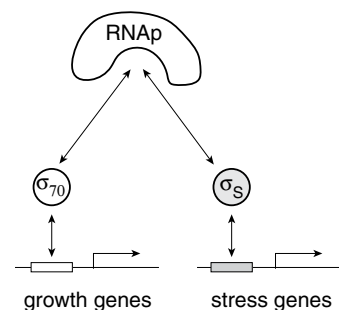


FIGURE 14.20

Thus, the position on the line is given by the fraction of RNAP bound to σ_s , namely $\theta = \sigma_s/(\sigma_s + \sigma_{70})$. *E. coli* has signaling systems that read the environment and accordingly produce and degrade the two sigma factors, in order to determine where the cell lies between the tasks of growth and survival. The coordinates of the archetypes are encoded in the strength of the sites for the two sigma factors in each promoter (many promoters have binding sites for both sigma factors). A polytope with k vertices can be achieved by a similar design with k competing factors.

14.11 DIVISION OF LABOR IN THE INDIVIDUAL CELLS THAT MAKE UP AN ORGAN

We now turn from bacteria to gene expression in human cells. Human tissues are made of different types of specialized cells: brains are made of neurons and livers are made of hepatocytes. Having different cell types for each tissue allows a useful division of labor, assigning metabolic tasks to the liver and thinking tasks to the brain.

What about division of labor between cells of a given type, say the hepatocyte cells in the liver? Recall the ants, which divide labor toward a collective goal of colony survival and reproduction. Are there specialists and generalists also within a cell type?

Analysis of gene expression from individual cells all from the same organ indicates that division of labor is widespread. Gene expression of cells of a given cell type typically falls in a continuum bounded inside shapes with pointy vertices (Korem et al., 2015; Adler et al., 2019). The tasks of the cell type can thus be inferred.

For example, liver hepatocytes are famous for doing multiple functions. They synthesize blood proteins and other essential compounds, they detoxify the blood, get rid of ammonia by turning it into urea and regulate glucose levels by storing it into glycogen or making it from amino acids when needed (gluconeogenesis). Individual liver cells fill out a tetrahedron in gene expression space, where each axis is the expression of gene i , with $i = 1 \dots 20,000$. This tetrahedron is plotted in Figure 14.21, where each point is a cell, and the axes are the first three principal components of gene expression.

At the vertices of the tetrahedron are cells that specialize in four key tasks: synthesis of blood proteins (such as albumin), gluconeogenesis, detoxification and, surprisingly, lipid metabolism/iron homeostasis. Each archetype has additional secondary tasks, so that each specialist carries out a “syndrome of tasks”: for example, the gluconeogenesis archetype also produces the antioxidant glutathione.

The specialist cells have a particular arrangement in space across the liver (Halpern et al., 2017). The liver is made of repeating hexagonal columns called liver nodules, about 15 cells across. The cells that

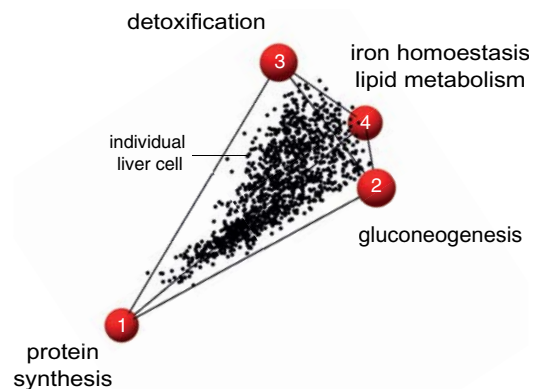


FIGURE 14.21 Adapted from (Adler et al., 2019).

specialize in synthesis (albumin, glutathione) tasks that require much oxygen, are found in the oxygen-rich boundary of the hexagons, near the portal veins. Cells that specialize in detoxification, which requires less oxygen, are found at the oxygen-poor center of the hexagon where the central vein drains the lobule. Lipid/iron specialist cells are found in the middle. This placement of specialists at positions best suited to their task helps maximize the organ performance at all tasks (Adler et al., 2019).

These individual liver-cell experiments are an example of the technologies that provide the ability, undreamed of when I was a postdoc 20 years ago, to measure thousands of numbers from each individual cell. They produce massive amounts of data. How can we analyze such information-rich experiments in biology? The challenge is that human beings cannot visualize high-dimensional data. We can deal with a 2D picture or sometimes a 3D volume, but 4D, not to speak of 20,000D, is alien to us. That is why we need approaches to reduce dimensionality into something we can comprehend.

The problem is that each dimensionality reduction method has an implicit model of how the data is structured. If the data is unlike that model, the method can be misleading. Consider a commonly used approach, called data **clustering**. Data clustering assumes that data is arranged in distinct, well-separated clouds, and clustering algorithms can easily detect those clouds. However, clustering does poorly when data is arranged in a continuum – there are no natural distinct clusters.

In this case, other algorithms, known as **archetype analysis algorithms** (Mørup and Hansen, 2012), can help detect whether data can be approximated as a continuum filling a polytope, and how many vertices the polytope has (Hart et al., 2015). These algorithms focus on the outside contours of the data. As in any approach, be wary of artifacts that make data spuriously look like a line or triangle (see Exercise 14.19).

14.12 VARIATION WITHIN A SPECIES LIES ON THE PARETO FRONT

Let's end by returning to animal morphology, in an example that opens up new questions. This example helped to start the ParTI framework, when Kathy Kavanagh showed me data on bird toes in 2009. The fourth toe of the bird has four bone segments called phalanges – similar to the bones in our fingers. Each bird species can be plotted in a trait space whose axes are the areas of three of the phalanges normalized by the area of the fourth. In this 3D trait space, birds fall approximately on a plane and on that plane they fill out a triangle (Figure 14.22).

If I wrote this chapter clearly, you can see that the triangle suggests that the bird toes have three tasks. We can infer the tasks by looking at the birds closest to each vertex of the triangle. We see parrots and other perching birds near one archetype. The task is grasping, and the toe shows the biomechanical optimum for grasping a branch, namely equal-sized (and curved) phalanges. The second archetype is close to ostriches and other walking birds – it is the walking archetype with phalanges that decrease in area with ratios 4:2:1:0. Again, a biomechanical optimum: If you ever wondered about the difference between your hands and feet, your hands are for grasping and have equal-sized parts, and your feet are for walking and have a long footpad with short toe for kicking off. The third archetype is

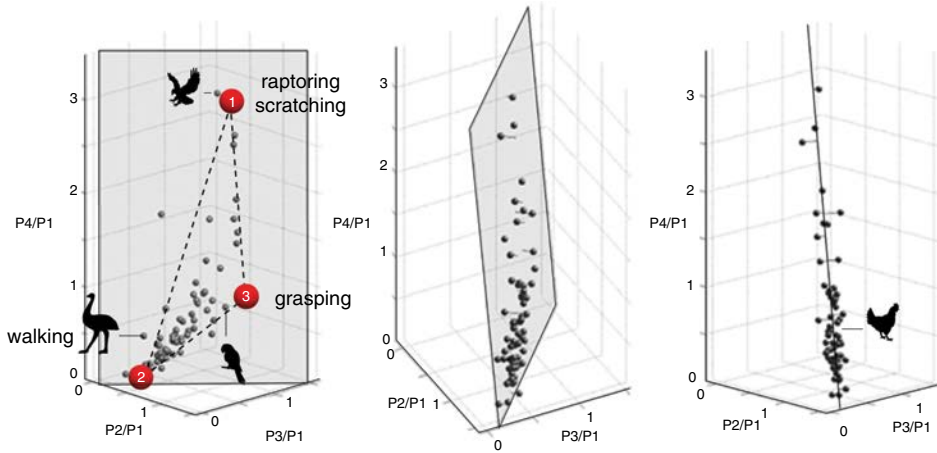


FIGURE 14.22 Adapted from (Kavanagh et al., 2013).

for raptoring/scratching, and has a long fourth phalanx that provides a good-sized lever for the talon.

But let's ask a different question. The point for a given species, say chicken, is the average over all chickens measured. What if we look at *individual* chickens? Each chick is born with slightly different bone ratios (and these ratios are set for life already in the egg). This variation is due to the combination of the parents' genomes that provide each individual with a unique combination of genetic differences called **polymorphisms**. Variation further arises from the randomizing effect of noise during development. So individuals form a cloud in trait space around the chicken average. Does this cloud go in all directions, or is it flattened like a pancake along the triangle defined by different species?

Kathy Kavanagh tested this by hatching 100 chicken eggs and 100 zebra-finch eggs (Kavanagh et al., 2013). She found that the cloud of variation is flattened like a pancake along the triangle defined by different species (Figure 14.23). Exaggerate the difference between two individuals and you get a caricature of another bird species.

How can this be? As mentioned above, variation between individuals comes from a combination of noise and polymorphisms. To maintain the chicks on the front, the prevalent polymorphisms must push the phenotype along the front, but importantly not off of the front. We will call such polymorphisms **aligned polymorphisms** (Figure 14.24), because their effect is aligned with the front. Aligned polymorphisms can be selected because chickens have a range of niches in which walking, grasping and scratching are differentially important (Sheftel et al., 2018). Each individual gets a mix of polymorphisms that create a cloud of variation aligned with the front. Polymorphisms that move the phenotype off of the front face the risk of a competitor on the front with higher performance at all tasks.

Furthermore, in order to produce the observed variation, developmental patterning mechanisms need to somehow focus the effects of noise along useful directions aligned with the front, a kind of "aligned canalization."

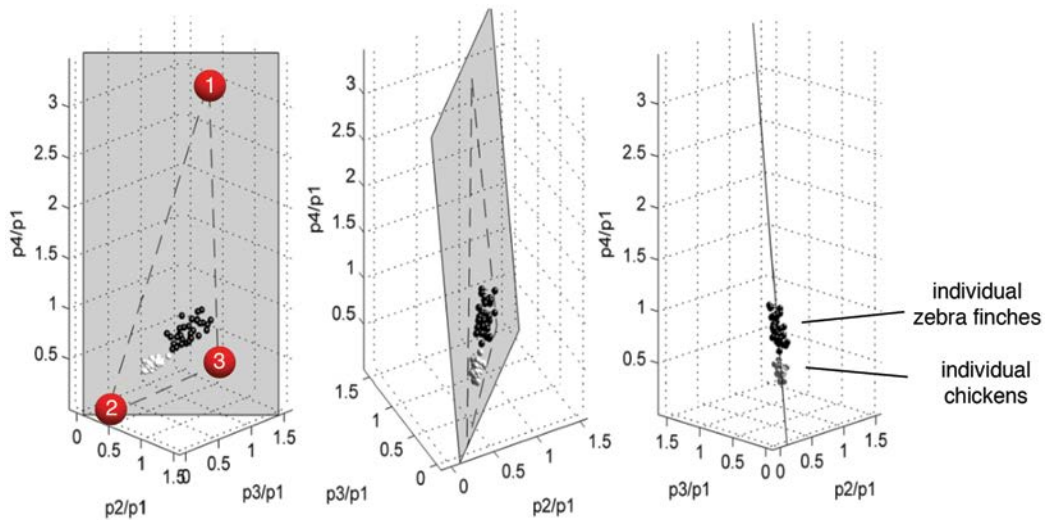


FIGURE 14.23 Adapted from (Kavanagh et al., 2013).

The importance of aligned genetic variation and canalization can be appreciated if we recall that every individual is a never-tried-before combination of mom and dad, a pastiche of their millions of genetic differences. It's like designing a new jet plane and flying it out of the hangar without ever testing it. Aligned polymorphisms and canalization mean that every offspring has a good chance to lie somewhere on the Pareto front, and, therefore, can be competitive in one of the niches available to the species.² What types of developmental mechanisms and genetic population structure is needed for this type of variation is an open question.

In sum, natural selection is usually a multi-objective optimization problem. Organisms are, therefore, rarely optimal for a single task, but instead evolve under trade-offs. Evolutionary trade-offs lead to patterns in phenotype space in which a continuum of possibilities is bounded within polyhedral-like shapes with pointy vertices. The pointy vertices can be used to infer the tasks at play. The position of each phenotype relative to the vertices (the archetypes) tells us how important each task was in its evolution. This notion applies in principle to any scale – molecules, circuits, cells, organisms – as long as natural selection has had enough time, population size and genetic variation to approach the optimal trade-offs between tasks.

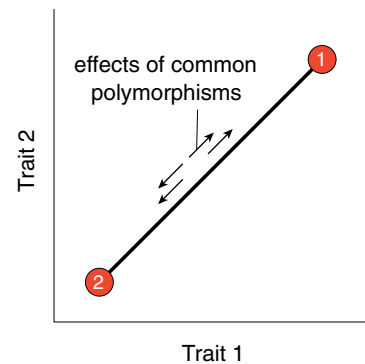


FIGURE 14.24

² One more requirement is that the genetic variation be additive in the sense that multiple polymorphisms do not generate components perpendicular to the front. Such additivity can itself be selected (Sheftel et al., 2018).

FURTHER READING

Natural Selection as a Multi-Objective Problem

(Arnold, 1983) “Morphology, performance, fitness.”

Evolutionary Trade-Offs and Pareto Task Inference

(Adler et al., 2019) “Continuum of gene-expression profiles provides spatial division of labor within a differentiated cell type.”

(Sheftel et al., 2013) “The geometry of the Pareto front in biological phenotype space.”

(Shoval et al., 2012) “Evolutionary Trade-Offs, Pareto Optimality, and the Geometry of Phenotype Space.”

Animal Morphology

(Grant, 1986) “The ecology and evolution of Darwin’s finches.”

(Kavanagh, Evans and Jernvall, 2007) “Predicting evolutionary patterns of mammalian teeth from development.”

(Kavanagh et al., 2013) “Developmental bias in the evolution of phalanges.”

(McGhee, 2006) “The geometry of evolution: adaptive landscapes and theoretical morphospaces.”

Algorithms for Archetype Analysis

(Hart et al., 2015) “Inferring biological tasks using Pareto analysis of high-dimensional data.”

(Mørup and Hansen, 2012) “Archetypal analysis for machine learning and data mining.”

R software package for ParTI by Vitalii Kleshchevnikov: <https://github.com/vitkl/ParetoTI>

EXERCISES

14.1 *Mathematical proof for the main theorem in this chapter:* Prove that if (i) fitness is an increasing function of k performance functions in trait space, and (ii) each performance function $i = 1 \dots k$ has a maximum at a point called archetype i , and (iii) performance drops with Euclidean distance from the archetype, then the point of maximum fitness is found inside the polytope defined by the k archetypes (Shoval et al., 2012).

Solution:

Each phenotype is described by a vector of traits T (for convenience we will drop the vector sign from now on). Fitness F is an increasing function of the performance at the k different tasks, $F(T) = F(P_1(T), P_2(T), \dots, P_k(T))$. Each performance function P_i has a maximum at archetype i , A_i , and performance decreases with Euclidean distance from the archetype $P_i(T) = P_i(\|T - A_i\|)$. We will show that the optimal phenotype (the phenotype that maximizes F) is a weighted average of the archetypes with weights that are positive and sum to one. This is equivalent to saying that the optimal phenotype is inside the polytope defined by the k archetypes. Another way to say this is that the optimal phenotypes are **convex combinations** of the archetypes.

The optimal phenotype maximizes fitness, $dF/dT = 0$. The second derivative also needs to be negative for a maximum (or zero with conditions on higher-order derivatives), see exercise 14.9. Let’s denote the distance from the archetype

chain rule, $\frac{dF}{dT} = \sum_i \frac{\partial F}{\partial P_i} \frac{dP_i}{dr_i} 2(T - A_i) = 0$. Solving this shows that the optimal T , T_{opt} , is a weighted average of the archetypes

$$T_{\text{opt}} = \sum_i \theta_i A_i \quad (\text{P14.1})$$

with weights

$$\theta_i = \frac{\frac{\partial F}{\partial P_i} \frac{dP_i}{dr_i}}{\sum_j \frac{\partial F}{\partial P_j} \frac{dP_j}{dr_j}} \quad (\text{P14.2})$$

where all derivatives are at T_{opt} . Note that the weights sum to one, $\sum \theta_i = 1$. The weights are positive $\theta_i > 0$ because F increases with performances (so that $\partial F / \partial P_i > 0$) and performance decreases with distance from the archetype ($dP_i / dr_i < 0$) so that all terms have a negative sign which cancels out. Hence,

- i. The optimal phenotype is a weighted average (convex combination) of the archetypes $T_{\text{opt}} = \sum \theta_i A_i$.
- ii. The weights are positive and sum to one ($\theta_i > 0$, $\sum \theta_i = 1$). Another way to say this is that the k maxima of the k performance functions define a $k - 1$ dimensional shape in trait space (the convex hull of those k points), and optimal phenotypes are trapped within inside this shape. For two tasks, $k = 2$, this shape is a line segment:

$$T = \theta A_1 + (1 - \theta) A_2, \quad 0 \leq \theta \leq 1 \quad (\text{P14.3})$$

14.2 Phenotype position is determined by fitness and performance gradients: This exercise shows that the position inside the polytope can provide information. Exercise 14.1 shows that the optimal phenotype for a given fitness function F is a weighted average of the archetypes.

- a. Show that the weight for each task can be interpreted as the importance of the task to fitness times the sensitivity of its performance to changes in the traits.
- b. Show that in a niche in which one task has much greater effect on fitness than the other tasks, the phenotype will be close to the corresponding archetype. This is a specialist phenotype.
- c. In the case of ammonites, how can the position in the triangle help us to understand what might have been the selective conditions for each ammonite?
- d. Discuss the effects of adding a new task to k existing tasks. This new task has a small effect on fitness and can be carried out effectively out by a large range of traits.

Solution to (a): the weights from Exercise 14.1 are

$$\theta_i = \frac{\frac{\partial F}{\partial P_i} \frac{dP_i}{dr_i}}{\sum_j \frac{\partial F}{\partial P_j} \frac{dP_j}{dr_j}} \quad (\text{P14.4})$$

The weight for task i is therefore the normalized product of the importance of the task to fitness $\partial F/\partial P_i$ times the sensitivity of the performance to changes in the trait dP_i/dr_i .

14.3 *Multiple tasks break the symmetry of neutral spaces:*

Consider a system with two tasks. Task 1 has a performance function whose maximum is at point A, the archetype for that task, and performance decays with Euclidean distance from the archetype. Task 2, however, has a maximum performance not attained at a single point, but instead *in an entire region N* in trait space (Figure 14.25). Such a region, in which all points have the same performance, is called a **neutral space**. Performance decreases with Euclidean distance from N (distance to the closest point in N).

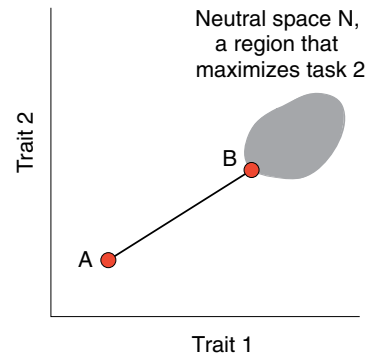


FIGURE 14.25

- Show that only one point in N is on the Pareto front. It's as if task 1 “chooses” one point in N . It thus breaks the symmetry between the points in N .
- Show that the Pareto front is a line segment that connects point A with a point on the boundary of N .
- Give a biological example for a task which is maximized at a region and not a point in trait space.
- What would happen if task 2 was the only task affecting fitness?

Solution to a:

- The chosen point is the point in N closest to A. To see this, let's call this point B. Choose a point C in N other than B. Its performance in task 1 is lower than B (because B is closer to A by definition), and its performance at task 1 is the same as B since both are in the neutral space. Hence, C is dominated by B and can be removed. We are left with point B.

14.4 *The Pareto front is where performance contours are tangent:*

- Show that the Pareto front is the set of points in which the contours of the performance functions are externally tangent.

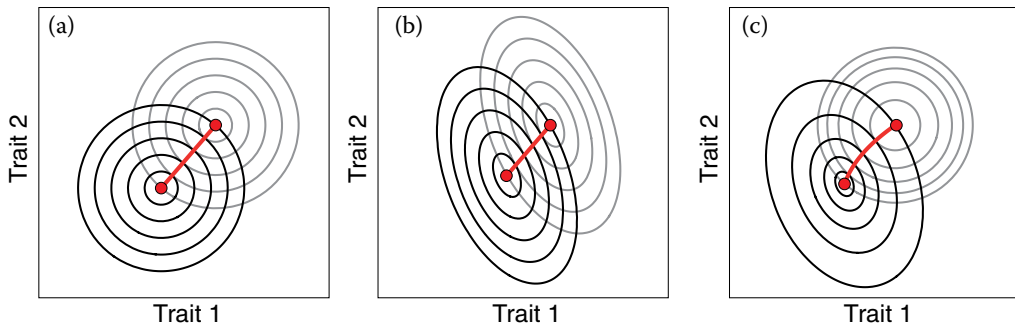


FIGURE 14.26

- b. Use this to explain why, when performance functions decline with Euclidean distance from their maxima, contours are circular and the set of tangent points is a line (Figure 14.26a).
- 14.5 *Pareto front is a line if both performances decay with the same inner-product norm:* This exercise shows that when the performance functions decline not with Euclidean distance, but all decline with the same inner-product norm (giving different traits differential impact on performance, with elliptical contours), the theorem of Exercise 14.1 still applies.

Consider the case of two tasks, in which performance functions decay with distance given by an inner product norm, $r_i = (T - A_i)^T Q (T - A_i)$ where Q is a positive definite matrix. The inner product norm is the same for both tasks. Their contours are, therefore, concentric parallel ellipses around the archetype, such that both tasks have ellipses of the same orientation and eccentricity (Figure 14.26).

