

A STUDY OF SMALL GENE REGULATORY NETWORKS:
ROBUSTNESS, DYNAMICS, AND BEHAVIOR

A Dissertation

Presented to the Faculty of the Graduate School
of Cornell University

in Partial Fulfillment of the Requirements for the Degree of
Doctor of Philosophy

by

Sarah Robin Stockwell

May 2010

© Sarah Robin Stockwell

A STUDY OF SMALL GENE REGULATORY NETWORKS:
ROBUSTNESS, DYNAMICS, AND BEHAVIOR

Sarah Robin Stockwell, Ph.D.

Cornell University 2010

Gene regulatory networks, like any evolving biological system, are subject to potentially damaging mutations. Much work has been done to study what types of networks are more robust to node deletions (knockouts of entire genes). Less well understood, however, is the question of which networks best maintain consistent behavior in the face of smaller mutations that affect binding affinity, protein half-life, and other regulatory parameters. Such mutations have subtler effects than whole-gene knockouts do, but because they are far more common than knockout mutations, their impact on network evolution may be substantial. The first chapter investigates the expression patterns of simulated gene regulatory networks as these types of parameters are varied, and explores which topologies allow the networks to "ignore" parameter-changing mutations and maintain their expression patterns relatively unchanged. In the simulations, networks containing mutual repression feedback consistently displayed a more robust response to simulated mutation. The presence of this variety of feedback in well-studied developmental regulatory networks suggests that it may be a widespread mechanism for reducing the phenotypic consequences of both noise and mutational perturbations.

The second chapter also uses feedback loop module networks as a means to investigate and compare modeling approaches. It describes an algorithm to infer the best Boolean representation of the differential equation network models, as well as metrics for measuring how closely the Boolean model approached the dynamics of the continuous one. Using these tools allowed testing of the "Booleanizability" of

networks containing mutual-activator and mutual-repressor feedback loops. The investigation revealed that Boolean models are better approximations of networks with repressor loops than of those without them, and this is explained in terms of the characteristics explored in the investigation of network robustness.

Chapter 3 contains a model of the gap genes in *Drosophila melanogaster*, based on published experimental findings on the interactions among these genes and their products. The mechanistic mathematical model of gap gene expression was fitted to experimental data and placed in the context of other gap gene models. The chapter discusses the advantages and limitations of the various modeling techniques that have been employed for this system.

BIOGRAPHICAL SKETCH

Sarah Robin Stockwell was born and grew up in San Diego, California, where she attended Gompers Secondary School. She still feels nostalgic about her box of medals from the Science Olympiad, or “Geek Olympics,” in which she prepared her nerves for the eventual ordeal of graduate school by competing in the quiz-show Science Bowl event, as well as the arrestingly titled “Name That Organism!” competition – her first encounter with taxonomy. She also competed in numerous science fairs, winning prizes but perhaps earning her most lasting renown as “that student who filled the biology storeroom three inches deep in isopropyl alcohol when part of her experimental apparatus dissolved.” Wisely retreating from a career in experimental physics, she did theoretical simulations of vortex interactions for subsequent science fairs.

She matriculated at Swarthmore College in 1990. After her graduation, she spent a year in rural El Salvador teaching math, English, and basic chemistry to schoolteachers and clinic workers. Upon returning to the United States, she began work at Qualcomm Inc. She worked first in the company’s Eudora division, then as a Unix systems administrator. She began taking ecology and evolutionary biology courses part-time at UC San Diego, eventually cutting her Qualcomm hours to part-time in order to do research in the lab of the generous professor Dr. Chris Wills. With his encouragement, she applied to graduate school, and began Cornell’s Ecology and Evolutionary Biology Ph.D. program with Dr. Andrew G. Clark. While her dissertation research somehow turned out to be almost entirely genetics, with a bit of applied math thrown in, she is nevertheless an evolutionary biologist at heart, and looks forward to teaching and doing research in that field.

This work is dedicated to my mom, Nancy Watts Stockwell:

April 23, 1942 – December 13, 2009.

Thank you for all you gave me. I love you.

ACKNOWLEDGEMENTS

I would like to give my heartfelt thanks to my family: my mom, Nancy, who gave me her strength and her love of biology; my dad, Ross, who gave me his humor and his sense of responsibility; and my sister, Ellie, who gave me any hipness I possess, and who is my best and closest friend. Without their unconditional love and support, I would not be here. My loving thanks, also, to my husband, Andre, who has been a joy and a steady support, who encouraged and reassured me throughout the dissertation, and who typeset the equations to boot.

I also want to express my appreciation for all the teachers who have supported, encouraged, and of course educated me, especially: Jane Carney Schulze, Jean Bryant, Jane Fulkerson, Lana Davies, Barbara Murray, Martin Teachworth, Robert Cizek, Toni Wisehart, Chris Baldwin, Joan Petties, and Dana Shelburne. Many scientists have given their time to help teach me how to do good research, for which I am grateful: Carl Grossman, Chris Wills, Carl Bergstrom, George von Dassow, Garry Odell, Ed Munro, Lisa Nagy, Chris Myers, Chip Aquadro, and of course Andy Clark. Special thanks to Steve Strogatz, an inspirational teacher and leader who taught me to enjoy math again, and convinced me that it's respectable to work on fun problems. Rick Harrison took time to meet an unknown prospective applicant for coffee in La Jolla and give her the scoop on grad school. Chip Aquadro gave me a solid foundation from which to think about evolutionary genetics. Chris Myers was generous with his time and his creative insights. Ryan Gutenkunst was a cheerful and tremendously helpful collaborator. And Andy Clark generously accepted a grad student who wanted to work on odd problems outside the scope of his lab; made his time, experience, and resources available to me; and helped ground my work in biological reality.

I am indebted to a number of funding sources: A fellowship from the NSF IGERT program in Nonlinear Systems at Cornell; an NSF Graduate Research Fellowship; a fellowship from Friday Harbor Laboratories and the Center for Cell Dynamics; and NIH grant HL072904.

TABLE OF CONTENTS

| | |
|--|------|
| Biographical Sketch..... | iii |
| Dedication..... | iv |
| Acknowledgements..... | v |
| Table of Contents..... | vii |
| List of Figures..... | viii |
| List of Tables..... | x |
| Preface..... | xi |
| Chapter 1: Predicting the mutational robustness of gene regulatory networks..... | 1 |
| Chapter 2: Boolean models of modular networks..... | 66 |
| Chapter 3: Modeling the <i>Drosophila</i> gap gene network..... | 104 |

LIST OF FIGURES

| | | |
|--------------|---|----|
| Figure 1.1. | The feedback loops used as modules to construct the networks..... | 4 |
| Figure 1.2. | Assembling loop modules into two-module networks by adding cross-regulation..... | 5 |
| Figure 1.3. | Mathematical representation of loop modules..... | 9 |
| Figure 1.4. | Phase portraits..... | 10 |
| Figure 1.5. | Simulated mutations (parameter tweaks) alter the destinations of some trajectories, and thus the basins of attraction..... | 11 |
| Figure 1.6. | Static-signal networks: A constant external “push” activating or repressing an isolated module..... | 15 |
| Figure 1.7. | Two-module networks with RL (mutual inhibition) modules have higher local robustness to simulated mutations than those with AL (mutual activation) modules..... | 16 |
| Figure 1.8. | Two-module networks with RL (mutual inhibition) modules have higher global robustness to simulated mutations than those with AL (mutual activation) modules..... | 17 |
| Figure 1.9. | External regulation often pushes bistable repressor loop (RL) modules into monostability. Pictured here: RL2..... | 19 |
| Figure 1.10. | External regulation often pushes bistable repressor loop (RL) modules into monostability. Pictured here: RL1..... | 20 |
| Figure 1.11. | Activator loops (here, AL2) often retain their bistability even under the influence of external regulation..... | 21 |
| Figure 1.12. | Activator loops often remain bistable despite external signals. Pictured here: AL1..... | 22 |
| Figure 1.13. | Monostability helps confer robustness to parameter variation..... | 23 |
| Figure 1.14. | Monostability helps confer robustness to parameter variation (2-node modules)..... | 24 |
| Figure 1.15. | As the strength of the external signal increases (horizontal axis), the size of the basin of attraction (vertical axis) changes more abruptly for RL modules (above and Figure 1.16) than for AL modules (Figure 1.17, 1.18)..... | 26 |
| Figure 1.16: | Size of 10 basin for module RL1 as it receives constant external activation (green circles) or repression (red crosses) of increasing intensity..... | 27 |
| Figure 1.17: | Size of 1111 basin for module AL2 as it receives external activation (green circles) or repression (red crosses) signals of increasing intensity..... | 28 |
| Figure 1.18. | Size of 11 basin for module AL1 as it receives external activation (green circles) or repression (red crosses) signals of increasing intensity..... | 29 |
| Figure 1.19. | Repressor loops readily become <i>robust</i> in response to both activating and repressing external signals..... | 31 |
| Figure 1.20. | Repressor loops readily become <i>monostable</i> in response to both activating and repressing external signals..... | 31 |

| | | |
|-------------|--|-----|
| Figure 2.1. | The algorithm for finding the optimal Boolean representation of a DE network model..... | 73 |
| Figure 2.2. | Choosing the optimal (lowest cost) threshold position..... | 76 |
| Figure 2.3. | The four representative parameter sets for the single-gene activator loop (AL1) module, and their Boolean state tables and rulesets..... | 78 |
| Figure 2.4. | Comparing predicted and inferred state tables for network AL1actAL1..... | 82 |
| Figure 2.5. | RL networks are more likely to lose one of their steady states, and are easier to Booleanize as a result..... | 87 |
| Figure 3.1. | Regulators combining to regulate a target in ways that cannot be adequately represented by a linear summation of their effects..... | 110 |
| Figure 3.2. | Comparing protein concentrations in our model with smoothed protein concentrations measured in fixed embryos..... | 116 |

LIST OF TABLES

| | | |
|-------------|--|-----|
| Table A1.1. | Local robustness of static-signal networks to internal parameter perturbation..... | 55 |
| Table A1.2. | Ideal pluripotency scores for static-signal modules with X held constant at 0.5..... | 57 |
| Table A1.3. | Ideal pluripotency scores for static-signal modules with X held constant at 1.0..... | 57 |
| Table 2.1. | Mean values of “Booleanizability” estimators, and comparisons across classes of two-module networks..... | 85 |
| Table A2.1. | Mean values of “Booleanizability” estimators for classes of two-module networks..... | 97 |
| Table 3.1. | A summary of gap gene models..... | 115 |
| Table A3.1. | Fitted half-life parameters from the model..... | 132 |

PREFACE

This dissertation contains three investigations into the nature of transcriptional genetic networks. Taken together, they form a study of the dynamical behavior of small groups of genes that regulate one another's transcription, with feedback between the genes. All are idealizations of how real eukaryotic genes interact, even the gap gene model, which is the most data-driven of the three. The goal has been to simplify the concrete (gap genes project) and abstract (robustness and “Booleanization” projects) questions of how feedback shapes gene regulation. By simplifying the problem, we can hope to make some progress in studying it.

Feedback makes network behavior much more complex and thus more difficult for us to understand, and yet it also seems to be the key to how organisms respond appropriately to their environments (Krishna et al. 2006; Krishna et al. 2007) and lock in cell differentiation decisions during development (Davidson 2006). This dissertation is therefore an attempt to balance tractability with biological relevance, in order to shed light on the questions that gene regulatory feedback raises.

All three projects use some form of the mathematical representation of combinatorial transcriptional regulation worked out by George von Dassow and the other members of Garrett Odell's research group at Friday Harbor Laboratories, the Center for Cell Dynamics (von Dassow et al. 2000; Meir et al. 2002). The author is indebted to these faculty, who spent an intensive semester teaching her and several others how to model gene networks in a way that would be as credible as possible to both applied mathematicians (like Dr. Odell) and experimental biologists (like Dr. von Dassow and Dr. Ed Munro).

Chapter 1 (“Predicting the Mutational Robustness of Gene Regulatory Networks”) presents and mathematically describes a simple set of feedback motifs,

which are then combined into larger networks in order to study what properties of small networks makes large networks more robust to parameter variation (i.e., to simulated mutation). We develop two metrics of robustness to simulated mutation: global robustness, which characterizes overall behavior by sampling parameter space; and local robustness, which measures the system's response to doubling and halving individual parameters. We introduce the idea of isolated modules with constant external regulation acting upon them ("static-signal modules") to simplify the analysis of feedback loops under regulation.

By simulating thousands of parameter sets of individual and static-signal modules to measure their local and global robustness, we determine that modules with mutual repression feedback are more robust to simulated mutations than those composed of mutually activating feedback, but only when subject to external regulation. We establish that the greater robustness of networks containing mutual repression feedback is due to their greater tendency to become monostable for one steady state when receiving an external regulatory signal. Finally, we extrapolate this finding to evolutionary expectations for gene regulatory networks, especially in development.

In Chapter 2 ("Boolean models of modular networks"), we investigate modeling approaches for the networks presented in Chapter 1. We discuss the reasons why Boolean representations of gene regulatory networks can be useful, and why they can also give misleading results for networks containing feedback. We develop an algorithm to use the flow of trajectories in state space to find the best possible threshold to divide expression levels into "low" and "high" for each kind of molecule. This gives the best possible Boolean representation of each differential equation network, and also offers three quantitative metrics for measuring how good an approximation the Boolean version is.

We score the feedback motifs from the robustness project, and the two-module networks composed of them, for "Booleanizability," via the algorithm above. We establish that networks with repressor-loop modules submit more gracefully to a Boolean approximation, generating fewer modeling artifacts, than those with activator loops, for the same reasons of monostability that led to mutational robustness in the preceding chapter's investigation.

Finally, Chapter 3 ("Modeling the *Drosophila* gap gene network") discusses the uses of models of gene regulatory networks, and the abstractions they employ to investigate different questions about such systems. After a brief review of the segmentation patterning system in the *Drosophila melanogaster* embryo, we synthesize modeling work that has been done on the gap gene system, reviewing how each kind of abstraction choice affects the kinds of questions addressable by each model. We also offer a new model of the gap gene network, using a previously successful approach that has been neglected for this system. We report the insights gained from of the new model, in the context of work that has been done before.

This dissertation is primarily the work of Sarah R. Stockwell, and she conceived the problems and hypotheses, developed the simulations, and wrote the manuscript. She is also responsible for any errors herein. However, several people collaborated on these projects who deserve mention here, and who will be co-authors on the papers that emerge from this work. Dr. Ryan Gutenkunst, a fellow participant in Cornell's IGERT Nonlinear Systems Program, co-wrote the SloppyCell program (Gutenkunst et al. 2007) that was used as the ODE solver for the robustness and "Booleanization" projects. He, Dr. Chris R. Myers, and Dr. Andrew G. Clark also met with Ms. Stockwell on many occasions to discuss ideas and brainstorm solutions to obstacles that arose in the analysis of those two projects. Dr. Lisa Nagy of the

University of Arizona was one of the other students in the Center for Cell Dynamics course, and her experimental experience and knowledge of the developmental literature was invaluable for researching and deciding on how to codify the various transcriptional co-regulatory relationships in the gap gene project. She and Dr. Clark will be co-authors on the gap gene paper when it is submitted for publication.

REFERENCES

- Davidson, EH. 2006. The regulatory genome: gene regulatory networks in development and evolution. Burlington, MA ; San Diego: Academic. p. xi, 289 p.
- Gutenkunst, RN, Atlas, JC, Casey, FP, Kuczynski, RS, Waterfall, JJ, Myers, CR, Sethna, JP. 2007. SloppyCell. <http://sloppycell.sourceforge.net>
- Krishna, S, Andersson, AM, Semsey, S, Sneppen, K. 2006. Structure and function of negative feedback loops at the interface of genetic and metabolic networks. *Nucleic Acids Res* 34:2455–2462.
- Krishna, S, Semsey, S, Sneppen, K. 2007. Combinatorics of feedback in cellular uptake and metabolism of small molecules. *Proc Natl Acad Sci U S A* 104:20815–20819.
- Meir, E, Munro, EM, Odell, GM, Von Dassow, G. 2002. Ingeneue: a versatile tool for reconstituting genetic networks, with examples from the segment polarity network. *J Exp Zool* 294:216–251.
- von Dassow, G, Meir, E, Munro, EM, Odell, GM. 2000. The segment polarity network is a robust developmental module. *Nature* 406:188–192.

CHAPTER 1
PREDICTING THE MUTATIONAL ROBUSTNESS OF GENE REGULATORY
NETWORKS

Abstract

Gene regulatory networks, like any evolving biological system, are subject to potentially damaging mutations. Much work has been done to identify classes of networks that are more robust to node deletions -- knockouts of entire genes. Less well understood, however, is the question of which networks best maintain consistent behavior in the face of smaller mutations that affect binding affinity, protein half-life, and other regulatory parameters. Such mutations have subtler effects than whole-gene knockouts do, but because they are far more common than complete loss-of-function mutations, their impact on network evolution may be substantial. We modeled regulatory mutations in simulated gene networks and determined which topologies displayed consistent expression patterns in the face of mutation. We found that networks containing mutual repression feedback loops consistently displayed a more robust response to simulated mutation than we observed in networks where positive feedback motifs predominated, preserving gene expression patterns despite changes in the strength and timing of their interactions. In addition, these mutual-repression topologies were more consistent and complete in their responses to external signals, and more robust to noise. The presence of mutual-repression feedback in experimentally well-characterized developmental regulatory networks suggests that it may be a widespread mechanism for reducing the phenotypic consequences of both environmental and genetic perturbations that would interfere with the process of development.

Introduction

The development of an organism from a single cell to a fully functional adult is one of the most intricately choreographed dances in nature: hundreds of genes turning on and off at the right times, passing signals to one another within and between cells. This complicated ballet can suffer from small missteps along the way, when minor changes in cis-regulatory regions or protein binding affinities change the timing or strength of the signals. Yet the overall pattern of the dance is usually unbroken, and the adult emerges with tissues correctly arranged, and appendages in all the right places (Gerhart and Kirschner 1997; Wagner 2005). In recent years, new methods of experimental manipulation and computational simulation have made it possible to subtly perturb the gene regulatory networks responsible for early embryonic development and study the effects on the expression patterns produced. One observation of these studies has been the surprising level of mutational robustness – that is, genetic canalization -- exhibited by many networks, especially those involved in early embryonic development (von Dassow et al. 2000; Wagner 2005). Such networks can often withstand mutations that affect the quantitative interactions of their component genes and still maintain the correct stable expression patterns that direct the phenotype of the organism.

The degree to which robustness is an evolved property of gene regulatory systems (as opposed to being a side effect of the network topology itself) remains an open question. While tolerance of mutations is clearly a useful property in a process as essential as development, it is difficult for populations to evolve robustness to mutations that have not yet occurred; secondary selection is a weak force (Wright 1934). However, some researchers have argued (Wagner et al. 1997; Rao et al. 2002; de Visser et al. 2003) that selection for robustness to noise and environmental

perturbation -- a helpful feature during the lifetime of each organism, and thus a trait exposed to direct selection -- may have the side benefit of conferring mutational robustness as well. A genotype that confers robustness to mutations has a fitness advantage only when it co-occurs with a deleterious mutation (Wright 1934), making it rare for a mechanism that confers *only* mutational robustness to rise to high frequency in the population. However, developing organisms are subject to a variety of environmental insults, and recent research has documented the substantial stochasticity in cellular processes (McAdams and Arkin 1999; Elowitz et al. 2002; Kaern et al. 2005; Raser and O'Shea 2005; Chang et al. 2008) that causes the concentrations of gene products to fluctuate even in cells inhabiting the comfortably controlled environment of a laboratory. Selection for tolerance of constant, random perturbations like these is therefore ubiquitous and constant. When we look for mechanisms of genetic robustness, then, we might begin our search among the modes of environmental and noise tolerance that organisms have evolved over time.

Here we present results showing that particular regulatory network topologies fulfill this expectation by conferring both dynamical robustness to noise and genetic robustness to quantitative mutations. Selection for such motifs on their noise-buffering merits may have helped give developmental networks the remarkable mutational robustness they exhibit today.

Materials and Methods

Gene regulatory networks can be composed of dozens or hundreds of genes, regulating one another in complex webs of feedback. This feedback, while important to the function of the network, makes it more difficult to understand how the effects of quantitative mutations percolate through the network. Recent evidence that

modularity is an important property of genetic networks (Hartwell et al. 1999; von Dassow and Munro 1999; Raff 2000; Winther 2001; Csete and Doyle 2002; Ihmels et al. 2002; Lipson et al. 2002; Wagner 2002; Alon 2003; Segal et al. 2003; Wolf and Arkin 2003; Schlosser and Wagner 2004; Kashtan and Alon 2005; Qi and Ge 2006; Slonim et al. 2006; Hinman and Davidson 2007; Singh et al. 2008) suggests a way to break the problem down into more manageable components while preserving the properties of the feedback and crosstalk in the larger network. We modeled small gene network modules, each consisting of a single feedback loop (Figure 1.1), and then assembled these modules into pairs, with one module regulating the other and, in some cases, being regulated by it in return (Figure 1.2). The resulting two-module networks were small enough to permit a comprehensive survey of their behavior, while the inter-module regulation let us study the emergent properties of large gene regulatory networks that might arise from crosstalk between modules.

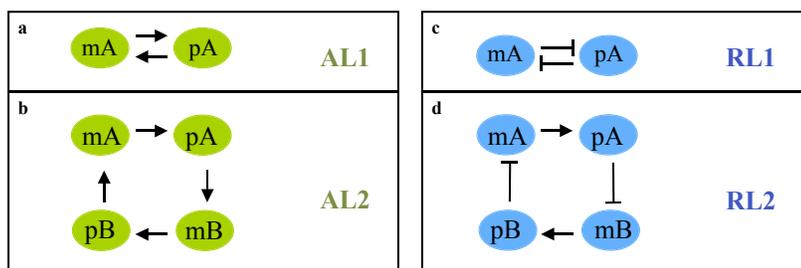


Figure 1.1. The feedback loops used as modules to construct the networks. a,b “Activator Loops,” with positive feedback. **c,d**, “Repressor Loops,” with mutually repressing feedback. The modules are named for whether their internal regulation is activating (AL) or repressing (RL), and whether they have one or two genes (AL1, AL2, RL1, RL2). “mA” represents mRNA for gene A; pA represents the protein product of that gene.

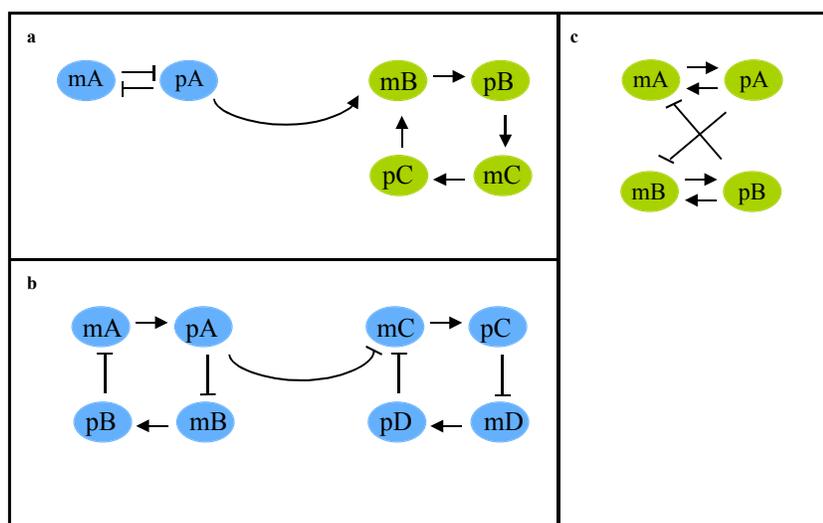


Figure 1.2. Assembling loop modules into two-module networks by adding cross-regulation. **a**, A repressor loop (RL1) activating an activator loop (AL2). **b**, A repressor loop repressing a repressor loop (RL2). **c**, Two activator loops (AL1) repressing each other.

Network topology

We chose feedback loops as the building blocks of our networks because of their biological importance, and because feedback can play a critical role in amplifying or damping changes to networks. Bistable feedback loops, in particular, have received widespread attention as a simple and biologically important variety of gene regulatory module that arises in contexts ranging from development to metabolism (Bhalla and Iyengar 1999; Ferrell and Xiong 2001; Ferrell 2002; Pomerening et al. 2003; Xiong and Ferrell 2003; Ozbudak et al. 2004; Legewie et al. 2006). In gene regulatory loops, two or more nodes (two genes, or an mRNA and its protein) interact with each other so as to give the network two widely separated steady states. Such a loop can act as a kind of switch, changing states in response to external regulation and locking in a response to a transitory signal. We modeled the simplest biologically plausible examples of two classes of transcriptional bistable switch

feedback loops: “activator” and “repressor” loops (AL and RL). Each loop consists of 2-4 nodes, representing mRNAs and proteins that regulate each other’s transcription or translation.

“Activator” loops (AL1, Figure 1.1a; AL2, Figure 1.1b) are positive feedback loops, in which each mRNA or protein node encourages the production of the next node’s mRNA/protein via translation or transcriptional regulation. Stable steady states for these switches are those in which all nodes are at low concentration (low/low) or all nodes are at maximum concentration (high/high). The inclusion of first-order decay terms for all molecules in our models ensured that, for intermediate-value parameters, AL modules were as likely to settle on the low/low state as on the high/high state. Specimens of these loops have been well studied in nature (Alon 2007). For example, autoactivating transcription factors can help strengthen and amplify signals, as Krox and GCM do in the development of sea urchin endomesoderm (Davidson et al. 2002).