 - a. Sketch the contours.
 - b. Show that the Pareto front is a line segment (use Exercise 14.4).
 - c. Compute the Pareto front for the case of k tasks. Show that it is exactly the same as in the case of Euclidean norms – a polytope with vertices at the archetypes. (*Hint:* Rotate and dilate space until the contours are circular.)
- 14.6 *Pareto front is curved if performances decay with different norms:* Consider the case of two tasks, and performances that each decays by a different inner-product norm. Their contours are therefore concentric parallel ellipses around each archetype, but with different orientation and eccentricity (Figure 14.26c).

 - a. Explain why the Pareto front is curved. Show graphically that the most curved front occurs when the long axes of the ellipses are at 45 degrees to each other, and the long axes are much longer than the short axes.
 - b. Consider the case where the long axes of the ellipses for the two tasks are orthogonal to each other. Sketch the contours. Is the front curved or straight?
 - c. What is the biological meaning of the difference in norms and contour shape?

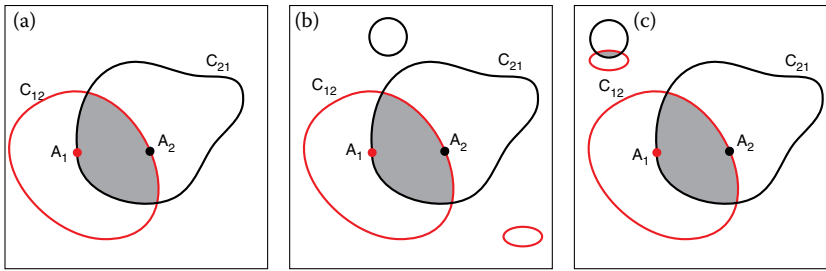


FIGURE 14.27

14.7 *Bounds for general performance functions:* This exercise shows that even when we know nothing about the shape of the performance functions (no norms or even no monotonic decay from the archetype), one can still bound the Pareto front in a region between the archetypes that lies between certain contours (Sheftel et al. 2013).

Consider the case of two traits and two tasks. Performance functions can have a general form, with global maxima at the archetypes A_1 and A_2 .

- Show that the Pareto front is bounded inside the region between two special contours, C_{12} , the contour of performance 1 that goes through A_2 , and C_{21} , the contour of performance 2 that goes through A_1 (Figure 14.27a).
 - Suppose that the performance functions have local maxima in addition to the global maximum, such that the contours C_{12} and C_{21} have disconnected pieces that go around the local maxima (Figure 14.27b). Show that generally, the local maxima do not affect the Pareto front. Show that only when the local maxima of the two performance functions happen to be close to each other, the Pareto front can have multiple disconnected pieces (Figure 14.27c).
- 14.8 *Too few traits measured:* Suppose a system has four tasks, but only two traits are measured experimentally. What kind of shapes would describe the optimal phenotypes in the 2D trait space?

Solution:

Triangle or kite. The four tasks generally lead to a tetrahedron with four archetypes at the vertices. Measuring two traits means projecting the tetrahedron on a plane. Projections of a tetrahedron are shaped as a quadrangle (kite) or a triangle. The latter case is when one of the archetypes is occluded.

14.9 *Empty regions in the polytope:* The theorems we discussed so far are silent on the question of where in the polytope the points can lie. This exercise will show that some regions of the polytope can be empty (forbidden) if the performance functions have certain curvatures. We will use a 1D example, with a single trait T and two tasks.

- Show that the Pareto front is the line segment between the two archetypes.
- Show that a condition for optimality is $d^2F/dT^2 < 0$.

- c. Show that this requires a condition on the curvature of the performance functions d^2P_i/dT^2 .
 - d. Show that when both curvatures are positive, there is an empty region with no phenotypes.
 - e. What happens when the performance functions are Gaussians that decay with distance from the archetype?
 - f. What other reasons might explain an empty region inside a polytope. (*Hint: Consider physical constraints on the phenotype.*)
- 14.10 *Mass-longevity triangle* (Szekely et al., 2015): Plotting the longevity of mammalian and bird species versus their mass shows a continuum inside a triangle-like shape (Figure 14.28). At the three vertices are shrews (that weigh a few grams and live about 2 years), elephants and whales (tens of tons, ~ 100 years) and small bats (a few grams, ~ 50 years). Near the bat archetype are mammals that live in trees and social mammals that live underground (e.g. naked mole rat). Flying birds are found near the bat archetype and walking birds near the bottom edge of the triangle. Interpret these findings in terms of trade-offs and tasks.
- 14.11 *Different modules of tasks*: Suppose that an organism has two parts or modules, each with a different set of tasks and traits. For example, a bird has beak traits devoted to

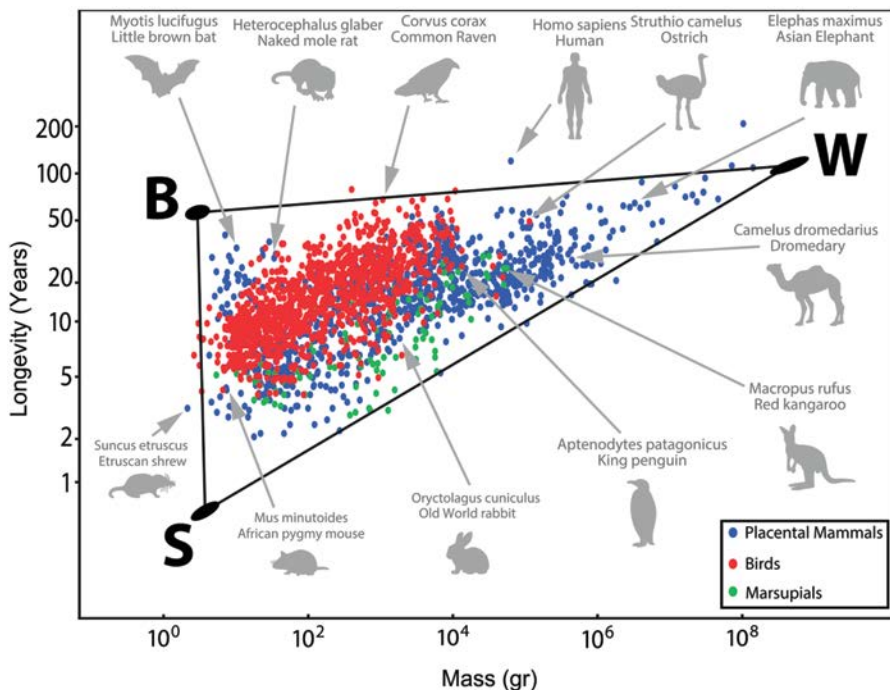


FIGURE 14.28 Adapted from (Szekely et al., 2015).

tasks of eating and toe traits devoted to the tasks of walking/grasping. Suppose that each module has two traits and two tasks.

- a. What would the Pareto front look like in the 4D trait space?
 - b. What would happen in a ParTI analysis if we didn't realize that toes and beaks are separate and mixed the traits together into one big dataset?
 - c. *Harder problem:* Devise an algorithm to detect in a large dataset whether there are multiple separable modules of traits and tasks, each with each own Pareto front (give outline of algorithm, 100 words).
- 14.12 *Molecular mechanism for polytopes in gene expression:* In this exercise, we consider a mechanism that can generate a polytope with k vertices for gene expression. It is a generalization of the σ -factor mechanism in Figure 14.20. Suppose that k transcription factors X_i , with $i = 1 \dots k$, regulate genes, but are active only when bound to protein Y , in the complex $[X_i Y]$. The binding is strong with affinity K_i . Each gene promoter has binding sites for one, some or all of the X_i . The expression of gene j when $[X_i Y]$ binds its site is w_{ij} , and the effects of the regulators add up (SUM gate).
- a. Show that the expression of gene j is $T_j = Y_T \sum_i w_{ij} X_i K_i^{-1} / \sum_i X_i K_i^{-1}$ where Y_T is the total level of Y .
 - b. Show that gene expression lies in a polytope with k vertices in gene expression space whose axes are T_j .
 - c. What are the coordinates of the archetypes in terms of the mechanism parameters?
 - d. In each condition, the cell regulates the concentrations of X_i . What is the concentration combination that leads to gene expression near vertex 1? Near the middle of the polytope? Outside of the polytope?
 - e. What happens to the polytope if we delete one of the regulators?
 - f. What happens if we change the level of Y , Y_T ?
- 14.13 *Aligned mutations in the polytope mechanism:* Consider the mechanism of Exercise 14.12. Suppose a mutation can change one regulator concentration X_i , one binding site strength w_{ij} or Y_T .
- a. Which mutations move the phenotype inside the front?
 - b. Which mutations move the phenotype off of the front?
 - c. Which mutations change the archetype coordinates?
 - d. Which mutations would you expect to see at high prevalence (i.e., common polymorphisms) in a population of organisms facing niches that share the same k tasks, but where each niche gives different weighting to each task?

- 14.14 *Pareto optimality in engineering*: Consider the performance space of car designs, with the performances of acceleration and economy (Figure 14.5).
- Where are the archetypes in performance space?
 - What would be examples of relevant traits in a trait space of cars?
 - How is ParTI different from Pareto analysis in performance space?
- 14.15 *Trade-offs in a network motif*: Consider the negative autoregulation network motif of Chapter 2, with dynamics $\frac{dX}{dt} = \frac{\beta}{1 + \left(\frac{X}{K}\right)^n} - \alpha X$ where X is a stable protein. Suppose the tasks are speed (fast response time) and economy (minimal protein production integrated over one cell generation, $\log(2)/\alpha$).
- Plot performance space and trait space, with the traits β , K , n .
 - What is the Pareto front?
 - When is simple regulation selected?
- 14.16 *Performance space of fold-change-detection (FCD) designs*: Read Adler et al. (2017). Explain how the Pareto front concept is used to define a handful of FCD circuit designs.
- 14.17 *Aligned canalization*: Analyze the French flag model of morphogen pattern formation (Chapter 12). In this model, morphogen X is produced at position $x = 0$ and diffuses with diffusion coefficient D and is degraded at rate α . Cell fate decisions are in these regions defined by the points in space where morphogen crosses the thresholds T_1 and T_2 .
- What is the effect of varying thresholds, D , and α on the patterns?
 - Define a trait space given by the ratio of the lengths of cell-fate regions. What suite of variations occurs upon changes in threshold D , and α ?
 - Could such a design provide aligned canalization for a given set of tasks?
- 14.18 *Optimal arrangement in space* (Adler et al., 2019): Consider a tissue with a spatial coordinate x , with gradients of oxygen and nutrients across x . Cells in the tissue have two tasks, whose performance depends on space and on gene expression, $P_i(x) = \phi_i(x)P(|T(x) - A_i|)$ where $T(x)$ is gene expression of the cells at position x and A_i are the archetypes. The collective performance at task i , summed over all cells, is S_i . The overall function of the tissue is an increasing function of the collective performances $F = f(S_1, S_2)$.
- Suppose that performance functions P have negative curvature. Solve for the gene expression profiles as a function of position $T(x)$.
 - Are there specialist cells and generalist cells?
 - Task 1 is performed best near $x = 0$, and task 2 is insensitive to space ($\phi_1 = 1 - x$, $\phi_2 = 1$). What is the spatial expression profile?

- d. Relate this problem to the case of hepatocytes in the main text.
 - e. What Pareto front shapes might be expected in a tissue with 3D spatial gradients?
- 14.19 *Triangles can result from other reasons:* A coin is tossed N times, and the number of heads H is recorded. This is repeated 100 times for $N = 1, 2, \dots, 100$.
- a. Sketch the data in a trait space whose axes are the number of tosses N versus the number of heads H .
 - b. Explain why the data resembles a triangle.
 - c. Is there anything special about the data points near the vertices?
- 14.20 *Why only 2–4 archetypes, and not more, in most datasets?*

In the systems analyzed in this chapter, we saw evidence for 2, 3 or 4 tasks, namely lines, triangles or tetrahedra. Why don't we see many more tasks, say 10 or 100?

Solution:

To see k archetypes requires k tasks to have the same, large, influence on fitness. More precisely, the traits on the Pareto front are convex combinations of the archetypes $\sum_i \theta_i A_i$ with the weight for archetype i proportional to the impact of task i on fitness $f_i = \partial F / \partial P_i$ times the sensitivity of the performance at task i to the traits: $s_i = dP_i / dr_i$ namely $\theta_i = f_i s_i / \sum_j f_j s_j$ (Exercise 14.2). It is reasonable to assume that tasks have widely distributed impacts and sensitivities, so that it is unlikely that a large number of different tasks will have large weights with the same order of magnitude. Biological systems usually have many additional tasks of small impact and/or small sensitivity (i.e., can be performed well by many phenotypes), and these tasks exert a weak “gravitational pull” on the data. They are undetectable without very precise data. Can you think of such low-impact tasks, for example, for bird beaks?

BIBLIOGRAPHY

- Adler, M., Korem Kohanim, Y., Tendler, A., Mayo, A. and Alon, U. 2019. ‘Continuum of gene-expression profiles provides spatial division of labor within a differentiated cell type’. *Cell Systems*, 8(1), pp. 43–52.e5. <https://doi.org/10.1016/j.cels.2018.12.008>
- Adler, M. et al. 2017. ‘Optimal regulatory circuit topologies for fold-change detection’, *Cell Systems*, 4(2), pp. 171–181. doi: 10.1016/j.cels.2016.12.009.
- Alon, U. 2009. ‘How to choose a good scientific problem’, *Molecular Cell*, 35(6), pp. 726–728. doi: 10.1016/j.molcel.2009.09.013.
- Arnold, S.J. 1983. ‘Morphology, performance and fitness’, *Integrative and Comparative Biology*, 23(2), pp. 347–361. doi: 10.1093/icb/23.2.347.
- Grant, P.R. 1986. ‘Ecology and evolution of Darwin’s finches’, Grant, P. R. *Ecology and Evolution of Darwin’s Finches*. Princeton University Press.
- Halpern, K.B. et al. 2017. ‘Single-cell spatial reconstruction reveals global division of labour in the mammalian liver’, *Nature*, 542(7641), pp. 352–356. doi: 10.1038/nature21065.
- Hart, Y. et al. 2015. ‘Inferring biological tasks using Pareto analysis of high-dimensional data’, *Nature Methods*, 12(3), pp. 233–235. doi: 10.1038/nmeth.3254.

- Kavanagh, K.D. et al. 2013. 'Developmental bias in the evolution of phalanges', *Proceedings of the National Academy of Sciences*, 110(45), pp. 18190–18195. doi: 10.1073/pnas.1315213110.
- Kavanagh, K.D., Evans, A.R. and Jernvall, J. 2007. 'Predicting evolutionary patterns of mammalian teeth from development', *Nature*, 449(7161), pp. 427–432. doi: 10.1038/nature06153.
- Kennedy, W.J. 1989. 'Thoughts on the evolution and extinction of Cretaceous ammonites', *Proceedings of the Geologists Association*, 100(3), pp. 251–279.
- Korem, Y. et al. 2015. 'Geometry of the gene expression space of individual cells', *PLoS Computational Biology*, 11(7). doi: 10.1371/journal.pcbi.1004224.
- McGhee, G.R. 2006. *The Geometry of Evolution: Adaptive Landscapes and Theoretical Morphospaces*, Cambridge University Press.
- Mørup, M. and Hansen, L.K. 2012. 'Archetypal analysis for machine learning and data mining', *Neurocomputing*, 80, pp. 54–63. doi: 10.1016/j.neucom.2011.06.033.
- Raup, D.M. 1967. 'Geometric analysis of shell coiling: coiling in Ammonoids', *J. Paleo.*, 41(1), pp. 43–65. doi: 10.2307/1301903.
- Savir, Y. et al. 2010. 'Cross-species analysis traces adaptation of Rubisco toward optimality in a low-dimensional landscape', *Proceedings of the National Academy of Sciences*, 107(8), pp. 3475–3480. doi: 10.1073/pnas.0911663107.
- Sheftel, H. et al. 2013. 'The geometry of the Pareto front in biological phenotype space', *Ecology and Evolution*, 3(6). doi: 10.1002/ece3.528.
- Sheftel, H. et al. 2018. 'Evolutionary trade-offs and the structure of polymorphisms', *Philosophical Transactions of the Royal Society B: Biological Sciences*, 373(1747), p. 20170105. doi: 10.1098/rstb.2017.0105.
- Shoval, O., Sheftel, H., Shinar, G., Hart, Y., Ramote, O., Mayo, A., Dekel, E., Kavanagh, K. and Alon, U. 2012. 'Evolutionary trade-offs, pareto optimality, and the geometry of phenotype space', *Science*, 336(6085), pp. 1157–1160. doi: 10.1126/science.1217405.
- Szekely, P. et al. 2015. 'The mass-longevity triangle: Pareto optimality and the geometry of life-history trait space', *PLoS Computational Biology*, 11(10). doi: 10.1371/journal.pcbi.1004524.
- Tendler, A., Mayo, A. and Alon, U. 2015. 'Evolutionary tradeoffs, Pareto optimality and the morphology of ammonite shells', *BMC Systems Biology*, 9(1). doi: 10.1186/s12918-015-0149-z.
- Wilson, E.O. 1980. 'Caste and division of labor in leaf-cutter ants (Hymenoptera: Formicidae: Atta)', *Behavioral Ecology and Sociobiology*. Springer-Verlag, 7(2), pp. 157–165. doi: 10.1007/BF00299521.

Modularity

15.1 THE ASTOUNDING SPEED OF EVOLUTION

After the extinction of the dinosaurs, a few small mammalian species survived. They found a world with many empty niches. In about 10 million years, these ancestors evolved to organisms as different as whales, bats and primates.

Ten million years is about 10^7 generations. In this chapter, we'll see that this is extremely fast if you compare it to simulations of evolution. For example, we will simulate evolving a logic-gate circuit from scratch. In 10^7 generations, you can barely evolve a logic circuit with a dozen inputs. Thus, we need to explain the speed at which evolution can generate complex organisms.

15.2 MODULARITY IS A COMMON FEATURE OF ENGINEERED AND EVOLVED SYSTEMS

Clues can be found by considering a different process that generates complex systems: human engineering. Engineering is unlike biological evolution in many ways. Evolution tinkers by making random changes to available parts, whereas engineers work with top-down principles and learn from the vast experience of other engineers. But although evolution did not go to engineering school, its outcomes, evolved biological systems, have striking similarities to certain aspects of engineered systems.

We have seen these similarities throughout this book. One similarity is the reuse of a small set of elementary circuits. Engineers use tested tried-and-true circuit elements like amplifiers and logic gates to build complex devices. As we saw in Part 1 of this book, biological regulation networks are also made of a small set of network motifs, which can be connected to generate elaborate computational functions. A second shared principle is robustness, a major concern of engineers who use principles such as integral feedback control. As we saw in Part 2 of this book, the need for robustness shapes the way that biological systems are built. In Part 3, we have discussed optimality, and were inspired by concepts of cost-benefit analysis and Pareto optimality used by engineers.

In this chapter, we focus on an additional powerful principle that allows both engineers and biological evolution to scale up and produce remarkably complex systems that work in the real world: **modularity**.

A modular system is a system that can be decomposed, at least to a first approximation, into structurally independent parts. Each part also has a specific function. Modules have input and output ports to interface with other modules. Inside the module are internal connections that are shielded from the rest of the system. For example, a car can be decomposed into modules such as the motor and the battery, and well-written software can be decomposed into independent subroutines.

Figure 15.1 shows an example of two networks, one with a non-modular structure and another that is modular. The structural modularity of a network can be quantified by algorithms, such as the Newmann–Girvan algorithm (Appendix C). These algorithms attempt to partition the network into parts, such that the parts have more connections

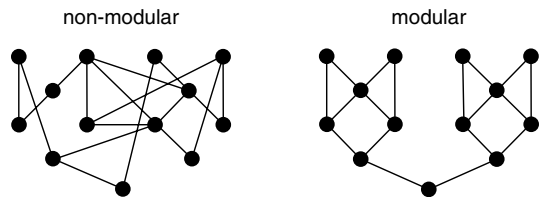


FIGURE 15.1

inside them than to other parts. The degree to which a network can be decomposed in such a way can be used to provide a modularity score. With this quantification, networks can be placed on a continuum between modular and non-modular. The vast majority of random networks are non-modular. But biological systems on all scales show modularity.

15.3 MODULARITY IS FOUND AT ALL LEVELS OF BIOLOGICAL ORGANIZATION

The body is made of organs. Each organ is a module with special structural organization and specific functions. For example, the liver module supplies glucose and blood proteins to the rest of the body (Figure 15.2). The lung supplies oxygen and removes CO_2 . These two modules are linked by their functional output (glucose from the liver to the lung, oxygen from the lung to the liver). You can evolve the lung without need to modify the liver, as long as the lung supplies the needed oxygen.

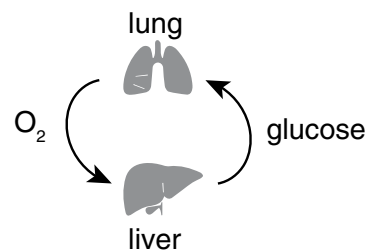


FIGURE 15.2

The organs are made of cells, a classic biological module. Each cell has subparts called organelles with specific function such as the nucleus that houses the genes. Genes, in turn, are organized into promoters, introns and exons. Promoters have modules such as binding sites. Delete a binding site and you lose a specific regulatory feature. For example, deleting a DNA region (the *eve stripe 2* enhancer) from the promoter of the *eve* gene causes the loss of stripe number 2 of the 7 expression stripes in the fruit-fly embryo (Figure 15.3), keeping the other 6 stripes of expression intact.