“Repressor” loops (RL1, Figure 1.1c; RL2, Figure 1.1d) contain mutually inhibitory interactions, resulting in switches for which the stable states are high/low or low/high. The nodes/genes in each half of the switch repress those in the other half, so that the nodes cannot simultaneously coexist in a highly expressed state. In contrast to the activator loop’s feedback, which aligns the expression states of the nodes, the positive feedback within the repressor loop works to separate the nodes’ expression levels. When one gene is weakly expressed, it has little power to inhibit the transcription of its partner, and its partner can rise from an intermediate concentration to a higher one, which represses the first gene still further until both genes have reached equilibrium at the low/high steady state. RL modules, also, have a substantial literature (Alon 2007). Some of the best-characterized examples are found in the early segmentation patterning network of *Drosophila melanogaster*, where mutual

repression between gap genes (Niessing et al. 1997; Schroeder et al. 2004) helps limit the boundaries of their expression domains (Manu et al.; Pankratz and Jackle 1993). Specific examples of RL modules in the gap gene network include the mutual repression between *hunchback* and *knirps* (Clyde et al. 2003; Schroeder et al. 2004), and between *giant* and *Krüppel* (Eldon and Pirrotta 1991; Kraut and Levine 1991).

These switches are well studied in isolation (Gardner et al. 2000; Becskei et al. 2001; Atkinson et al. 2003; Isaacs et al. 2003; Ingolia and Murray 2007), but their behavior when embedded in a larger regulatory context is more difficult to characterize. Choosing these modules as the "building blocks" of our networks allowed us to study their emergent properties along with the general question of robustness to parameter-altering mutations. We combined the loop modules into two-module networks with transcriptional regulation between the modules in one or both directions (Figure 1.2). We assembled 62 unique pairs of the modules: all possible combinations of AL1/RL2 modules regulating one other, as well as the mirror RL1/AL2 network for each pair to allow a symmetrical, unbiased comparison with the opposite topology (A1.1). This exhaustive approach allowed us to study the full behavioral repertoire of the resulting simulated networks.

The RL1 modules have perhaps the most tenuous biological plausibility of the four motifs; but they provide a symmetrical comparison to the AL1 modules, which are quite common (autoregulatory transcription factors). To compare AL and RL networks without bias, we required topologies and equations that were symmetrical and balanced. The trends found in the more realistic AL2 and RL2 modules were mirrored in the AL1 and RL1 modules, and the two variants of each style of feedback loop ensured that our findings were not an artifact of the particular topology or equations we chose.

Real gene regulatory networks are, of course, much larger than the 2-, 3-, and 4-gene networks studied here. However, this set of networks encompasses all combinations of incoming and outgoing transcriptional regulation of the two categories of loop modules. Restricting ourselves to simple networks allowed us to be exhaustive. The computational approach also allowed completeness; by simulating the networks, we were able to explore thousands of variations that would have been impossible in vivo, giving us a comprehensive picture of each network's dynamical repertoire. Rather than focusing on the peculiarities of a particular biological network, we surveyed the whole landscape of these networks' behavior in a complete and unbiased way.

We represented each network as a set of ordinary differential equations, following the formalism of von Dassow and colleagues (von Dassow et al. 2000; Meir et al. 2002). The mathematical representation of individual modules is illustrated in Figure 1.3; see A1.1 and A1.2 for more detail and for the method of combining multiple regulators affecting a single gene. We used tools from the SloppyCell package (Gutenkunst et al. 2007; Myers et al. 2007) to solve the equations. The parameters in the equations are the half-lives of the mRNA and protein molecules (H); the binding affinity of proteins for the genes they transcriptionally regulate (k , the half-maximal concentration of the regulator, where a large k denotes a weak regulator); and the cooperativity with which regulators bind DNA (n , the Hill coefficient). We varied these parameters to simulate cis- and trans-regulatory mutations (k , n) and mutations in the decay rates of the gene products themselves (H). For example, k captures a gene's sensitivity to a particular transcriptional regulator, so we model mutations that alter the affinity of a transcription factor for its cis-regulatory binding site by varying k . The simulated mutations changed the strength and timing of

interactions between the molecules. We then measured the robustness of each network's mRNA and protein concentrations to the mutations.

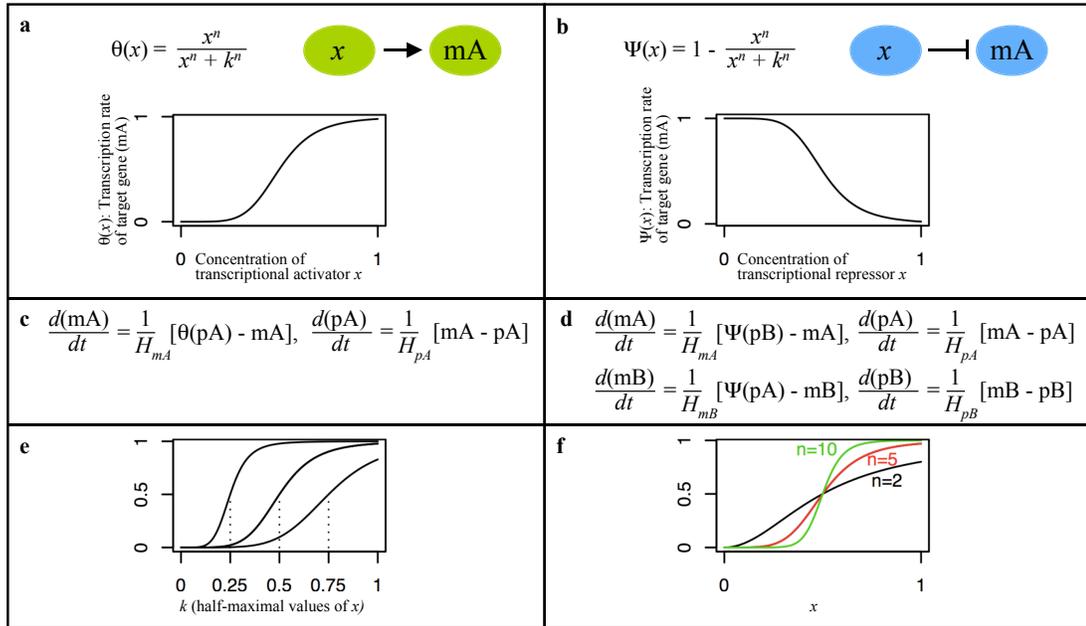


Figure 1.3. Mathematical representation of loop modules. **a**, Transcriptional activation as modelled in AL modules. The transcription rate of the target gene is a saturating function of the concentration of its transcriptional activator. **b**, Transcriptional repression as modelled in RL modules. **c**, The mathematical representation of module AL1 (Figure 1.1a). mA (messenger RNA) change in concentration is a function of its transcription (first term) and first-order decay (second term). The translation rate of pA is a linear function of its messenger RNA, mA. **d**, Module RL2 (Figure 1.1d). **e**, The effect of varying parameter k , a gene's sensitivity to its transcriptional regulator. Changing this parameter simulates mutations in transcription factor binding site affinity. **f**, The effect of varying parameter n , the binding cooperativity of a transcriptional regulator. All equations are normalized by the half-life (H) of the molecule (see A1.2).

Measuring robustness

We developed two quantitative metrics for measuring the robustness of a gene regulatory network to simulated mutation. A traditional way of measuring robustness to parameter variation in a system with steady states is to begin from some initial

condition, let the system evolve until it reaches equilibrium, and note the steady-state condition; then change the parameters, repeat the experiment or simulation, and calculate how much the steady state has shifted as a result of the parameter change (Wall et al. 2004). Our methods are a generalization of that approach, expanded to consider 1000 initial concentrations for each parameter set instead of only one.

For each network, we first set the parameters to biologically plausible values, chose the initial concentrations of the mRNAs and proteins in the network, and then solved the equations numerically to find how the concentrations changed over time (Figure 1.4). Each initial concentration eventually settled into a steady state (with

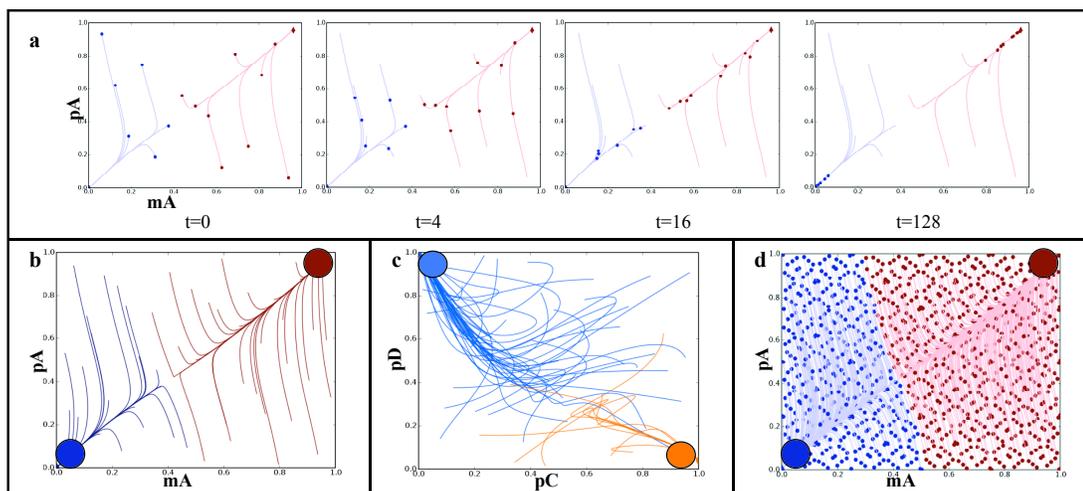


Figure 1.4. Phase portraits. Colors indicate basins of attraction. These portraits are of single modules, for visual clarity. Actual simulations were performed on single modules, static-signal networks, and two-module networks (with up to 8 dimensions). **a**, Dots represent the current concentrations of mRNA and protein, and lines show the trajectories of the concentrations over time. Each initial concentration eventually settles into a steady state. **b**, AL1. Activator loops can settle upon a state where both concentrations are low (bottom left), or a state where both are high (upper right). **c**, RL2. Repressor loop steady states with one gene highly expressed and the other unexpressed. (Only protein node concentrations shown; trajectories appear to cross because of the projection of 4-dimensional space.) **d**, Each network was solved starting from 1000 initial concentrations (dots) to define the basins of attraction.

extremely rare exceptions; see A1.3). We can characterize the overall behavior of a particular network by observing the set of initial concentrations that reaches each stable expression state (i.e., the basin of attraction of the steady states).

If the parameters change – a simulated mutation – some initial concentrations will change their trajectories and reach the steady state in a different corner of phase space. As a result, the basins of attraction change, and a basin may disappear altogether (Figure 1.5). We measured how much the basins shifted as parameters

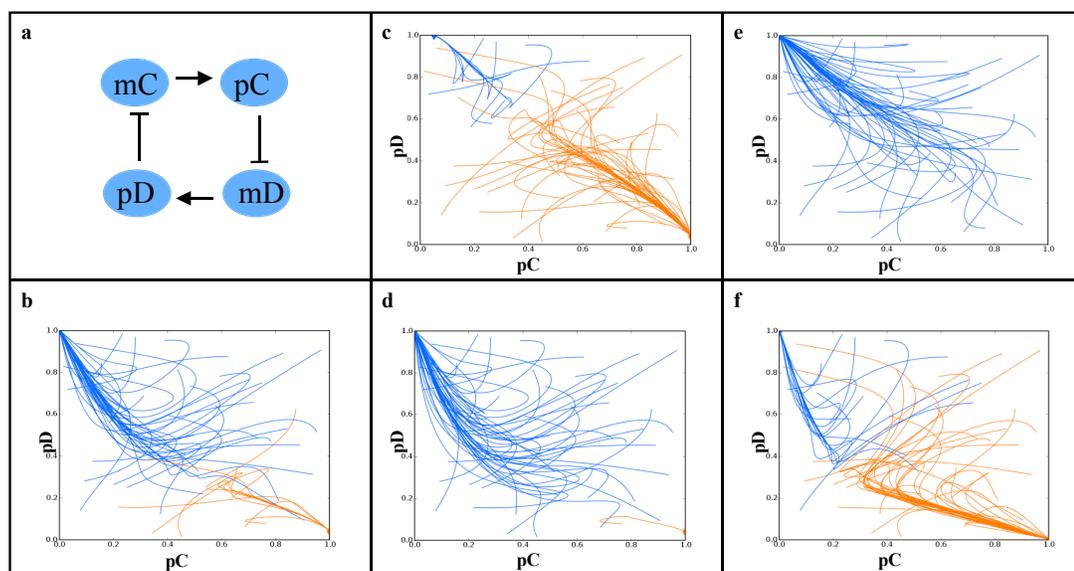


Figure 1.5. Simulated mutations (parameter tweaks) alter the destinations of some trajectories, and thus the basins of attraction. **a**, An isolated repressor loop module (RL2). **b**, Its phase portrait at a given parameter set. **c-f**, The same network’s phase portrait as k (half-maximal repression) parameters are varied. **c**, k for the pD \dashv mC repression edge is doubled. **d**, The same k is halved. **e**, k for the pC \dashv mD repression edge is doubled. **f**, The same k is halved. Taken together, 1000 initial concentrations and their shifting trajectories (of which 50 are shown here) illustrate how the basins of attraction are altered by mutation. The consistency of the basins of attraction constitutes the local robustness of this network module. Similar simulations were carried out for two-module networks to measure their robustness.

varied, and we used these changes to create a quantitative metric for measuring robustness to mutation. We define robustness as the extent to which the basins of

attraction stay constant as the network is buffeted by mutations. (Mutations that alter transient molecular concentrations, but not the steady state at which the concentrations eventually settle or the set of initial concentrations that lead to each steady state, do not lower the robustness score.) Biologically, this robustness value measures how consistent steady-state expression patterns remain as mutations change binding affinities, half-lives, and so on.

We investigated the effects of mutation in two ways. First, to simulate individual mutations, we started at particular parameter sets and individually doubled and halved each parameter. This probed the local parameter space around the chosen parameter set, and the variation altered the original phase portrait of the network, sending some trajectories off to a different steady state. For each steady state, we calculated the proportion of the initial concentrations that were in its basin of attraction in the original parameter set which remained in its basin of attraction for each tweaked parameter set. This yielded the *local robustness* metric for each network. More precisely, for a particular steady state s , the local robustness L is:

$$L_s = \frac{1}{2M} \sum_{p=1}^M \left[\frac{\sum_{i=1}^N \delta(I_{ip}) \delta(I_{i0})}{\sum_{i=1}^N \delta(I_{ip})} + \frac{\sum_{i=1}^N \delta(I_{ip}) \delta(I_{i0})}{\sum_{i=1}^N \delta(I_{i0})} \right] \quad (1)$$

where I is a vector of initial concentrations of the network components (and thus a point in state space), N is the number of initial concentrations for which we simulated the network to characterize state space (1000), and M is the number of tweaked parameter sets. $\delta(I) = 1$ if initial concentration I lies in the basin of steady state s , and $\delta(I) = 0$ if it does not (that is, if integrating the network equations from the initial state I leads to an alternate steady state). In the equation above, parameter set 0 is the original, untweaked parameter set, so L_s represents the average proportion of the basin

of steady state s that was shared between mutated parameter sets and the original parameter set. To calculate the overall local robustness of a network for a particular parameter set, we sum over the list of all steady states (S) that were found by integrating trajectories for the original and mutated versions of that parameter set:

$$L = \sum_{s=1}^S w_s L_s \quad (2)$$

where w_s is a scaling factor reflecting the mean basin size for steady state s across all variants of this parameter set:

$$w_s = \frac{1}{N(M+1)} \left[\sum_{p=1}^M \sum_{i=1}^N \delta(I_{ip}) + \sum_{i=1}^N \delta(I_{i0}) \right] \quad (3)$$

We calculated the local robustness of each of sixteen representative parameter sets for each two-module network (see A1.3), and for 250 representative parameter sets for each isolated module. See A1.3 for more details.

Secondly, to explore the overall repertoire of the networks, we sampled 250 points throughout parameter space for each two-module network, and calculated the *global robustness* of the network. This metric was analogous to the local robustness measurement, except that we calculated the mean basin consistency across all pairwise comparisons of the 250 parameter sets, rather than comparing a given parameter set to its tweaked variants:

$$G_s = \frac{1}{M(M-1)} \sum_{p=1}^M \sum_{q=p+1}^M \left[\frac{\sum_{i=1}^N \delta(I_{ip}) \delta(I_{iq})}{\sum_{i=1}^N \delta(I_{ip})} + \frac{\sum_{i=1}^N \delta(I_{ip}) \delta(I_{iq})}{\sum_{i=1}^N \delta(I_{iq})} \right] \quad (4)$$

$$G = \sum_{s=1}^S w_s G_s, \quad w_s = \frac{1}{NM} \left[\sum_{p=1}^M \sum_{i=1}^N \delta(I_{ip}) \right] \quad (5,6)$$

This approach gave us a global picture of the extent to which different binding affinities and other regulatory characteristics changed the network's behavior. The networks with relatively consistent expression patterns under varying regulatory parameters were considered to have higher global robustness. (More details in A1.3.)

We used these specific, quantitative metrics for robustness to uncover trends in the behavior of different classes of networks.

Global and local robustness have been investigated using alternate metrics in a recent study (Hafner et al. 2009) that measured the robustness of a model of a circadian oscillator. The metrics employed in that study are amenable to stable oscillators, while ours work best for systems dominated by steady-state behavior (stable nodes and damped oscillators). A more substantial difference in our approach is that we considered a more general problem; rather than studying a particular example of a molecular network and fitting parameters to its specific behavior, we sought to characterize the general robustness properties of transcriptional networks.

Individual real gene networks may each explore only a portion of the possible parameter space, rendering part of our analysis irrelevant for a given instance of a regulatory network. However, our goal is to delineate the overall behavior that each kind of feedback motif is capable of, and the exhaustive analysis afforded by simulations lets us do this. The general trends we find when we explore all of parameter and initial-concentration space will lend insight into what kinds of behaviors we can expect from real biological networks with the same topological patterns.

Performing these experiments “*in silico*” allowed us to explore hundreds of thousands of network realizations, mutation types and sizes, and initial concentrations to characterize the networks’ behavior with a comprehensiveness and precision that would not have been possible experimentally.

Static-signal modules

We also simulated isolated RL and AL modules with a single input from outside, mimicking the kind of external regulation they receive when embedded in the

larger networks (Figure 1.6). Such circuits have been studied in metabolic control networks, though in pursuit of different questions than those addressed here (Savageau 1976; Wall et al. 2003; Wall et al. 2004). In these “static-signal” modules, we kept the external regulatory signal constant with time; by contrast, the signal from a true second module changes in strength as the concentration of the signaling protein alters. The static-signal external regulation could help enhance ("actAL," "actRL") or inhibit ("repAL", "repRL") the transcription of a gene in the module it affected. Static-signal modules reproduced the robustness trends we observed with modules embedded in larger networks, but were simpler to study. We varied the strength of the input signal to study how static-signal modules made the transition from bistability (a pluripotent cell) to monostability (a cell with only one possible fate) across 250 parameter sets for each static-signal module. Details are in A1.4.

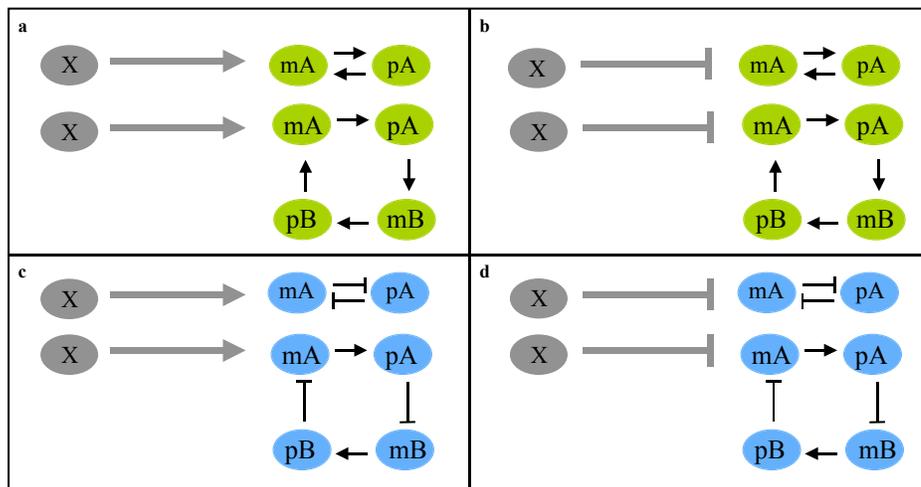


Figure 1.6. Static-signal networks: A constant external “push” activating (a, c) or repressing (b, d) an isolated module. Activator loop (AL) modules are in green; repressor loop (RL) modules are in blue. **a**, actAL1, actAL2. **b**, actRL1, actRL2. **c**, repAL1, repAL2. **d**, repRL1, repRL2.

Results and Discussion

Two-module networks

We found that networks containing repressor loop modules (mutually repressing genes) were consistently more robust to simulated mutation than those with activator loops, by both local robustness (Figure 1.7) and global robustness measures (Figure 1.8). Networks with two AL modules were the least robust, followed by those

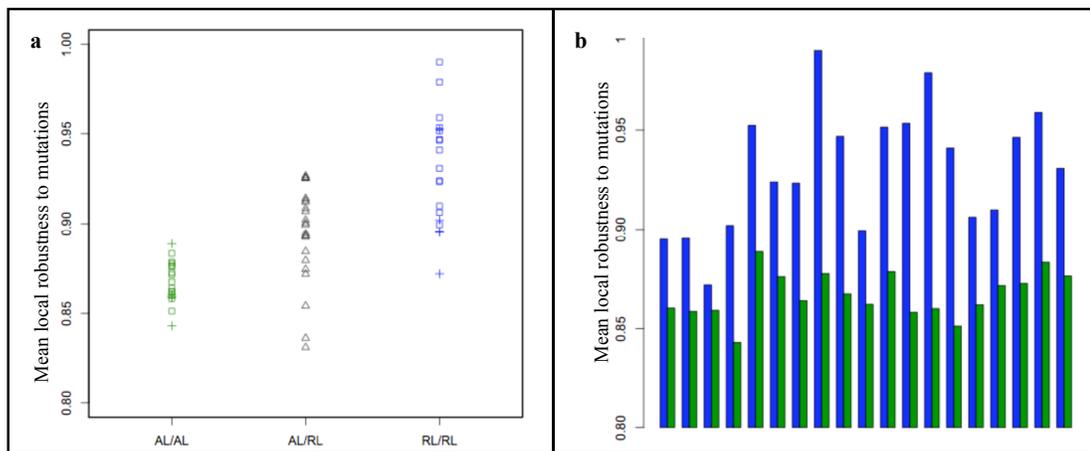


Figure 1.7. Two-module networks with RL (mutual inhibition) modules have higher local robustness to simulated mutations than those with AL (mutual activation) modules. **a**, Local robustness to simulated mutations rises with the number of repressor loops in the network. Local robustness is the proportion of simulated mutations (parameter halving/doubling) for which a given initial concentration gave rise to the same steady state as before the change, averaged over 1000 initial concentrations. The value for each network is the mean local robustness across a sampling of parameter sets. Crosses are 4-node networks, triangles are 6-node networks, and squares are 8-node networks. **b**, Paired comparison: Each RL/RL network (blue) has higher local robustness than the equivalent network with AL modules (green). Networks are in the order listed in A1.1. For global robustness data, see Figure 1.8.

with one AL and one RL module (permutation test: local robustness, p -value = $4.00e-4$; global robustness, $p < 2.00e-4$). Networks with no AL modules were more robust

still (AL/RL vs. RL/RL, permutation test: local robustness, $p = 8.00e-4$; global robustness, $p = 1.60e-3$. AL/AL vs. RL/RL, paired bootstrap comparison: local robustness, $p < 2.00e-4$; global robustness, $p < 2.00e-4$. Details in A1.5).

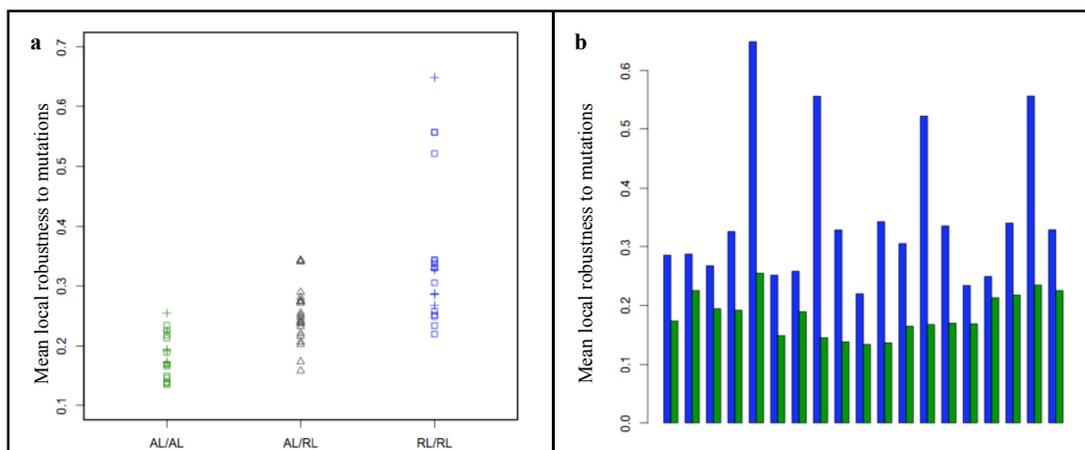


Figure 1.8. Two-module networks with RL (mutual inhibition) modules have higher global robustness to simulated mutations than those with AL (mutual activation) modules. **a**, Global robustness to simulated mutations rises with the number of repressor loops in the network. Global robustness is the average proportion of a steady state's basin of attraction that was constant across pairs of parameter sets (see A1.3). Symbols are as in Figure 1.7. **b**, Paired comparison of RL/RL networks (blue) with the equivalent network with AL modules (green). Networks are in the order listed in section A1.1.