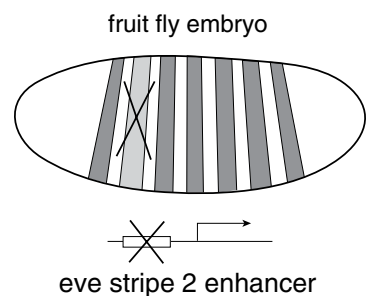
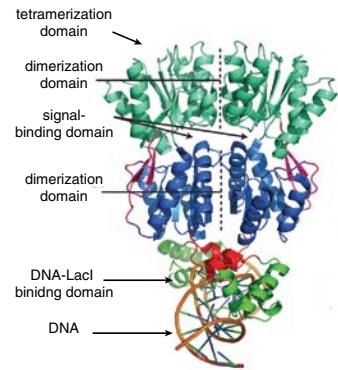


FIGURE 15.3

Proteins themselves are made of modular domains: for example, a typical transcription factor such as the lac repressor LacI has a domain that binds to its input signal, a domain that binds DNA and a protein-binding domain to form LacI tetramers (Figure 15.4). Biologists routinely glue together domains from different proteins to make new proteins that recombine the features of the original proteins.



15.4 MODULARITY IS NOT FOUND IN SIMPLE COMPUTER SIMULATIONS OF EVOLUTION

FIGURE 15.4 Adapted from (Raman et al., 2004).

Why does modularity exist in biological networks, and how did it evolve? It is important to realize that not all evolved networks are modular. The opposite is true: **non-modular solutions are the norm in simple computer simulations of evolution.**

In evolutionary simulations, a population of networks is evolved by randomly adding, removing and changing connections between nodes – and even duplicating and recombining parts of the networks – until the networks perform a given computation goal, that is, until the networks give the correct output-to-input relationship.

Unlike biological networks, simulated networks evolved in this way are usually non-modular. They have a highly interconnected structure that cannot be decomposed into nearly independent subsystems (Thompson, 1998). Since the evolved networks are non-modular it is difficult to understand how they work. These non-modular solutions are often more highly optimized than their modular, human-engineered counterparts.

The fundamental reason for the lack of modularity in these evolved networks is that modular structures are far rarer and usually less optimal than non-modular ones for a given task. Typically, there are many possible connections that break modularity and increase fitness. Thus, even an initially modular solution rapidly evolves into one of many possible non-modular solutions.

Viewed in this perspective, the modularity of biological networks is puzzling. To understand how biological modularity might have evolved, we need to add something to these computer simulations that is biologically plausible.

Here, we will explore a mechanism for the evolution of modularity, using simulated evolution of circuits made of logic gates. These simulations will serve as a metaphor for understanding biological evolution.

15.5 SIMULATED EVOLUTION OF CIRCUITS MADE OF LOGIC GATES

Evolving circuits is fun and can teach us about basic aspects of evolution. We will evolve circuits made of Boolean logic gates. Each gate has two inputs that can be 0 or 1. For example, an AND gate outputs 1 only if both inputs are 1, and outputs zero otherwise (Figure 15.5). An XOR gate (exclusive-OR) outputs 1 if either input is 1, but not both

X	Y	X AND Y	X XOR Y	X NAND Y
0	0	0	0	1
0	1	0	1	1
1	0	0	1	1
1	1	1	0	0

FIGURE 15.5

(Figure 15.5). A NAND gate (not-AND) is the inverse of AND, and outputs a 1 unless both inputs are 1 (Figure 15.5).

We will use NAND gates as our basic units, because they are universal gates: you can build any logic function from NAND gates. Our circuits will be built of NAND gates wired together – an example with five gates is shown in Figure 15.6. The circuits have input ports, like the four inputs X, Y, Z and W in Figure 15.6. One gate is specified as the circuit’s output.

We’ll evolve the circuits by rewiring them until they meet a certain **goal**. For example, the goal can be to compute the logic statement

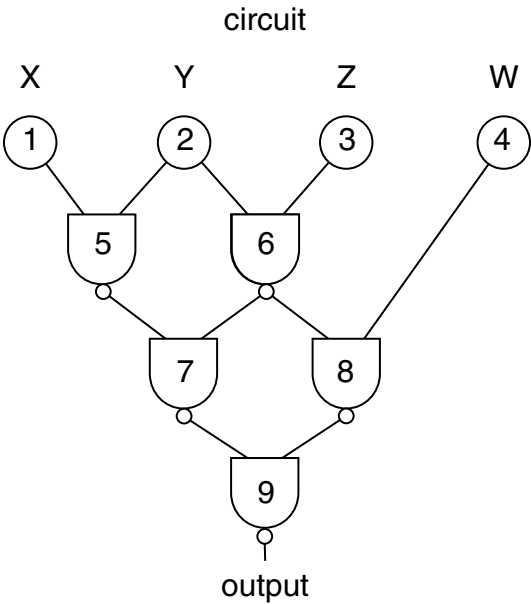


FIGURE 15.6

$$G1 = (X \text{ XOR } Y) \text{ AND } (Z \text{ XOR } W) \tag{15.5.1}$$

The fitness of a circuit can be evaluated by inputting all possible combinations of input values ($2^4 = 16$ combinations for the case of four inputs, because each input can be 0 or 1), and counting the fraction of times that the output node gives the correct output according to the goal G1. To mimic the biological cost of components, we can add a small fitness cost proportional to the number of gates n , $\text{cost} = \varepsilon n$. thus

$$\text{fitness} = \text{fraction of correct outputs} - \varepsilon n \tag{15.5.2}$$

The simulations start with an initial population of N random circuits, say $N = 100$ (Figure 15.7). A fraction p of the circuits is mutated, by randomly rewiring a connection. Each circuit is then evaluated for fitness by providing all possible input combinations and seeing which outputs it gives. The circuits with the highest fitness are selected (say the top 50%), and the rest are discarded. The surviving circuits are replicated, by making an identical copy of the circuit, until we again have a population of N circuits. This is the

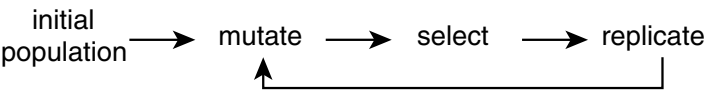


FIGURE 15.7

initial population for the next iteration (Figure 15.7). We repeat this process of mutation, selection, making copies, until we find a circuit that satisfies the goal G1 perfectly. Each such round represents one generation.

The mutation step is inspired by biological mutations. The circuits are described by a genome in which each gate is represented by two numbers that specify its two inputs. The circuit of Figure 15.6 has the genome shown in Figure 15.8. The genome specifies, for example, that gate 7 receives inputs from gates 5 and 6.

A **point mutation** means changing one of the connections in the circuit (Figure 15.9), by changing one number in the genome. Other types of mutations add a new gate, or remove a gate. One can also duplicate a gate together with its inputs, a change called **gene duplication**. A special type of mutation, that occurs in biological mating, is **recombination**: a new circuit is made by taking the genomes of two circuits, cutting them at a certain point and pasting the first part of one genome to the second part of the other (Figure 15.10).

Running our simulation, we can plot the fitness of the best circuit in the population as a function of generations (Figure 15.11). The circuits initially have low fitness, as expected for random circuits. Then, after a few tens of generations, fitness begins to rise. It rises again after a few hundred generations, then again after a few thousand, until a perfect circuit is found in about 10,000 generations. Note that the rate of improvement slows down,

genome

inputs	1	2	2	3	5	6	6	4	7	8
gate	5	6	6	7	8	8	9			

FIGURE 15.8

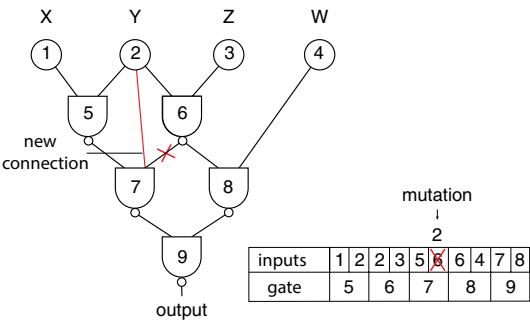


FIGURE 15.9

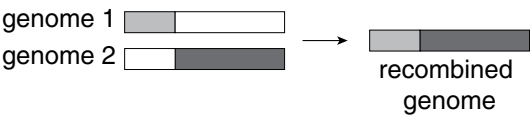


FIGURE 15.10

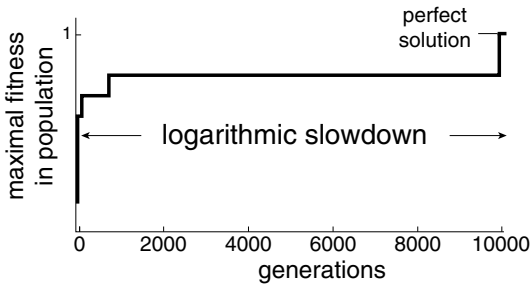


FIGURE 15.11 Adapted from (Kashtan and Alon, 2005).

a phenomenon known as **logarithmic slowdown** that is typical of both simulated evolution and laboratory evolution, such as serial dilution experiments on bacteria discussed in [Chapter 13](#).

As expected, the structure of the evolved perfect solution is non-modular, [Figure 15.12](#). Different simulation runs find different non-modular solutions. The evolved solutions use the minimal possible number of gates for this goal, $n = 10$, due to the cost term in the fitness.

The evolved circuit is non-modular despite the fact that we chose a goal G1 that is modular – it is made of two XORs combined by an AND. Modularity does not evolve in the simulations even if we use different population sizes, or different rates for point mutations, gene duplications and recombinations. Gene duplication and recombination can generate new modules if modules existed before, but they cannot generate functional modules from scratch.

So how do modules arise in biology? To make progress, let's consider the reasons that engineers use modules. Modules are useful in engineering when the design goals change over time. If you use modules, you don't have to build a new device or write new software from scratch. Instead you can reuse old modules and wire them together to meet new goals (Wagner and Altenberg, 1996; Lipson, Pollack and Suh, 2002).

So let's try to switch goals once in a while and see if modularity evolves.

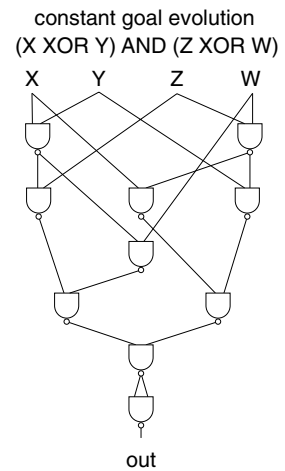


FIGURE 15.12

15.6 RANDOMLY VARYING GOALS CAUSE CONFUSION

We can repeat the same simulations, but switch the goal every $E = 20$ generations. Thus, the circuits start evolving toward one goal G1 for E generations, and then the goal is switched to G2 for E generations, then back to G1, and so on. Suppose we choose goal G2 arbitrarily, for example by generating a Boolean logic function with random 0 and 1 values for its outputs. Switching back and forth between G1 and G2 typically causes fitness to remain low. Perfect solutions are not found. Switching confuses evolution by making it switch directions incessantly.

15.7 MODULARLY VARYING GOALS LEAD TO SPONTANEOUS EVOLUTION OF MODULARITY

Something different happens if we choose a goal G2 that uses some of the same sub-goals of G1. Recall that G1 was

$$G1 = (X \text{ XOR } Y) \text{ AND } (Z \text{ XOR } W) \quad (15.7.1)$$

For G2, let us choose, for example

$$G2 = (X \text{ XOR } Y) \text{ OR } (Z \text{ XOR } W) \quad (15.7.2)$$

This goal uses two XOR functions like G1, but combines them using an OR gate instead of an AND gate.

The biological rationale for using G2 with the same sub-goals as G1 is that the changing goals of biological organisms and molecules often have shared sub-goals. For example, animals need to move, eat and reproduce. But in a new environment some of these sub-goals might change while others remain the same, for example eating different foods without affecting reproduction. Similarly, a protein that needs both to bind a signaling molecule and bind DNA may face evolutionary goals in which one type of binding needs to change without the changing the other.

Switching between two such modular goals is called **modularly varying goals** or **MVG**. Switching between G1 and G2 every $E = 20$ generations, fitness increases and reaches a perfect solution within about 2000 generations (Figure 15.13). Similar results are found for other switching rates E between 5 and 1000 generations.

A perfect solution in MVG means that when the goal is G1, the circuits perfectly solve G1. When the goal switches to G2, fitness drops (because the circuits solve G1 and not G2) but within about a few generations, a perfect solution to G2 evolves and takes over the population (Figure 15.13, inset). Then when the goal switches back to G1, fitness drops and within a few generations a perfect solution to G1 is again found.

Strikingly, the solutions found in MVG evolution are **modular in structure** (Figure 15.14). Each of the modules also has a specific function. There are two XOR modules made of four NAND gates. The outputs of these two XOR modules go to a three-gate module that can rewire between AND and OR by changing two connections (shown in red in Figure 15.14).

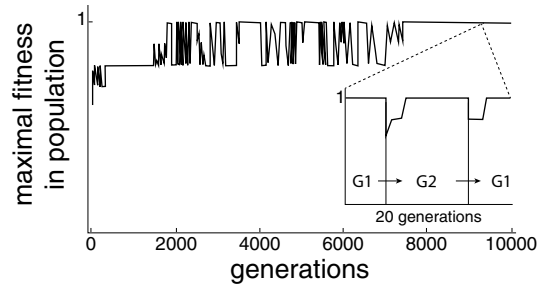


FIGURE 15.13 Adapted from (Kashtan and Alon, 2005).

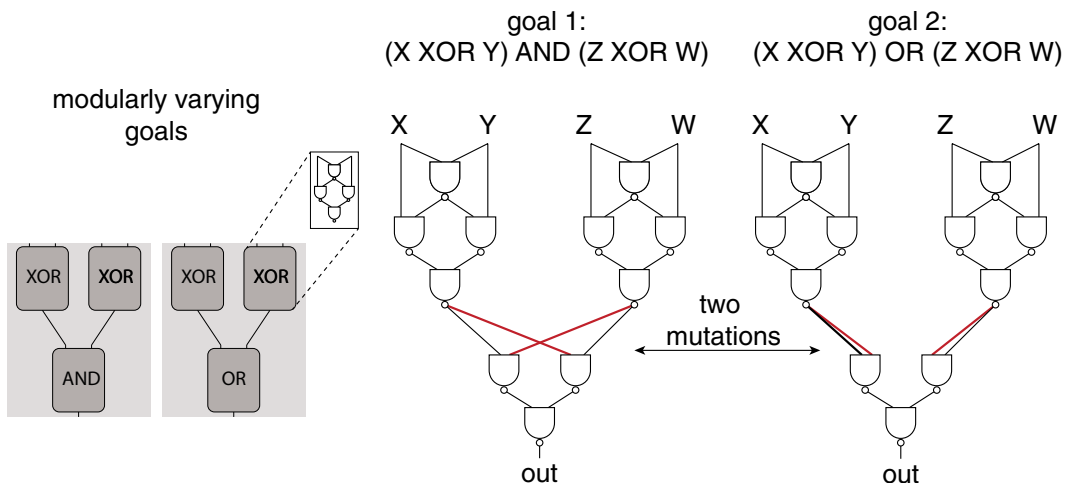


FIGURE 15.14

The modular solution found by MVG uses more gates than the non-modular solutions found when the goal is constant ($n = 11$ instead of $n = 10$). Thus, in terms of the cost of components, modularity is suboptimal. Indeed, if we start with a modular solution and stop switching the goals, modularity decays within tens of generations and evolution converges on a non-modular solution (Figure 15.15). Thus, the need to adapt to a changing environment creates selection pressure to maintain modularity against forces that would wash it out.

Why does modularity evolve in MVG?

It's as if evolution **learns the shared sub-**

goals by seeing two examples, G1 and G2. If only one of the goals is constantly presented, the sub-goals cannot be picked out. It's like a student learning to solve equations: after seeing enough examples, the basic recurring steps can be mastered.

At first sight, MVG seems to be a tougher task than evolution toward a constant goal, such as presenting G1 constantly. MVG is tougher because it needs to generate perfect solutions to two goals instead of one. Remarkably, MVG does this in one-fifth the time that it takes constant-goal evolution to find a perfect solution to one goal (about 2000 versus 10,000 generations)!

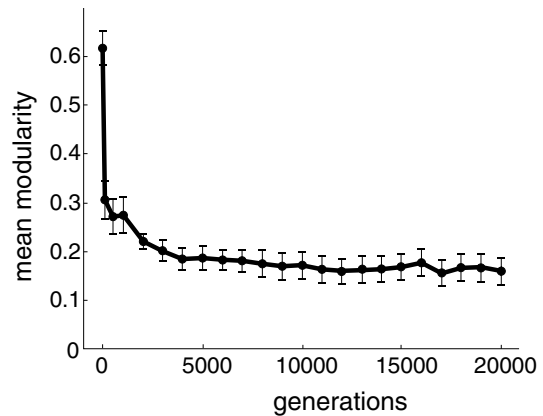


FIGURE 15.15 Adapted from (Kashtan and Alon, 2005).

15.8 THE MORE COMPLEX THE GOAL, THE MORE MVG SPEEDS UP EVOLUTION

The speedup by MVG is also found when evolving circuits that are more complex. For example, one can evolve toward goals with six inputs, such as $G1 = (X \text{ XOR } Y) \text{ AND } (Z \text{ XOR } W) \text{ OR } (U \text{ XOR } V)$. Here, constant-goal evolution takes much longer than the four-input goal G1 above, rising to millions of generations. As always, it gives rise to non-modular circuits. Using MVG that periodically switches between several goals G1, G2, G3 and G4 that each change only one of the sub-goals (e.g., an AND to an OR) leads to evolution of modular circuits. And it does so faster than when the goal is constant.

It turns out that the speedup provided by MVG is larger the more complex the goal. To quantify this, we can define goal

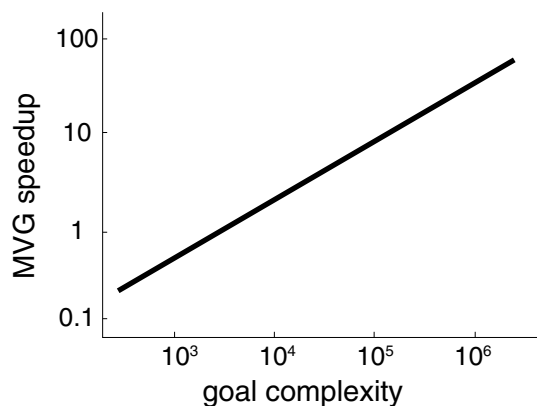


FIGURE 15.16

complexity C as the number of generations it takes to evolve it on average in constant-goal evolution. Speedup S is how much faster MVG evolves a perfect solution for the same goal. Simulations show that speedup rises with complexity, $S \sim C^a$ with $a \sim 0.7 - 0.9$ (Figure 15.16; Kashtan, Noor and Alon, 2007). For example, MVG can evolve a circuit that would take 10^9 generations under a constant goal in only 10^6 generations, a thousand times faster.

How does speedup work? Constant-goal evolution is slow because it often gets stuck in local fitness maxima (Figure 15.17a). This is the reason for the logarithmic slowdown of Figure 15.10. MVG helps evolution get unstuck by changing the fitness landscape every time the goal changes (Figure 15.17b). Thus, a local fitness maximum for the previous goal is no longer a local maximum for the new goal. When the varying goals share the same sub-goals, it seems that a local maximum for G1 is usually replaced by a high-slope region for G2 and vice-versa, because circuit modules useful for one goal provide benefit also for the other goal. Varying the goals again and again thus creates an effective fitness ramp. Evolution climbs this ramp until it reaches perfect modular solutions for the two goals that can reach each other by a few mutations (Figure 15.17c).

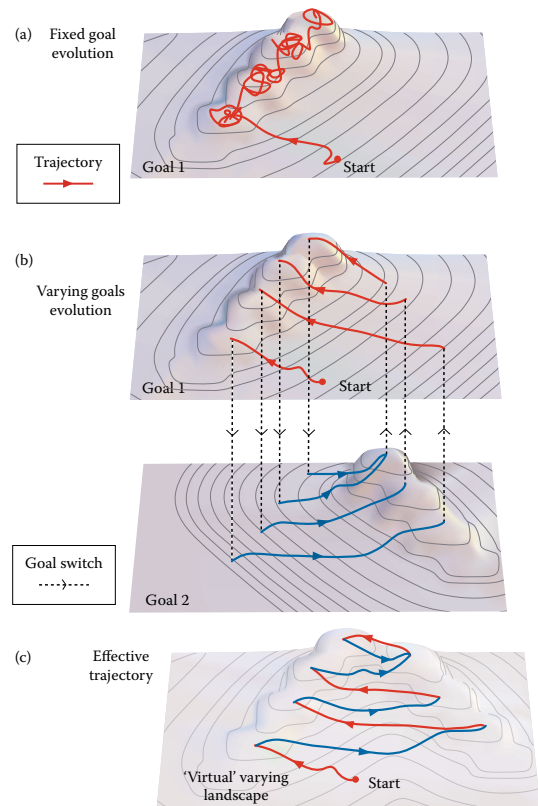


FIGURE 15.17

15.9 MODULAR GOALS AND BIOLOGICAL EVOLUTION

We used simulations of circuit evolution as a metaphor for biological evolution. MVG is one mechanism for evolution of modularity and speedup of evolution, but by no means the only one. Modularity is enhanced, for example, when there is a “wiring” cost for connections between components, as in neuronal and vascular systems (Wagner, Pavlicev and Cheverud, 2007; Clune, Mouret and Lipson, 2013). Simulation is a good way to experiment and test mechanisms for modularity and other hallmarks of evolved systems.