Interestingly, isolated modules did not show this pattern (permutation test on local robustness at 250 parameter sets: AL1 vs. RL1, $p = 0.62$; AL2 vs. RL2, $p = 0.45$. See A1.5 for details). The trend appeared only when modules are embedded in larger networks. The robustness phenomenon, then, is not something we could have discovered by studying the feedback loops in isolation. It appears to be an emergent property of larger networks.

Static-signal modules

The RL modules contribute higher mutational robustness to the two-module networks of which they are a part, but are no more robust than AL modules when isolated. The robustness seems to arise as a byproduct of the RL modules' being subject to regulation from outside (i.e., from another module), or perhaps from the regulation they themselves impose on neighboring modules. When embedded in a larger network, a loop module experiences regulatory signals from other module(s) that vary over time, as the concentration of the regulator changes. To simplify the question of how external regulation affects the modules, we replaced the time-varying regulatory signal with a constant regulatory "push" to the module, creating a "static-signal" module (Figure 1.6). The static input signal exerted the same kind of transcriptional control over the loop that a neighboring module would have, but without the complicating factor of the strength of the regulation changing over time as the network settled to equilibrium.

We then conducted two kinds of tests. First, we asked whether this simplified external regulation was sufficient to recapture the robust-RL phenomenon we had noted in larger networks. We varied the internal parameters of the loop module while maintaining a constant external "push" at a concentration of 1.0, and measured the local robustness of the module. We did this for all 8 of the static-signal modules illustrated in Figure 1.6, for 250 parameter sets each. We found that RL modules did indeed show higher robustness to simulated mutation in this scenario (permutation test on local robustness: p-value < 2.00e-4 for each of actAL1 vs. actRL1, repAL1 vs. repRL1, actAL2 vs. actRL2, and repAL2 vs. repRL2), demonstrating that the emergent robustness of RL-containing networks is due, at least in part, to the input that RL modules receive.

Secondly, we found that the static external regulation was substantially more likely to push RL modules into a monostable condition -- that is, into a mode where all initial concentrations led to the same steady-state expression levels. When isolated, RL and AL networks had approximately equal numbers of pluripotent (bistable) cases (Figures 1.9d-e, 1.10d-e, 1.11d-e, 1.12d-e; $X=0$). However, RL modules responded

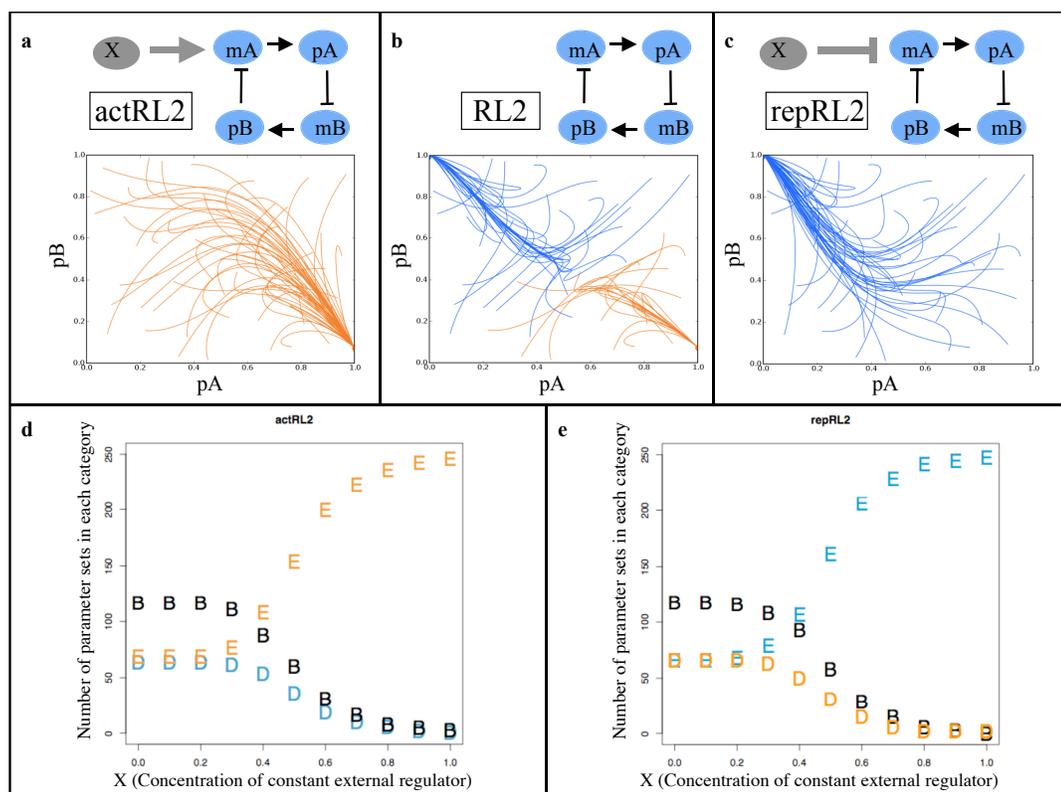


Figure 1.9. External regulation often pushes bistable repressor loop (RL) modules into monostability. Pictured here: RL2. In **c, d**: Orange points represent the number of parameter sets that are bistable at the steady state 1100; blue points are those bistable at 0011. In **c**, 1100 is encouraged (E) by the external regulator, and 0011 is discouraged (D); in **d**, the external regulator is a repressor so the pressures are reversed. B indicates the number of bistable parameter sets.

much more vigorously to external regulation than AL modules did, and nearly all of their bistable parameter sets became monostable for the steady-state encouraged by the

external regulator (Figures 1.9, 1.10). AL modules often stayed bistable when under external regulation, or shifted from monostable for the stable state discouraged by the regulator to a bistable condition (Figures 1.11, 1.12).

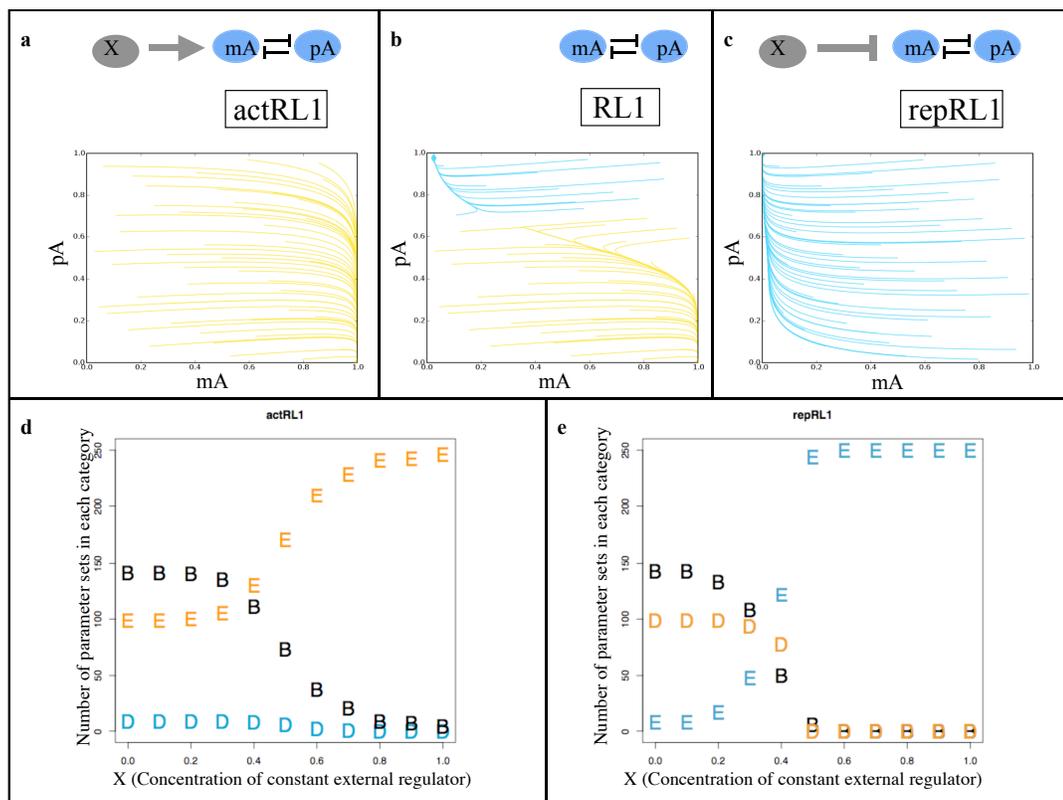


Figure 1.10. External regulation often pushes bistable repressor loop (RL) modules into monostability. Pictured here: RL1. In c, d: Orange points represent the number of parameter sets that are bistable at the steady state 10; blue points are those bistable at 01.

In biological terms, a monostable cell is one that is committed to a particular cell fate. Changes in the concentrations of the gene network components cannot bring a monostable cell to the basin of attraction of another steady state expression level, because no such alternate steady state exists. As a result, following a perturbation, the

network settles back into its original steady state. Such a cell is *dynamically robust* to transient events.

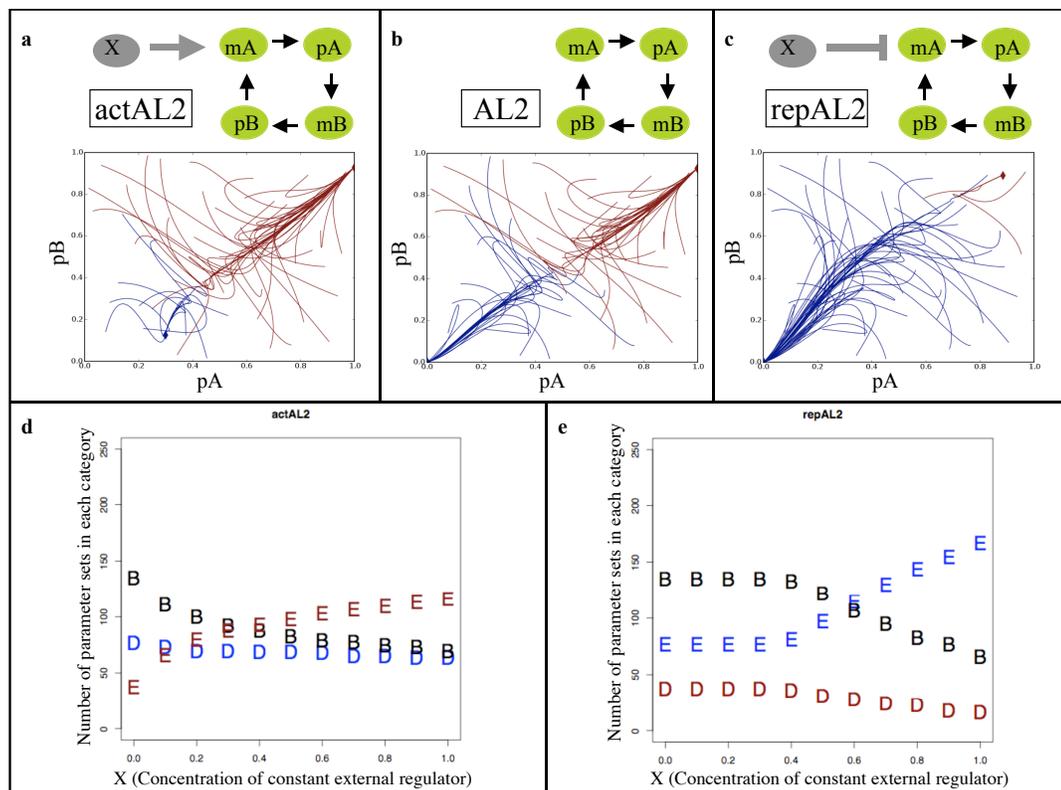


Figure 1.11. Activator loops (here, AL2) often retain their bistability even under the influence of external regulation. In **c, d**: Red points represent the number of parameter sets that are bistable at the steady state 1111; blue points are those bistable at 0000. In **c**, 1111 is encouraged (E) by the external regulator, and 0000 is discouraged (D); in **d**, the external regulator is a repressor so the pressures are reversed. B represents the number of bistable parameter sets.

Bistable cells, by contrast, are vulnerable to internal noise and environmental changes that affect gene expression. The discrete, often small numbers of molecules involved in intracellular reactions can cause substantial shifts in the concentrations of RNA and proteins in a cell (Thattai and van Oudenaarden 2001; Elowitz et al. 2002; Ozbudak et al. 2002). Such shifts can knock the system out of one basin of attraction and into the other, triggering switching between steady-state expression levels (Hasty

et al. 2000), as has been demonstrated in yeast for an activator loop (Becskei et al. 2001).

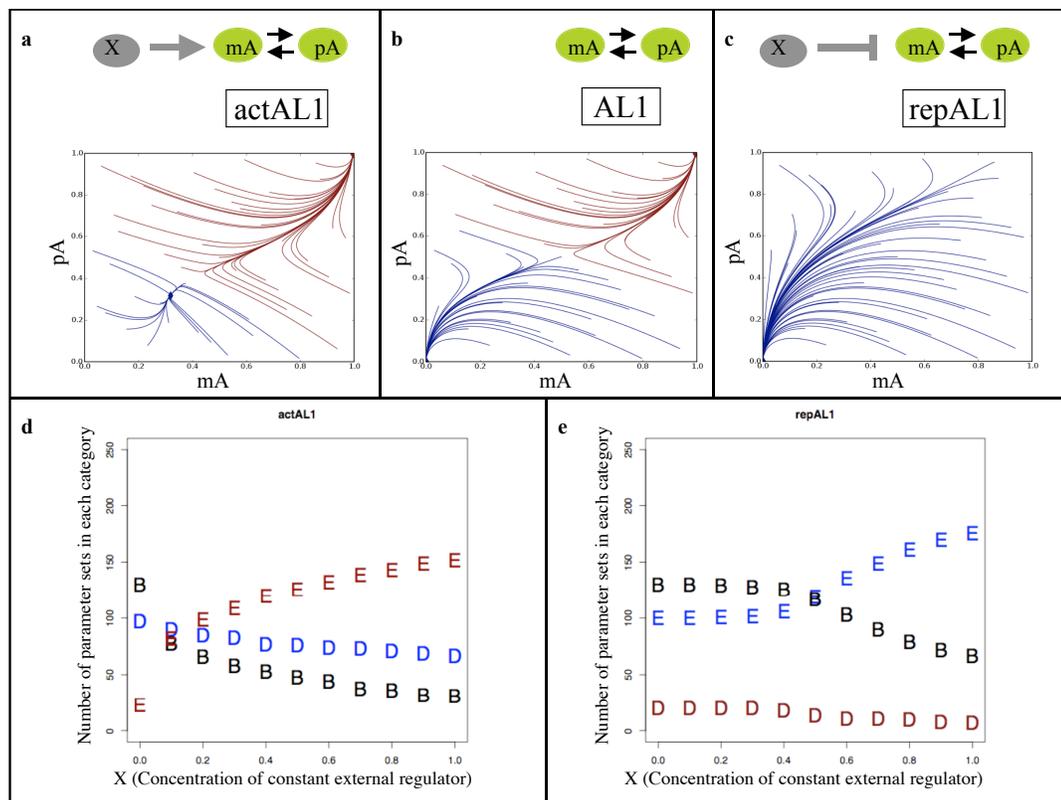


Figure 1.12. Activator loops often remain bistable despite external signals. Pictured here: AL1. In c, d: Red points represent the number of parameter sets that are bistable at the steady state 11; blue points are those bistable at 00.

A pluripotent *repressor loop* network can readily be pushed into either monostable (committed) state by an external regulator. In that state, even substantial noise or environmental perturbation to the network will not persuade the cell to switch to the alternate expression pattern. Pluripotent *activator loop* networks, by contrast, often retain their pluripotency (bistability) when subject to external regulation. The result is cells that, even when pushed toward a particular fate by outside signals, retain their vulnerability to stochastic events that can knock the network out of the basin of

attraction of one steady-state and into the other, changing the cell fate. A gene regulatory network constructed of repressor loops, then, is one that will adopt the expression state its regulators impose upon it, and keep that state despite noise or variation in the system.

In addition to being dynamically robust to transient events and noise, monostable cases are also more robust to parameter variation - that is, to perturbations in the genetic regulatory effects within the network itself, such as mutations (Figures 1.13, 1.14). This trend holds for isolated modules without external regulation as well

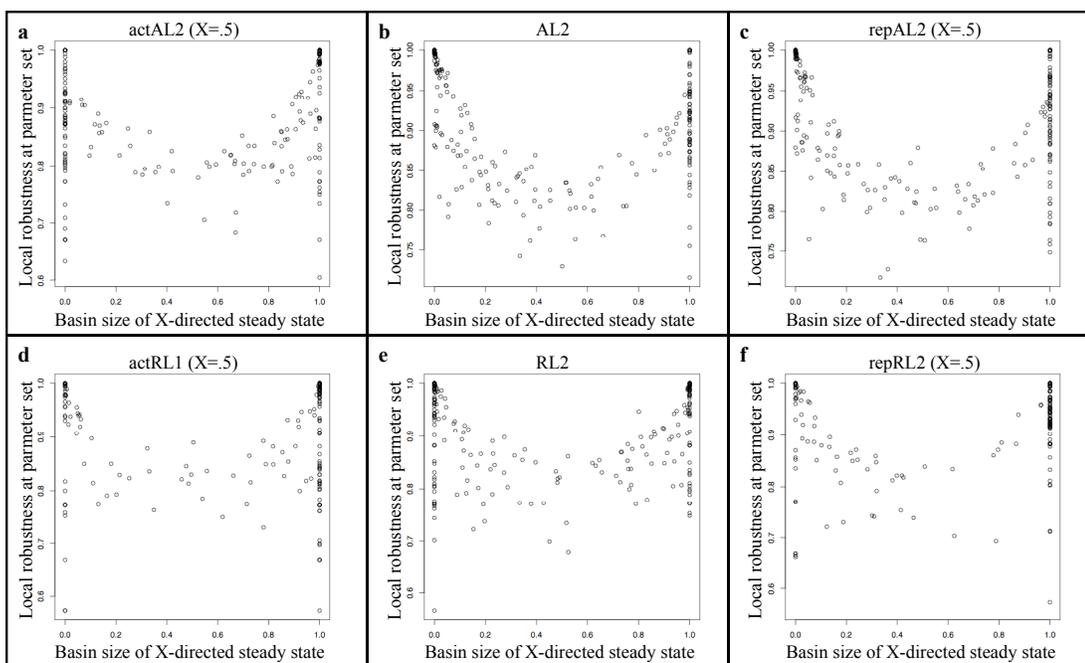


Figure 1.13. Monostability helps confer robustness to parameter variation.

Vertical axis of each panel: Local robustness to simulated mutation at each of 250 parameter sets for a static-signal module. Horizontal axis: Basin size for the steady state “encouraged” by constant external regulator X (at X=0.5) for a given parameter set, measured as the proportion of initial concentrations that settled on that steady state. Basin sizes of 0 (far left of each panel) or 1.0 (far right) are monostable. In each panel, parameter sets that are monostable for either steady state tend to be more robust to the simulated mutations, resulting in a U-shaped distribution.

(Figure 1.13b,e; Figure 1.14b,e). For monostable cases, small changes affect the shapes of the trajectories, but generally not their destined steady state. Therefore, mutations change the transient concentrations, but leave the final, stable expression pattern unchanged.

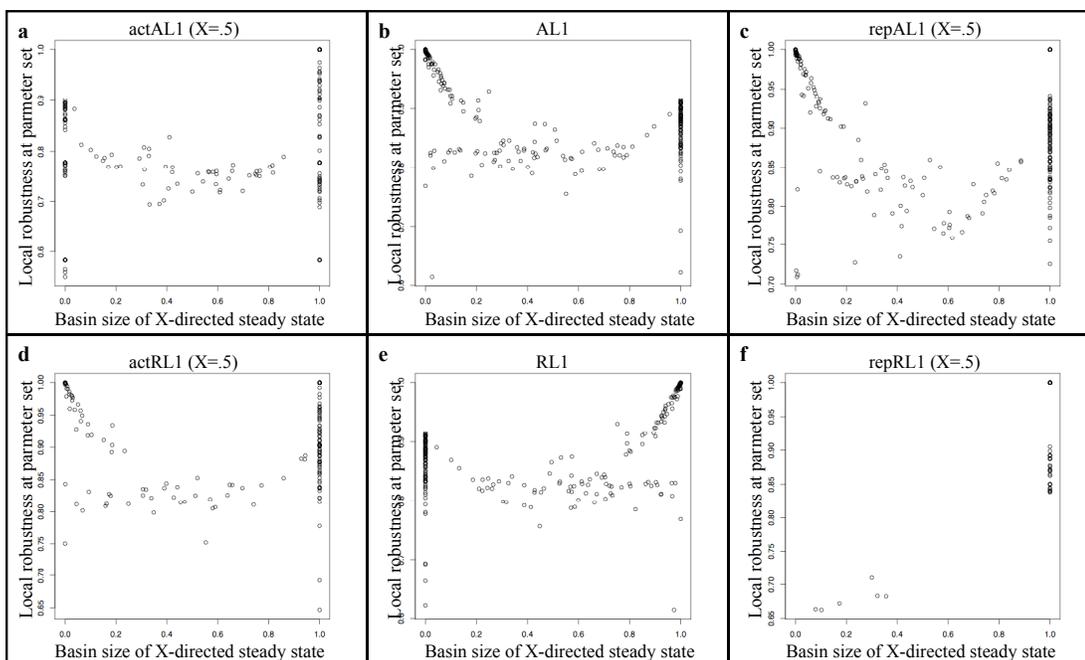


Figure 1.14. Monostability helps confer robustness to parameter variation (2-node modules). Analogous to Figure 1.13, but with the 2-node modules AL1 and RL1.

The effect of external input is often to move bistable isolated modules to a monostable state, which is more robust to parameter variation. RL modules are more susceptible to this pressure toward monostability. This explains their greater robustness to simulated mutation, which we observe only when they are embedded in a larger regulatory context.

Repressor loops can adopt either monostable state when stimulated to do so by external regulation (Figures 1.9, 1.10). The symmetry of the feedback within the

repressor loop is what causes this "ambidextrous" response. Repressor loops have internal reinforcement of both up-regulation and down-regulation. An external repressor, of course, is reinforced directly by pA's influence on mA. Less obviously, an external activator also encounters internal reinforcement of its signal, albeit indirectly. It up-regulates mA, which down-regulates pA, which in turn allows mA's concentration to rise. The result is that even the activating "pushes" on RL modules induces many transitions to monostability.

The balance between steady states in activator loops is of a different kind. Instead of two nodes jockeying to knock down each other's expression, activator loops achieve the unexpressed state (00/0000) when the decay rates of both nodes are balanced with the strength of mutual activation. External regulation can nudge this balance upward or downward, but tends not to have the precipitous effect of boosting one node into complete dominance over the other, as in the repressor loop.

This difference is visible in the transitions made by individual modules as external regulation gradually grows more intense (Figures 1.15-1.18). Repressor loops tend to reach a "tipping point" and flip from one monostable state to the other. For example, in parameter set 19 of Figure 1.15, the RL2 module is monostable when the activating external signal is at value 0.0, shown at the left side of the box (green circles). None of the trajectories end at steady state 1100. The size of the 1100 basin of attraction remains essentially unchanged as the external activation increases from 0 to 0.5, but abruptly increases to 1 – monostable at the other steady state – when the signal strength reaches 0.6. By contrast, in the corresponding parameter set for the activator loop AL2 (Figure 1.17), the basin of attraction of steady state 1111 gradually increases in size (green circles), and the system remains bistable even at X=1.0. Interestingly, RL modules' extra sensitivity to external signals near the tipping point – a kind of instability – confers extra robustness to parameter variation, because it

promotes the switch into monostability. Activator loops, by contrast, are more likely to experience a gradual movement of the boundary between the basins of attraction, while retaining bistability.

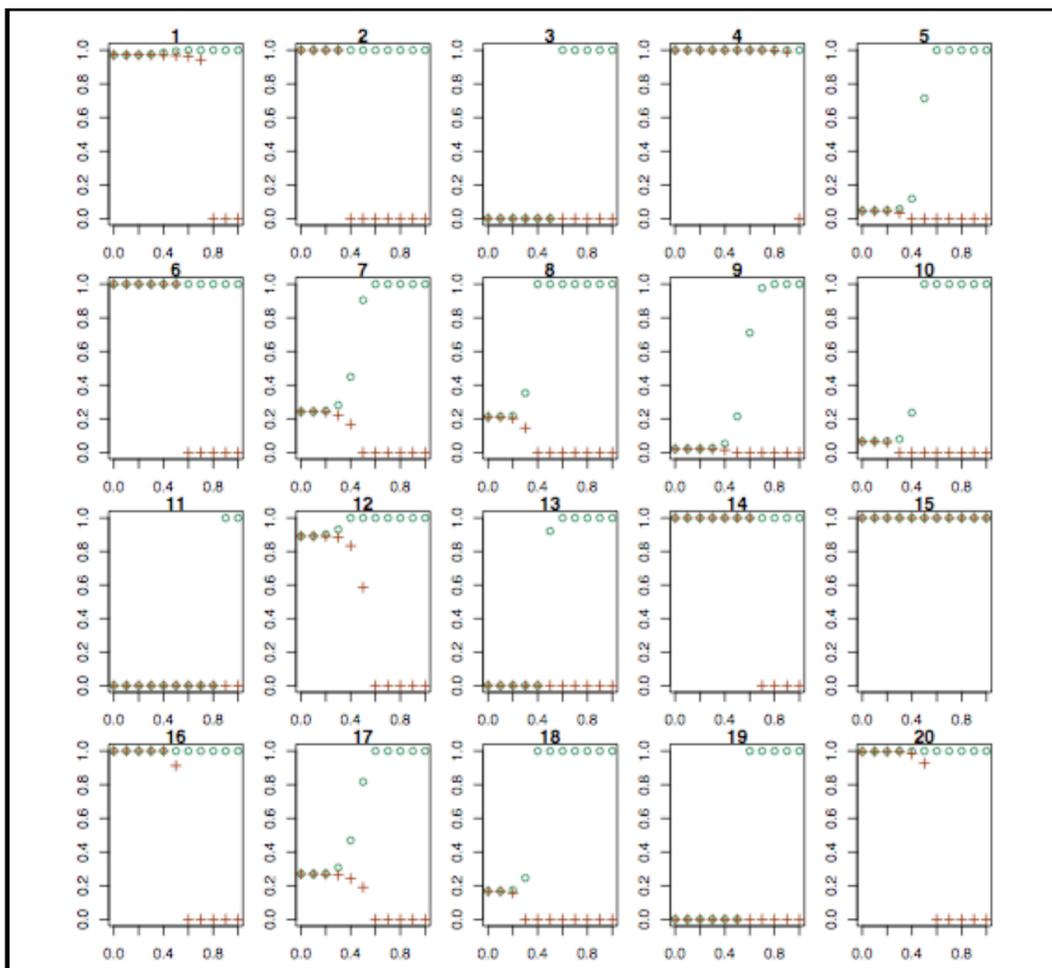


Figure 1.15. As the strength of the external signal increases (horizontal axis), the size of the basin of attraction (vertical axis) changes more abruptly for RL modules (above and Figure 1.16) than for AL modules (Figure 1.17, 1.18). Each panel in this series of four figures represents a single parameter set of an RL or AL module. **Figure 1.15:** Size of 1100 basin for module RL2 as it receives constant external activation (green circles) or repression (red crosses) of increasing intensity. Initially bistable parameter sets of the repressor loop module abruptly flip to the monostable state (top or bottom of panel) that is encouraged by the external signal.

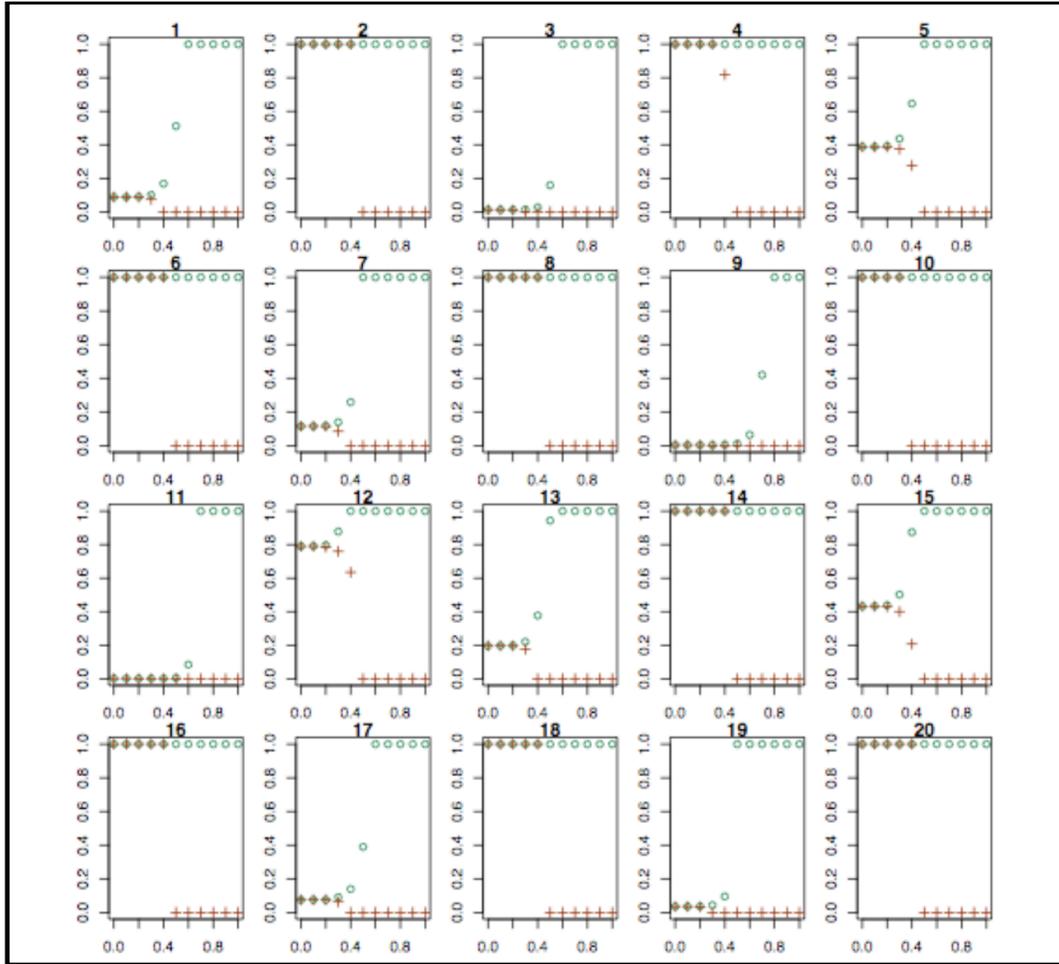


Figure 1.16: Size of 10 basin for module RL1 as it receives constant external activation (green circles) or repression (red crosses) of increasing intensity. Initially bistable parameter sets of the repressor loop module abruptly flip to the monostable state (top or bottom of panel) that is encouraged by the external signal.

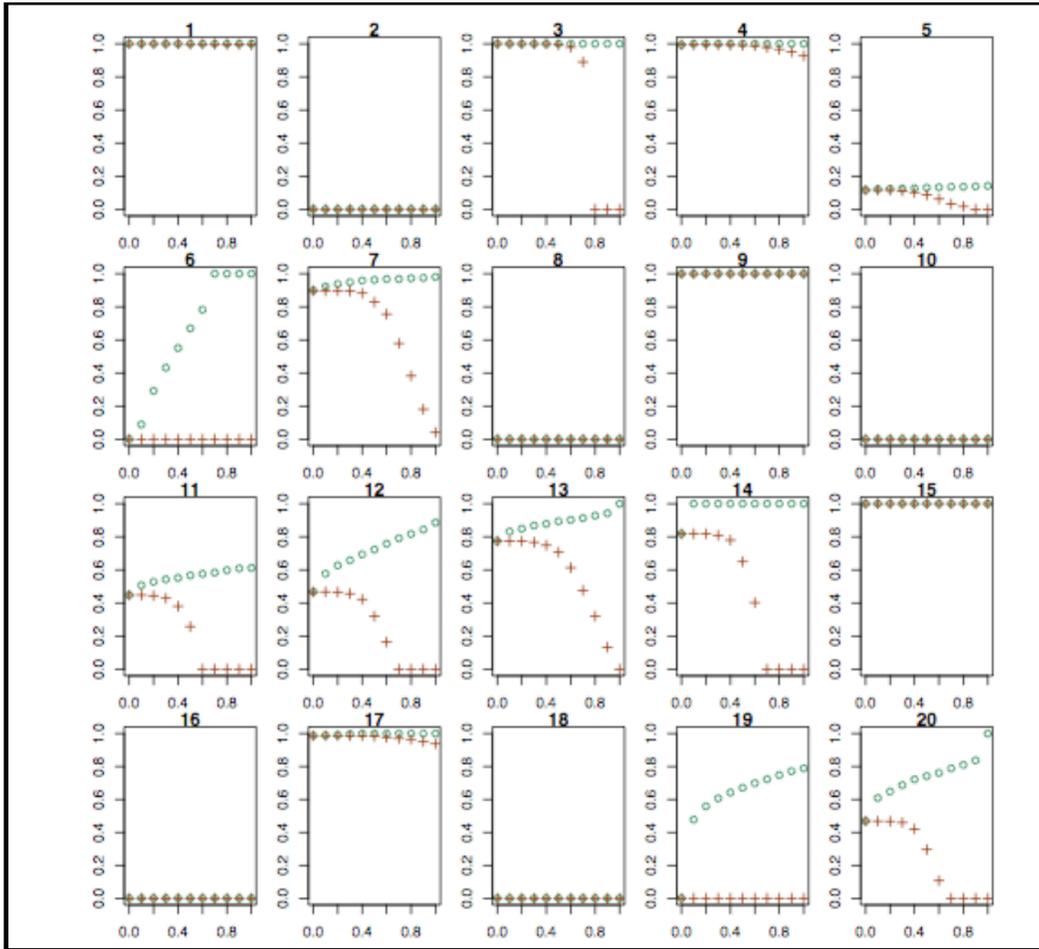


Figure 1.17: Size of 1111 basin for module AL2 as it receives external activation (green circles) or repression (red crosses) signals of increasing intensity. Activator loop basins change much more gradually than repressor loop basins, and sometimes remain bistable even at maximum signal strength.

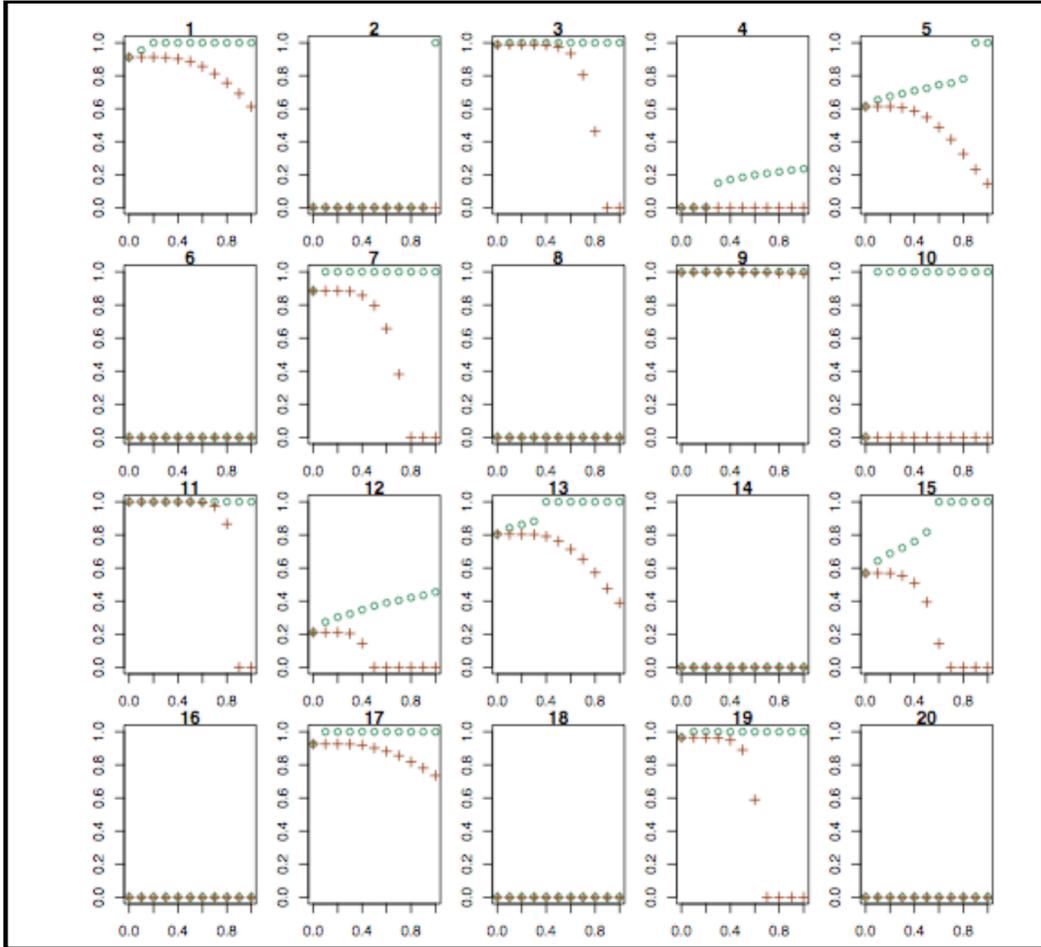


Figure 1.18. Size of 11 basin for module AL1 as it receives external activation (green circles) or repression (red crosses) signals of increasing intensity. Activator loop basins change much more gradually than repressor loop basins, and sometimes remain bistable even at maximum signal strength.

Because repressor loops can easily be pushed into either monostable condition, they are well suited to gene networks controlling cell fate decisions. The ideal decision-making mechanism for a cell undergoing differentiation would be one that integrates external signals, and then responds to them in a complete and unambiguous way that is not compromised by stochastic variations in molecular concentrations, or by quantitative genetic variation in the details of its interior interactions. In the language of dynamics, it would be monostable at the “encouraged” steady-state when pushed that way by an external regulator, monostable at the other steady-state when stimulated in that direction instead, and robust to mutation in both cases. Regulatory networks that meet all four criteria fall into the special category of optimal decision-making mechanisms. We scored each module type for these four characteristics (see A1.6), and found that RL modules are dramatically closer to the cell-differentiating ideal (permutation test, details in A1.6: AL1 (mean = 1.60) vs. RL1 (mean = 1.88), p -value $< 2.00e-4$; AL2 (mean = 1.64) vs. RL2 (mean = 1.76), $p < 2.00e-4$) as illustrated in Figures 1.19 and 1.20. Repressor loops, then, not only can achieve robust commitment to one cell fate when signaled to do so; they can also achieve the other cell fate when pushed in the other direction. If we could design a network for making developmental decisions, we would choose to assemble it out of mutually repressing genes.

Of course, the developmental networks in nature are not designed, but are the product of evolution. While selection for mutational robustness may be a weak force, selection for succeeding at the precise and delicate task of building a body -- despite intrinsic and extrinsic noise due to variation in diffusion rates, molecule numbers, temperature, and so on -- is surely a strong selective pressure. We speculate that

selection for tolerating variation within an organism's lifetime has led, in part, to the abundance of repressor loops we see in well-studied developmental networks

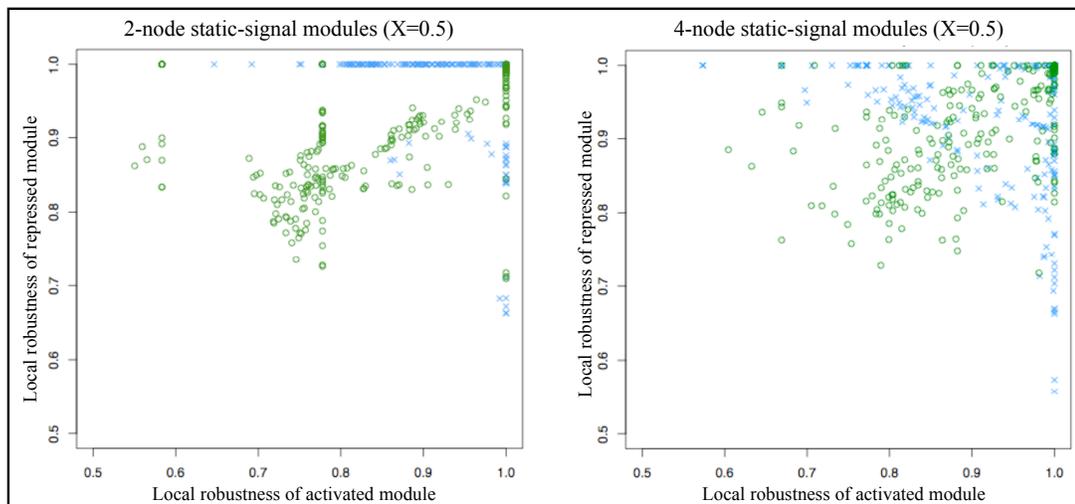


Figure 1.19. Repressor loops readily become *robust* in response to both activating and repressing external signals. At each of the 250 parameter sets of AL and RL modules, we measured the local robustness to parameter variation as the module was activated or inhibited by a static external signal. Green: AL1, AL2. Blue: RL1, RL2.

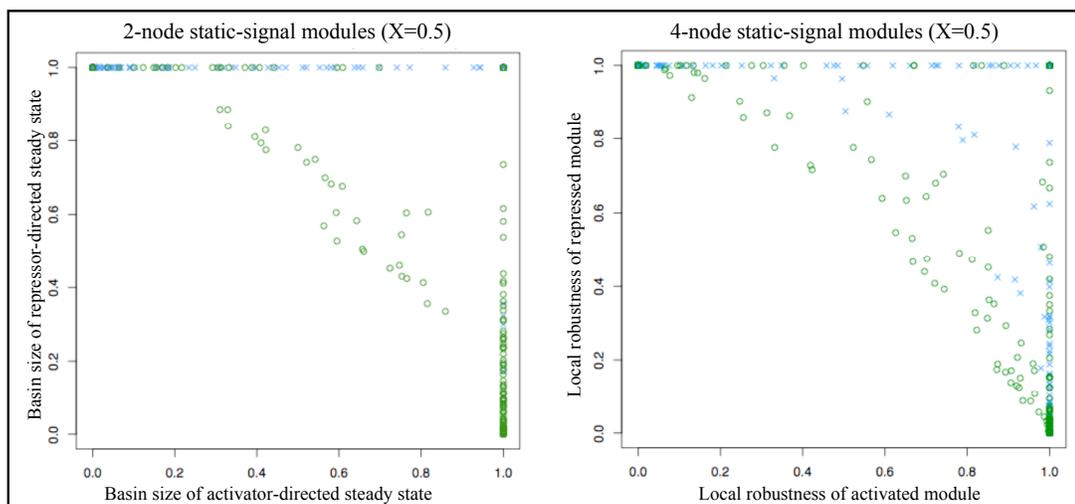


Figure 1.20. Repressor loops readily become *monostable* in response to both activating and repressing external signals. At each of the 250 parameter sets of AL and RL modules, we measured the basin size (proportion of initial concentrations reaching a given steady state) as the module was activated or inhibited by a static external signal. Green: AL1, AL2. Blue: RL1, RL2.

(Brandman et al. 2005) and, as a fortunate side-effect, to the toleration of genetic perturbation. The gap gene network in early *Drosophila* segmentation patterning, for example, is composed almost entirely of interlocking repressor loops (Niessing et al. 1997; Clyde et al. 2003; Schroeder et al. 2004), and the downstream segment polarity network contains them as well (Von Dassow and Odell 2002).

The especially thorough elucidation of the regulatory network of sea urchin development (Davidson et al. 2002) has uncovered many feedback loops employed to “lock in” an expression pattern transiently specified by a signal from outside the loop module. Repressor loops, tending toward monostability when either activated or repressed by an external signal, are especially well suited to performing this task. A feedback loop that is bistable when isolated (as both RL and AL modules often are) can occupy either of the two steady-state expression levels. When an external signal pushes the feedback module from a bistable phase into monostability, however, it moves the cell to the only steady state expression allowed now, regardless of which steady state it had occupied before. When the external signal fades away, the module relaxes back into the bistable state – but now the cell stays at the steady state to which it was pushed, because of the hysteresis in the system. If the feedback loop had retained both steady states during the outside regulation, any cells occupying the other steady-state could remain there during the signal, and retain that state when the signal disappeared, leaving no record of what the signal had been. Thus, only a module that becomes monostable when signaled can reliably record the transient information. Our simulations indicate that repressor loops are far more apt to do so, and in fact they appear in the urchin regulatory networks as delimiters of developmental regions. For example, upstream specification signals set the switch position of the *gooseoid/deadringer* repressor loop (Davidson et al. 2003), which serves to demarcate the oral ectoderm region. We predict that as more developmental networks

are mapped in detail, repressor loops will turn up frequently in the role of memory circuits, recording transient signals in order to direct downstream differentiation. In the meantime, synthetic repressor loops in microbes have experimentally demonstrated that RL modules can be readily pushed into either stable state by means of transient external signals, and that such modules will remember their new state after the signal has ceased (Gardner et al. 2000).

In summary: We developed quantitative metrics of the robustness of gene expression in the face of regulatory variation. Using these measurements, we found that two-module networks which contained more mutual repression (RL) loops were more robust to simulated regulatory mutations (parameter variation) than those with activator loop (AL) modules. Isolated loop modules showed no such tendency. Analysis of static-signal modules revealed that the robustness trend arises, at least in part, from the fact that RL modules are more likely to become monostable in response to external regulation, such as that received by another gene or module in the network. That is, the repressor loops settle exclusively on whichever steady state is encouraged by the outside signal, regardless of the initial concentrations of the network components. AL modules, by contrast, often retain their bistable (pluripotent) status despite external signals to the contrary. This incomplete response to regulation renders them more vulnerable to noise (which can stochastically flip the state of the feedback loop from one steady state to another) and regulatory mutations (which can change the persistent boundary between basins of attraction). Repressor loops are thus better suited for the precise and consistent cell fate decisions required by much of development, while activator loops may play a role in regulatory networks where variable responses to identical conditions are desirable.

Further exploration

Critical tasks that must proceed in a stereotypical fashion every time, like many of those in development, should be selected to employ robust motifs like repressor loops. At the same time, our findings suggest that where repressor loops appear, they may buffer more regulatory variation than other motifs can. An experimental prediction thus arises from our findings: Because they can mask deleterious variation in regulatory parameters, we predict that repressor loops in nature should tend to have more standing regulatory variation in the population than activator loops in gene regulatory networks of similar functional importance. Additionally, the converse should be true of activator loops: when they appear in networks that lead to highly canalized phenotypes, we expect to find very little regulatory variation in their internal feedback interactions.

We have considered feedback loops in the context of developmental networks that require precision and repeatability, but some genetic networks have the task of amplifying stochastic effects instead of damping them. For example, bacterial persistence is a phenomenon whose mechanistic details are not yet clear, wherein the members of a genetically identical population of bacteria adopt either a slow-growing, antibiotic-resistant "persister" state or a virulent, fast-growing, antibiotic-vulnerable phenotype (Balaban et al. 2004; Kussell and Leibler 2005; Kussell et al. 2005). Cells can switch back and forth between these states, apparently stochastically. This behavior has been observed in a number of bacterial species, including some important pathogens (Balaban et al. 2004). A phenomenon like this, where it is actually desirable for stochastic signals to flip the expression pattern from one state to another, would be an excellent fit for activator loops and their tendency to retain bistability. There is tantalizing evidence of just such an activator loop between TNF- α and

antigen 85B, a protein involved in the active-growth phase of *Mycobacterium tuberculosis* (Wilkinson et al. 2001; Zahrt 2003). Other systems in which diversity is beneficial, such as gene networks involved in the immune response, would also be expected to employ the noise-magnifying abilities of activator loops.

The topology of repressor loop modules lends them the ability to respond with alacrity to external regulation in either direction. This property of RL modules – the ability to cleanly and robustly adopt either steady state when pushed toward it by an external signal – is invaluable in a context like development, when switch-like decisions are common and each cell must reliably adopt the correct expression state. Organisms require gene regulatory networks that maintain their functions in many genetic backgrounds and despite small intrinsic noise and mutational insults. Our data suggest that the robust class of repressor loop modules will be common in large-scale gene regulatory networks, especially those where the correct expression state is important to the fitness of the organism. Where amplification of stochastic events is useful, we predict that activator loops will predominate instead.

Some developmental processes may even benefit from a combination of the two motifs' special abilities: a bistable activator loop adopting an expression state nearly at random, which is then relayed as a regulatory signal to a repressor loop entrusted with robustly committing the cell to that fate. When a mosaic of different cell types in a single tissue is desirable, interlocked activator and repressor loops could help generate that pattern. As Csete and Doyle (2002) point out, "An important use of positive feedback is to deliberately destabilize equilibria and amplify small differences to create switches and to break symmetries and homogeneities." Becksei et al. (2001) experimentally demonstrated that an activator loop in a eukaryote could switch cells from one stable state to another in response to noise. Differentiation, initiated by activator loops and stabilized by robust repressor loops, could thus help determine

developmental cell lineages. For example, a recent paper (Chang et al. 2008) showed that stochastic switching between stable states in mouse hematopoietic progenitor cells played an important role in differentiating the population into erythroid or myeloid cell fates.

We limited our study to four mathematical representations of the AL and RL topology, in order to explore those models exhaustively. The dose-response curves of the Hill functions we employed have been widely used to represent many different kinds of regulation, with quite different biological mechanisms underlying the regulation. Because of the general nature of our mathematical representation, we anticipate that our findings about repressor vs. activator loops will generalize outside transcriptional regulatory networks. However, there are many levels of interaction that we omitted in our simple models, such as translational and post-translational regulation. Additional regulatory mechanisms such as RNAi are emerging as important influences of gene expression, and feedback loops in biological gene networks are invariably more complex than those studied here. In addition, there are of course other ways to mathematize transcriptional regulation: stochastic models, kinetic models, Boolean models, and other styles of the continuous differential equations we chose, for example. We hope others will continue to explore the question of which topological motifs confer environmental and genetic robustness to the networks in which they are embedded. In this way, we can begin to break down the dauntingly complex dynamics of genome-scale regulatory networks into understandable components, and improve our ability to predict the expression patterns of large feedback-containing networks based on their topology.