Does MVG happen in real biological evolution? One prediction of MVG is that the more variable the goal, the more modular the structure. For example, the ribosome, a machine that has had the same primary goal for billions of years – to translate the genetic code – is an example of a non-modular complex of about 50 proteins and RNA molecules. It is difficult to say what each individual protein subunit does in the ribosome. Lack of variation in goals leads to a non-modular, fully wired solution.

In contrast, a system like bacterial chemotaxis appears in almost all bacterial species. In each case, different ligands are detected, and often different types of motor systems are engaged. The chemotaxis system is modular, with receptors for detecting inputs, chemotaxis kinases and phosphatases for information processing and motors for moving. You can replace a receptor and get chemotaxis to a new ligand. A non-modular design might, for example, have put signal-detection in the same protein complex as signal-processing and moving.

We conclude this chapter by returning to our mammalian ancestor that evolved to bats, whales, cats and primates. Animals have universal sub-goals such as breathing, eating, moving and reproducing. Each of these sub-goals is met by specialized organs – limbs for moving, lungs for breathing. Our mammalian ancestor had modules for each of these sub-goals. Its new niches presented a host of new combinations and modifications of these sub-goals. Because of the modularity, each module, such as the limb, could evolve without ruining the other modules, such as the lung.

Let's zoom in on the mammalian limb (Figure 15.18). All mammals share the same basic bone structure in their forelimb: one humerus, then the pair of bones called radius and ulna, then many carpals and then fingers with metacarpals and small phalanges. The same bones were tuned to make a whale fin, a bat wing and a primate hand by changing the bone proportions. Note the serial duplication of the phalanges to make the whale flipper. In this way, the limb could be shaped to swim, fly or grasp using the same modular pattern.

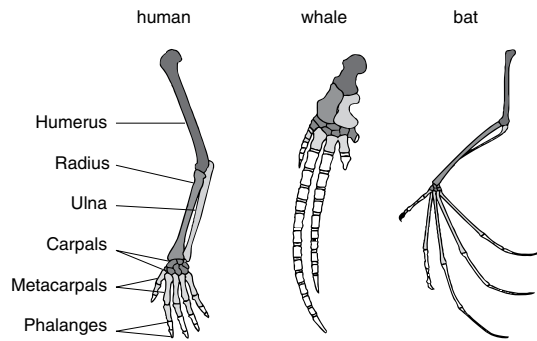


FIGURE 15.18

One might ask how limbs can evolve, given that a mutation that changes a bone length in the limb can be deadly: you need to accordingly change the muscles, nerves, blood vessels and brain region to control the limb. Otherwise, you get a useless appendage, and low fitness: the mutation would not get passed to the next generation. How is coordination between these processes achieved to obtain a useful limb?

Here is a place where engineers can perhaps learn from biology. Development uses a “follow the bone” principle (Kirschner and Gerhart, 2005). The bone lays down the structure, and the rest of the tissues use a random search strategy to find their place. For example, cells lacking oxygen emit a signal that attracts blood vessels. Blood vessels grow and shrink until they find this signal, which stabilizes the blood vessels and provides oxygen where needed. If you look at the veins in your arms, you will see that each arm has a different vein pattern as a result of this random search strategy. Similar exploratory processes allow muscles to grow and shrink until they find bones that have not yet been connected. Nerves grow and shrink until they find muscle that has not been innervated. Modular brain regions such as cortical barrels develop and expand when they get the proper neural input. In this way, a change in the bone can allow the coordinated formation of an entire functional limb.

It is likely that many more principles remain to be discovered in biological systems and their evolution.

FURTHER READING

Detecting and Quantifying Modularity

(Girvan and Newman, 2002) “Community structure in social and biological networks.”

(Sporns and Betzel, 2016) “Modular brain networks.”

Modularity in Biology

(Bhattacharyya et al., 2006) “Domains, motifs and scaffolds: the role of modular interactions in the evolution and wiring of cell signaling circuits.”

(Clune, Mouret and Lipson, 2013) “The evolutionary origins of modularity.”

(Kirschner and Gerhart, 2005) “The plausibility of life.”

(Simon, 1969) “The architecture of complexity.”

(Wagner, Pavlicev and Cheverud, 2007) “The road to modularity.”

Modularly Varying Goals

(Kashtan and Alon, 2005) “Spontaneous evolution of modularity and network motifs.”

(Kashtan, Noor and Alon, 2007) “Varying environments can speed up evolution.”

(Parter, Kashtan and Alon, 2008) “Facilitated variation: how evolution learns from past environments to generalize to new environments.”

EXERCISES

- 15.1 *Spectrum of modularity*: Describe an example of a modular biological system, and an example of a biological system that is more non-modular. Can you estimate which system has more variable goals?
- 15.2 *Hierarchical modularity*: A hierarchically modular system can be decomposed into modules, each of which can be decomposed into further modules.
 - a. Give an example from engineering, in a device or software. Give an example from biology.
 - b. What might be the use of hierarchical modularity?
 - c. How would you design an MVG simulation to test how hierarchical modularity might evolve?
- 15.3 *Evolving XOR*: Write a code that does simulated evolution on circuits of NAND gates, with two input ports, toward the goal $G1 = X \text{ XOR } Y$.
 - a. Plot the mean and maximal fitness as a function of generations.
 - b. Do you observe steps in the evolutionary search?
- 15.4 *Universal gate*: Use NAND gates to produce circuits that act as AND, OR, XOR and EQ gates (EQ outputs 1 only if both inputs are equal).
- 15.5 *All languages have common features such as nouns and verbs*: The brain displays regions which, to some approximation, can be considered as dedicated to different aspects of language. Discuss this organization in terms of MVG.

15.6 *MVG switching rate*: In MVG, goals are changed every E generations.

- a. What determines the fastest rate at which goals can be changed, and the slowest rate, in order to obtain modular solutions?
- b. What happens if E is too small or too large?

15.7 *Bowties* (Csete and Doyle, 2004; Friedlander et al., 2015): Many biological systems show the following feature, called a bowtie or hourglass: Many inputs funnel into a few intermediates, which fan out into many outputs. Examples include metabolism in which many input nutrients are broken down to build twenty amino acid intermediates that form thousands of proteins in the cell.

- a. Provide three other examples of bowties in biology.
- b. Bowties appear in engineering. Diverse energy sources (coal, solar, wind, etc.) are converted to 110/220 V electricity, which powers many output devices. What might be the use of such a bowtie design?
- c. What was the design used to couple energy sources to devices before the electrical revolution?
- d. Provide another example of bowties in technology or economics, and discuss their historical precedents.
- e. How might a bowtie structure evolve in biological systems? Suggest a simulation study to explore this.
- f. Given a set of varying goals, what might determine the number of intermediates that evolve (the size of the ‘waist’ in the bowtie)?

BIBLIOGRAPHY

- Bhattacharyya, R.P. et al. 2006. ‘Domains, motifs, and scaffolds: the role of modular interactions in the evolution and wiring of cell signaling circuits’, *Annual Review of Biochemistry*, 75(1), pp. 655–680. doi: 10.1146/annurev.biochem.75.103004.142710.
- Clune, J., Mouret, J.B. and Lipson, H. 2013. ‘The evolutionary origins of modularity’, *Proceedings of the Royal Society B: Biological Sciences*, 280(1755), pp. 20122863–20122863. doi: 10.1098/rspb.2012.2863.
- Csete, M. and Doyle, J. 2004. ‘Bow ties, metabolism and disease’, *Trends in Biotechnology*, 22(9), pp. 446–450. doi: 10.1016/j.tibtech.2004.07.007.
- Friedlander, T. et al. 2015. ‘Evolution of bow-tie architectures in biology’, *PLoS Computational Biology*, 11(3), p. e1004055. doi: 10.1371/journal.pcbi.1004055.
- Girvan, M. and Newman, M.E.J. 2002. ‘Community structure in social and biological networks’, *Proceedings of the National Academy of Sciences of the United States of America*. National Academy of Sciences, 99(12), pp. 7821–7826. doi: 10.1073/pnas.122653799.
- Kashtan, N. and Alon, U. 2005. ‘Spontaneous evolution of modularity and network motifs’, *Proceedings of the National Academy of Sciences*, 102(39), pp. 13773–13778. doi: 10.1073/pnas.0503610102.
- Kashtan, N., Noor, E. and Alon, U. 2007. ‘Varying environments can speed up evolution’, *Proceedings of the National Academy of Sciences*, 104(34), pp. 13711–13716. doi: 10.1073/pnas.0611630104.

- Kirschner, M.W. and Gerhart, J.C. 2006. *The Plausibility of Life: Resolving Darwin's Dilemma*, illustrated by John Norton. Yale University Press, ISBN 0300119771, 9780300119770.
- Lipson, H., Pollack, J.B. and Suh, N.P. 2002. 'On the origin of modular variation', *Evolution*, 56(8), pp. 1549–1556. doi: 10.1111/j.0014-3820.2002.tb01466.x.
- Parter, M., Kashtan, N. and Alon, U. 2008. 'Facilitated variation: how evolution learns from past environments to generalize to new environments', *PLoS Computational Biology*, 4(11), p. e1000206. doi: 10.1371/journal.pcbi.1000206.
- Raman S. et al. 2014. 'Engineering allostery', *Trends Genet.*, 30(12), pp. 521–528. doi:10.1016/j.tig.2014.09.004.
- Sporns, O. and Betzel, R.F. 2016. 'Modular brain networks', *Annual Review of Psychology*, 67(1), pp. 613–640. doi: 10.1146/annurev-psych-122414-033634.
- Thompson, A. 1998. ICES '98 *Proceedings of the Second International Conference on Evolvable Systems: From Biology to Hardware* Pages 13–24.
- Wagner, G.P. and Altenberg, L. 1996. 'Complex adaptations and the evolution of evolvability', *Evolution*, 50(3), pp. 967–976. doi: 10.1111/j.1558-5646.1996.tb02339.x.
- Wagner, G.P., Pavlicev, M. and Cheverud, J.M. 2007. 'The road to modularity', *Nature Reviews Genetics*, 8(12), pp. 921–931. doi: 10.1038/nrg2267.



Taylor & Francis

Taylor & Francis Group

<http://taylorandfrancis.com>

The Input Functions of Genes

Michaelis–Menten and Hill Equations

A.1 BINDING OF A REPRESSOR TO A PROMOTER

This appendix introduces basic models in biochemistry. We begin with understanding the interaction of a repressor protein with DNA and with its inducer. The repressor X binds to a specific DNA site, D , in a promoter. Thus, X and D bind to form a complex, $[XD]$. Transcription of the gene occurs only when the repressor is *not* bound, that is, when D is free. The DNA site can be either free, D , or bound, $[XD]$, resulting in a **conservation equation**:

$$D + [XD] = D_T \quad (\text{A.1.1})$$

where D_T is the total concentration of the site. For example, a single DNA binding site per bacterial cell means that $D_T = 1/\text{cell volume} \sim 1/\mu\text{m}^3 \sim 1 \text{ nM}$. In eukaryotic cells, the volume of the nucleus is on the order of 10–100 μm^3 , making D_T 10–100 times smaller.

The repressor X and its target D diffuse and occasionally collide to form the complex $[XD]$. This process can be described by **mass-action kinetics**: X and D collide and bind each other at a rate k_{on} . The rate of complex formation is thus proportional to the collision rate, given by the product of the concentrations of X and free D :

$$\text{rate of complex information} = k_{\text{on}} XD$$

The complex $[XD]$ falls apart (dissociates) at a rate k_{off} . The rate of change of $[XD]$ based on these collision and dissociation processes is described by

$$d[XD]/dt = k_{\text{on}} XD - k_{\text{off}}[XD] \quad (\text{A.1.2})$$

The rate parameter for the collisions, k_{on} , describes how many collision events occur per second per protein at a given concentration of D , and thus has units of 1/time/concentration. It is useful to remember that k_{on} in biochemical reactions is often limited by the rate of collisions of a diffusing molecule hitting a protein-size target, and has a **diffusion-limited** value of about $k_{\text{on}} \sim 10^8\text{--}10^9 \text{ M}^{-1} \text{ sec}^{-1}$, independent of the details of the reaction. For the case of a transcription factor and DNA, the diffusion limit is usually about ten times higher because of one-dimensional diffusion effects due to sliding of the transcription factor along the DNA (Berg, Winter and von Hippel, 1981).

The off-rate k_{off} , on the other hand, has units of 1/time and can vary over many orders of magnitude for different reactions, because k_{off} is determined by the strength of the chemical bonds that bind X and D .

Equation A.1.2 approaches a steady state in which concentrations do not change with time, $d[XD]/dt = 0$. Solving Equation A.1.2 at steady state, we find that the balance between the collision of X and D and the dissociation of $[XD]$ leads to the chemical equilibrium equation:

$$K_d[XD] = XD \quad (\text{A.1.3})$$

where K_d is the **dissociation constant**,

$$K_d = k_{\text{off}}/k_{\text{on}}$$

The dissociation constant K_d has units of concentration. The larger the dissociation constant, the higher the rate of dissociation of the complex, that is, the weaker the binding of X and D .

Solving for the concentration of free DNA sites, D , using Equations A.1.1 and A.1.3, we find $K_d(D_T - D) = XD$, which yields

$$\frac{D}{D_T} = \frac{1}{1 + X/K_d} \quad (\text{A.1.4})$$

For many repressors, $[XD]$ complexes dissociate within less than 1 sec, (i.e., $k_{\text{off}} > 1 \text{ sec}^{-1}$). Therefore, we can average over times much longer than 1 sec and consider D/D_T to be the probability that site D is free, averaged over many binding and unbinding events.

The probability that the site is free, D/D_T , is a decreasing function of the concentration of repressor X . When there is no repressor, $X = 0$, the site is always free, $D/D_T = 1$. The site has a 50% chance of being free, $D/D_T = 1/2$, when $X = K_d$.

When site D is free, RNA polymerase can bind the promoter and transcribe the gene. The rate of transcription (number of mRNAs per second) from a free site is given by the maximal transcription rate β . (In the main text we used β to denote the rate of protein production. This rate is proportional to the transcription rate times the number of proteins translated per mRNA.)

The maximal transcription rate depends on the DNA sequence and position of the RNA polymerase binding site in the promoter and other factors. It can be tuned by evolutionary selection, for example, by means of mutations that change the DNA sequence of the RNAP binding site. In different genes, $\beta \sim 10^{-4} - 1$ mRNA/sec. The rate of mRNA production, called the **promoter activity**, is β times the probability that site D is free:

$$\text{promoter activity} = \frac{\beta}{1 + \frac{X}{K_d}} \quad (\text{A.1.5})$$

Figure A.1 shows the promoter activity as a function of X (here X is the repressor in its active, DNA binding form, denoted X^* in the main text). When X is equal to K_d , transcription is reduced by 50% from its maximal value. The value of X needed for 50% maximal repression is called the **repression coefficient**.

For efficient repression, enough repressor is needed so that site D is almost always occupied with repressor. From Equation A.1.4, this occurs when repressor concentration greatly exceeds the dissociation constant, such that $X/K_d \gg 1$. This is the case for many repressors, including the *lac* repressor LacI.

So far we've discussed how the repressor binds the promoter and inhibits transcription. To turn the gene system ON, a signal S_x must cause X to unbind from the DNA. We will treat the simplest case, in which a small molecule (an inducer) is the signal. The inducer directly binds to protein X and causes it to assume a molecular conformation where it does not bind D with high affinity. Typically, signals can reduce the affinity of X to its DNA sites by a factor of 10–100. Thus, the inducer frees the promoter and allows transcription of the gene. We now consider the binding of inducer to X .

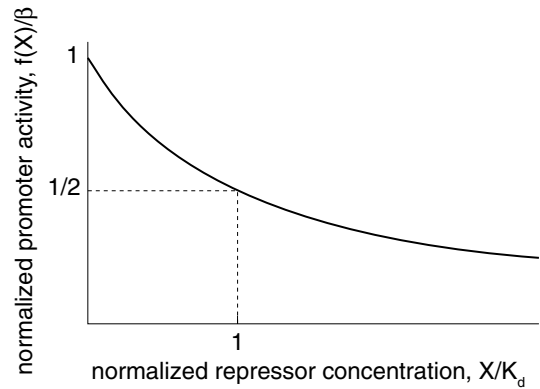


FIGURE A.1

A.2 BINDING OF AN INDUCER TO A REPRESSOR PROTEIN: THE MICHAELIS–MENTEN EQUATION

The repressor protein X is designed to bind a small-molecule inducer S_x , which can be considered its input signal. The two can collide to form a bound complex, $[XS_x]$. The repressor is therefore found in either free form, X , or bound form, $[XS_x]$. A conservation law states that the two forms sum up to the total concentration of repressor protein X_T :

$$X_T = X + [XS_x] \quad (\text{A.2.1})$$

We do not need a conservation equation for S_x because usually the number of S_x molecules is much larger than the number of X molecules, and so almost all molecules of S_x are unbound. For example, in the *lac* system, the number of LacI repressors, each made of a tetramer of LacI proteins, is $X_T \sim 10$ units/cell, which is negligible relative to S_x , which is at least 1000/cell for a detectable response.

According to mass-action kinetics, X and S_x collide to form the complex $[XS_x]$ at a rate k_{on} , and the complex $[XS_x]$ falls apart (dissociates) at a rate k_{off} . Thus, the equation is:

$$d[XS_x]/dt = k_{\text{on}}XS_x - k_{\text{off}}[XS_x] \quad (\text{A.2.2})$$

At steady state, $d[XS_x]/dt = 0$, and we find the chemical equilibrium relation:

$$K_x[XS_x] = XS_x \quad (\text{A.2.3})$$

where $K_x = k_{\text{off}}/k_{\text{on}}$ is the dissociation constant. For the *lac* repressor, $K_x \sim 1 \mu\text{M} \sim 1000$ inducer (IPTG) molecules/cell. Using the diffusion-limited value for $k_{\text{on}} \sim 10^9/\text{M}/\text{sec}$, we find the lifetime of the complex is $1/k_{\text{off}} \sim 1$ msec.

Using the conservation of total repressor X (Equation A.2.1), we arrive at a useful equation that recurs throughout biology. This equation is known as the **Michaelis–Menten equation** in the context of enzyme kinetics; we use the same name in the present context of inducer binding:

$$[XS_x] = \frac{X_T S_x}{S_x + K_x} \quad \text{Michaelis–Menten equation} \quad (\text{A.2.4})$$

The Michaelis–Menten term (Figure A.2) has three notable features:

1. It rises approximately linearly with S_x when S_x is low ($S_x \ll K_x$).
2. It reaches saturation (stops rising) at high S_x .
3. It is half maximal when $S_x = K_x$.

The dissociation constant thus provides the scale for detection of signal: S_x concentrations far below K_x are not detected; concentrations far above K_x saturate the repressor at its maximal binding.

Recall that in cases like LacI, only X *unbound* to S_x , is active, X^* , in the sense that it can bind the promoter D to block transcription. Because free X is active, we denote it by X^* . Active repressor, $X^* = X_T - [XS_x]$, decreases with increasing inducer levels:

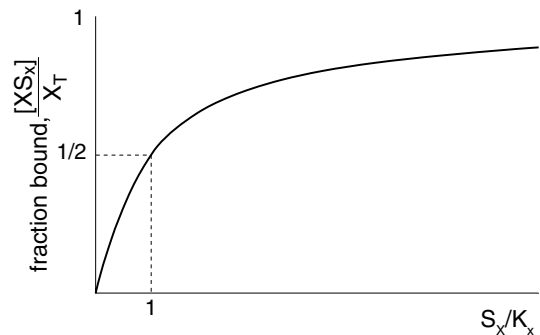


FIGURE A.2

$$X^* = \frac{X_T}{1 + S_x/K_x} \quad \text{concentration of } X \text{ not bound to } S_x \quad (\text{A.2.5})$$

A.3 COOPERATIVITY OF INDUCER BINDING AND THE HILL EQUATION

Before returning to the input function, we comment on a more realistic description of inducer binding. Most transcription factors are composed of several repeated protein subunits, for example, dimers or tetramers. Each of the protein subunits can bind inducer molecules. Often, full activity is only reached when multiple subunits bind the inducer. A useful phenomenological equation for this process, called the **Hill function**, can be derived by assuming that n molecules of S_x can bind X .

To describe the binding process, we assume a simple case: the protein (multimer) X can either be bound to n molecules of S_x , described by the complex $[nS_x X]$, or unbound, denoted X_o (thus, in this simple treatment, intermediate states where fewer than n molecules are bound are neglected). The total concentration of bound and unbound X is X_T , and the conservation law is thus

$$[nS_x X] + X_o = X_T \quad (\text{A.3.1})$$

The complex $[nS_x X]$ is formed by collisions of X with n molecules of S_x . Thus, the rate of the molecular collisions needed to form the complex is given by the product of the concentration of free X , X_o and the concentration of S_x to the power n (the probability of finding n copies of S_x at the same place at the same time):

$$\text{collision rate} = k_{\text{on}} X_o S_x^n \quad (\text{A.3.2})$$

where the parameter k_{on} describes the on-rate of complex formation. The complex $[nS_x X]$ dissociates with rate k_{off} :

$$\text{dissociation rate} = k_{\text{off}} [nS_x X] \quad (\text{A.3.3})$$

The total rate of change of the concentration of the complex is thus the difference between the rate of collisions and dissociations:

$$d[nS_x X]/dt = k_{\text{on}} X_o S_x^n - k_{\text{off}} [nS_x X] \quad (\text{A.3.4})$$

This equation reaches equilibrium within milliseconds for typical inducers. Hence, we can make a steady-state approximation, in which $d[nS_x X]/dt = 0$, to find that dissociations balance collisions:

$$k_{\text{off}} [nS_x X] = k_{\text{on}} X_o S_x^n \quad (\text{A.3.5})$$

We can now use the conservation equation (Equation A.3.1) to replace X_0 with $X_T - [nS_x X]$, to find

$$(k_{\text{off}}/k_{\text{on}})[nS_x X] = (X_T - [nS_x X])S_x^n \quad (\text{A.3.6})$$

Finally, we can solve for the fraction of bound X , to find a binding equation known as the Hill equation:

$$\frac{[nS_x X]}{X_T} = \frac{S_x^n}{K_X^n + S_x^n} \quad \text{Hill equation} \quad (\text{A.3.7})$$

where we have defined the constant K_X such that

$$K_X^n = k_{\text{off}}/k_{\text{on}} \quad (\text{A.3.8})$$

Equation A.3.7 can be considered the probability that the site is bound, averaged over many binding and unbinding events of S_x .

The parameter n is known as the Hill coefficient. When $n = 1$, we obtain the Michaelis–Menten function (Equation A.2.4). As shown in Figure A.3, both the Michaelis–Menten and Hill equations reach half-maximal binding when $S_x = K_X$.

The larger the Hill coefficient n , the steeper the Hill curve (Figure A.3). In the

lac system, $n = 2$ with the inducer IPTG (Yagil and Yagil, 1971). Reactions described by Hill coefficients $n > 1$ are often termed cooperative reactions.

The concentration of unbound repressor X is given by:

$$\frac{X^*}{X_T} = \frac{1}{1 + \left(\frac{S_x}{K_X}\right)^n} \quad (\text{A.3.9})$$

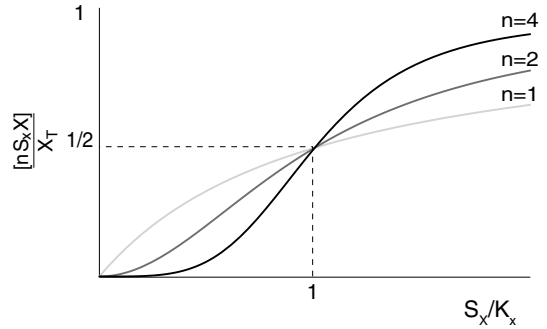


FIGURE A.3

A.4 THE MONOD–CHANGEUX–WYMAN MODEL

We note that a more rigorous and elegant analysis of cooperative binding based on symmetry principles is due to Monod, Changeux and Wyman, in a paper well worth reading (Monod, Wyman and Changeux, 1965), usually also described in biochemistry textbooks. This approach includes all of the intermediate states in which a multimer can bind different numbers of inducer molecules. In this model X switches to an active state X^*

and back. The signal S_x binds X with dissociation constant K_x , and binds X^* with a lower dissociation constant K_x^* . Up to n molecules of S_x can bind to X . The two states, X and X^* spontaneously switch such that in the absence of S_x , X is found at a probability larger by L than X^* . The result is:

$$\frac{X^*}{X_T} = \frac{\left(1 + \frac{S_x}{K_x^*}\right)^n}{L \left(1 + \frac{S_x}{K_x}\right)^n + \left(1 + \frac{S_x}{K_x^*}\right)^n} \quad (\text{A.4.1})$$

Interesting extensions to this model make analogies to Ising models in physics (Duke et al., 2001). One difference between the rigorous models and the Hill curve is that binding at low concentrations of S_x is linear in S_x rather than a power law with coefficient n , as in Equation A.3.7. This linearity is due to the binding of a single site on X , rather than all sites at once.

A.5 THE INPUT FUNCTION OF A GENE REGULATED BY A REPRESSOR

We can now combine the binding of inducer to the repressor (Equation A.2.5) and the binding of the repressor to the DNA (Equation A.1.4) to obtain the input function of the gene. The input function in this case describes the rate of transcription as a function of the input inducer concentration S_x :

$$f(S_x) = \frac{\beta}{1 + X^*/K_d} = \frac{\beta}{1 + \frac{X_T}{K_d} \frac{1}{1 + \left(\frac{S_x}{K_x}\right)^n}} \quad (\text{A.5.1})$$

Figure A.4 shows how the transcription rate of a gene repressed by X increases with increasing inducer concentration S_x . Note, when no inducer is present, there is a **leakage transcription rate**, $f(S_x = 0) = \beta/(1 + X_T/K_d)$, also called the **basal promoter activity**. This leakage is smaller the stronger X binds its DNA site, that is the larger X_T/K_d . In Figure A.4, the parameter values are $X_T/K_d = 10$ (top curve) and $X_T/K_d = 50$ (bottom curve), both with $n = 2$. Half-maximal induction is reached at $S_x = 3K_x$ and $S_x = 7K_x$, respectively. The half-maximal induction point, $S_x = S_{1/2}$, is approximately (when $X_T \gg K_d$)

$$S_{1/2} \sim \left(\frac{X_T}{K_d}\right)^{\frac{1}{n}} K_x \quad (\text{A.5.2})$$

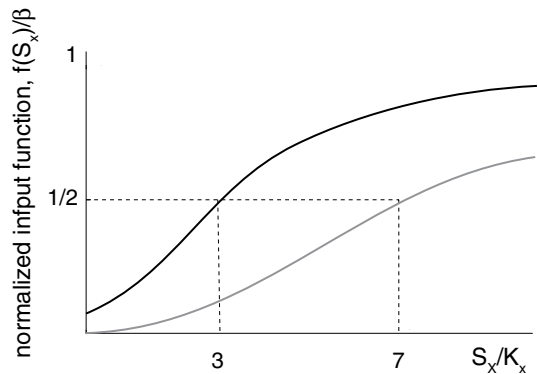


FIGURE A.4

The halfway inducer concentration $S_{1/2}$ can be significantly larger than K_X (Figure A.4). For LacI, for example, $X_T/K_d \sim 100$ and $n = 2$, so that $S_{1/2} \sim 10 K_X$.

We now turn to describe transcription activators.

A.6 BINDING OF AN ACTIVATOR TO ITS DNA SITE

In the decade following the discovery of the *lac* repressor, other gene systems were found to have repressors with a similar principle of action. It is interesting that it took several years for the scientific community to accept evidence that there also existed transcriptional activators.

An activator protein increases the rate of transcription when it binds to its DNA site in the promoter. The rate of transcription is thus proportional to the probability that the activator X is bound to D . Using the same reasoning as above, the binding of X to D is described by a Michaelis–Menten function:

$$\text{promoter activity} = \frac{\beta X^*}{X^* + K_d} \quad (\text{A.6.1})$$

Many activators have a specific inducer, S_x , such that X is active, X^* , in the sense that it can bind DNA to activate transcription, only when it binds S_x .¹ Thus, we obtain

$$X^* = \frac{X_T S_x^n}{K_X^n + S_x^n} \quad (\text{A.6.2})$$

The gene's input function is

$$f(S_x) = \beta X^* / (K_d + X^*) \quad (\text{A.6.3})$$

This function, shown in Figure A.5, is an increasing function of signal. The basal transcription level is zero in this regulation function, $f(S_x = 0) = 0$. Simple activators thus can have lower leakage than repressors. If needed, however, a nonzero basal level can be readily achieved by allowing RNAP to bind and activate the promoter to a certain extent even in the absence of activator.

The inducer level needed for half-maximal induction of an activator can be much smaller than K_X :

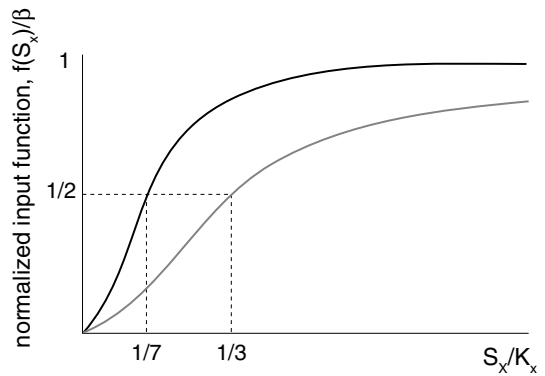


FIGURE A.5

² In other cases the activator is active when it is not bound to S_x and inactive when it is bound. In such cases, S_x is an inhibitor of X . Similarly, some repressors can be activated by binding S_x . These cases can be readily described using the reasoning in this appendix.

$$S_{1/2} \sim \left(\frac{K_d}{X_T} \right)^{\frac{1}{n}} K_X \quad (\text{A.6.4})$$

in contrast to the repressor case (Equation A.5.2). In [Figure A.5](#), for example, $S_x \sim 1/3 K_x$ and $S_x \sim 1/7 K_x$ for the cases of $X_T/K_d = 10$ (bottom curve) and $X_T/K_d = 50$ (top curve), both with $n = 2$.

Overall, similar input function shapes as a function of inducer S_x can be obtained with either activator or repressor proteins. Rules that govern the choice of activator or repressor for a given gene are discussed in [Chapter 7](#).

In this appendix, we described a simplified model that captures the essential behavior of a simple gene regulation system, in which proteins are transcribed at a rate that increases with the amount of inducer S_x . Many real systems have additional important details that make them tighter and sharper switches. The present description is sufficient, however, to understand basic circuit elements in transcription networks.

A.6.1 Comparison of Dynamics with Logic and Hill Input Functions

How good is the approximation of using logic input functions (see Section 1.3.4) instead of graded functions like Hill functions? In [Figure A.6](#), the dynamics of accumulation of a simple one-step transcription cascade are shown, using three different forms of the input function $f(X)$. The input functions are Hill functions with $n = 1$ and $n = 2$, and a logic input function. At time $t = 0$, X^* starts to be produced, and its concentration increases gradually with time. The graded input functions show expression as soon as X^* appears, whereas the logic input function shows expression only when X^* crosses the threshold K . Overall, the qualitative dynamics in this cascade are similar for all three input functions.

A.7 MICHAELIS–MENTEN ENZYME KINETICS

We now briefly describe a useful model of the action of an enzyme X on its substrate S , to catalyze formation of product P . Enzyme X and substrate S bind with rate k_{on} to form

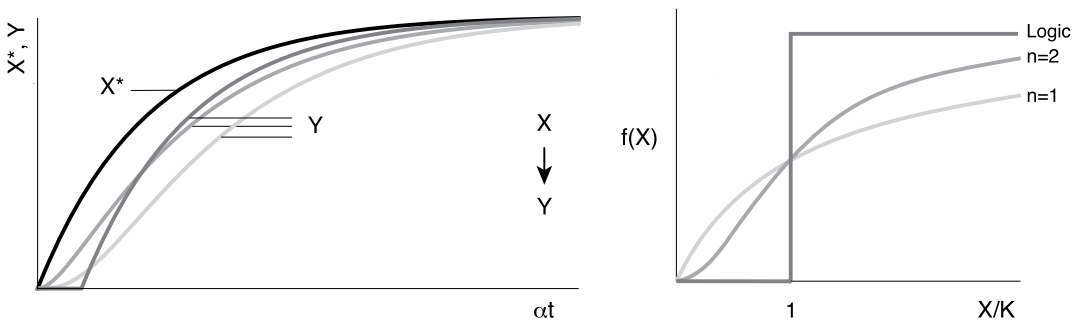


FIGURE A.6

a complex $[XS]$, which dissociates with rate k_{off} . This complex has a small rate ν to form product P , so that



The rate equation for $[XS]$, taking into account the dissociation of $[XS]$ into $X + S$, as well as into $X + P$, is

$$d[XS]/dt = k_{\text{on}}XS - k_{\text{off}}[XS] - \nu[XS] \quad (\text{A.7.2})$$

At steady state, we obtain

$$[XS] = k_{\text{on}}/(\nu + k_{\text{off}})XS \quad (\text{A.7.3})$$

If substrate S is found in excess, we need only worry about the conservation of enzyme X :

$$X + [XS] = X_T \quad (\text{A.7.4})$$

Using this in Equation A.7.3, we find the Michaelis–Menten equation:

$$\nu[XS] = \nu \frac{X_T S}{K_m + S} \quad \text{Michaelis – Menten equation} \quad (\text{A.7.5})$$

where the Michaelis–Menten coefficient of the enzyme is:

$$K_m = (\nu + k_{\text{off}})/k_{\text{on}} \quad (\text{A.7.6})$$

This constant has units of concentration and is equal to the concentration of substrate at which the production rate is half-maximal. When substrate is saturating, $S \gg K_m$, production is at its maximal rate, equal to νX_T . Thus, the production rate does not depend on S , (i.e., it depends on S to the power zero) and is known as **zero-order kinetics**:

$$\text{production rate} = \nu X_T \quad \text{zero-order kinetics} \quad (\text{A.7.7})$$

In the main text we will sometimes make an approximation to this function when the substrate S is found in low concentrations, $S \ll K_m$. In this case, the production rate becomes linear in S , as can be seen from Equation A.7.5 by neglecting S in the denominator. This regime is known as **first-order kinetics**:

$$\text{production rate} = \nu X_T \frac{S}{K_m} \quad \text{first-order kinetics} \quad (\text{A.7.8})$$

FURTHER READING

- (Ackers, Johnson and Shea, 1982) “Quantitative model for gene regulation by lambda phage repressor.”
 (Berg et al., 2002) “Biochemistry enzymes: basic concepts and kinetics.”
 (Monod, Wyman and Changeux, 1965) “On the nature of allosteric transitions: a plausible model.”
 (Ptashne, 2004) “Genetic switch: phage lambda revisited.”
 (Setty et al., 2003) “Detailed map of a cis-regulatory input function.”

EXERCISES

- A.1 Given a simple repressor with parameters β , X_p , K_d , K_x and n , design an activator that best matches the performance of the repressor. That is, assign values to the parameters for the activator so its input function will have the same maximal expression, and the same $S_{1/2}$, and the same slope around $S_{1/2}$ as the repressor input function.
- A.2 Derive the approximate value of diffusion-limited k_{on} based on **dimensional analysis**. Dimensional analysis seeks a combination of the physical parameters in the problem that yields the required dimensions. If only one such combination exists, it often supplies an intuitive approximate solution to otherwise complicated physical problems. Assume a target protein with a binding site of area $a = 1 \text{ nm}^2$, and a small molecule ligand that diffuses with diffusion constant $D = 1000 \text{ } \mu\text{m}^2/\text{sec}$. The affinity of the site is so strong that it binds all ligand molecules that collide with it.

Solution:

To study the on-rate k_{on} , imagine a single protein in a solution of 1 M ligand L (concentration of ligand is $\rho = 1 \text{ M} = 6 \cdot 10^{23} \text{ molecules/liter} \sim 10^9 \text{ molecules}/\mu\text{m}^3$). The number of L molecules colliding with the binding site of the protein has dimensions of molecules/sec and should be constructed from ρ , D and a . The combination with the desired dimensions is $k_{on} \sim \rho D a^{1/2}$, because D has units of $[\text{length}]^2/[\text{time}]$ and a has units of $[\text{length}]^2$. This combination makes sense: it increases with increasing ρ , a and D as expected. Inserting numbers, we find $k_{on} \sim \rho D a \sim 10^9 \text{ molecules}/\mu\text{m}^3 \cdot 1000 \text{ } \mu\text{m}^2/\text{sec} \cdot 10^{-3} \text{ } \mu\text{m} = 10^9 \text{ molecules/sec}$, hence $k_{on} \sim 10^9/\text{M}/\text{sec}$. Note that dimensional analysis neglects dimensionless prefactors and is often only accurate to within an order of magnitude (Milo and Philipps, 2015).

- A.3 What is the expected diffusion-limited k_{on} for a protein sliding along DNA to bind a DNA site? The protein is confined to within $r = 1 \text{ nm}$ of the DNA. The total length of DNA in a bacterium such as *E. coli* is on the order of 1 mm (!), and the volume of the *E. coli* cell is about $\sim 1 \text{ } \mu\text{m}^3$. Discuss the biological significance of the increase in k_{on} relative to free diffusion in space.
- A.4 *Off-times*
- Estimate the off-time ($1/k_{off}$) of a diffusion-limited repressor that binds a site with $K_d = 10^{-11} \text{ M}$.

- b. What is the off-time of a small-molecule ligand from a receptor that binds it with $K_d = 10^{-6} M$ (bacterial chemotaxis attractants), $K_d = 10^{-12} M$ (mammalian hormone binding to receptors)?
- c. Mammalian ligands that bind a receptor on the cell surface are often taken up into the cell and destroyed or recycled together with the receptor, in a process called endocytosis. Explain how the ligand can remain bound for long enough if endocytosis takes minutes?

BIBLIOGRAPHY

- Ackers, G.K., Johnson, A.D., and Shea, M.A. 1982. 'Quantitative model for gene regulation by lambda phage repressor', *Proceedings of the National Academy of Sciences*, 79(4), pp. 1129–1133. doi: [10.1073/pnas.79.4.1129](https://doi.org/10.1073/pnas.79.4.1129).
- Berg, J.M. et al. 2002. *Biochemistry*. W.H. Freeman.
- Berg, O.G., Winter, R.B. and von Hippel, P.H. 1981. 'Diffusion-driven mechanisms of protein translocation on nucleic acids. 1. Models and theory'. *Biochemistry*, 20(24), pp. 6929–6948. doi: [10.1021/bi00527a028](https://doi.org/10.1021/bi00527a028).
- Duke, T.A.J., Le Novère, N. and Bray, D. 2001. 'Conformational spread in a ring of proteins: A stochastic approach to allostery', *Journal of Molecular Biology*, 308(3), pp. 541–553.
- Milo, R. and Phillips, R. 2015. *Cell Biology by the Numbers*. Garland Science, New York.
- Monod, J., Wyman, J. and Changeux, J.P. 1965. 'On the nature of allosteric transitions: a plausible model'. *Journal of Molecular Biology*, 12(1), pp. 88–118. doi: [10.1016/S0022-2836\(65\)80285-6](https://doi.org/10.1016/S0022-2836(65)80285-6).
- Ptashne, M. 2004. *A genetic switch: phage lambda revisited*. Cold Spring Harbor Laboratory Press, Cold Spring Harbor NY. doi: [10.1016/j.molcel.2017.02.028](https://doi.org/10.1016/j.molcel.2017.02.028).
- Setty, Y. et al. 2003. 'Detailed map of a cis-regulatory input function', *Proceedings of the National Academy of Sciences of the United States of America*, 100(13), pp. 7702–7707. doi: [10.1073/pnas.1230759100](https://doi.org/10.1073/pnas.1230759100).
- Yagil, G. and Yagil, E. 1971. 'On the relation between effector concentration and the rate of induced enzyme synthesis'. *Biophysical Journal*, 11(1), pp. 11–27. doi: [10.1016/S0006-3495\(71\)86192-1](https://doi.org/10.1016/S0006-3495(71)86192-1).

Multi-Dimensional Input Functions

Many genes are regulated by more than one transcription factor. The combined effect of these regulators is described by a multi-dimensional input function. As an example, we consider a simple case and then discuss the more general forms of the input function.

B.1 INPUT FUNCTION THAT INTEGRATES AN ACTIVATOR AND A REPRESSOR

Let's take a look at an input function that integrates an activator X and a repressor Y at a promoter. How can an activator and repressor work together?

A common situation is that the activator and repressor bind the promoter independently on two different sites. Thus, there are four binding states of promoter D : D , D_X , D_Y and D_{XY} , where D_{XY} means that both X and Y bind to D . Transcription occurs mainly from the state D_X , in which the activator X but not the repressor Y bind. In the following, we use X and Y to denote the active forms X^* and Y^* .