Existing databases of transcriptional interactions, such as TRANSFAC (Wingender 2008) or RegulonDB (Gama-Castro et al. 2008), consist mostly of unidirectional motifs -- chains, feed-forward loops, and cascades. However, many of

the well-studied gene regulatory networks such as *Drosophila* segmentation patterning (Niessing et al. 1997; Von Dassow and Odell 2002) abound with feedback loops. Part of the answer to this paradox is in the bias that partial sampling brings to the databases. A feedback loop, if we omit one of its edges because of incomplete knowledge, unravels into a linear pathway. As a result, the sampling bias in any set of transcriptional interactions with missing data should artificially elevate the number of cascade and chain motifs, and lower that of loops. Indeed, the most famous gene regulatory module of all, the *lac* operon, is a repressor loop: *lacI*'s protein suppresses transcription of the mRNA for lactose permease, which increases allolactose concentrations in the cell, and allolactose allosterically inhibits *lacI*. However, the *lac* operon would appear to be a chain instead of a feedback loop if we omitted allolactose -- a molecule not present in transcriptional databases. Hybrid datasets including both transcription factors and small molecules like nutrients do, in fact, find an abundance of feedback loop motifs (Babu and Teichmann 2003), though these surveys have so far only been done in microbes, so far as we are aware, which limits their applicability to developmental questions. We therefore eagerly await the arrival of integrated developmental databases that encompass genes, proteins, ligands, and metabolites, and look forward to exploring the regulatory motifs that will be revealed therein.

APPENDIX

A1.1. Details of the two-module networks

To enable a clean comparison of the effects of activator vs. repressor loops (AL vs. RL) while controlling for the effects of network size, we constructed all topologically unique regulatory pairs of AL1 and RL2 modules, and then made the mirror image of each two-module network using RL1 and AL2 modules. All regulation between modules was transcriptional, i.e., from a protein node in one module to an mRNA node in the other. The complete list of two-module networks follows. “p” nodes represent the concentration of proteins, and “m” nodes represent the concentration of mRNAs. For convenience, genes in 2-node modules (AL1, RL1) are labeled A (in the first module in the pair; thus, mA and pA for the mRNA and protein nodes of that gene) or B (in the second module of the pair). Genes in 4-node modules (AL2, RL2) are labeled C and D (first module) or E and F (second module).

Mathematical representation of the networks may be found in A1.2.

- 1a. AL1actAL1 (Node pA in the first module transcriptionally activates node mB in the second module.)
- 1b. RL1actRL1 (Same between-module regulation as in 1a, but the individual modules are now repressor loops rather than activator loops.)
- 2a. AL1repAL1 (Node pA in the first module transcriptionally represses node mB in the second module.)
- 2b. RL1repRL1
- 3a. AL1actAL1act (Feedback between the modules. Node pA in the first AL1 module transcriptionally activates node mB in the second AL1 module; node pB in the second AL1 activates node mA in the first AL1.)
- 3b. RL1actRL1act

4a. AL1actAL1rep (pA in the first AL1 activates mB in the second AL1; pB in the second AL1 represses mA in the first AL1.)

4b. RL1actRL1rep

5a. AL1repAL1rep

5b. RL1repRL1rep

6a. AL1actRL2 (pA in AL1 transcriptionally activates mC in RL2.)

6b. RL1actAL2

7a. AL1repRL2

7b. RL1repAL2

8a. RL2actAL1 (pC in RL2 transcriptionally activates mA in AL1.)

8b. AL2actRL1

9a. RL2repAL1

9b. AL2repRL1

10a. AL1actRL2act1 (pA in AL1 activates mC in RL2; pC in RL2 activates mA in AL1.)

10b. RL1actAL2act1 (The same nodes are involved in the same between-module interactions, though within-module regulation is of course the opposite of 10a.)

11a. AL1actRL2act2 (pA in AL1 activates mC in RL2; pD in RL2 activates mA in AL1.)

11b. RL1actAL2act2

12a. AL1actRL2rep1 (pA in AL1 activates mC in RL2; pC in RL2 represses mA in AL1.)

12b. RL1actAL2rep1

13a. AL1actRL2rep2 (pA in AL1 activates mC in RL2; pD in RL2 represses mA in AL1.)

13b. RL1actAL2rep2

14a. AL1repRL2rep1 (pA in AL1 represses mC in RL2; pC in RL2 represses mA in AL1.)

14b. RL1repAL2rep1

15a. AL1repRL2rep2 (pA in AL1 represses mC in RL2; pD in RL2 represses mA in AL1.)

15b. RL1repAL2rep2

16a. AL1repRL2act1 (pA in AL1 represses mC in RL2; pC in RL2 activates mA in AL1.)

16b. RL1repAL2act1

17a. AL1repRL2act2 (pA in AL1 represses mC in RL2; pD in RL2 activates mA in AL1.)

17b. RL1repAL2act2

18a. RL2actRL2 (pC in the first RL2 module activates mE in the second RL2 module.)

18b. AL2actAL2

19a. RL2repRL2 (pC in the first RL2 represses mE in the second RL2.)

19b. AL2repAL2

20a. RL2actRL2act1 (pC in the first RL2 activates mE in the second RL2; pE in the second RL2 activates mC in the first RL2.)

20b. AL2actAL2act1

21a. RL2actRL2act2 (pC in the first RL2 activates mE in the second RL2; pE in the second RL2 activates mD in the first RL2.)

21b. AL2actAL2act2

22a. RL2actRL2act3 (pC in the first RL2 activates mE in the second RL2; pF in the second RL2 activates mD in the first RL2.)

22b. AL2actAL2act3

23a. RL2actRL2act4 (pC in the first RL2 activates mE in the second RL2; pF in the second RL2 activates mC in the first RL2.)

23b. AL2actAL2act4

24a. RL2actRL2rep1 (pC in the first RL2 activates mE in the second RL2; pE in the second RL2 represses mC in the first RL2.)

24b. AL2actAL2rep1

25a. RL2actRL2rep2 (pC in the first RL2 activates mE in the second RL2; pE in the second RL2 represses mD in the first RL2.)

25b. AL2actAL2rep2

26a. RL2actRL2rep3 (pC in the first RL2 activates mE in the second RL2; pF in the second RL2 represses mD in the first RL2.)

26b. AL2actAL2rep3

27a. RL2actRL2rep4 (pC in the first RL2 activates mE in the second RL2; pF in the second RL2 represses mC in the first RL2.)

27b. AL2actAL2rep4

28a. RL2repRL2rep1 (pC in the first RL2 represses mE in the second RL2; pE in the second RL2 represses mC in the first RL2.)

28b. AL2repAL2rep1

29a. RL2repRL2rep2 (pC in the first RL2 represses mE in the second RL2; pE in the second RL2 represses mD in the first RL2.)

29b. AL2repAL2rep2

30a. RL2repRL2rep3 (pC in the first RL2 represses mE in the second RL2; pF in the second RL2 represses mD in the first RL2.)

30b. AL2repAL2rep3

31a. RL2repRL2rep4 (pC in the first RL2 represses mE in the second RL2; pF in the second RL2 represses mC in the first RL2.)

31b. AL2repAL2rep4

A1.2. Mathematical representation of the networks

We adapted the Hill-function centered equations of the Center for Cell Dynamics, which are described in (von Dassow et al. 2000; Meir et al. 2002). These are the most biologically well-grounded mathematical representations of transcription and translation of which we are aware, and they have the additional advantage of being easy to mix and match as different combinations of regulators conspire to modify the transcription of a single gene. We followed their method for non-dimensionalizing and normalizing the equations, so that all variables are confined to values between 0 and 1.

Biologically plausible parameter ranges were adapted from von Dassow et al. (2000). All ranges were sampled from linearly:

k (half-maximal activation/repression coefficient): 0 – 1

n (Hill/cooperation coefficient): 1 – 10

H (half life, in minutes): 5 – 100

Each transcriptional regulatory edge between a protein and an mRNA node had its own n and k parameter, with a value independent of other n and k parameters. For example, for an RL2 module with gene A regulating gene B and vice versa, the parameters would be:

For $d(\text{mA})/dt$: n_{pB_mA} , k_{pB_mA} , H_{mA}

For $d(\text{pA})/dt$: H_{pA}

For $d(\text{mB})/dt$: n_{pA_mB} , k_{pA_mB} , H_{mB}

For $d(\text{pB})/dt$: H_{pB}

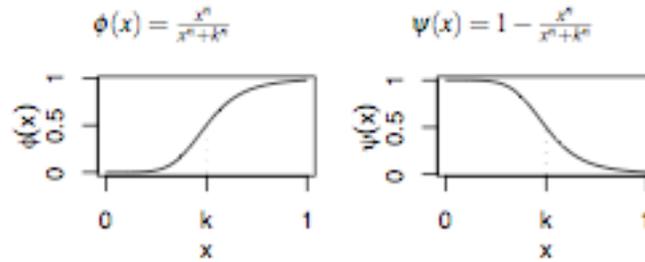
Transcriptional regulatory edges between modules inherited parameters from within modules. For example, if protein pB also regulated a gene in a neighboring module, its parameters for that regulatory relationship would be the same as those for its target gene within its own module. However, these inherited parameters were halved and doubled independently of the within-module parameters for local robustness tests.

We chose initial concentration points to cover phase space evenly, via the Sobol quasirandom number generator algorithm (Press 1992), as implemented in the qrng module for pygsl (Gädke et al. 2007). We also used this algorithm to sample the 250 points in parameter space.

Mathematical representation:

These equations are adapted from the "building-block" equation guide in the appendix of Meir et al. (2002).

Building blocks



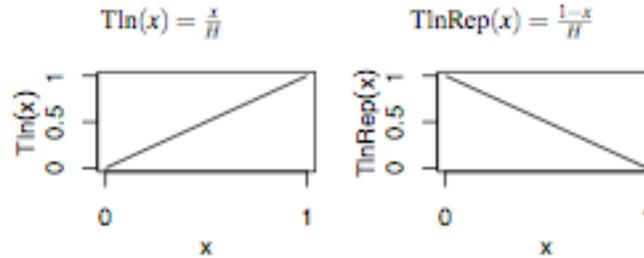
$\phi(x)$ is the standard Hill function.

Transcriptional activation & repression terms

$$\text{TxnAct}(a) = \frac{\phi(a)}{H} \quad \text{TxnRep}(r) = \frac{\psi(r)}{H}$$

H is the half-life parameter.

Translation & translational repression terms



Two transcriptional activators co-regulating a gene

$$\text{TxnAct.Act}(a_1, a_2) = \frac{1 - \psi(a_1)\psi(a_2)}{H}$$

Two transcriptional repressors co-regulating a gene

$$\text{TxnRep.Rep}(r_1, r_2) = \frac{\psi(r_1)\psi(r_2)}{H}$$

Transcription of a gene with a within-loop activator (a) and an external repressor (r)

$$\text{TxnAct_Rep}(a, r) = \frac{\phi(a)\psi(r)}{1 - \phi(a)\psi(r)} \left(\frac{1}{H} \right)$$

Transcription of a gene with a within-loop repressor (r) and an external activator (a)

$$\text{TxnRep_Act}(a, r) = \left[1 - \frac{(r\psi(a))^{n_r}}{(r\psi(a))^{n_r} + k_r^{n_r}} \right] \left(\frac{1}{H} \right)$$

Using these terms to make loop modules

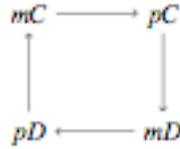
AL1



$$\frac{d}{dt} mA = \text{TxnAct}(pA) - mA \left(\frac{1}{H_{mA}} \right)$$

$$\frac{d}{dt} pA = \text{Tln}(mA) - pA \left(\frac{1}{H_{pA}} \right)$$

AL2



$$\frac{d}{dt} mC = \text{TxnAct}(pD) - mC \left(\frac{1}{H_{mC}} \right)$$

$$\frac{d}{dt} pC = \text{Tln}(mC) - pC \left(\frac{1}{H_{pC}} \right)$$

$$\frac{d}{dt} mD = \text{TxnAct}(pC) - mD \left(\frac{1}{H_{mD}} \right)$$

$$\frac{d}{dt} pD = \text{Tln}(mD) - pD \left(\frac{1}{H_{pD}} \right)$$

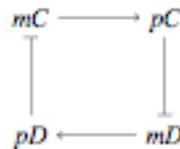
RL1



$$\frac{d}{dt} mA = \text{TxnRep}(pA) - mA \left(\frac{1}{H_{mA}} \right)$$

$$\frac{d}{dt} pA = \text{TlnRep}(mA) - pA \left(\frac{1}{H_{pA}} \right)$$

RL2



$$\frac{d}{dt} mC = \text{TxnRep}(pD) - mC \left(\frac{1}{H_{mC}} \right)$$

$$\frac{d}{dt} pC = \text{Tln}(mC) - pC \left(\frac{1}{H_{pC}} \right)$$

$$\frac{d}{dt} mD = \text{TxnRep}(pC) - mD \left(\frac{1}{H_{mD}} \right)$$

$$\frac{d}{dt} pD = \text{Tln}(mD) - pD \left(\frac{1}{H_{pD}} \right)$$

Examples of external regulation (e.g., from another module in the network); used for two-module networks and static-signal modules:

AI.1, with external regulation

$$X \longrightarrow mA \begin{array}{c} \curvearrowright \\ \curvearrowleft \end{array} pA \quad \frac{d}{dt}mA = \text{TxnAct-Act}(pA, X) - mA \left(\frac{1}{H_{mA}} \right)$$

$$X \longrightarrow \mid mA \begin{array}{c} \curvearrowright \\ \curvearrowleft \end{array} pA \quad \frac{d}{dt}mA = \text{TxnAct-Rep}(pA, X) - mA \left(\frac{1}{H_{mA}} \right)$$

RI.1, with external regulation

$$X \longrightarrow mA \begin{array}{c} \curvearrowright \\ \curvearrowleft \end{array} pA \quad \frac{d}{dt}mA = \text{TxnRep-Act}(pA, X) - mA \left(\frac{1}{H_{mA}} \right)$$

$$X \longrightarrow \mid mA \begin{array}{c} \curvearrowright \\ \curvearrowleft \end{array} pA \quad \frac{d}{dt}mA = \text{TxnRep-Rep}(pA, X) - mA \left(\frac{1}{H_{mA}} \right)$$

A1.3. Robustness metrics

Global robustness

We explored bifurcations in parameter space by sampling trajectories to delineate the basins of attraction. We decided on this approach rather than a formal bifurcation analysis for two primary reasons. First, our goal was to survey all of parameter space, rather than the neighborhood of a particular parameter set (as we might have if we had been investigating a particular biological network and had fitted our equations to experimental data). A traditional method of bifurcation analysis is to hold all parameters constant except one, and calculate the bifurcations that occur as that parameter is varied. Given our global approach, there was no natural point at which to hold all n-1 parameters while we changed one; we would have needed to make these “slices” throughout all of parameter space. With some of the two-module

networks reaching 20 parameters, this was not practical. Instead, we selected 250 points in parameter space for each network and determined the attractors there.

Second, many gene networks formally had more stable states than would be practically accessible to a cell possessing such a network. For example, all AL1 modules have a stable node at $m_A=0, p_A=0$ because of the decay terms in the equations. However, for many parameter sets, the basin of attraction of this node is so tiny (e.g., a separatrix less than $1e-3$ away from the node at 0,0) that that part of phase space would never be explored by a biological system. We elected to ignore attractors like this as biologically meaningless, if mathematically valid. We considered a module “monostable” if all 1000 trajectories settled on the same steady state, even if another attractor formally existed in the system, because such a gene regulatory network would act monostable in a real cell. “Bistable” modules were those where at least one (usually more) of the 1000 trajectories settled on a second attractor.

The steady states reached by the different parameter sets all tend to concentrate in the corners one would expect for the network in question, and we labeled them as such. For example, the steady states for a two-module network containing an AL1 module and an RL2 module tend to be near one of these four corners of phase space, where m_A/p_A are the nodes in AL1 and $m_C/p_C/m_D/p_D$ are the nodes in RL2:

- 1) $m_A=0, p_A=0; m_C=0, p_C=0, m_D=1, p_D=1$ (00; 0011)
- 2) $m_A=0, p_A=0; m_C=1, p_C=1, m_D=0, p_D=0$ (00; 1100)
- 3) $m_A=1, p_A=1; m_C=0, p_C=0, m_D=1, p_D=1$ (11; 0011)
- 4) $m_A=1, p_A=1; m_C=1, p_C=1, m_D=0, p_D=0$ (11; 1100)

We sorted the steady states found for each parameter set into the corners of phase space, and labeled them by the corner they were closest to.

To measure *global robustness* for each network, we used the Sobol algorithm mentioned above to generate 250 points evenly distributed across parameter space.

For each parameter set, we did the following:

For each of 1000 initial concentrations (IC) of the network components, we integrated the equations to find the steady state each initial concentration settled at. We labeled each IC by the steady state it reached.

Thus, for each parameter set, we had a list of 1000 steady-state corner labels. The set of ICs with the label for a particular steady state is the basin of attraction for that steady state.

We compared the basins across parameter sets, and measured how much they varied. For each steady state with label L , we examined each unique pair of parameter sets and calculated the proportion of parameter set i 's basin for L that was also L 's basin for parameter set j ; and the proportion of parameter set j 's basin for L that was also L 's basin for parameter set i (see eq. 4, main text). We took the mean of these values over all unique pairs of parameter sets to find the proportion of L 's basin that matched across the average pair of parameter sets (eq. 5, main text). This produced a global robustness score for steady state L : the average proportion of L 's basins that were shared across parameter sets.

We summed the scores for all steady states, weighting each one by its average basin size (eq. 6, main text). This produced an overall global robustness metric for the network in question. Global robustness scores varied between 0 (no consistency of basins among parameter sets) and 1 (every parameter set has the same basins for each steady state).

Local robustness

Global robustness is a measure of how consistent basins of attraction are across all of parameter space. We also developed a metric to mimic more closely the mutational process. To calculate this *local robustness*, we began with a single parameter set, and then doubled and halved each parameter value in turn. The calculation is similar to that for global robustness, but instead of comparing basins across all pairs of parameter sets, we compared the basins resulting from each parameter tweak to those of the original, un-tweaked parameter set.

Each network has one measurement of global robustness, across all 250 parameter sets. By contrast, each network has multiple measurements of local robustness; there is one local robustness measurement for each parameter set and its tweaks. We measured local robustness for 250 parameter sets for each isolated module and static-signal network (2-4 nodes each). Because of computational limitations, we measured local robustness for 16 of the larger two-module networks (4-8 nodes each), in the following way. For each isolated module (AL1, RL1, AL2, RL2) we selected four parameter sets to illustrate the range of behavior it could exhibit. Each representative parameter set was bistable, to more easily allow the two-module network containing the modules to range in any direction. Each two-module network was assembled with all combinations of the representative parameter sets for its two component modules, yielding 16 parameter sets for each two-module network. These were used as the original parameter values that could be tweaked to generate a measurement of local robustness. To avoid pseudoreplication from the four representative parameter sets for each module, each of which appears in four of the 16 parameter sets for the two-module networks, the local robustness reported for each two-module network is the mean measurement across its 16 representative parameter sets.

IC-centered robustness

An alternative way of measuring robustness, which avoids the possible artifacts involved in grouping steady states by their closest corner, would have been to calculate it for each IC rather than for each steady state. This “IC-centered robustness” would measure, for each IC, how much the position of its steady-state point varied as parameters varied. We could then take the mean variation over all ICs as a measure of overall robustness. This method is in some ways more intuitive, and we did calculate IC-centered robustness, and found a strong correlation between it and the “steady-state-centered robustness” described above (data not shown). However, IC-centered robustness has a subtle drawback for the two-module networks. The interactions between the two modules of the two-module networks tend to discourage some of the steady states that would be represented in the modules if they were isolated. For example, in the AL1/RL2 pair above, we had four possible steady states. If the two modules were connected in the two-module network AL1actRL2act, we would have the between-module regulatory edges:

pA --> mC

pD --> mA

The first edge will tend to discourage the steady state where pA=1 and mC=0, because when pA is at high concentration it will raise the expression of mC. Thus, we lose steady state 3 in the list above. The second edge will discourage pD=1, mA=0 steady state (number 1 in the list). In practice, most trajectories in AL1actRL2act will occupy steady states 2 or 4.

Now consider AL1actRL2rep, with the edges:

pA --> mC

pD --| mA

Steady state 3 remains discouraged by the first edge, which is unchanged. The second edge discourages the steady state where $pD=1$, $mA=1$, because high pD will effectively suppress transcription of mA . Looking at the list of possible steady states, we find that the steady state “prohibited” by the second edge is also steady state 3. Both edges suppress the same steady state. Trajectories in this system will generally distribute themselves across the three remaining steady states.

As a result of this kind of effect, some two-module networks have two “encouraged” steady states and some have three. Those with three are biased toward lower IC-centered robustness, because each IC has three choices of destination instead of two. This almost invariably increases the average distance between the steady states that an IC selects as parameters vary, and consequently lowers its robustness. The “steady-state centered” robustness metric we used instead, for both local and global robustness, avoids this bias by measuring the robustness of each steady state, and thereby controlling for the number of steady states present in the network.

Oscillatory exceptions

The modules were chosen in part because their fixed points were stable nodes, and for most of the two-module networks (and all of the isolated and static-input modules), we did not observe periodic behavior for any parameter values. For a few of the two-module networks, we found damped oscillations indicating a bifurcation from a stable node to a stable spiral (damped oscillator, settling to a steady state). We did encounter stable oscillations for 2 out of 640 simulated mutation trials for `RL2actRL2rep1`, 1 out of 384 for `RL1actRL1rep`, and 2 out of 512 for `AL1actRL2rep1`. Since this represented such a tiny part of our data (less than 0.4% of the trials for each), we chose to restrict the extent of the simulated mutations that triggered the stable oscillations in those isolated cases. Instead of doubling or halving

the troublesome parameters in local robustness trials, we increased or decreased them to a point just short of that required to produce oscillations.

A1.4. Static-signal networks

These networks consisted of a single isolated module (AL1, AL2, RL1, or RL2) with one external regulator (“X”) that modified transcription of mA (AL1, RL1) or mC (AL2, RL2). X acted like a protein node from another module, using the same four terms as in the two-module network: TxnAct_Act, TxnRep_Rep, TxnAct_Rep, and TxnRep_Act. The difference was that X’s value, or “concentration,” was held constant, allowing us to observe the steady-state behavior of a single module while it was being regulated externally, without the confound of the regulator’s concentration changing with time. We varied X from 0 to 1, the same range as exhibited by the other variables in our model.

The external regulator “encourages” one steady state and “discourages” another. For actAL1, X helps activate the transcription of mA, so it pushes trajectories toward the steady state mA=1, pA=1 (11) and away from 00. Hence, the proportion of ICs leading to the encouraged steady-state of 11 tends to grow, and the basin of attraction for 00 tends to shrink. The opposite is true if X suppresses mA’s transcription (actAL1). The effects of the X regulator are:

actAL1 (actAL2): 11 (1111) encouraged, 00 (0000) discouraged
repAL1 (repAL2): 00 (0000) encouraged, 11 (1111) discouraged
actRL1 (actRL2): 10 (1100) encouraged, 01 (0011) discouraged
repRL1 (repRL2): 01 (0011) encouraged, 10 (1100) discouraged

A1.5. Statistical comparison of module types

Our set of two-module networks represented the entire population of networks that could be assembled out of the modules we used as building blocks (conforming to the rules we used for combining them: maximum of 1 edge from each module to each other module; transcriptional inter-module regulation only; AL1 modules paired with AL1 or RL2, and AL2 paired with AL2 or RL1 for balance). Because our measurements consisted of the entire population rather than a subsample thereof, we applied bootstrap techniques to the data. For metrics where we had a single value for each network (for example, the global robustness of a network over 250 parameter sets, or the mean local robustness measurement of a two-module network across its representative parameter sets), we proceeded as follows.

Each AL/AL network had a corresponding RL/RL network with the same edges between modules but the modules themselves replaced by RL interactions (the pairs of networks listed in section A1.1). These complementary topologies allowed paired comparisons. For each AL/AL and RL/RL pair, we calculated the difference in the metric under study between the two networks. A difference of 0 would have meant that the two networks scored the same with respect to the metric (e.g., global robustness). We sampled with replacement from the list of differences 10,000 times, calculating the mean of the sample each time. To measure the likelihood of the value 0 being drawn from the resulting distribution of means, we calculated the two-tailed p-value as follows:

$$\text{right-tailed p-value} = (1 + S (t \geq t^*)) / (N + 1)$$

$$\text{left-tailed p-value} = (1 + S (t \leq t^*)) / (N + 1)$$

$$\text{2-tailed p-value} = 2 (\min (\text{right-tailed p-value}, \text{left-tailed p-value}))$$

where N is the number of samplings with replacement (10,000), t is the list of differences, and t^* is the value whose position with respect to the distribution we are measuring (0). With 10,000 sampling iterations, the minimum p-value is $2.00e-4$.

The mixed-module networks, AL1/RL2 and AL2/RL1, did not allow for natural pairwise comparisons with the other network types. To compare AL/AL or RL/RL networks to mixed-module networks, we used an unpaired permutation test instead. Here, we simply make a list of the metric values for each network, labeling each with its network class (AL/AL vs. AL/RL, for example). We permute the labels, take the mean of each class with the permuted labels, and find the difference between the means. The resulting set of values gives us the distribution we would expect if there were no significant difference between the two classes of networks. As with the paired test, we compare the critical value – here, the difference between the means of the correctly labeled network classes – to the distribution to calculate a two-tailed p-value.