The probability that X is bound is given by the (now familiar) Michaelis–Menten function (Appendix A):

$$P_{X \text{ bound}} = \frac{X}{K_1 + X} = \frac{X/K_1}{1 + X/K_1} \quad (\text{B.1.1})$$

The probability that Y is not bound is given by the Michaelis–Menten term equal to 1 minus the probability of binding:

$$P_{Y \text{ not bound}} = 1 - \frac{Y}{K_2 + Y} = \frac{1}{1 + Y/K_2} \quad (\text{B.1.2})$$

Since the two binding events are independent, the probability that the promoter D is bound to X and not to Y is given by the product of the two probabilities:

$$P_{X \text{ bound AND } Y \text{ not bound}} = P_{X \text{ bound}} P_{Y \text{ not bound}} = \frac{X/K_1}{1 + X/K_1 + Y/K_2 + XY/K_1K_2} \quad (\text{B.1.3})$$

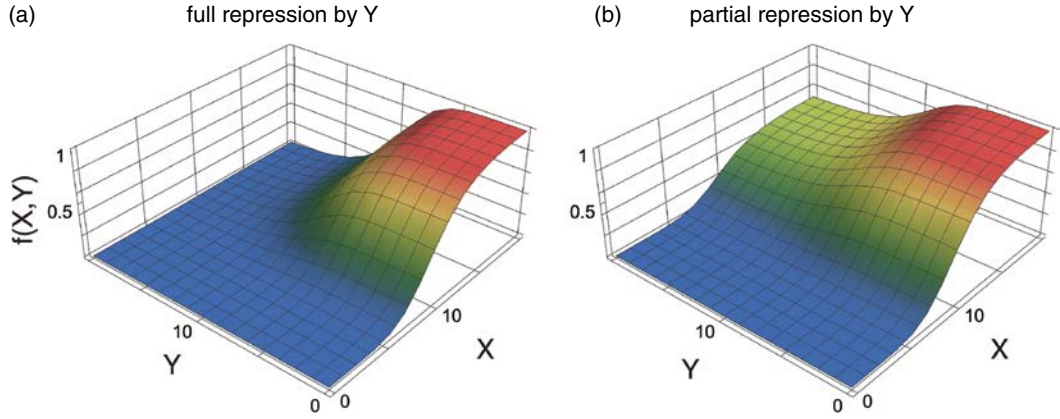


FIGURE B.1

and the output promoter activity is given by the production rate β_z times this probability:

$$P_z = \beta_z X/K_1 / (1 + X/K_1 + Y/K_2 + XY/K_1K_2) \quad (\text{B.1.4})$$

This results in an input function that resembles the logic function *X* AND NOT *Y* shown in Figure B.1a. Fans of statistical physics will recognize the partition function in the denominator of this expression. The relation of partition functions to promoters has been worked out by Ackers, Johnson and Shea (1982) and Buchler, Gerland and Hwa (2003).

In many promoters, when the repressor binds, repression is only partial and there is basal transcription (leakage). In such cases, the state in which both *X* and *Y* bind, D_{XY} , also contributes a transcription rate, $\beta'_z < \beta_z$, to the promoter activity of *Z*:

$$P_z = \frac{\beta_z X/K_1 + \beta'_z XY/K_1K_2}{1 + X/K_1 + Y/K_2 + XY/K_1K_2} \quad (\text{B.1.5})$$

This results in an input function with three plateau levels: zero when $X = 0$, β_z when *X* is high but *Y* is low and β'_z when both are high (Figure B.1b, with $\beta'_z = 0.3$ and $K_x = K_y = 10$). This continuous input function can be approximated by a logic function

$$P_z = \theta(X > K_1)(\beta_z[1 - \theta(Y > K_2)] + \beta'_z \theta(Y > K_2)) \quad (\text{B.1.6})$$

where θ is the step function, equal to 0 or 1.

These results have some generality. The input functions can often be described by the ratio of polynomials of the active concentrations of the input transcription factors X_i , $i = 1, \dots, N$. For example, with activation/repression coefficients K_i and Hill coefficients n_i , one finds

$$f(X_1, \dots, X_N) = \frac{\sum_i \beta_i \left(\frac{X_i}{K_i} \right)^{m_i}}{1 + \sum_i \beta_i \left(\frac{X_i}{K_i} \right)^{n_i}} \quad (\text{B.1.7})$$

For each activator $m_i = n_i$ and for each repressor $m_i = 0$. These types of functions match well with experimentally determined input functions (Setty et al., 2003). More complicated expressions are possible if the different transcription factors interact with each other on the protein level (Buchler, Gerland and Hwa, 2003).

EXERCISE

- B.1 *This promoter ain't big enough*: Activator X and repressor Y bind a promoter. The repressor and activator sites overlap so that X and Y cannot both bind at the same time. What is the resulting input function? How does it differ from the input function obtained from independent binding?

BIBLIOGRAPHY

- Ackers, G.K., Johnson, A.D. and Shea, M.A. 1982. 'Quantitative model for gene regulation by lambda phage repressor', *Proceedings of the National Academy of Sciences*, 79(4), pp. 1129–1133. doi: 10.1073/pnas.79.4.1129.
- Buchler, N.E., Gerland, U. and Hwa, T. 2003. 'On schemes of combinatorial transcription logic', *Proceedings of the National Academy of Sciences*, 100(9), pp. 5136–5141. doi: 10.1073/pnas.0930314100.
- Setty, Y. et al. 2003. 'Detailed map of a cis-regulatory input function'. *Proceedings of the National Academy of Sciences of the United States of America*, 100(13), pp. 7702–7707. doi: 10.1073/pnas.1230759100.



Taylor & Francis

Taylor & Francis Group

<http://taylorandfrancis.com>

Graph Properties of Transcription Networks

C.1 TRANSCRIPTION NETWORKS ARE SPARSE

What is the maximal number of arrows (called in this appendix edges according to the usage in graph theory) in a network with N nodes? Each node can have an outgoing edge to each of the $N - 1$ other nodes, for a total of $E_{\max} = N(N - 1)$ edges. If we also allow self-edges, there are an additional N possible edges, for a total of $E_{\max} = N^2$. Note that a maximally connected network has a pair of edges in both directions (mutual edges) between every two nodes.

The number of edges actually found in transcription networks, E , is much smaller than the maximum possible number of edges. The networks are thus **sparse**, in the sense that $E/E_{\max} \ll 1$. Typically, less than 0.1% of possible edges are found in the network.

Transcription networks are the product of evolutionary selection. It is easy to lose an edge in the network: a single mutation in the binding site of X in the promoter of Y can cause the loss of the interaction. The sparse nature of the network reflects the fact that only very few and specific interactions, with useful function, are selected and appear in the network.

C.2 TRANSCRIPTION NETWORKS HAVE LONG-TAILED OUT-DEGREE SEQUENCES AND COMPACT IN-DEGREE SEQUENCES

Each node in a transcription network represents a gene (or operon). Incoming edges to a node in the network correspond to transcription factors that regulate the gene. The number of edges that point into a node is called the node's **in-degree**. Similarly, the **out-degree** is the number of edges pointing out of a node, corresponding to the number of genes regulated by the transcription factor protein that is encoded by the gene that corresponds to the node.

The mean number of edges per node, called the **mean connectivity** of the network, is $\lambda = E/N$. Typically, λ is on the order of 2–10 edges/node.

Do all nodes have similar degrees? Transcription networks have nodes that show much higher out-degrees than the average node. Many transcription factors regulate a few genes, fewer regulate tens of genes and even fewer regulate hundreds of genes. The latter are called **global regulators** and usually respond to key environmental signals to control

large ensembles of genes. Examples of global regulators in bacteria include CRP, which responds to glucose starvation, and RpoS, which responds to general stresses. Thus, the out-degree distribution has a long tail and can be roughly described as a power law, at least over a certain range (Barabási and Oltvai, 2004). That is, the number of nodes with out-degree k is roughly $P(k) \sim k^{-\gamma}$, with $\gamma \sim 1-2$. Note that the out-degree distribution is only approximately a power law; for example, it is bounded by the total number of genes N .

In contrast to the long tail of the out-degree distribution, the in-degree distribution is concentrated around its average value (Thieffry and Thomas, 1998; Guelzim et al., 2002; Shen-Orr et al., 2002). The in-degrees range between zero and a few times the mean connectivity, λ . There is little chance of finding a node regulated by 10 or 100 times more inputs than the average node. In other words, the in-degree distribution does not have a long tail, and instead resembles compact distributions such as the Poisson distribution, whose standard deviation is about the same as the mean.

The compact distribution of in-degrees may be due in part to a physical limitation. In simple organisms, promoters are short. The region near the RNAP binding site that participates in regulation is on the order of a few hundred base-pairs (DNA letters). There is no space in the promoter region to accommodate more than a few binding sites for transcription factors (each on the order of 10 base-pairs). In more complex organisms, transcription factors can affect a gene even if bound far away on the DNA, through DNA-looping interactions and other effects. Such action-at-a-distance can increase the number of input transcription factors to a given gene. Animal cells often display larger in-degrees than microorganisms, accommodating the computations needed during development – the same DNA must serve many different cell types.

Networks with long-tailed degree distribution are sometimes called “scale-free” because the number of edges per node has no typical scale. Nodes with many more connections than the average are called **hubs**. Many natural and engineered networks have hubs and degree distributions that appear to be power laws over a certain range (Barabási and Oltvai, 2004).

This power-law-like behavior can stem from multiple different reasons, and probably has a different origin in each type of network. A general mechanism for generation of power law connectivity was proposed in the context of networks by Barabási and Albert (1999). In this model, called preferential attachment, new nodes are added to a growing network and connect with higher probability to nodes that already have many connections. This process generates networks with scale-free degree distributions. However, this is not a reasonable model of the evolution of transcription networks in which edges are continually selected for function.

In some communication networks, scale-free distributions have been proposed to afford robustness of network connectivity with respect to the deletion of random nodes. However, robustness to node removal does not appear to be the function of the degree distribution in transcription networks. These networks are often not robust to mutations (deletion of nodes), especially in bacteria. We believe that the origin of long-tailed degree distribution lies in a broad distribution of the benefit of the functions that need to be performed by the cells, and which require partitioning of gene resources into co-regulated modules of widely differing sizes. An interesting theory on the origin of power laws in designed or optimized systems along these lines has been suggested by Carlson and Doyle (2000).

C.3 CLUSTERING COEFFICIENTS

An additional statistical property of graphs is the clustering coefficient, which corresponds to whether the neighbors of a given node are connected to each other. Let us consider the network as nondirected; that is, disregard the direction of the edges. A node with k neighbors can be a part of at most $k(k-1)/2$ triangles, one for each possible pair of neighboring nodes. The clustering coefficient C is the average number of triangles that a node participates in, divided by this maximal number. Transcription networks have average clustering coefficients larger than those of randomized networks.

As described in [Chapter 3](#), network motifs in sensory transcription networks generally include one main type of triangle, the feedforward loop. The major contribution to the clustering coefficient of transcription networks therefore stems from feedforward loops. This pattern appears to be selected due to its functions.

The clustering coefficient can also be measured as a function of the number of neighbors that each node has, resulting in a clustering sequence $C(k)$. Often, $C(k) \sim 1/k$ over a certain range, so that the more neighbors a node has, the lower its clustering coefficient (Barabási and Oltvai, 2004). In transcription networks, this tendency appears to correspond to the way that feedforward loops connect to each other. The chief arrangement of feedforward loops in sensory transcription networks is the multi-output FFL, discussed in [Chapter 4](#). In the multi-output FFL, node X regulates (and is thus a neighbor of) Y , and both X and Y regulate k output nodes. These output nodes are typically not neighbors. Thus, node X has $k+1$ neighbors (Y and the k output nodes), with only k connections between these neighbors (the connections of Y to the outputs), resulting in a clustering coefficient $C \sim k/k^2 \sim 1/k$.

Generally, it appears that global statistical properties of biological networks such as degree sequences and clustering sequences are the result of selection working on the detailed circuit patterns in each individual system. Different networks have different selection constraints, which must be understood in order to understand their graph properties.

C.4 QUANTITATIVE MEASURE OF NETWORK MODULARITY

Network modularity is the degree to which it can be separated into nearly independent sub-networks ([Chapter 15](#)). A quantitative measure of modularity was developed by Newman and Girvan (Newman and Girvan, 2004). Briefly, the Newman and Girvan algorithm finds the division of the nodes into modules that maximizes a measure Q . This measure is the fraction of the edges in the network that connect between nodes within a module minus the expected value of the same quantity in a network with the same assignment of nodes into modules but random connections between the nodes:

$$Q = \sum_{s=1}^K \left(\frac{l_s}{L} - \left(\frac{d_s}{2L} \right)^2 \right) \quad (\text{C.4.1})$$

where K is the number of modules, L is the number of edges in the network, l_s is the number of edges between nodes in module s and d_s is the sum of the degrees of the nodes

in module s . The rationale for this modularity measure is as follows (Guimerà and Amaral, 2005): a good partition of a network into modules must include many within-module edges and as few as possible between-module edges. However, if we try to minimize the number of between-module edges (or equivalently maximize the number of within-module edges), the optimal partition consists of a single module and no between-module edges. Equation C.4.1 addresses this difficulty by imposing $Q = 0$ if nodes are placed at random into modules or if all nodes are in the same module.

This measure can be further refined by normalizing it with respect to randomized networks. The normalized measure Q_m is (Kashtan and Alon, 2005):

$$Q_m = \frac{Q_{\text{real}} - Q_{\text{rand}}}{Q_{\text{max}} - Q_{\text{rand}}} \quad (\text{C.4.2})$$

where Q_{real} is the Q value of the network, Q_{rand} is the average Q value of randomized networks and Q_{max} is the maximal possible Q value of a network with the same degree sequence as the real network. The values of Q_{real} , Q_{rand} and Q_{max} can be calculated by efficient algorithms (Kashtan and Alon, 2005).

The Q_m measure of modularity normalizes out the effects of network size and connectivity. Biological networks show high modularity according to this measure: The transcription network of the bacterium *Escherichia coli* has $Q_m = 0.54$, the neuronal synaptic network of *C. elegans* has $Q_m = 0.54$ and a human signal transduction network has $Q_m = 0.58$.

BIBLIOGRAPHY

- Barabási, A.L. and Albert, R. 1999. 'Emergence of scaling in random networks', *Science*, 286(5439), 509–512, doi: 10.1126/science.286.5439.509.
- Barabási, A.L. and Oltvai, Z.N. 2004. 'Network biology: understanding the cell's functional organization', *Nature Reviews Genetics*, 5(2), 101–113. doi: 10.1038/nrg1272.
- Carlson, J.M. and Doyle, J. 2000. 'Highly optimized tolerance: a mechanism for power laws in designed systems', *Physical Review E*, 60(2), 1412–1427. doi: 10.1103/PhysRevE.60.1412.
- Guelzim, N. et al. 2002. 'Topological and causal structure of the yeast transcriptional regulatory network', *Nature Genetics*, 31(1), 60–63. doi: 10.1038/ng873.
- Guimerà, R. and Amaral, L.A.N. 2005. 'Functional cartography of complex metabolic networks', *Nature*, 433(7028), 895–900. doi: 10.1038/nature03288.
- Kashtan, N. and Alon, U. 2005. 'Spontaneous evolution of modularity and network motifs', *Proceedings of the National Academy of Sciences*, 102(39), 13773–13778. doi: 10.1073/pnas.0503610102.
- Newman, M.E.J. and Girvan, M. 2004. 'Finding and evaluating community structure in networks', *Physical Review E*, 69(2), 268. doi: 10.1103/PhysRevE.69.026113.
- Shen-Orr, S.S. et al. 2002. 'Network motifs in the transcriptional regulation network of *Escherichia coli*', *Nature Genetics*, 31(1), 64–68. doi: 10.1038/ng881.
- Thieffry, D. and Thomas, R. 1998. 'Qualitative analysis of gene networks', in *Proceedings of the Third Pacific Symposium on Biocomputing*.

Noise in Gene Expression

D.1 INTRODUCTION

The concentration of each protein in a population of genetically identical cells varies from cell to cell due to stochastic processes (reviewed in McAdams and Arkin [1997] and Kærn et al. [2005]). The concentration of a given protein often has a **coefficient of variation** (standard deviation divided by the mean) in the range $CV = 0.1$ to 1 (Elowitz et al., 2002; Ozbudak et al., 2002; Blake et al., 2003; Raser and O’Shea, 2004). That is, the cell–cell variations are on the order of tens of percents of the mean. When a protein is

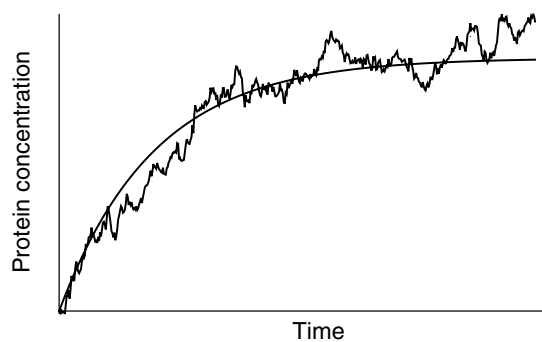


FIGURE D.1

produced and degraded such that its levels change with time, each cell shows fluctuations around the mean trajectory (Figure D.1). This stochastic behavior is called gene-expression **noise**.

Noise is not just a fluke, it’s a major fact of life in biological systems. An organism without any stochastic behavior would seem to us robotic and un-lifelike.

Resisting noise is a central constraint that shapes biological circuits, as described in Part 2 of the book, on robustness. Cells also use noise as a random number generator to produce oscillations, stochastic differentiation, and to provide bet-hedging: noise adds diversity that allows organisms to deal with unexpected situations. The uses of noise are reviewed in Eldar and Elowitz (2010).

D.2 EXTRINSIC AND INTRINSIC NOISE

One important source of noise is **extrinsic noise**, in which the cellular capacity to produce proteins, and the regulatory systems that regulate a gene, fluctuate over time. For example, fluctuations in a transcription factor concentration can affect the expression rate of its target genes. The correlation time of these variations in production rates is often on the scale of a cell generation: that is, a cell with high production levels often tends to stay high for a cell cycle or more (Rosenfeld et al., 2005).

In addition to **extrinsic** noise, there is also **intrinsic noise**, which is due to stochastic variations in the transcription and translation events of the gene. An elegant experiment by Michael Elowitz and colleagues (Elowitz et al., 2002) measured the relative level of intrinsic and extrinsic noise. Elowitz measured the levels of two fluorescent

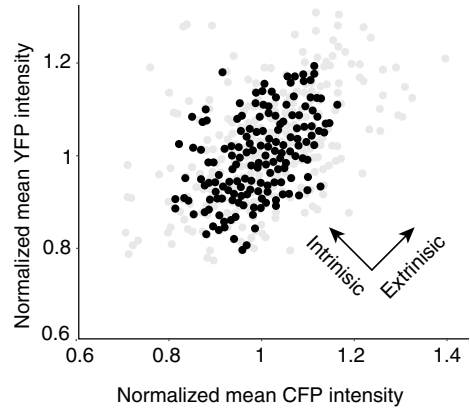
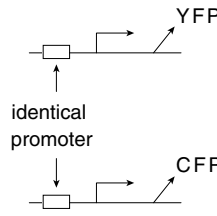


FIGURE D.2 Adapted from (Elowitz et al., 2002).

proteins, one yellow and one cyan (YFP and CFP), expressed from identical promoters (Figure D.2) in the same bacterial cell. Genetically identical cells grown in the same conditions (dots in Figure D.2) showed different amounts of yellow and cyan fluorescence. This variation can be decomposed into a shared component in which yellow and cyan are both high or both low, and a perpendicular component in which the ratio between yellow and cyan varies. The shared component is due to extrinsic noise: the variation shared by the genes due to upstream factors such as variations in regulators and the cells' metabolic capacity. The perpendicular component is due to intrinsic noise that changes the ratios of the two colors, due to stochastic steps in transcription and translation of each gene. Intrinsic noise appears to fluctuate on a timescale of minutes in bacteria (Rosenfeld et al., 2005). The two types of noise are not independent: intrinsic noise in the levels of a regulator can show up as extrinsic noise in expression of its targets.

D.3 DISTRIBUTION OF PROTEIN LEVELS

The cell–cell distribution of the number of copies of a given protein, $P(X)$, is captured by a well-studied model of noise, in which protein is produced in stochastic **bursts**. The bursts originate from various processes (Golding and Cox, 2006), such as a transition of the gene between two states: an ON state in which mRNA is transcribed, and an OFF state in which transcription is zero. The frequency of bursts is a per protein half-life, and in each bursts b proteins are produced on average.

In the case where mRNA lifetime is much shorter than protein lifetime, as is typical of bacteria and yeast, this process results in a Gamma distribution of protein numbers per cell at steady state:

$$P(X) = C X^{a-1} e^{-x/b} \quad (\text{D.3.1})$$

When there are many bursts per protein half-life, $a > 1$, this distribution is approximately bell-shaped and slightly skewed to the right (Figure D.3a). When there is less than one burst per protein half-life, protein levels fluctuate widely over time, and $P(X)$ is sharply skewed with a maximum at $X = 0$ (Figure D.3b). These distributions agree with precise measurements

of protein variation between cells in microorganisms (Friedman, Cai and Xie, 2006) and mammalian cells (Cohen et al., 2009).

A more precise calculation makes adjustments for proteins of very low abundance (b close to one) yielding a negative binomial distribution (Paulsson, 2004), which closely resembles Gamma distributions when b is large.

Different proteins have different burst frequencies and sizes, and hence different mean abundance $\langle X \rangle$ and standard deviation σ . Measurements show a typical relationship between the abundance of a protein and its standard deviation between cells. The coefficient of variation drops with protein abundance as

$$CV = \sigma / \langle X \rangle = 1 / \sqrt{\langle X \rangle} \quad (\text{D.3.2})$$

until hitting a minimal noise floor CV_0 for proteins of high abundance due to extrinsic noise. More precisely,

$$(\sigma / \langle X \rangle)^2 = 1 / \langle X \rangle + \alpha_p / \beta_m + CV_0^2 \quad (\text{D.3.3})$$

where α_p is the protein removal rate and β_m is the transcription rate (Paulsson, 2004; Taniguchi et al., 2010; Bar-Even et al., 2006).

D.4 NETWORK MOTIFS AFFECT NOISE

Regulatory circuits affect the noise. For example, protein levels can be made to fluctuate less by means of negative autoregulation, as discussed in Chapter 2. Conversely, positive autoregulation can increase cell-cell variability. Positive feedback can lead to bistability as discussed in Chapter 5 (Figure D.4). In fact, stochastic effects can lead to bistability even when deterministic equations do not show it: a non-cooperative positive autoregulation loop (Hill coefficient = 1) can lead to bistability in the presence of noise (Berg et al., 2002; Friedman, Cai and Xie, 2006).

Bistability leads to a bimodal distribution (Figure D.4), with two cell

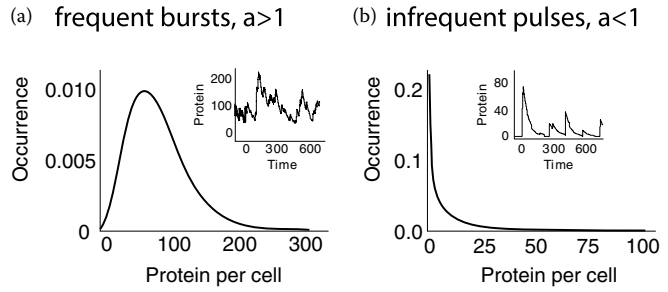


FIGURE D.3

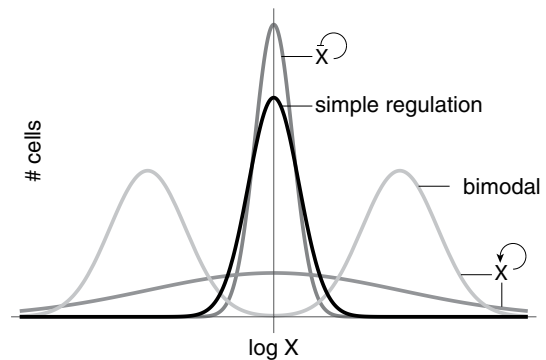


FIGURE D.4

populations, with high and low expression (Novick and Weiner, 1957; Siegle and Hu, 1997; Ferrell and Machleder, 1998; Ozbudak et al., 2002). Bimodality is fundamental to the ability of a cell to randomly take one of two different fates in a given condition. These effects of positive and negative autoregulation can be calculated exactly (Friedman, Cai and Xie, 2006). The effects of noise on gene circuits can be efficiently simulated on the computer using the Gillespie algorithm (Gillespie, 1976).

Noise can also be amplified by regulatory cascades: each step in the cascade receives

variability from its upstream regulator, and hence variation increases down the cascade (Figure D.5; Blake et al., 2003; Hooshangi, Thiberge and Weiss, 2005; Pedraza and Van Oudenaarden, 2005). Rapidly degraded proteins can have narrower distributions than stable proteins, because stable proteins integrate the noise in production rates over longer times. Stable proteins also have longer autocorrelation times in their noise, remaining higher or lower than average for longer times than short-lived proteins (Sigal et al., 2006). As a rule of thumb, the faster the response time of a system, the smaller the fluctuations.

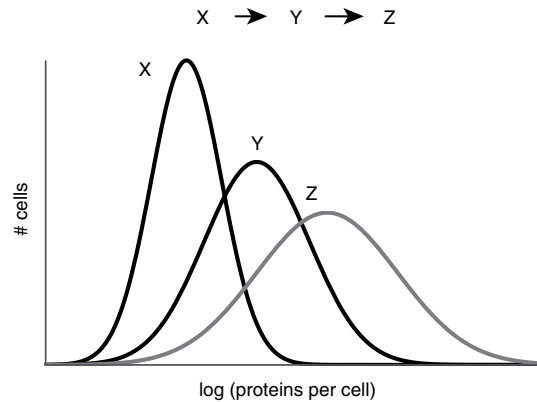


FIGURE D.5

D.5 POSITION OF NOISIEST STEP

One interesting observation is that the position of the noisiest step in a pathway can influence the overall noise (McAdams and Arkin, 1999; Ozbudak et al., 2002). This is because each step in the pathway usually amplifies noise in the previous steps.

For example, consider two mechanisms that produce 100 proteins per hour: In mechanism A, one mRNA molecule is made and is translated to 100 proteins on average. In mechanism B, 100 mRNAs are made and are each translated to one protein on average. The fluctuations in protein production are much larger in mechanism A, because an average of one mRNA normally means that in some cells either 0 or 2 mRNAs will be made in a given hour, resulting in 0 or ~200 proteins. In mechanism B, there is little chance to make zero mRNAs during an hour, and fluctuations are smaller.

The chromosomal position of a gene can also affect noise, due to local differences in chromatin regulation (Blake et al., 2003; Becskei, Kaufmann and Van Oudenaarden, 2005). Generally, noise level can be tuned over evolutionary timescales by changing the parameters of the noisy steps in the expression of each gene. It appears that essential proteins and complex-forming proteins are less noisy than other proteins, whereas stress-response proteins are noisier (Bar-Even et al., 2006).

Noise in biological systems can be modeled using stochastic mathematical equations. Such theoretical treatment of stochastic effects is beyond the present scope – if you want to learn more, good places to start are reviews on noise in biology (Paulsson, 2004; Kærn et al., 2005). Excellent texts on stochastic processes, such as those by Gardiner and Van Kampen,

can give access to the highly developed field of stochastic theory in physics, chemistry and engineering.

FURTHER READING

- (Acar, Becskei and Van Oudenaarden, 2005) “Enhancement of cellular memory by reducing stochastic transitions.”
- (Bar-Even et al., 2006) “Noise in protein expression scales with natural protein abundance.”
- (Berg, 1978) “A model for the statistical fluctuations of protein numbers in a microbial population.”
- (Blake et al., 2003) “Noise in eukaryotic gene expression.”
- (Elowitz et al., 2002) “Stochastic gene expression in a single cell.”
- (Friedman, Cai and Xie, 2006) “Linking stochastic dynamics to population distribution: an analytical framework of gene expression.”
- (Kærn et al., 2005) “Stochasticity in gene expression: from theories to phenotypes.”
- (McAdams and Arkin, 1997) “Stochastic mechanisms in gene expression.”
- (Novick and Weiner, 1957) “Enzyme induction as an all-or-none phenomenon.”
- (Ozbudak et al., 2002) “Regulation of noise in the expression of a single gene.”
- (Raser and O’Shea, 2004) “Control of stochasticity in eukaryotic gene expression.”
- (Rosenfeld et al., 2005) “Gene regulation at the single-cell level.”

BIBLIOGRAPHY

- Acar, M., Becskei, A. and Van Oudenaarden, A. 2005. ‘Enhancement of cellular memory by reducing stochastic transitions’, *Nature*, 435(7039), 228–232. doi: 10.1038/nature03524.
- Bar-Even, A. et al. 2006. ‘Noise in protein expression scales with natural protein abundance’, *Nature Genetics*, 38(6), 636–643. doi: 10.1038/ng1807.
- Becskei, A., Kaufmann, B.B. and Van Oudenaarden, A. 2005. ‘Contributions of low molecule number and chromosomal positioning to stochastic gene expression’, *Nature Genetics*, 37(9), 937–944. doi: 10.1038/ng1616.
- Berg, J.M. et al. 2002. *Biochemistry*. W.H. Freeman.
- Berg, O.G. 1978. ‘A model for the statistical fluctuations of protein numbers in a microbial population’, *Journal of Theoretical Biology*, 71(4), 587–603. doi: 10.1016/0022-5193(78)90326-0.
- Blake, W.J. et al. 2003. ‘Noise in eukaryotic gene expression’, *Nature*, 422(6932), 633–637. doi: 10.1038/nature01546.
- Cohen, A.A., Kalisky, T., Mayo, A., Geva-Zatorsky, N., Danon, T., Issaeva I. et al. 2009. ‘Protein dynamics in individual human cells: Experiment and theory’, *PLoS ONE* 4(4): e4901. <https://doi.org/10.1371/journal.pone.0004901>.
- Eldar, A. and Elowitz, M.B. 2010. ‘Functional roles for noise in genetic circuits’, *Nature*, 467(7312), 167–173. doi: 10.1038/nature09326.
- Elowitz, M.B. et al. 2002. ‘Stochastic gene expression in a single cell’, *Science*, 297(5584), 1183–1186. doi: 10.1126/science.1070919.
- Ferrell, J.E. and Machleder, E.M. 1998. ‘The biochemical basis of an all-or-none cell fate switch in xenopus oocytes’, *Science*, 280(5365), 895–898. doi: 10.1126/science.280.5365.895.
- Friedman, N., Cai, L. and Xie, X.S. 2006. ‘Linking stochastic dynamics to population distribution: an analytical framework of gene expression’, *Physical Review Letters*, 97(16), 3116. doi: 10.1103/PhysRevLett.97.168302.
- Gillespie, D.T. 1976. ‘A general method for numerically simulating the stochastic time evolution of coupled chemical reactions’, *Journal of Computational Physics*, 22(4), 403–3434. doi: 10.1016/0021-9991(76)90041-3.
- Golding, I. and Cox, E.C. 2006. ‘Physical nature of bacterial cytoplasm’, *Physical Review Letters*, 96(9), 675. doi: 10.1103/PhysRevLett.96.098102.

- Hooshangi, S., Thiberge, S. and Weiss, R. 2005. 'Ultrasensitivity and noise propagation in a synthetic transcriptional cascade', *Proceedings of the National Academy of Sciences*, 102(10), 3581–3586. doi: 10.1073/pnas.0408507102.
- Kærn, M. et al. 2005. 'Stochasticity in gene expression: from theories to phenotypes', *Nature Reviews Genetics*, 6(6), 451–464. doi: 10.1038/nrg1615.
- McAdams, H.H. and Arkin, A. 1997. 'Stochastic mechanisms in gene expression', *Proceedings of the National Academy of Sciences*, 94(3), 814–819. doi: 10.1073/pnas.94.3.814.
- McAdams, H.H. and Arkin, A. 1999. 'It's a noisy business! genetic regulation at the nanomolar scale', *Trends in Genetics*, 15(2), 65–69. doi: 10.1016/S0168-9525(98)01659-X.
- Novick, A. and Weiner, M. 1957. 'Enzyme induction as an all-or-none phenomenon', *Proceedings of the National Academy of Sciences*, 43(7), 553–566. doi: 10.1073/pnas.43.7.553.
- Ozbudak, E.M. et al. 2002. 'Regulation of noise in the expression of a single gene', *Nature Genetics*, 31(1), 69–73. doi: 10.1038/ng869.
- Paulsson, J. 2004. 'Summing up the noise in gene networks', *Nature*, 427(6973), 415–418. doi: 10.1038/nature02257.
- Pedraza, J.H. and Van Oudenaarden, A. 2005. 'Noise propagation in gene networks', *Science*, 307(5717), 1965–1969. doi: 10.1126/science.1109090.
- Raser, J.M. and O'Shea, E.K. 2004. 'Control of stochasticity in eukaryotic gene expression', *Science*, 304(5678), 1811–1814. doi: 10.1126/science.1098641.
- Rosenfeld, N. et al. 2005. 'Gene regulation at the single-cell level', *Science*, 307(5717), 1962–1965. doi: 10.1126/science.1106914.
- Siegele, D.A. and Hu, J.C. 1997. 'Gene expression from plasmids containing the araBAD promoter at subsaturating inducer concentrations represents mixed populations', *Proceedings of the National Academy of Sciences*, 94(15), 8168–8172. doi: 10.1073/pnas.94.15.8168.
- Sigal, A. et al. 2006. 'Variability and memory of protein levels in human cells', *Nature*, 444(7119), 643–646. doi: 10.1038/nature05316.
- Taniguchi, Y., Choi, P.J., Li, G.W., Chen, H., Babu, M., Hearn, J., Emili, A. and Xie, X.S. 2010. 'Quantifying *E. coli* proteome and transcriptome with single-molecule sensitivity in single cells,' *Science*. 329(5991), pp. 533–538. doi: 10.1126/science.1188308.

Words of Thanks

It is a pleasure to thank my teachers. First, my mother, Pnina, who gave much loving care, for teaching me, among many things, math and physics throughout my childhood, and my father, Hanan, for humor and humanism. To my PhD adviser Dov Shvarts, with his impeccable intuition and pedagogy, who offered, when I was confused about what subject to pursue after graduation, the unexpected suggestion of biology. To my PhD adviser, David Mukamel for teaching the power of rigorous yet simple models and for the freedom to make a mess in the labs of the biology building. To Tsiki Kam and Yossi Yarden, who helped me fall in love with biology. To my postdoctoral adviser, Stan Leibler, who introduced me to the study of design principles in biology with caring, generosity and many inspiring ideas. To Mike Surette and Arnie Levine for teaching love of experimental biology and for answers to almost every question. And to my other first teachers of biology, Michael Elowitz, Eldad Tzahor and Tal Raveh, who provided unforgettable first experiences of things such as centrifuge and pipette.

I have learned much from my wonderful students, research associates and postdocs, some of whose research can be found in this book: Miri Adler, Anat Bren, Ariel Cohen, Erez Dekel, Tamar Danon, Eran Eden, Naama Geva-Zatrosky, Shalev Itzkovitz, Yuval Hart, Jean Hausser, Shiraz Kalir, Omer Karin, Nadav Kashtan, Yael Korem, Galit Lahav, Daniel Madar, Shmoolik Mangan, Ron Milo, Elad Noor, Merav Parter, Michal Ronen, Nitzan Rosenfeld, Vered Sasson, Irit Shachrai, Hila Sheftel, Shai Shen-Orr, Guy Shinar, Oren Shoval, Alex Sigal, Pablo Szekely, Avichai Tendler, Benjamin Towbin and Alon Zaslaver.

Thanks also to the students in the course “Introduction to Systems Biology” from 2000 to 2018 for questions and suggestions. Special thanks to Naama Barkai for developing the lectures that make the basis for the chapter on robust patterning.

To my friends for much laughter mixed with wisdom: Michael, Tsvi, Yuvalal, Lior, Ela and Arnon, Dana and Kobi, Eliav and Vered, Nirit and Boaz, and Arik and Uri Moran. To Edna and Ori, Dani and Heptzibah, Nili and Guy and Gidi with love. To Galia Moran with love and appreciation. To Gefen, Tamar and Carmel with hugs.

For reading and commenting on parts of the manuscript, thanks to Dani Alon, Michael Elowitz, Ron Milo, Shalev Itzkovitz, Hannah Margalit, Ruth Scherz-Shouval, Ariel Cohen and all members of my lab, especially Avi Mayo, David Glass and Tomer Milo.

To Shalev Itzkovitz for devoted help with the first edition. Special thanks to Avi Mayo for incredible dedication with the second edition, including making all of the figures.

To the Weizmann Institute, and especially to Benny Geiger, Haim Harari and Daniel Zajfman, for keeping our institute a place to play.



Taylor & Francis

Taylor & Francis Group

<http://taylorandfrancis.com>

Index

Note: Page numbers followed by “*n*” with numbers indicate footnotes.

A

Activation coefficient, 9
Activator(s), 5, 8
 accumulation, 222
 binding to DNA site, 294–295
 input function integrates repressor and, 299–301
Activator and repressor demand rule, 130
Aligned canalization, 262
Aligned polymorphisms, 261–262
Allometric relationships, 254
Alternating stability of fixed points, 92
Amino acids, 118, 120
Ammonite shells, 256
AM strategies, 112
Antagonistic reactions, 144
APC, 105
Arabinose, 129
 system of *E. coli*, 45–46
AraC, 45
Archetype, 252–253, 256–257
 analysis algorithms, 260
Arginine (Arg), 118
Astounding speed of evolution, 273
Attractants, 153
Autogenous control, 23
Autokinase, 140
Autoregulated cascade, 33
Autoregulation, 21, 23–24; *see also* Negative
 autoregulation (NAR); Positive
 autoregulation (PAR)
 with Hill input function, 31–32
Avidity effect, 149

B

Bacillus subtilis, 70
 interlocked feedforward loops in sporulation
 network, 70–73

Bacterial chemotaxis, 144, 153, 282
 adaptation, 158–159
 attractants lowering activity of X, 157–158
 behavior, 153–155
 E. coli, 176–177
 fold-change detection in, 176–178
 individuality and robustness in, 165–166
 protein circuit, 156–159
 response and exact adaptation,
 155–156
Bacterial two-component systems, 140
 bifunctional components, 144
 black-box approach, 143–144
 limits of robustness, 143
Bang-bang control, 111
Barkai–Leibler model of exact adaptation,
 159–162
 experiments, 164–165
 robust adaptation and integral feedback,
 162–164
Basal expression level, 10
Basal promoter activity, 293
Basins of attraction, 79, 81
Beta-cell–Insulin–Glucose model
 (BIG model), 196
 compensation time in, 205
 and DC, 197–200
 linear analysis of fixed points, 205
Beta cells, 192
Bet hedging strategy, 166, 235
Bi-fan, 64
 dynamics, 74
 signal integration by, 68–69
Bifunctional components, 140, 144
Bifunctional enzymes, 140*n*1
BIG model, *see* Beta-cell–Insulin–Glucose model
Bimodality, 310
Biological oscillations, 97

Biological oscillators
 delay oscillators, 102–103
 noise-induced oscillations, 101–102
 oscillations requiring negative feedback and delay, 97–101
 positive and negative feedback loop motif, 103–107
 robust bistability using two positive feedback loops, 107–109
 Biphasic mechanism, 206–207
 Biphasic response, 50–51, 57–58, 203
 curves, 202–203, 219
 Bistability, 79, 105, 309–310
 Black-box approach, 143–144
 Black-box calculation, 147–148
 Blood proteins, 259
 Brain uptake of glucose, 205
Buchnera, 235
 Bursts of gene expression, 308

C

C1 (transcription factor), 83
 C1-FFL, *see* [Coherent type-1 FFL](#)
Caenorhabditis elegans, 89–90
 Calcium homeostasis
 compensation time, 206
 mutant resistance, 206
 cAMP, 45
 Canalization, 255
 Cascades, 18
 of repressors, 84
 Cdc25 kinase, 105
 CDK1, 105
 Cell-cycle
 clock, 108
 oscillator circuits, 105
 CFP, *see* [Cyan fluorescent protein](#)
 CheA, 157
 CheB, 158
 Chemostat, 232*n*3
 Chemotaxis, 66
 circuit providing FCD, 178–180
 system, 221–222
 CheR, 158
 Clustering coefficients, 305
 Codon, 119
 Coefficient of variation, 307, 309
 Cognitive problem of cell, 3–4
 Coherent FFLs, 40, 51, 55–57, 87
 Coherent type-1 FFL (C1-FFL), 40–41, 72; *see also* [Incoherent type-1 FFL \(I1-FFL\)](#)
 dynamics of C1-FFL with AND logic, 41–42

 OR-gate C1-FFL, 46–47
 sign-sensitive delay element, 42–46
 Collision rate, 291
 Compact in-degree sequences, 303–304
 Compensation time in BIG model, 205
 Competitive exclusion, 246
 Conformational proofreading, 128, 219; *see also* [Kinetic proofreading](#)
 optimal handicap in, 134
 specificity without consuming energy, 128–129
 Conservation equation, 287
 Conservation law, 139, 291
 Conservative evolution, 51
 Constitutive expression, 234
 Convergent evolution of FFLs, 51–52
 Cooperativity, 100
 Cortisol, 50–51
 Cost
 function, 240
 of LacZ protein, 229–230
 of noise, 246
 cro (transcription factor), 83
 Crosstalk, 130
 avoiding crosstalk by efficient coding, 135
 CRP, 45, 49, 69, 304
 Cyan fluorescent protein (CFP), 308
 Cyclin, 105

D

DAGs, *see* [Directed acyclic graphs](#)
 Dale's rule, 8*n*1
 Damped oscillations, 98–99
 Data clustering, 260
 DC, *see* [Dynamical compensation](#)
 De-ubiquitinating enzyme, 127
 Degradation rate, 49
 Degree-preserving random networks (DPRNs), 55
 Delay
 in metabolic pathways, 74–75
 oscillators, 102–103
 Demand, 234
 Demand rules, 219
 for gene regulation, 129–130
 for genes of concordant and opposing functions, 135
 for phosphorylation, 135
 Dense overlapping regulons (DORs), 69–70
 signal integration by, 68–69
 Dephosphorylation, 139
 Desert rodent, *see* *Psammomys obesus*
 Detailed balance conditions, 132

Developmental patterning mechanisms, 261
 Developmental timing, 84
 Developmental transcription networks, 210
 long transcription cascades and developmental timing, 84
 network motifs in, 77
 PAR, 78–80
 regulating feedback and regulated feedback, 83–84
 two-node positive feedback loops for decision-making, 80–83
 Diamond pattern, 56, 58
 Diffusion, 220–222
 and degradation process, 212
 diffusion-limited value, 288
 equation, 185
 Diffusion–degradation equations, 211–213
 Dimensionality reduction method, 260
 Dimensionless variables, 179
 Directed acyclic graphs (DAGs), 73
 Dissociation
 balance collisions, 291
 rate, 291
 Dissociation constant (K_d), 288
 DNA site, activator binding to, 294–295
 DORs, *see* [Dense overlapping regulons](#)
 Dorsal region (DR), 215
 Double-negative feedback loop, 80–81
 Double-positive feedback loop, 80
 positive autoregulation in, 92
 Double-positive feedback, nullclines for, 92
 Downstream protein half-life, 112
 DPRNs, *see* [Degree-preserving random networks](#)
 DR, *see* [Dorsal region](#)
Drosophila (fruit fly), 215
 patterning mechanisms, 215–220
 Dynamical compensation (DC), 197–200
 for beta-cell secretion rate, 205
 for blood volume, 206
 Dynamic measurements, FCD circuit
 identification from, 183
 Dynamics
 of cell populations, 195
 compensation, 219
 of robust mechanism, 149
 of transcription networks, 13–15