The approach above was employed for two-module networks, where we had a single value for each network. Global robustness tests produce a single value across all parameter sets. Local robustness tests produce one value per parameter set (the robustness as those parameters are halved and doubled), but since the parameter sets used for local robustness testing for two-module networks consisted of all possible combinations of 4 representative parameter sets for each module (see A1.3, “Local robustness”), we used the mean local robustness score across all combinations of representative parameter sets to avoid pseudoreplication. Thus, each metric for two-module networks yielded a single value for each network.

When it is appropriate to use multiple measurements for each network, we have a slightly different problem with the same solution. Because of the smaller size of single modules and static-signal molecules, it was tractable to expand our local

robustness testing to 250 parameter sets distributed across parameter space, instead of the combinations of representative parameter sets used for two-module networks. The measurements at various parameter sets were much more likely to be independent in this context than with the two-module networks, but incomplete independence was still a concern. Also, the sample size of 250 points was somewhat arbitrary, making it inappropriate to use traditional statistical methods of estimating the power of our tests – we could simply have sampled twice as many points in parameter space, and lowered our p-values as a result. The standard way around this difficulty, which arises often in research involving simulations, is permutation testing, as above. To compare the measurements for two types of networks (e.g., the isolated AL1 module vs. the RL1 module), we used unpaired bootstrap tests to permute the network labels of the measurements, and compared the true difference between the two sets of measurements with the permuted distribution.

There are 4 types of isolated modules (AL1, RL1, AL2, RL2) and 8 types of static-signal molecules (actAL1, repAL1, actRL1, repRL1, etc.). Since global robustness would yield only a single measurement for each module, trends (or lack thereof) would not be apparent when we used this metric, because of the small number of data points. Since global and local robustness had similar patterns for two-module networks, we confined our comparisons between AL and RL single and static-signal modules to local robustness measurements only. We have 250 local robustness measurements for each isolated and static-signal module, enabling a robust comparison of module types. We compared the distribution of local robustness measurements for each sample type, using the permutation testing method described above. The results, as reported in the main text, were: AL1 (mean local robustness = 0.886) vs. RL1 (mean = 0.882): $p = 0.62$. AL2 (mean = 0.913) vs. RL2 (mean = 0.907): $p = 0.45$. Results for static signal networks are in table A1.1, below.

Table A1.1. Local robustness of static-signal networks to internal parameter perturbation. The concentration of the external signal (“X”) was kept constant at 1.0.

| Network name | Mean local robustness (250 parameter sets) | Local robustness: 2-tailed p-value of bootstrap comparison |
|--------------|--|--|
| actAL1 | 0.829 | 2.00e-4 |
| actRL1 | 0.997 | |
| repAL1 | 0.921 | 2.00e-4 |
| repRL1 | 1.00 | |
| actAL2 | 0.900 | 2.00e-4 |
| actRL2 | 0.995 | |
| repAL2 | 0.935 | 2.00e-4 |
| repRL2 | 0.999 | |

A1.6. Scoring ideal pluripotency

The ideal cell-fate decision mechanism would assume a monostable (and thus noise-intolerant) condition for one steady state when stimulated in that direction by an outside signal, and a monostable condition for the other steady state when pushed in that direction as well. It would also be perfectly robust to simulated mutations in both states. To test whether RL modules did a better job of approximating this “special category” of regulatory motifs, we tested each static-signal network’s score on four metrics simultaneously, for each of 250 parameter sets. Each metric varies between 0 and 1, and X has the concentration of 1.0 for these tests.

- Proportion of ICs that lead to the steady state encouraged by the external regulator X when X is an *activator*. This value is 0 if all trajectories lead to other steady states, and 1 if the network is monostable for the X-encouraged steady state. It has an intermediate value if the network is bistable

- Proportion of ICs that lead to the steady state encouraged when X is a *repressor*.
- Local robustness to simulated mutation when X is an *activator*. Robustness measurements also vary between 0 and 1.
- Local robustness to simulated mutation when X is a *repressor*.

To collapse the four metrics into one value that we could use to score each network for each parameter set, we calculated the Euclidean distance between each parameter set's values for the four measurements and 0,0,0,0 (a score of 0 in each category). A module that was perfectly monostable and perfectly locally robust under activation and repression would have the maximum score of 2.00 ($\sqrt{1+1+1+1}$), and be the closest approximation of the ideal cell differentiation network.

Having condensed the metrics into a single score measuring how good an approximation of the ideal each network was, we were able to test whether RL modules' scores were drawn from a different distribution than that of AL modules' scores. We used the unpaired permutation test described above. Results are below. RL modules were dramatically closer to the ideal score, for both medium ($X = 0.5$) and strong ($X = 1.0$) external signals.

Our results for static-signal modules suggest that larger networks with mutual repression modules in them will be better suited for developmental differentiation tasks than similar networks containing mutual activation modules.

Table A1.2. Ideal pluripotency scores for static-signal modules with X held constant at 0.5.

| Module type | AL1 | RL1 | AL2 | RL2 |
|---|-----------|------|-----------|------|
| Mean local robustness with external activation | .833 | .937 | .899 | .922 |
| Mean local robustness with external repression | .892 | .979 | .918 | .928 |
| Mean size of “encouraged” basin with external activation | .598 | .758 | .605 | .733 |
| Mean size of “encouraged” basin with external repression | .600 | .981 | .543 | .707 |
| Mean overall score (max=2.0) | 1.6 | 1.88 | 1.64 | 1.76 |
| 2-tailed p-value of bootstrap comparison | < 2.00e-4 | | < 2.00e-4 | |

Table A1.3. Ideal pluripotency scores for static-signal modules with X held constant at 1.0.

| Module type | AL1 | RL1 | AL2 | RL2 |
|---|-----------|------|-----------|------|
| Mean local robustness with external activation | .829 | .997 | .901 | .995 |
| Mean local robustness with external repression | .921 | 1.00 | .935 | .999 |
| Mean size of “encouraged” basin with external activation | .658 | .986 | .631 | .989 |
| Mean size of “encouraged” basin with external repression | .785 | 1.00 | .731 | .992 |
| Mean overall score (max=2.0) | 1.71 | 1.99 | 1.73 | 1.99 |
| 2-tailed p-value of bootstrap comparison | < 2.00e-4 | | < 2.00e-4 | |

APPENDIX REFERENCES

- Gädke, A, Küpper, J, Schnizer, P. 2007. pygsl: Python interface for the GNU scientific library. <http://sourceforge.net/projects/pygsl/>
- Meir, E, Munro, EM, Odell, GM, Von Dassow, G. 2002. Ingeneue: a versatile tool for reconstituting genetic networks, with examples from the segment polarity network. *J Exp Zool* 294:216–251.
- Press, WH. 1992. *Numerical recipes in C : the art of scientific computing*. Cambridge, New York: Cambridge University Press. p. xxvi, 994.
- von Dassow, G, Meir, E, Munro, EM, Odell, GM. 2000. The segment polarity network is a robust developmental module. *Nature* 406:188–192.

REFERENCES

- Alon, U. 2003. Biological networks: the tinkerer as an engineer. *Science* 301:1866–1867.
- Alon, U. 2007. Network motifs: theory and experimental approaches. *Nat Rev Genet* 8:450–461.
- Atkinson, MR, Savageau, MA, Myers, JT, Ninfa, AJ. 2003. Development of genetic circuitry exhibiting toggle switch or oscillatory behavior in *Escherichia coli*. *Cell* 113:597–607.
- Babu, MM, Teichmann, SA. 2003. Evolution of transcription factors and the gene regulatory network in *Escherichia coli*. *Nucleic Acids Res* 31:1234–1244.
- Balaban, NQ, Merrin, J, Chait, R, Kowalik, L, Leibler, S. 2004. Bacterial persistence as a phenotypic switch. *Science* 305:1622–1625.
- Becskei, A, Seraphin, B, Serrano, L. 2001. Positive feedback in eukaryotic gene networks: cell differentiation by graded to binary response conversion. *EMBO J* 20:2528–2535.
- Bhalla, US, Iyengar, R. 1999. Emergent properties of networks of biological signaling pathways. *Science* 283:381–387.
- Brandman, O, Ferrell, JEJ, Li, R, Meyer, T. 2005. Interlinked fast and slow positive feedback loops drive reliable cell decisions. *Science* 310:496–498.
- Chang, HH, Hemberg, M, Barahona, M, Ingber, DE, Huang, S. 2008. Transcriptome-wide noise controls lineage choice in mammalian progenitor cells. *Nature* 453:544–547.
- Clyde, DE, Corado, MS, Wu, X, Pare, A, Papatsenko, D, Small, S. 2003. A self-organizing system of repressor gradients establishes segmental complexity in *Drosophila*. *Nature* 426:849–853.

- Csete, ME, Doyle, JC. 2002. Reverse engineering of biological complexity. *Science* 295:1664–1669.
- Davidson, EH, McClay, DR, Hood, L. 2003. Regulatory gene networks and the properties of the developmental process. *Proc Natl Acad Sci USA* 100:1475–1480.
- Davidson, EH, Rast, JP, Oliveri, P, Ransick, A, Calestani, C, Yuh, CH, Minokawa, T, Amore, G, Hinman, V, Arenas-Mena, C, Otim, O, Brown, CT, Livi, CB, Lee, PY, Revilla, R, Rust, AG, Pan, Z, Schilstra, MJ, Clarke, PJ, Arnone, MI, Rowen, L, Cameron, RA, McClay, DR, Hood, L, Bolouri, H. 2002. A genomic regulatory network for development. *Science* 295:1669–1678.
- de Visser, JA, Hermisson, J, Wagner, GP, Ancel Meyers, L, Bagheri-Chaichian, H, Blanchard, JL, Chao, L, Cheverud, JM, Elena, SF, Fontana, W, Gibson, G, Hansen, TF, Krakauer, D, Lewontin, RC, Ofria, C, Rice, SH, von Dassow, G, Wagner, A, Whitlock, MC. 2003. Perspective: Evolution and detection of genetic robustness. *Evolution Int J Org Evolution* 57:1959–1972.
- Eldon, ED, Pirrotta, V. 1991. Interactions of the *Drosophila* gap gene giant with maternal and zygotic pattern-forming genes. *Development* 111:367–378.
- Elowitz, MB, Levine, AJ, Siggia, ED, Swain, PS. 2002. Stochastic gene expression in a single cell. *Science* 297(5584):1183–1186.
- Ferrell, JE, Xiong, W. 2001. Bistability in cell signaling: How to make continuous processes discontinuous, and reversible processes irreversible. *Chaos* 11:227–236.
- Ferrell, JEJ. 2002. Self-perpetuating states in signal transduction: positive feedback, double-negative feedback and bistability. *Curr Opin Cell Biol* 14(2):140–148.
- Gama-Castro, S, Jimenez-Jacinto, V, Peralta-Gil, M, Santos-Zavaleta, A, Penaloza-Spinola, MI, Contreras-Moreira, B, Segura-Salazar, J, Muniz-Rascado, L,

- Martinez-Flores, I, Salgado, H, Bonavides-Martinez, C, Abreu-Goodger, C, Rodriguez-Penagos, C, Miranda-Rios, J, Morett, E, Merino, E, Huerta, AM, Trevino-Quintanilla, L, Collado-Vides, J. 2008. RegulonDB (version 6.0): gene regulation model of Escherichia coli K-12 beyond transcription, active (experimental) annotated promoters and Textpresso navigation. *Nucleic Acids Res* 36:D120–4.
- Gardner, TS, Cantor, CR, Collins, JJ. 2000. Construction of a genetic toggle switch in Escherichia coli. *Nature* 403(6767):339–342.
- Gerhart, J, Kirschner, M. 1997. Cells, embryos, and evolution: Toward a cellular and developmental understanding of phenotypic variation and evolutionary adaptability. Malden, Mass.: Blackwell Science. p. xiii, 642.
- Gutenkunst, RN, Atlas, JC, Casey, FP, Kuczynski, RS, Waterfall, JJ, Myers, CR, Sethna, JP. 2007. SloppyCell. <http://sloppycell.sourceforge.net>
- Hafner, M, Koepl, H, Hasler, M, Wagner, A. 2009. 'Glocal' robustness analysis and model discrimination for circadian oscillators. *PLoS Comput Biol* 5:e1000534.
- Hartwell, LH, Hopfield, JJ, Leibler, S, Murray, AW. 1999. From molecular to modular cell biology. *Nature* 402:C47–52.
- Hasty, J, Pradines, J, Dolnik, M, Collins, JJ. 2000. Noise-based switches and amplifiers for gene expression. *Proc Natl Acad Sci U S A* 97:2075–2080.
- Hinman, VF, Davidson, EH. 2007. Evolutionary plasticity of developmental gene regulatory network architecture. *Proc Natl Acad Sci U S A* 104:19404–19409.
- Ihmels, J, Friedlander, G, Bergmann, S, Sarig, O, Ziv, Y, Barkai, N. 2002. Revealing modular organization in the yeast transcriptional network. *Nat Genet* 31:370–377.
- Ingolia, NT, Murray, AW. 2007. Positive-feedback loops as a flexible biological module. *Curr Biol* 17:668–677.

- Isaacs, FJ, Hasty, J, Cantor, CR, Collins, JJ. 2003. Prediction and measurement of an autoregulatory genetic module. *Proc Natl Acad Sci USA* 100:7714–7719.
- Kaern, M, Elston, TC, Blake, WJ, Collins, JJ. 2005. Stochasticity in gene expression: from theories to phenotypes. *Nat Rev Genet* 6:451–464.
- Kashtan, N, Alon, U. 2005. Spontaneous evolution of modularity and network motifs. *Proc Natl Acad Sci U S A* 102:13773–13778.
- Kraut, R, Levine, M. 1991. Mutually repressive interactions between the gap genes giant and Kruppel define middle body regions of the *Drosophila* embryo. *Development* 111:611–621.
- Kussell, E, Kishony, R, Balaban, NQ, Leibler, S. 2005. Bacterial persistence: a model of survival in changing environments. *Genetics* 169:1807–1814.
- Kussell, E, Leibler, S. 2005. Phenotypic diversity, population growth, and information in fluctuating environments. *Science* 309:2075–2078.
- Legewie, S, Bluthgen, N, Herzog, H. 2006. Mathematical modeling identifies inhibitors of apoptosis as mediators of positive feedback and bistability. *PLoS Comput Biol* 2:e120.
- Lipson, H, Pollack, JB, Suh, NP. 2002. On the origin of modular variation. *Evolution Int J Org Evolution* 56:1549–1556.
- Manu, Surkova, S, Spirov, AV, Gursky, VV, Janssens, H, Kim, A-RK, Radulescu, O, Vanario-Alonso, CE, Sharp, DH, Samsonova, M, Reinitz, J. Canalization of Gene Expression in the *Drosophila* Blastoderm by Gap Gene Cross Regulation. *PLoS Biology* 7:e49.
- McAdams, HH, Arkin, A. 1999. It's a noisy business! Genetic regulation at the nanomolar scale. *Trends Genet* 15:65–69.

- Meir, E, Munro, EM, Odell, GM, Von Dassow, G. 2002. Ingeneue: a versatile tool for reconstituting genetic networks, with examples from the segment polarity network. *J Exp Zool* 294:216–251.
- Myers, CA, Gutenkunst, RN, Sethna, JP. 2007. Python Unleashed on Systems Biology. *Computing in Science & Engineering* 9:34–37.
- Niessing, D, Rivera-Pomar, R, La Rosee, A, Hader, T, Schock, F, Purnell, BA, Jackle, H. 1997. A cascade of transcriptional control leading to axis determination in *Drosophila*. *J Cell Physiol* 173:162–167.
- Ozbudak, EM, Thattai, M, Kurtser, I, Grossman, AD, van Oudenaarden, A. 2002. Regulation of noise in the expression of a single gene. *Nat Genet* 31(1):69–73.
- Ozbudak, EM, Thattai, M, Lim, HN, Shraiman, BI, Van Oudenaarden, A. 2004. Multistability in the lactose utilization network of *Escherichia coli*. *Nature* 427:737–740.
- Pankratz, MJ, Jackle, H. 1993. Blastoderm segmentation. In: Bate, MA, Martinez Arias, A, editors. *The Development of Drosophila melanogaster*. Cold Spring Harbor, NY: Cold Spring Harbor Laboratory Press. p. 467–516.
- Pomerening, JR, Sontag, ED, Ferrell, JEJ. 2003. Building a cell cycle oscillator: hysteresis and bistability in the activation of Cdc2. *Nat Cell Biol* 5:346–351.
- Qi, Y, Ge, H. 2006. Modularity and dynamics of cellular networks. *PLoS Comput Biol* 2:e174.
- Raff, RA. 2000. Evo-devo: the evolution of a new discipline. *Nat Rev Genet* 1:74–79.
- Rao, CV, Wolf, DM, Arkin, AP. 2002. Control, exploitation and tolerance of intracellular noise. *Nature* 420(6912):231–237.
- Raser, JM, O'Shea, EK. 2005. Noise in gene expression: origins, consequences, and control. *Science* 309:2010–2013.

- Savageau, MA. 1976. Biochemical systems analysis: A study of function and design in molecular biology. Reading, Mass: Addison-Wesley Pub. Co., Advanced Book Program. p. xvii, 379 p.
- Schlosser, G, Wagner, GP. 2004. Modularity in development and evolution. Chicago: University of Chicago Press. p. x, 600 p.
- Schroeder, MD, Pearce, M, Fak, J, Fan, H, Unnerstall, U, Emberly, E, Rajewsky, N, Siggia, ED, Gaul, U. 2004. Transcriptional control in the segmentation gene network of *Drosophila*. PLoS Biol 2:E271.
- Segal, E, Shapira, M, Regev, A, Pe'er, D, Botstein, D, Koller, D, Friedman, N. 2003. Module networks: identifying regulatory modules and their condition-specific regulators from gene expression data. Nat Genet 34:166–176.
- Singh, AH, Wolf, DM, Wang, P, Arkin, AP. 2008. Modularity of stress response evolution. Proc Natl Acad Sci U S A 105:7500–7505.
- Slonim, N, Elemento, O, Tavazoie, S. 2006. Ab initio genotype-phenotype association reveals intrinsic modularity in genetic networks. Mol Syst Biol 2:2006.0005.
- Thattai, M, van Oudenaarden, A. 2001. Intrinsic noise in gene regulatory networks. Proc Natl Acad Sci USA 98:8614–8619.
- von Dassow, G, Meir, E, Munro, EM, Odell, GM. 2000. The segment polarity network is a robust developmental module. Nature 406:188–192.
- von Dassow, G, Munro, E. 1999. Modularity in animal development and evolution: elements of a conceptual framework for EvoDevo. J Exp Zool 285:307–325.
- Von Dassow, G, Odell, GM. 2002. Design and constraints of the *Drosophila* segment polarity module: robust spatial patterning emerges from intertwined cell state switches. J Exp Zool 294:179–215.
- Wagner, A. 2002. Estimating coarse gene network structure from large-scale gene perturbation data. Genome Res 12:309–315.

- Wagner, A. 2005. Robustness and evolvability in living systems. Princeton, N.J: Princeton University Press. p. xii, 367 p.
- Wagner, GP, Booth, G, Bagheri-Chaichian, H. 1997. A population genetic theory of canalization. *Evolution* 51(2):329–347.
- Wall, ME, Hlavacek, WS, Savageau, MA. 2003. Design principles for regulator gene expression in a repressible gene circuit. *J Mol Biol* 332:861–876.
- Wall, ME, Hlavacek, WS, Savageau, MA. 2004. Design of gene circuits: lessons from bacteria. *Nat Rev Genet* 5:34–42.
- Wilkinson, RJ, DesJardin, LE, Islam, N, Gibson, BM, Kanost, RA, Wilkinson, KA, Poelman, D, Eisenach, KD, Toossi, Z. 2001. An increase in expression of a *Mycobacterium tuberculosis* mycolyl transferase gene (*fbpB*) occurs early after infection of human monocytes. *Mol Microbiol* 39:813–821.
- Wingender, E. 2008. The TRANSFAC project as an example of framework technology that supports the analysis of genomic regulation. *Brief Bioinform* 9:326–332.
- Winther, RG. 2001. Varieties of modules: kinds, levels, origins, and behaviors. *J Exp Zool* 291:116–129.
- Wolf, DM, Arkin, AP. 2003. Motifs, modules and games in bacteria. *Curr Opin Microbiol* 6:125–134.
- Wright, S. 1934. Physiological and evolutionary theories of dominance. *The American Naturalist* 68:24–53.
- Xiong, W, Ferrell, JEJ. 2003. A positive-feedback-based bistable 'memory module' that governs a cell fate decision. *Nature* 426:460–465.
- Zahrt, TC. 2003. Molecular mechanisms regulating persistent *Mycobacterium tuberculosis* infection. *Microbes Infect* 5:159–167.

CHAPTER 2

BOOLEAN MODELS OF MODULAR NETWORKS

Abstract

How simple can a gene network model be without sacrificing the essential dynamics of the system? Real genetic networks can often withstand noise and parameter variation with their dynamics qualitatively intact, effectively “ignoring” minor tweaks to the system. How much detail, then, can we ignore in our models and still accurately capture their behavior? These are pressing questions if we hope to construct genomic models of interacting genes and proteins, because tractability at such large scales demands that we use simple rules to describe how the components interact. To help guide modeling choices for gene regulatory networks, we created detailed differential equation (DE) models of common gene network motifs – small feedback loops – whose switch-like behavior might be expected to accommodate a Boolean representation. We developed an algorithm for finding the best Boolean approximation of each DE network module, and of larger networks assembled from combinations of these modules. We created three quantitative metrics for determining how closely the Boolean model approximated the more detailed DE dynamics. We used these methods to determine which topologies worked best under a Boolean model, and which were more susceptible to the artifacts introduced by Boolean simplifications. These results can help guide the simplifications necessary for constructing models of extended genetic networks.

Introduction

The genome of every organism defines a network of interacting genes. The genes and their protein products negotiate among themselves to determine which will be expressed and which silenced, in each tissue and at each moment in time. The sum of these negotiations is the phenotype of the organism.

Interpreting those conversations among the genes, and thereby the map between genotype and phenotype, is a central challenge of modern biology. It is not sufficient to know the identity of all the participants, nor even the regulatory interactions among them; we must also try to determine what *behavior* a particular gene network will produce. Which set of genes will be expressed, at what levels, in response to which conditions? How will those expression patterns change with time? What steady-state expression levels are within the dynamical repertoire of each regulatory network? This question becomes formidable when the network includes feedback loops between the genes, and with the increasing complexity it soon surpasses our ability to understand intuitively. To study its behavior, we must codify the interactions within the network and model their behavior.

What kind of model should we use for this task? A perfectly detailed description of the gene network, such as a set of differential equations (DEs) with experimentally ascertained parameter values, would in some ways be ideal. It would tell us all the behavior to expect, for any of hundreds of different initial concentrations of the molecules involved, and could even help us predict how that behavior would change if mutation or environmental effects changed some of the parameters – a drop in temperature, for example, or an increase in binding affinity between a transcription factor and its DNA target. Certainly such a model would tell us how a deletion of one of the genes involved would affect the expression patterns.

However, we almost never possess that level of detail about the network. Parameters such as the binding affinity and binding cooperativity of transcriptional regulators are difficult to measure. Noise intrinsic in the system, due to small numbers of molecules (McAdams and Arkin 1999; Elowitz et al. 2002) and bursty transcription and translation (Ozbudak et al. 2002; Swain et al. 2002) complicates the interactions, and measurements of parameters have noise in them as well. Finally, for most organisms, the experiments to start the network from hundreds of initial concentrations and track its behavior to characterize the system would be prohibitive. Instead, we tend to know only the genes involved and portions of the network topology (e.g., which proteins activate and repress the synthesis of which others).

Even if we had perfect knowledge of all the parameters in the genomic regulatory network, it would be intractable to simulate the dynamics of a large network at the level of detail encapsulated in fitted DEs. Such a model would, in any case, be unverifiable outside the restricted range of parameters that have been measured experimentally. To study the behavior of genome-scale (or even simply large) gene regulatory networks, we need to employ more tractable, less detailed models.

Both the level of resolution of the data and the limitations of our computers prompt us toward a simpler model. But how simple is too simple? What level of abstraction can capture the essential dynamics while remaining tractable and relying on approximate data?

There have been some indications that a very simple kind of model, a Boolean representation, might suffice in some cases (Szallasi and Liang 1998; Smolen et al. 2000; Covert and Palsson 2002; Albert and Othmer 2003; Setty et al. 2003; Li et al. 2004; Istrail and Davidson 2005; Mayo et al. 2006; Sudarsan et al. 2006; Istrail et al. 2007; Covert et al. 2008). The idea of using Boolean models for large-scale networks

is appealing, because computer scientists have developed many tools for working with Boolean systems, and computational analysis of such networks is very fast. However, Boolean models are too simple for many kinds of gene networks. The simultaneous updating of all nodes tends to introduce timing artifacts and artificial oscillations (Glass and Kauffman 1973; Smolen et al. 2000). Two states, “on” or “off,” can be insufficient to capture important details of gene expression – for example, it would be hopelessly inadequate for modeling a morphogen gradient. More subtly, the 0/1 limitation means that a transcriptional regulator cannot be perceived as “on” by one of its targets and as “off” by a different target that has a higher threshold of activation. Boolean models are therefore only appropriate for switch-like networks with widely separated, discrete steady states.