E

Endocytosis, 215
 Energy consumption, 148

Environmental selection of FFL network motif, 236–238
 EnvZ/OmpR, 140
 Equal timing, 73
 Equilibrium binding of tRNAs, 119–121
 Erdős–Rényi model (ER model), 23, 31
 ER model, *see* [Erdős–Rényi model](#)
 Error rate, 120
 in delay mechanism, 126
Escherichia coli, 3–4, 66, 69, 129, 154, 156, 228, 232, 235, 238, 245, 258, 306
 chemotaxis system of, 176–177
 network, 23
 sign-sensitive delay in arabinose system of, 45–46
 Evolutionary dynamics analysis, 233
 Evolutionary optimization, 227
 Evolutionary theory, 249
 Exact adaptation, 155–156
 Barkai–Leibler model, 159–162
 Exact adaptation, FCD and, 180
 Exclusive-OR gate (XOR gate), 275–276
 Expansion–repression mechanism, 223
 Exponential pulse distribution, 244
 Extrinsic noise, 307–308

F

Fan out, 18
 FBLs, *see* [Feedback loops](#)
 FCD, *see* [Fold-change detection](#)
 Feedback, 183
 inhibition, 62, 74
 mechanism, 195, 200, 213, 215
 strength, 99–100
 Feedback loops (FBLs), 54, 215
 Feedforward, 183
 Feedforward loop (FFL), 39, 61, 77, 87, 236; *see also* [Multi-output feedforward loop](#)
 convergent evolution, 51–52
 dynamics of coherent type-1 FFL with AND logic, 41–42
 environmental selection of FFL network motif, 236–238
 incoherent type-1 FFL generating pulses of output, 47–51
 network motif, 37–39
 no oscillations in, 94
 OR-gate C1-FFL, 46–47
 selection, 242–245
 structure of FFL gene circuit, 39–41
 types, 51
 FFL, *see* [Feedforward loop](#)

Field, 209
 FIFO order, *see* First-in-first-out order
 Fine-tuned model, 159
 for exact adaptation, 166–169
 First-in-first-out order (FIFO order), 64
 First-order kinetics, 296
 Fitness, 229, 237, 250
 of circuit, 276
 landscape picture for single task, 249–250
 Fitness functions, 227–228
 and optimal expression level, 231–232
 Fixed point of equation, 28
 Flagellar motor, 66–67
 Flagellar phases, 246
 Fluorescence resonance energy transfer (FRET), 177
 Fluorescence system, 177
 Flux boundary condition, 222
 FM strategies, 112
 Fold-change detection (FCD), 177–178, 187–188, 219, 251
 in bacterial chemotaxis, 176
 chemotaxis circuit providing, 178–180
 and exact adaptation, 180
 general condition for, 182–183
 identifying FCD circuits from dynamic measurements, 183
 incoherent feedforward loop, 180–181
 provides robustness to input noise and allows scale-invariant searches, 184–186
 universal features of sensory systems, 175–176
 Fold change, 179
 “Follow the bone” principle, 282
 Frameshift errors, 134
 French flag model, 209–210
 Frequency of bursts, 308
 FRET, *see* Fluorescence resonance energy transfer
 Fruit fly, *see* *Drosophila*

G

Gain control, 184
 Galactose utilization genes, 49
 Gamma distribution of protein abundance, 308
 Gene
 duplication, 277–278
 gene-expression noise, 307
 recombination, 277–278
 Gene expression
 trade-offs in, 258–259
 Generalists, 254
 Generalized diamond, 74
 Gene input function
 binding of activator to DNA site, 294–295

 binding of inducer to repressor protein, 289–291
 binding of repressor to promoter, 287–289
 cooperativity of inducer binding and Hill equation, 291–292
 input function of gene regulated by repressor, 293–294
 Michaelis–Menten enzyme kinetics, 295–296
 Monod–Changeux–Wyman model, 292–293
 Genetic code, 118
 Genetic drift, 228
 Genome, 277
 Genotype, 249
 GFP, *see* Green fluorescent protein
 Gillespie algorithm, 310
 Glucose, 45, 191–192
 dynamics, 197–198
 tolerance test, 191
 Glucotoxicity, 196, 200–201, 203
 Glycolysis, 202
 Goal, 276
 MVG, 278–280
 randomly varying goals cause confusion, 278
 Green fluorescent protein (GFP), 45, 102

H

Hill coefficient, 9, 292
 Hill function, 9–10, 291–292, 295
 Homeostasis, 86
 Homogeneity conditions, 186
 for FCD, 182
 Homologous genes, 51
 Hopf bifurcation, 104
 Hubs, 304
 in networks, 54–55
 Hybrid FFL motifs, 87
 Hybrid network motifs, 85–87
 Hysteresis, 106

I

I1-FFL, *see* Incoherent type-1 FFL
 IFFLs, *see* Incoherent feedforward loops
 IL-2, 144
 Immune FCD, 187
 Immune system, recognition of self and non-self by, 123–127
 Incoherent feedforward loops (IFFLs), 40, 49, 51, 57–58, 87, 180–181
 with microRNA, 58

Incoherent type-1 FFL (I1-FFL), 40, 47, 72, 180–181, 183; *see also* Coherent type-1 FFL (C1-FFL)
 biphasic steady-state response curves, 50–51
 as FCD circuit, 187
 generating pulses of output, 47–51
 speeds response times, 48–50
 Individuality in bacterial chemotaxis, 165–166
 Induced fit, 128
 Inducer binding
 cooperativity and Hill equation, 291–292
 to repressor protein, 289
 Inhibition function, 81
 Input dynamic range, 34–35
 Input functions, 9–10
 Input-output robustness, 142
 Insulin, 192
 minimal model non robust to changes in, 193–194
 resistance, 193
 sensitivity, 192–193
 Insulin-glucose
 feedback loop, 191, 204
 Integral feedback, 162–164, 171, 219
 Integrated fitness, 243
 Interconnected FFLs, 58
 Interlocked feedforward loops in *B. subtilis*
 sporulation network, 70–73
 Intermediate states, 127
 Intrinsic noise, 307–308
 Inverse ecology, 238
 Islets of Langerhans, 203

J

Jacobian matrix, 99, 104
 Just-in-time production, 63

K

Kinetic proofreading, 117–118, 219; *see also* Conformational proofreading
 in diverse processes in cell, 127–128
 equilibrium binding of tRNAs, 119–121
 of genetic code reducing error rates, 118–123
 increasing fidelity of T-cell recognition, 125–127
 low error rate of immune recognition, 124–125
 recognition of self and non-self by immune system, 123–127

L

Labor division in individual cells, 259–260
lac promoter, 164

lac repressor (LacI), 289
 tetramers, 275
Lac system, 12
 Lactose system, 129
 LacZ protein, 228
 benefit of, 230–231
 cost of, 229–230
 Last-in-first-out order (LIFO order), 62, 68
 Leakage transcription rate, 293
 LEGI mechanism, *see* Local excitation, global inhibition mechanism
 LIFO order, *see* Last-in-first-out order
 Ligands, 156
 Limit cycle, 104
 Linear integral feedback cannot filter out ramps of input, 172
 Linearity of output curve, limits to, 148
 Linearized positive autoregulation, 34
 Linear ordinary differential equation, 211
 Linear stability analysis, 35, 98, 111
 Linear systems, 188
 Linkage rate, 120
 Liver hepatocytes, 259
 Local excitation, global inhibition mechanism (LEGI mechanism), 223
 Lock-on mechanism, 80
 Lock and key, 128
 Logarithmic slowdown, 278
 Logic gates, simulated evolution of circuits made of, 275–278
 Logic input functions, 10–11, 180, 295
 Long-tailed out-degree sequences, 303–304
 Long-term memory, 79
 Long transcription cascades, 84
 Lysogenic mode, 83
 Lytic mode, 83

M

Mammalian limb, 282
 Mass-action
 equations for two-component circuit, 146
 kinetics, 139, 287
 for non-robust circuit, 145
 Mathematically controlled comparison, 27
 Maximal promoter activity, 9–10
 Memory
 of multiple past inputs, 94
 in regulated feedback network motif, 93
 Metabolic pathways, 62
 Methylation modifications, 158
 Methyl group, 158
 MHCs, 123

Michaelis–Menten
 enzyme kinetics, 295–296
 equation, 289, 294, 296
 functions, 142, 188, 299
 kinetics, 167

Minimal model, 192–194, 204

Modified-ER (MER), 31

Modular goals, 281–282

Modularity, 273
 in all levels of biological organization, 274–275
 of components, 8
 modular goals and biological evolution, 281–282
 non-modular solutions in simple computer simulations, 275
 simulated evolution of circuits made of logic gates, 275–278

Modularly varying goals (MVG), 279
 leads to spontaneous evolution of modularity, 278–280
 speeds up evolution, 280–281

Modular system, 274

Molecular network, 216

Monod–Changeux–Wyman Model, 292–293

Monotonic response functions, 50

Morphogen, 209–210, 218, 221
 gradients, 209–210

Morphology, trade-offs in, 254–256

mRNA dynamics, 16–17

Multi-cellular organisms, 77

Multi-dimensional input functions, 11–12, 299–301

Multi-input FFL, 74
 as coincidence detector, 93–94
 in neuronal networks, 89–91

Multi-objective optimality
 archetypes, 256–257
 fitness landscape picture for single task, 249
 labor division in individual cells, 259–260
 multiple tasks characterized by performance functions, 250–251
 Pareto optimality in performance space, 251–252
 Pareto optimality in trait space leads to simple patterns, 252–253
 trade-offs between tasks, 253–254
 trade-offs for proteins, 257
 trade-offs in gene expression, 258–259
 trade-offs in morphology, 254–256
 variation within species lies on Pareto front, 260–262

Multi-output AND C1-FFL, 74

Multi-output FFL, 64–66
 generating FIFO temporal programs, 66–68
 persistence detector, 68

Multi-output OR C1-FFL, 74

Multiple tasks, 249–251

Mutant cells, 201

Mutational targets, 245–246

MVG, *see* [Modularly varying goals](#)

N

NAND gate, *see* [Not-AND gate](#)

NAR, *see* [Negative autoregulation](#)

Nautilus, 257

Near-irreversibility, 121*n*2

Negative autoregulation (NAR), 21, 23, 49; *see also* [Positive autoregulation \(PAR\)](#)
 as linear filters, 35
 promoting robustness to fluctuations in production rate, 29–30
 rate analysis, 28–29
 speeds response time of gene circuits, 24–29

Network
 dynamics, 10–11

Network modularity, quantitative measure of, 305–306

Network motifs, 21–23, 37, 70, 214–215
 affect noise, 309–310
 algorithm for detecting, 54
 detection by comparison to randomized networks, 23
 in developmental transcription networks, 77
 in neuronal networks, 88–91
 PPI networks, 85–87

Neuronal networks
 multi-input FFLs in, 89–91
 network motifs in, 88–91

Neurons, 89

Neurotoxicity, 203

Newmann–Girvan algorithm, 274

NF- κ B protein, 184

Niche, 250

NLIFBL, *see* [Nonlinear integral feedback loop](#)

Noise, 29, 307
 cost, 246
 distribution of protein levels, 308–309
 extrinsic and intrinsic noise, 307–308
 in gene expression, 307
 network motifs affect noise, 309–310
 noise-induced oscillations, 101–102
 position of noisiest step, 310–311

Nonlinear degradation, 213

Nonlinear integral feedback loop (NLIFBL), 178–180, 182–183

Non-monotonic response, 50

Non-robust, 138

Not-AND gate (NAND gate), 276
 Notch-Delta lateral inhibition, 222–223
 Nullcline
 analysis, 81
 for double-positive feedback, 92
 Numerical screening approach, 217–218

O

OFF steps of signal, 42–44, 47
 One-dimensional diffusion–degradation
 equation, 211
 One-node circuits, 110–111
 ON steps of signal, 42–44, 47
 Operons, 67, 172
 Optimal gene circuit design
 environmental selection of FFL network motif,
 236–238
 evolutionary optimization, 227
 inverse ecology, 238
 optimal expression level of protein, 228–234
 optimal regulation in changing environments,
 234–236
 Optimal genetic code for minimizing errors,
 133–134
 Optimal handicap in conformational proofreading,
 134
 Optimality theory, 227
 Optimal mismatch, 135
 Optimal position of stop codons, 134
 Optimal tRNA concentrations, 132–133
 OR-gate
 C1-FFL, 46–47
 logic, 55
 Organelles, 274
 Oscillations, 97
 noise-induced oscillations, 101–102
 requiring negative feedback and delay, 97–101
 sizable delay to negative feedback loop, 97–101
 transcription factors, 112
 Oscillator motif, equations for, 111
 Osmotic response system of *E. coli*, 140
 Overdamped system, 100

P

p53 (guardian of genome), 100
 PAR, *see* Positive autoregulation
 Paradoxical components, 219
 Paradoxical control in *E. coli* carbon/nitrogen
 balance, 149
 Paradoxical enzymes, *see* Antagonistic reactions
 Paradox of plankton, 246

Parameter sensitivity coefficient, 32–33
 Pareto front, 251, 253
 variation within species lies on,
 260–262
 Pareto optimality
 in performance space, 251–252
 in trait space, 252–253
 Pareto task inference (ParTI), 252, 258, 260
 ParTI, *see* Pareto task inference
 Patterning, 210
 Patterns, 21–23
 PCA, *see* Principal component analysis
 Performances, 250
 functions, 250–251
 Performance space, 251
 Pareto optimality in, 252–252
 Persistence detector, 44
 Phage lambda, 82
 Phalanges, 260
 Phase diagram, 235, 238, 245
 Phase portraits, 93
 of circuit, 81
 Phenotype, 249–250
 Phosphatase, 139
 Phosphorylated messenger, 139
 Phosphorylation, 87, 139
 Phosphotransferase, 140
 Piezo1, 144
 Plasmid, 141
 Poincaré–Bendixson theorem, 104
 Point mutation, 277
 Polymorphisms, 261
 Polynomial self-enhanced degradation, 222
 Polytope, 254
 Population dynamics, 245–246
 Positive autoregulation (PAR), 78, 82, 149; *see also*
 Negative autoregulation (NAR)
 in double-positive feedback loop, 92
 as linear filters, 35
 slows responses and lead to bistability,
 78–80
 with step function, 34
 Positive feedback, 103, 309
 Power-law morphogen profile, 214
 PPI networks, *see* Protein–protein interaction
 networks
 Preferential attachment, 304
 Principal component analysis (PCA), 254
 Production rate, 78, 104
 change in, 16
 of gene, 29–30
 Promoter, 5
 activity, 289

Protein, 3, 13n2, 275

- benefit of LacZ Protein, 230–231
- cells reach optimal LacZ levels, 232–234
- cost of LacZ protein, 229–230
- fitness function and optimal expression level, 231–232
- levels distribution, 308–309
- optimal expression level of, 228
- trade-offs for, 257

Protein–protein interaction networks (PPI networks), 85

- feedforward loops, 87
- hybrid FFL motifs, 87
- hybrid network motifs, 85–87
- network motifs in, 85–87

Psammomys obesus (desert rodent), 207

PTH, 199

Pulse of activation, 18

Pulse trains, 111

Q

Quantitative measure of network modularity, 305–306

R

Randomized networks, 21–23, 31

Randomizing forces, 23, 38, 246, 250

Rate analysis, 28–29

Rate plot, 28

Reactive oxygen species (ROS), 201

RecA, 83

Recognition, 117

- of self and non-self by immune system, 123–127

Reduction in attractant, 166

Regulated feedback, 83–84

Regulating feedback, 83–84

Relative changes, 33, 48, 175–176

Relative sensing, 175

Relaxation oscillators, 106

Removal rate, 78

Repellents, 153, 166

Repressilator, 58, 102

- stability analysis, 112

Repression

- coefficient, 289
- threshold, 30

Repressors, 5, 8

- binding to promoter, 287–289
- decay, 222
- gene input function regulated by repressor, 293–294

inducer binding to repressor protein, 289

input function integrates activator and, 299–301

Resisting noise, 307

Response adaptation, 155–156

Response time, 187

of simple regulation, 13–15

of stable proteins, 15

Reverse phosphotransfer, 148

Ribosome, 118

RNA polymerase (RNAP), 5, 258, 288

Robust(ness), 117, 220, 273

adaptation, 162–164

in bacterial chemotaxis, 165–166

of biological functions, 219

bistability using positive feedback loops, 107–109

design, 166

expression ratios, 111

input–output relations, 149

loss of, 148

principle, 215–220

self-enhanced morphogen degradation, 212–214

shuttling mechanism, 218

timing, 73–74, 222

Robust model

for bacterial chemotaxis, 178

with two methylation sites, 169–171

Robust signaling by bifunctional

components, 137

bacterial two-component systems, 140–144

robust input–output curves, 137–138

simple signaling circuits, 138–139

Robust spatial patterning

French flag model, 209–212

increased robustness by self-enhanced morphogen degradation, 212–214

network motifs, 214–215

robustness principle, 215–220

ROS, *see* Reactive oxygen species

Runs, 154–155

S

Salmonella typhimurium, 246

Salt-and-pepper pattern, 222–223

Scale-free distributions, 304

Scale-invariant searches, FCD allowing, 184–186

Scaling, 223

Self-arrows in random networks with transcription factors, 31

Self-enhanced degradation, 219

Self-enhanced morphogen degradation, 212–214

- Sensing of relative changes, 175
- Sensory adaptation, 156, 166
- Sensory systems, universal features of, 175–176
- Sensory transcription networks, 73, 77
 - global structure of, 70
- Serial dilution, 232
- Series of pulses, 94
- Serine receptor (Tsr), 162
- Shuttling mechanism, 218–220
- Sigma factors, 258
- Sign-sensitive delay
 - in arabinose system of *E. coli*, 45–46
 - element, 42–46
- Signal-transduction circuits, 137
- Signal integration by bi-fans and dense-overlapping
 - regulons, 68–69
- SIM, *see* [Single-input module](#)
- Simple regulation as filter for high-frequency noise,
 - linear analysis, 35
- Simulated networks, 275
- Simulations of circuit evolution, 281
- Single-input module (SIM), 61, 68, 77
 - with autoregulation, 74
 - network motif, 61–62
 - software, 74
 - temporal gene expression programs, 62–64
- Single objective, 249
- Single task, 249–250
- Slow dynamics, 78
- Slow feedback loop, 196, 199–200
- Sparse, 303
 - network, 53
- Speedup, 28
 - for I1-FFL, 57
- Spiral fixed point, 99
- Spontaneous dephosphorylation, 148
- Spontaneous symmetry breaking, 223
- Statistical property of graphs, 305
- Steady-state
 - glucose depends on insulin sensitivity, 194
 - response, 57
 - tumbling frequency, 155
- Step-like stimulation, 41
- Stepping stone evolution, 257
- Subgraphs, 37
 - in random networks, 53–54
- Synthetic circuit, 49
 - increasing fidelity of recognition, 125–127
 - receptors, 125
- Temporal gene expression programs, 62–64
- Temporal order, 62–64
- TFs, *see* [Transcription factors](#)
- Thermodynamic box conditions, 132
- Thermostat, 86
- Third-base wobble, 134
- Three-component negative feedback loop, 102
- Three-node feedback loops, 73, 89
- Tight control over blood glucose, 191
- Time-dependent production and decay, 17–18
- Timescales, separation of, 7–8
- Tissue-level feedback loops, 201–202
- Tissue size control, 195
- TNF, *see* [Tumor-necrosis factor](#)
- Toggle switch, 81–82
- Trade-off, 202, 249
 - in gene expression, 258–259
 - in morphology, 254–256
 - for proteins, 257
 - between tasks, 253–254
- Traits, 249
- Trait space, 249
 - Pareto optimality in, 252–253
- Transcription factors (TFs), 3–4, 102, 137, 291
 - proteins, 6
- Transcription networks, 5, 61, 69, 303
 - biological parameter values, 4
 - clustering coefficients, 305
 - cognitive problem of cell, 3–4
 - dynamics and response time of simple
 - regulation, 13–15
 - elements, 5–12
 - quantitative measure of network modularity,
 - 305–306
 - long-tailed out-degree sequences and compact
 - in-degree sequences, 303–304
- Transient memory, 87
 - in FFL, 94
- Translational symmetry, 212
- Translation machinery, high fidelity of, 121
- Tricarboxylic acid cycle (TCA cycle), 149
- Tristability, 92–93
- tRNA, 118–123, 127–128
- Tumbles, 154–155
- Tumbling frequency, 154
- Tumor-necrosis factor (TNF), 184
- Tunable frequency, 107
- Turing patterns, 223
- Turn-OFF order, 68
- Turn-ON order, 68
- Turning off autoregulation, 34

T

- Takeover by mutant cells, 201
- TCA cycle, *see* [Tricarboxylic acid cycle](#)
- T-cells, 123–124, 144

Two-node positive feedback
 for decision-making, [34](#), [80–83](#)
Two-output FFL, [64](#)
Type 1 diabetes, [207](#)
Type 2 diabetes, [200–201](#)

U

Ubiquitin, [127–128](#)
Understandability, [72](#)
U-shaped death curve, [200–201](#)
U-shaped response curves, [202–203](#)

W

W-D trait space, [256](#)
Weber's law, [175](#)

Wee1, [105](#)
Wide input dynamic range, [184](#)

X

XOR gate, *see* [Exclusive-OR gate](#)

Y

Yeast, [69](#)
Yellow fluorescent protein (YFP), [308](#)

Z

Zero-order
 kinetics, [296](#)
 ultrasensitivity, [172](#)