Aware of these limitations, researchers have developed a variety of modifications to Boolean models that mitigate the artifacts they can introduce. René Thomas and colleagues (Thomas and D'Ari 1990; Sanchez et al. 1997; Thieffry and Thomas 1998; Sanchez and Thieffry 2001; Thieffry and Sanchez 2002; Sanchez and Thieffry 2003) have developed a “logical” modeling framework for genetic networks in which gene products may have one of a few integer concentration values, rather than merely 0 or 1. Necessarily, each gene also has multiple thresholds at which it can be affected by its regulators. The update rules in these models are Boolean in style. For example, in the group’s model of *Drosophila melanogaster* pair-rule genes (Sanchez and Thieffry 2003), the protein even-skipped represses *odd-skipped* if its value is greater than 1, *sloppy-paired* if it exceeds 2, and *paired* if even-skipped is higher than 3. These discrete thresholds are parameters used to fit the models to data. While variables are discrete, the time intervals between Boolean-style updates are real-valued, and this makes the cumulative model output continuous and avoids some of the oscillatory effects that can plague Boolean models.

Researchers have also introduced random asynchronous updating into more traditional, 0/1-variable Boolean approaches to model gene networks (Chaves et al. 2006; Thakar et al. 2007), and in fact asynchronous updating can fundamentally change the dynamics of such networks (Husbands and Harvey 1997). Probabilistic Boolean networks (Shmulevich et al. 2002) are another way of making Boolean models less rigid: In these, each node is subject to a number of update rules, one of which is randomly selected at each timestep. Finally, some developmental biologists have used hybrid models in which some *cis*-regulatory modules act as all-or-nothing switches and are represented via Boolean rules, while others produce a graded response in their gene targets (Yuh et al. 1998; Yuh et al. 2001). A variant of this approach is employed in McAdams and Shapiro (1995) for a prokaryotic network.

However, most of the tools from computer science for running and evaluating Boolean networks are designed for their original, simplest form. If we are to leverage the progress made in this field, or to simulate gene interaction networks on a genome-wide scale, it would be useful to discover heuristics that would help guide us in the application of Boolean models to genetic networks. There are a number of approaches that may be promising:

- 1) We can coarse-grain the network by dividing it into its component dynamic modules. This reductionist approach has the advantage of reducing the dimensionality of the system, and has shown promise as a way to begin understanding large-network behavior (Alon 2007).

- 2) We can restrict ourselves to applying Boolean methods only to topologies for which it is likely to introduce the fewest artifacts. Developmental networks in eukaryotes, and nutrient-sensing networks in prokaryotes, are replete with switch-like motifs that lock in an expression state in response to an external signal (Brandman et al. 2005; Davidson 2006; Aguda and Goryachev 2007). Positive feedback loops,

whether composed entirely of positive regulation or of an even number of repressors (Thomas and D'Ari 1990; Soulé 2003), have widely separated, non-oscillatory steady states at high or low values that can be well approximated by the 0s and 1s of a Boolean model.

3) When translating a real-valued system into one with values restricted to 0 or 1, we can look for ways to draw the threshold between values we call “0” and values we call “1” at the most natural possible value for each particular network.

To evaluate whether any of these approaches can be useful in distilling a gene regulatory network into a Boolean representation, we must also develop a metric to measure how good an approximation the Boolean model is of the original, real-valued network. Using this metric, we may also develop other heuristics to determine which kinds of gene regulatory networks lend themselves to a Boolean representation.

We developed three quantitative metrics of this kind, and measured them in simulated networks composed of Boolean-friendly positive feedback loop modules, which exhibit switch-like behavior. We created detailed, realistic DE models of the networks, with biologically plausible parameter values. We then created an algorithm to produce the best possible Boolean representation of the continuous-value model. The algorithm defines the best position for the threshold dividing "expressed" from "unexpressed," and it selects the best Boolean rule set to approximate the more complicated behavior of the DE model. Finally, and perhaps most importantly, it offers a measurement of how faithfully the optimal Boolean representation reproduces the more detailed continuous dynamics.

We applied the algorithm to a variety of simple networks made up of feedback switches (the kind of network we expect, *a priori*, to be most suitable for a Boolean model) and studied what characteristics of the networks make them amenable to a Boolean representation. We found that certain kinds of feedback loops were more

amenable to the Boolean approximation, and others tended to produce modeling artifacts in the transition. The results reported here may serve as heuristics to guide Boolean models of gene regulatory networks.

Methods

We examined simulated gene regulatory networks made up of simple kinds of feedback loop switches to find the best Boolean representation of each, and to discover whether the properties of component motifs affected the "Booleanizability" of the overall network in which they were embedded. The two simplest kinds of biologically plausible positive feedback loop are an "activator loop" consisting of a single transcription factor that activates its own transcription (AL1) and a "repressor loop" of two genes that repress each others' transcription (RL2). Each of these can have two steady states, widely separated: 00/11 for the activator loop and 01/10 for the repressor loop. We assembled these into two-module networks, keeping the system small enough that we could analyze its behavior exhaustively. There are 31 topologically unique combinations of the two loop modules. We also created the mirror image of each of these, with transcriptional activation replaced with repression and vice versa (AL2, RL1), to produce two topologies for each kind of positive feedback loop. We used these to generate the counterpart of each of the original two-module networks, for a total of 62 two-module networks. This allowed us to control for network size in our comparisons, and directly evaluate the effect of activating vs. repressing feedback. See Chapter 1 for more details on the modules and the two-module networks assembled from them, and for the DEs and biologically-grounded parameters used to model these networks.

Our goal was to map the behavior of the continuous, DE network models onto a Boolean representation of the networks. The first step was to characterize the behavior of the DE version of each network. We begin by considering the isolated modules.

For each isolated module, we solved the DE model for 1000 initial concentrations using the SloppyCell toolkit (Gutenkunst et al. 2007) to generate information about how the trajectories flow (Figure 2.1a, b). Each initial concentration proceeded toward one of the two steady states possible for that module type.

The next task was to find the best way to divide the continuous variables (mRNA and protein concentrations) into bins designated 0 or 1.

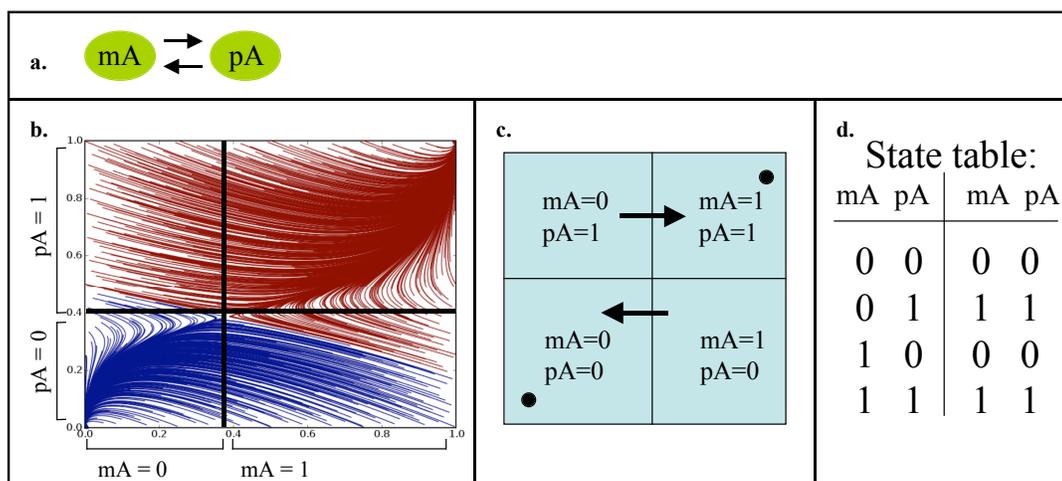


Figure 2.1. The algorithm for finding the optimal Boolean representation of a DE network model. **a**, the one-gene activator loop (AL1), in which a protein activates its own transcription. **b**, a phase portrait of the DE model of the loop. Trajectories, colored by basin of attraction, progress over time toward the stable fixed points in the corners. The optimal thresholds chosen by the algorithm are drawn in black. **c**, the transitions made by the majority of the trajectories in each threshold-delimited box. **d**, the Boolean state table derived from the transitions.

Finding the best threshold positions

Taking advantage of the feedback loops' switch-like property of cleanly separated "high" and "low" steady states, we followed the kind of on/off logical reasoning ("protein A represses gene B, so when A is highly expressed, B will likely be turned off") that geneticists use to think through and predict pathway behavior. This kind of reasoning divides the phase space into one region where protein A's concentration is considered low while B's is high, another where both are low, and so on (Figure 2.1b). The resulting "boxes" divide each continuous concentration variable into two discrete domains.

Of course, our model, like the mRNAs and proteins in a real genetic network, exhibits a much more continuous range of expression values. The steady states are high or low, but the process of separating high values into "expressed" (or 1) and low values into "unexpressed" (or 0) requires defining a threshold between the two categories for each node -- that is, deciding where the border of each "box" should be. If inappropriately chosen, the thresholds can introduce substantial modeling artifacts. To minimize such problems, we developed an algorithm to choose the most natural position of the threshold for each node by analyzing the properties of each network's phase portrait. The algorithm finds the natural boundaries between extrema by analyzing the flow of the trajectories themselves, and by using the nullclines of the system.

The algorithm weighted two factors equally in determining the optimal position of the thresholds:

- 1) We found, numerically, a point on the separatrix dividing the basins of attraction (the unstable fixed point occurring at the intersection of the nullclines), and penalized the thresholds if they were positioned too far from this. This helped align them with the natural divisions of the trajectories (see Figure 2.1b).

2) The discrete on/off representation of the system requires that the dynamics be discrete as well. In describing how the variables in the network in Figure 2.1 change over time, we say, "When we start with node mA "on" and node pA "off" (upper left box), we find that the system evolves to the state where both mA and pA are "on" (upper right box)." Such a description necessitates that all the trajectories in the first box move unanimously to the same new box. Of course, in a continuous system this is a simplification; in Figure 2.1b, a minority of the trajectories in the upper left box migrate to the bottom left box instead. Our boxes should be positioned such that a minimum of trajectories is neglected by the discretization in this way. Therefore, the second factor considered by the threshold-positioning algorithm minimized the number of "dissenting" trajectories that went to a box that was not the destination of the majority. This gave us the threshold position that produced the closest approximation of the unanimous-trajectories simplification.

The combination of these two factors provided a cost function for each potential threshold position. The algorithm explored phase space for the optimal threshold position (lowest value of the cost function) and placed the intersection of all the dimensions' thresholds at that position (Figure 2.2). By classifying the phase space into "on" and "off" regions as cleanly as possible, this discretization allowed us to apply traditional "if gene A is on, B will be off" pathway reasoning to the switch modules to predict how they would behave when combined.

Generating the Boolean rule set

Once the thresholds dividing "low" from "high" in each dimension have been established, they divide state space into "boxes" as described above. For a given box (e.g., "high mA/low pA" (1,0) box), the code simply calculates where the majority of the trajectories in that box go next. If most of them move next to the "low mA/low

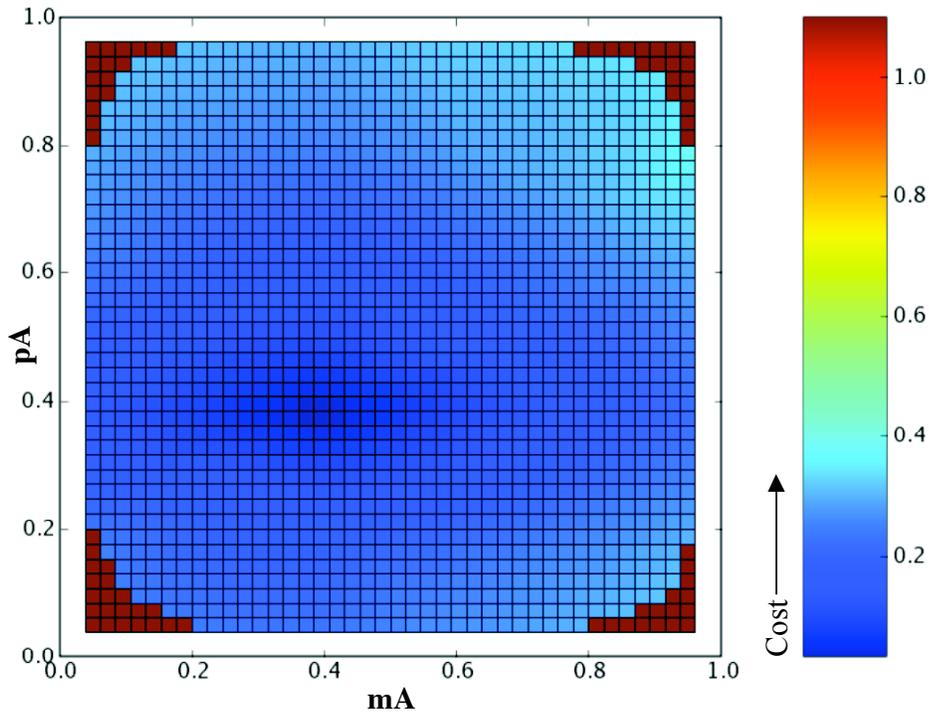


Figure 2.2. Choosing the optimal (lowest cost) threshold position. Dark red positions are excluded because each box is required to contain at least 1% of the initial conditions.

pA" box (1,1), we represent that in the state table (Figure 2.1c, d): $10 \rightarrow 00$. If, instead, most of them settle at a stable equilibrium point within the first box, which can be represented in the state table too: $10 \rightarrow 10$.

Performing this calculation for each box, the code generates a state table that represents the consensus of the 1000 trajectories characterizing phase space.

We can follow the state table rules to find, for each initial Boolean state, which box (state) it will eventually settle into. This gives us a way to measure how faithfully the simple Boolean representation replicates the dynamics of the more complex system.

A variety of state tables for each module

One way in which DE models can incorporate more detail about dynamics than Boolean models is that they have parameters describing nuances of the genetic interactions. When these parameters vary, they can change the paths by which trajectories reach their steady states (and sometimes even change the steady states themselves). Our equations contained biologically based parameters representing mRNA and protein half-lives, and the binding affinity and cooperativity of transcriptional regulators. Changing these parameters yielded different phase portraits. To approximate the DE models, we used our threshold-finding algorithm to infer the best Boolean state table for each kind of phase portrait (Figure 2.3).

For each module type (AL1, AL2, RL1, RL2), we identified the four Boolean state tables that summarized how its trajectories could flow in the DE model. We chose four *representative parameter sets* for the DE model of each kind of feedback loop, selecting those that best evoked the four characteristic state tables of that module (details in A2.1). We used these representative parameter sets when we assembled the single feedback loop modules into two-module networks.

The four representative parameter sets for each module were all *bistable*; that is, both steady states were represented. A feedback loop with particular parameters, or especially when subject to outside regulation, can move from a bistable condition like those shown in Figure 2.3 to a *monostable* phase portrait, in which all trajectories flow to a single steady state. For example, the module AL1 can settle on 00 or 11 as a steady state, depending on the initial concentrations of its mRNA and protein nodes. However, if it is embedded in a larger network where another module represses the transcription of the mRNA, it may shift to a monostable condition where all initial concentrations lead to the 00 steady state. Because we wanted our feedback loops to be free to vary in either direction when other modules sent them up-regulatory or

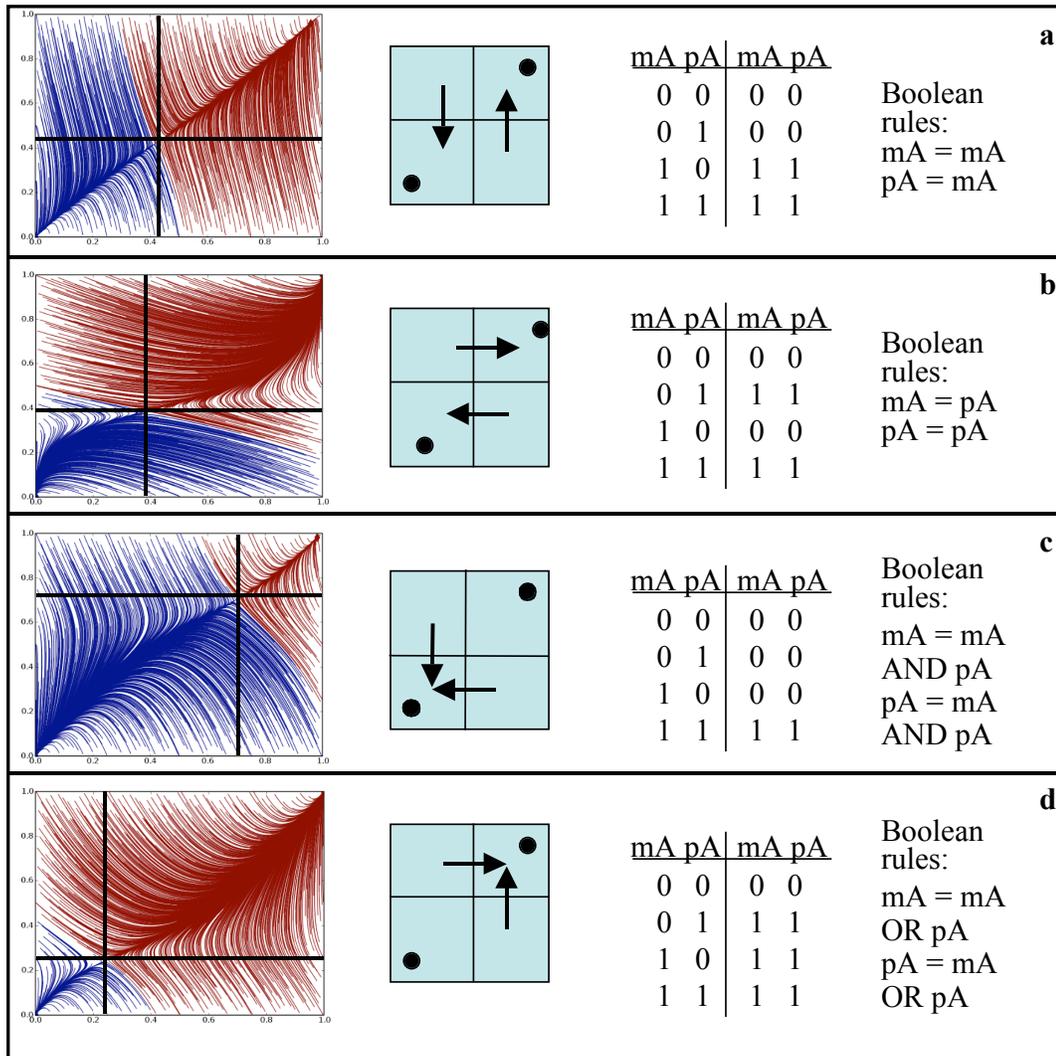


Figure 2.3. The four representative parameter sets for the single-gene activator loop (AL1) module, and their Boolean state tables and rulesets.

down-regulatory signals, we chose them to be bistable and thus maximally responsive to either kind of regulation from other elements in the network.

Two-module networks

We combined the feedback loop modules into two-module networks, in which a protein from one module could enhance or suppress the transcription of mRNA in another module (and vice versa, for many of the networks). To ensure that each two-

module network explored its full dynamical repertoire, we used four representative parameter sets for each of its modules. These encoded all the state table possibilities for a bistable module of a particular type. We assembled all unique pairwise combinations of the representative parameter sets of the modules in a two-module network, yielding 9 or 16 full parameter sets that encompassed all the behavior of a particular two-module network (see A2.1 for details). We simulated each of these full parameter sets, determined the best threshold positions, and generated an inferred Boolean state table to represent the behavior of the network under that particular parameter set. We measured the performance of the two-module network for each parameter set, and reported the mean measurement for that network across all its parameter sets.

Measuring the performance of the Boolean model for two-module networks

Because our ODE models, while mimicking real networks, have perfectly knowable behavior, we can precisely measure their departure from predicted behavior. We have three metrics for quantifying the “Booleanizability” of each two-module network.

Metric 1: Unanimity of trajectories

First, we have simply the measurement of how many “dissenting” trajectories there were in each threshold-defined box in phase space. For each isolated feedback loop and each two-module network, our algorithm chose thresholds that divided phase space into “boxes,” measured how trajectories flowed from one box to another, and gave us a Boolean representation of the DE network based on which box the majority of trajectories migrated to next. The algorithm positions the thresholds between boxes so as to minimize the number of dissenting trajectories. However, the trajectories are

never entirely unanimous in their destination, and the size of the dissenting minority is one measure of how much information we lose in the Boolean approximation. For example, in Figure 2.1b, some of the red trajectories in the $m_A=1/p_A=0$ (bottom right) box go to the top right box instead of joining the majority in the bottom left box. Dissenting trajectories like these lower the unanimity score for the network.

Metric 2: Agreement between predicted and simulated box transitions (state tables)

We can also measure how well a Boolean representation of a network's *component modules* (*i.e.*, the feedback loops) predicts the behavior of the larger network. While the “trajectory unanimity” metric described above is a measurement of how well we can represent a detailed DE system with a simpler Boolean rule set, we also want to quantify the artifacts that arise as the rule set is *executed* (iterated). We developed two metrics for doing so.

We made “predicted state tables” for each two-module network based on the Boolean rule sets for its component modules. For example, the two-module network AL1actAL1 is composed of two of the feedback loops illustrated in Figure 2.3. Consider the case in which the first AL1 module has the representative parameter set shown in Figure 2.3c:

$$m_A' = m_A \text{ AND } p_A$$

$$p_A' = m_A \text{ AND } p_A$$

and the second AL1 is at the representative parameter set from Figure 2.3a (using gene B to denote the second module in the network):

$$m_B' = m_B$$

$$p_B' = m_B$$

Adding the regulation between the modules, so that protein pA increases the transcription rate of mB, we have the overall rule set:

$mA' = mA \text{ AND } pA$

$pA' = mA \text{ AND } pA$

$mB' = mB \text{ OR } pA$ (*pA activates mB*)

$pB' = mB$

This gives us a new state table for the overall two-module network, which predicts what the transitions between boxes will be if the system behaves according to simple Boolean rules. We created these *predicted state tables* for each parameter set of each two-module network (Figure 2.4a). If pA's effect on mA had been that of a repressor instead (AL1repAL1), we would have applied the rule "mB = mB AND NOT pA." As mentioned above, our algorithm generated an "inferred state table" from the consensus of trajectory movements from one threshold-defined box to another (Figure 2.4b). We compared this inferred state table to the predicted state table that we had generated by combining the rule sets of the individual modules. We measured the Hamming distance between the two state tables (right-hand sides only) to determine how different the DE-derived state table was from the one predicted by Boolean logic. We weighted the comparisons by how many trajectories had "chosen" each box in the inferred state table (see A2.2 for details). The resulting score represented the proportion of digits that were consistent between the predicted and DE-inferred state tables (maximum score = 1.0). This gave us a second metric for measuring how cleanly a two-module network fitted a Boolean representation: the similarity of its state tables.

Metric 3: Agreement of predicted and simulated steady states

To find out where a Boolean trajectory beginning at a particular initial concentration will end, we can iterate through the state table to find its steady state. For example, in Figure 2.4a, the path from 1100 to its steady state is: 1100,

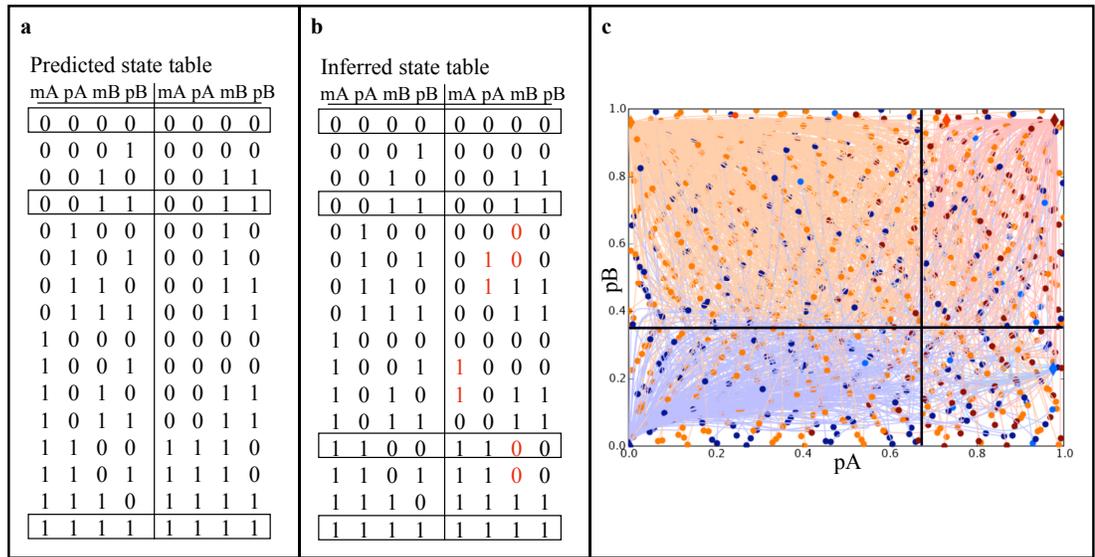


Figure 2.4. Comparing predicted and inferred state tables for network AL1actAL1. The AL1 modules have the representative parameter sets shown in fig. 3c and 3a. **a**, The state table predicted by applying Boolean logic to the rule sets of the loop module components (see text). **b**, The state table inferred from the DE model by noting the transition made by the majority of the trajectories in each box. Unpredicted values are shown in red. Steady states (LHS=RHS) are boxed. **c**, A phase portrait of the network. Dots are initial concentrations, colored according to their eventual steady state. Each dot is the starting point of a single trajectory.

1110, 1111. If a predicted state table is incorrect in a few of its details, the errors can accumulate as it is applied iteratively to determine a steady state. We designed a third measurement in order to capture the *cumulative* difference between the predicted state table and the DE model, as trajectories passed from their initial state to their final concentrations.

We compared the steady states dictated by the predicted Boolean state table to the steady states actually achieved by each trajectory in the DE model. For each parameter variant of each two-module network, we recorded the initial concentrations that fell within each box, then integrated their trajectories to determine in which box they reached a steady state. For example, in network AL1actAL1 (Figure 2.4c), 46% of the trajectories beginning in the "mA/pA high, mB/pB low" (1100, bottom right)

box ended in the same box, while 54% ended in the "mA/pA high, mB/pB high" (1111, top right) box. (Dots in bottom right box of Figure 2.4c whose trajectories go elsewhere have values of pA=1, pA=0 but also have mA \neq 1 and/or mB \neq 0 and thus are not in the 1100 box. The visual overlap arises from the two-dimensional projection of 4-dimensional space.)

We compared the initial/final box of the DE trajectories to the initial/final state in the predicted state table, and scored each two-module network for how close the two were, averaged across the representative parameter set combinations (see A2.3 for details). Binning the initial concentrations and final steady states of the DE model into boxes distilled the behavior of the continuous, detailed model into the same level of resolution supplied by our Boolean state tables, enabling us to compare the predictions of the simple model to the "reality" of the DE system.

Results and Discussion

We simulated DE representations of small feedback loops containing transcriptional activators (AL) or repressors (RL). We found the best Boolean representation of each, and then repeated the process for larger networks assembled from pairs of the original loop modules. We used the Boolean representations of individual modules to predict the Boolean state tables for the two-module networks, and compared these to the actual state tables derived by simulating the two-module DE. We also measured the ease with which each DE system was approximated by the Boolean model inferred directly from it, and compared the basins of attraction for the predicted state tables and the actual trajectories in the DE simulation.

Specifically, we measured three estimates of the "Booleanizability" of each two-module network: 1) the unanimity of its trajectory movements from box to box; 2)

the congruence between the state table predicted by Boolean assembly of its component modules and the state table inferred directly from the DE model; and 3) the agreement between the initial/steady state pairings calculated from the predicted steady state table and those recorded in the DE simulation.

Our goal was to find heuristics that would help clarify, for the benefit of modelers, which qualities of a network make it amenable to a Boolean representation. To that end, we compared the networks containing entirely repressor loop (RL) modules to those containing entirely activator loop (AL) modules. The exhaustive set of two-module networks meant that each AL/AL network had a corresponding RL/RL network with the same topology but reversed within-module regulatory signs (see pairs of networks in Chapter 1, A1.1). This correspondence allowed us to test the effect of activation *vs.* repression feedback directly, by comparing pairs of networks that differed with respect to that property but which were the same with respect to other characteristics, such as network size. We used paired bootstrap comparisons (see A2.4 for details). Our results are in Table 2.1.

For all three measures of “Booleanizability,” we found that the RL/RL networks had highly significantly greater values than those for AL/AL networks. The difference was most pronounced for the measures of how cleanly trajectories sorted themselves into boxes (metric 1, unanimity) and of the cumulative errors that arise as we iterated Boolean state tables (metric 3, steady-state comparisons).

We also compared two-gene, three-gene, and four-gene network metrics to study the effects of network size. We used unpaired permutation testing to determine how network size influenced each of the three “Booleanization” metrics (see A2.4). We found that across all size comparisons and metrics, smaller networks were highly significantly more amenable to a Boolean representation (Table 2.1).

Table 2.1. Mean values of “Booleanizability” estimators, and comparisons across classes of two-module networks. The value for each network is the mean metric across all combinations of representative parameter sets for the network. Means across network classes are presented here, e.g., the RL/RL mean encompassing all RL1/RL1 and RL2/RL2 networks. Activating vs. repressing feedback comparisons (RL/RL vs. AL/AL) are paired permutation tests; network size comparisons are unpaired bootstrap tests (see A2.4). **Highly significant.

| | | | | | | | | | | | | | | |
|--|--|---------------|---------------|-----------------|-----------------|---|-----------------|---------------|-----------------|-----------------|-----------------|---------------|-----------------|-----------------|
| <p>a) First metric: Unanimity of the trajectory box-transitions in the DE-simulated state table.</p> | <p>Effect of repressing vs. activating feedback</p> <table border="1"> <tr> <td>RL/RL > AL/AL</td> <td>p = 2.00e-4**</td> </tr> <tr> <td>0.845 (N=19)</td> <td>0.782 (N=19)</td> </tr> </table> | RL/RL > AL/AL | p = 2.00e-4** | 0.845 (N=19) | 0.782 (N=19) | <p>Effect of network size</p> <table border="1"> <tr> <td>2-gene > 3-gene</td> <td>p = 2.00e-4**</td> </tr> <tr> <td>0.903 (N=10)</td> <td>0.850 (N=24)</td> </tr> <tr> <td>3-gene > 4-gene</td> <td>p = 2.00e-4**</td> </tr> <tr> <td>0.850 (N=24)</td> <td>0.781 (N=28)</td> </tr> </table> | 2-gene > 3-gene | p = 2.00e-4** | 0.903 (N=10) | 0.850 (N=24) | 3-gene > 4-gene | p = 2.00e-4** | 0.850 (N=24) | 0.781 (N=28) |
| RL/RL > AL/AL | p = 2.00e-4** | | | | | | | | | | | | | |
| 0.845 (N=19) | 0.782 (N=19) | | | | | | | | | | | | | |
| 2-gene > 3-gene | p = 2.00e-4** | | | | | | | | | | | | | |
| 0.903 (N=10) | 0.850 (N=24) | | | | | | | | | | | | | |
| 3-gene > 4-gene | p = 2.00e-4** | | | | | | | | | | | | | |
| 0.850 (N=24) | 0.781 (N=28) | | | | | | | | | | | | | |
| <p>b) Second metric: Agreement between predicted and simulated state tables.</p> | <p>Effect of repressing vs. activating feedback</p> <table border="1"> <tr> <td>RL/RL > AL/AL</td> <td>p = 2.60e-3**</td> </tr> <tr> <td>0.766 (N=19)</td> <td>0.755 (N=19)</td> </tr> </table> | RL/RL > AL/AL | p = 2.60e-3** | 0.766 (N=19) | 0.755 (N=19) | <p>Effect of network size</p> <table border="1"> <tr> <td>2-gene > 3-gene</td> <td>p = 2.00e-4**</td> </tr> <tr> <td>0.810 (N=10)</td> <td>0.770 (N=24)</td> </tr> <tr> <td>3-gene > 4-gene</td> <td>p = 2.00e-4**</td> </tr> <tr> <td>0.770 (N=24)</td> <td>0.743 (N=28)</td> </tr> </table> | 2-gene > 3-gene | p = 2.00e-4** | 0.810 (N=10) | 0.770 (N=24) | 3-gene > 4-gene | p = 2.00e-4** | 0.770 (N=24) | 0.743 (N=28) |
| RL/RL > AL/AL | p = 2.60e-3** | | | | | | | | | | | | | |
| 0.766 (N=19) | 0.755 (N=19) | | | | | | | | | | | | | |
| 2-gene > 3-gene | p = 2.00e-4** | | | | | | | | | | | | | |
| 0.810 (N=10) | 0.770 (N=24) | | | | | | | | | | | | | |
| 3-gene > 4-gene | p = 2.00e-4** | | | | | | | | | | | | | |
| 0.770 (N=24) | 0.743 (N=28) | | | | | | | | | | | | | |
| <p>c) Third metric: Agreement between predicted and simulated estimates of initial/steady states.</p> | <p>Effect of repressing vs. activating feedback</p> <table border="1"> <tr> <td>RL/RL > AL/AL</td> <td>p = 2.00e-4**</td> </tr> <tr> <td>0.782 (N=19)</td> <td>0.760 (N=19)</td> </tr> </table> | RL/RL > AL/AL | p = 2.00e-4** | 0.782 (N=19) | 0.760 (N=19) | <p>Effect of network size</p> <table border="1"> <tr> <td>2-gene > 3-gene</td> <td>p = 4.00e-4**</td> </tr> <tr> <td>0.847 (N=10)</td> <td>0.792 (N=24)</td> </tr> <tr> <td>3-gene > 4-gene</td> <td>p = 2.00e-4**</td> </tr> <tr> <td>0.792 (N=24)</td> <td>0.744 (N=28)</td> </tr> </table> | 2-gene > 3-gene | p = 4.00e-4** | 0.847 (N=10) | 0.792 (N=24) | 3-gene > 4-gene | p = 2.00e-4** | 0.792 (N=24) | 0.744 (N=28) |
| RL/RL > AL/AL | p = 2.00e-4** | | | | | | | | | | | | | |
| 0.782 (N=19) | 0.760 (N=19) | | | | | | | | | | | | | |
| 2-gene > 3-gene | p = 4.00e-4** | | | | | | | | | | | | | |
| 0.847 (N=10) | 0.792 (N=24) | | | | | | | | | | | | | |
| 3-gene > 4-gene | p = 2.00e-4** | | | | | | | | | | | | | |
| 0.792 (N=24) | 0.744 (N=28) | | | | | | | | | | | | | |

The three metrics we examined measure subtly different qualities of the Boolean network representation. The “unanimity” metric tests how clean the process of dividing variables into 1/0 bins was. The state table comparison metric measures this as well, because we weight it by the proportions of trajectories that went to each box (see A2.2), but, more importantly, it measures to what extent the abstraction to 1/0 was *accurate* (not merely how precise it was). Finally, the third metric captures the *total* error that accumulates as we iterate the Boolean state table, and apply the successive layers of approximation. For example, in some networks, threshold boxes had dissenting trajectories (lowering the unanimity score), but the majority of

trajectories still transitioned to the predicted next box (resulting in a good state table comparison score, or metric 2). In some cases, small prediction errors that were revealed as minor reductions in the state table comparison metric accumulated as the state tables were iterated to find predicted Boolean steady states, and produced substantially lowered steady state comparison scores (metric 3). Overall, the three metrics encapsulate the various ways in which a network may (or may not) lend itself to a Boolean approximation. For all three metrics, we found that smaller networks and networks with repressor loops survived the approximation with their properties relatively intact.

The steady state comparison metric, in particular, gives us insight into why activator loops (AL) make networks less amenable to a Boolean representation. It measures steady states that we predicted would be lost due to the regulation coming from the other module in the network, but which were actually retained for many activator loop (AL) modules, albeit usually with smaller basins of attraction (see Figures 1.11, 1.12). These lingering steady states meant that modules that we had predicted to become monostable instead retained their bistability. The unexpectedly bistable cases caused divergence between actual and predicted steady states, as illustrated the steady state comparison scores and by the boxed state table elements in Figure 2.4.

We found that two-module networks containing RL loops are significantly more amenable to a Boolean representation than those containing AL loops. This result arises from the fact that repressor loops are substantially more likely to lose one of their two steady states when they receive regulation from outside (see Chapter 1). When isolated, the representative parameter sets we selected for repressor loops are all bistable: They can assume either the 01 or 10 steady state (0011/1100, for two-gene modules), depending on the initial concentrations in the network. We have found that

when repressor loops are subject to external regulatory “pushes,” such as those from the other module in the two-module network, they tend to comply completely with the external influence and lose the steady state that it discourages. For example, two RL1 modules have four steady states between them: $mA=0/pA=1/mB=0/pB=1$ (0101), 0110, 1001, and 1010. However, when pA in the first RL1 module activates transcription of mB in the second one (the RL1actRL1 two-module network), the steady state with $pA=1$, $mB=0$ disappears (Figure 2.5a).

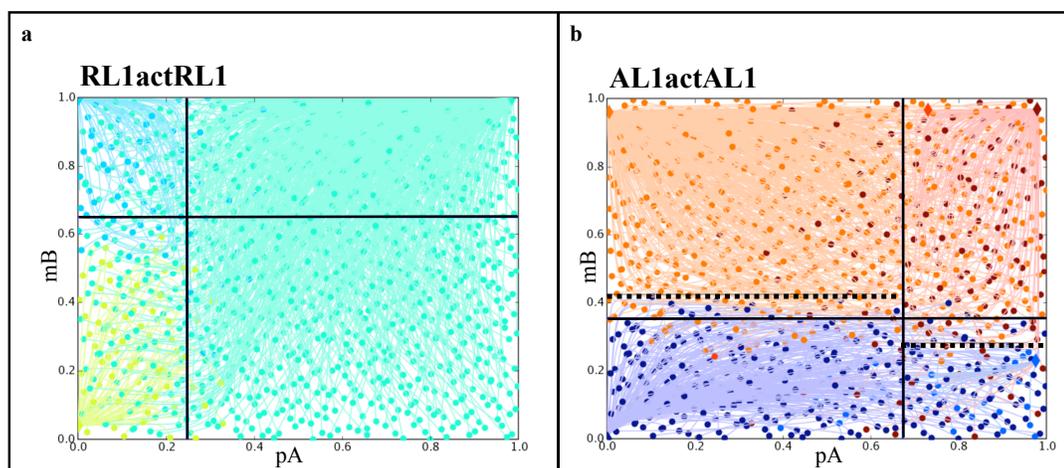


Figure 2.5. RL networks are more likely to lose one of their steady states, and are easier to Booleanize as a result. **a**, RL1actRL1 has lost the steady state at $pA=1$, $mB=0$ because pA activates the transcription of mB , preventing it from being unexpressed when pA is at 1. **b**, AL1actAL1 has not entirely lost the steady state at $pA=1$, $mB=0$, despite the fact that pA activates mB transcription. Solid lines indicate threshold positions chosen by the algorithm. Dotted lines in **b** indicate better threshold positions that would violate the Boolean assumption of a single threshold per variable.

Activator loops, by contrast, do not respond as thoroughly to external regulation, and often retain the “discouraged” steady state with its basin reduced in size (Figure 2.5b). We discuss the reasons for this in Chapter 1. These small, remnant basins are particularly problematic for a Boolean representation because of the conflict they provoke in the issue of where to draw the thresholds. They accentuate an

important artifact inherent in Boolean modeling: that a single threshold must have the same meaning in all contexts. If pA is in the state we classify as 0, then it must be classified as 0 regardless of the values of other variables, and regardless of which of its targets we are considering its effects upon. Only one threshold between 0 and 1 is allowed per variable. This is the problem Thomas and colleagues attempted to address with their discrete logical models (Thomas and D'Ari 1990).

Unequal basin sizes complicate the decision of where to draw the threshold in any network, but the situation arises frequently in AL networks because of the small, leftover basins caused by incomplete response to external regulation. This is an emergent effect, which arises because the AL modules are embedded in a larger regulatory context in which they receive signals from outside. Individual AL and RL loops are approximately equally amenable to a Boolean representation, as measured by the unanimity of their trajectories (see section A2.4). It is the combination of a feedback loop and the regulation coming from outside it that produce the greater “Booleanizability” of RL-containing networks.

To see why, consider a pair of isolated bistable feedback loops with no regulation between them. This system has four possible steady states – two per loop module. When the loops are joined to form a two-module network, the regulation between the modules can induce monostability in one both of the loops. A two-module network that loses one of its four steady states (or two of them, as often happens in the networks that have feedback between the modules) becomes easier to “Booleanize” in a number of ways. First, the trajectory paths can become simpler, because there are fewer destinations. This bring the network closer to the Boolean ideal of all trajectories crossing thresholds in the same direction.

More importantly, fewer basins mean fewer difficulties arising from conflicts in where to draw the thresholds. In Figure 2.5b, dotted lines indicate ideal threshold

positions for the variable mB on either side of the pA threshold. The dotted lines do not align because the two $mB=0$ basins of attraction are of different sizes. As a result, some of the trajectories are placed in the wrong “box” for their basin of attraction, and the Boolean approximation of the network diverges from the DE simulated behavior. When one side of phase space lacks a steady state, the conflict over mB 's threshold position evaporates, and the Boolean representation is cleaner and more predictive of the DE model's behavior.

We also found that smaller networks have some of the same advantages as those with fewer steady states (Table 2.1). When phase space has a smaller number of dimensions, trajectories are constrained in their flow and easier to partition. Each new variable added to the network means a new dimension in phase space and a new threshold that divides all pre-existing boxes in half, multiplying the possibilities of divergent trajectories and conflicts over threshold positions. We see this phenomenon at all levels of size increases in our model: 2-gene networks have better Boolean approximations than 3-gene networks, and 3-gene networks “Booleanize” better than 4-gene networks (Table 2.1). This is an ominous finding for those hoping to take advantage of the tractability of Boolean networks by applying them to large-scale regulatory networks.

However, more modestly sized networks may benefit from a Boolean model. Our finding that RL-rich networks lend themselves better to a Boolean representation has implications for the kinds of networks it may be most profitable for Boolean modelers to investigate. There is evidence, for example, that repressor loops play an important role in eukaryotic developmental networks (see Chapter 1). Especially when the detailed kinetic parameters of a system are unknown (as is often the case), Boolean models can be useful ways to make an “influence model” more rigorous, and

test intuitive hypotheses about how the individual interactions of the system produce tissue-scale patterns (Tomlin and Axelrod 2007).

Another fruitful area for Boolean modeling could be networks of prokaryotic regulatory switches, especially catabolic gene circuits such as the well-known *lac* operon (Ferrell 2002). These consist of mutually repressing genes that control whether a cell metabolizes certain nutrients, depending on their availability in the environment.

APPENDIX

A2.1: Choosing representative parameter sets for each module type

For the time-consuming “Booleanization” simulations of two-module networks, we needed to choose a limited number of parameter sets that summarized the behavior each network could produce. To this end, we selected four *representative parameter sets* for each module type (i.e., four each for AL1, AL2, RL1, and RL2). Then, for each two-module network, we simulated all 16 combinations of the representative parameter sets for each of its two component modules. For example, module AL1 has representative parameter sets labeled A, B, C, and D, while module RL2 has representative parameter sets E, F, G, and H. All two-module networks containing AL1 and RL2 (AL1repRL2rep1, for example) were simulated for the following overall parameter sets:

- AL1 parameter set A, RL2 parameter set E
- AL1 parameter set A, RL2 parameter set F
- AL1 parameter set A, RL2 parameter set G
- AL1 parameter set A, RL2 parameter set H
- AL1 parameter set B, RL2 parameter set E
- AL1 parameter set B, RL2 parameter set F
- AL1 parameter set B, RL2 parameter set G
- AL1 parameter set B, RL2 parameter set H
- AL1 parameter set C, RL2 parameter set E
- AL1 parameter set C, RL2 parameter set F
- AL1 parameter set C, RL2 parameter set G
- AL1 parameter set C, RL2 parameter set H
- AL1 parameter set D, RL2 parameter set E
- AL1 parameter set D, RL2 parameter set F
- AL1 parameter set D, RL2 parameter set G
- AL1 parameter set D, RL2 parameter set H

We found the best threshold positions and scored the “Booleanizability” of the two-module network at each of these 16 parameter sets.

Some two-module networks have inherent symmetry that meant some of the pairwise combinations of module parameter sets were redundant. For example, AL1actAL1act with module parameter sets A, B is the same as that network with parameter sets B, A. For these symmetric two-module networks, we simulated only the nine unique parameter set combinations.

To select the representative parameter sets for each module type, we first sampled parameter space using the Sobol algorithm (Press 1992; Gädke et al. 2007) and simulated each module type with 100 parameter sets. We calculated the consensus inferred state table and “Booleanization” score for each of these (see main text). Each module type has four possible consensus state tables (Figure 2.3, main text). From among the 100, we chose one parameter set to represent each state table based on high “Booleanization” scores and numerical tractability. We repeated this process for each module type, producing the four representative parameter sets for each of AL1, AL2, RL1, and RL2. The four representative rule sets for AL1 are pictured in Figure 2.3.

A2.2. Comparing predicted and inferred state tables.

We measured how well a Boolean network assembled from the rule sets of individual modules could predict the behavior of a two-module DE network. To do so, we first applied Boolean logic to the rule sets for each module to generate a “predicted state table” (see main text). We also inferred a state table from the DE simulation of the same network by calculating, for the trajectories in each threshold-defined “box,” which box the majority of them went to next. To measure the

difference between the predicted and inferred state tables, we calculated the Hamming distance (number of differing digits) between the right-hand side entries of the predicted state table and those of the inferred state table (see Figure 2.3a, b).

We then weighted each line of the tables by the proportion of trajectories that had gone to each box. The overall comparison score is the Hamming distance between the predicted and inferred state table right-hand side (RHS) entries, weighted by their frequency in the inferred state table. For example, for the network AL1actAL1 (representative parameter sets shown in Figures 2.3C and 2.3A, respectively), the DE-inferred state table is as follows. (Variables are mA, pA, mB, and pB.)

| Current box | Next box | Proportion of trajectories in current box that make that transition |
|-------------|----------|---|
| 0000 | → 0000 | 0.748 |
| 0000 | → 0010 | 0.238 |
| 0000 | → 0100 | 0.014 |
| 0001 | → 0000 | 0.923 |
| 0001 | → 0011 | 0.077 |
| 0010 | → 0000 | 0.007 |
| 0010 | → 0011 | 0.982 |
| 0010 | → 0110 | 0.011 |
| 0011 | → 0010 | 0.067 |
| 0011 | → 0011 | 0.933 |
| 0100 | → 0000 | 0.777 |
| 0100 | → 0110 | 0.085 |
| 0100 | → 1100 | 0.138 |
| 0101 | → 0100 | 0.941 |
| 0101 | → 0111 | 0.058 |
| 0110 | → 0010 | 0.714 |
| 0110 | → 0100 | 0.006 |

| | |
|-------------|-------|
| 0110 → 0111 | 0.106 |
| 0110 → 1110 | 0.174 |
| 0111 → 0011 | 0.521 |
| 0111 → 0110 | 0.292 |
| 0111 → 1111 | 0.188 |
| 1000 → 0000 | 0.084 |
| 1000 → 1010 | 0.048 |
| 1000 → 1100 | 0.867 |
| 1001 → 1000 | 0.933 |
| 1001 → 1101 | 0.067 |
| 1010 → 0010 | 0.086 |
| 1010 → 1000 | 0.008 |
| 1010 → 1011 | 0.039 |
| 1010 → 1110 | 0.867 |
| 1011 → 0011 | 0.030 |
| 1011 → 1010 | 0.242 |
| 1011 → 1111 | 0.727 |
| 1100 → 1110 | 1 |
| 1101 → 1100 | 1 |
| 1110 → 1111 | 1 |
| 1111 → 1110 | 0.054 |
| 1111 → 1111 | 0.946 |

Some boxes are unanimous; box 1100, for example, sends all its trajectories to box 1110. However, for most boxes the trajectories are not unanimous. For example, the predicted state table indicates that trajectories in box 1011 will go to box 0011. In the inferred state table above, the score for box 1011 is:

| Predicted RHS | Inferred RHS | Hamming distance | Freq. of traj. at this entry | Contrib. to total weighted Hamming dist. |
|------------------|-----------------|---------------------|---------------------------------|---|
| 0011 | 0011 | 0 | 0.030 | 0 |
| 0011 | 1010 | 2 | 0.242 | 0.484 |
| 0011 | 1111 | 2 | 0.727 | 1.454 |
| | | | | 1.938 |

Each line in the tables has a maximum contribution of 4. The final comparison score is the weighted proportion of digits that were *consistent* between the predicted and inferred table: $1 - \text{sum}(\text{weighted Hamming distances}) / (4 * \text{number of lines})$. The maximum score is 1.0. We calculated this score for each combination of representative parameter sets (16 for most two-module networks; 10 for some with symmetry that rendered some parameter set combinations redundant) and reported the mean score across all of the parameter sets for a given network.

A2.3. Comparing predicted and DE initial/final states.

We iterated each initial state in the predicted state table to find its final steady state. We compared these predictions to the boxes containing the actual initial and final states of each trajectory in the DE model. Analogously to the state tables, we weighted the scores by the proportion of each box of initial concentrations that ended at each particular steady state.

Occasionally, the synchronous updating of the Boolean state tables produced a cycle where none existed in the DE system. For example, in the predicted state table of RBRactRBRrep1 (representative parameter sets B, B), the initial state 0110 goes to 0100, then 1100, 1110, and back to 0110 again. These loops tended to be oscillations

among unstable states. We scored them as having the maximum Hamming distance between predicted and DE-measured states.

A2.4. Statistical comparisons.

Comparing two-module networks

Our set of two-module networks represented all combinations of the four types of feedback loops, within the rules we chose for combining the pairs of modules with regulatory relationships (see main text). Because our data are the entire population rather than a sample thereof, we must use resampling approaches to find estimates of confidence in the differences between classes of networks.

For the AL vs. RL comparisons, we were able to use paired bootstrap comparisons. Each AL/AL network had a corresponding RL/RL network which was identical in its inter-module regulation but had reversed regulatory signs within the modules. For example, AL1actAL1rep corresponded to RL1actRL1rep. The paired nature of our data allowed us to directly compare the effects of AL vs. RL modules, controlling for factors such as size. We resampled from the paired list 10,000 times and calculated the p-value as detailed in Chapter 1. The minimum p-value for 10,000 samples is $2.00e-4$.

The network size comparisons did not allow pairwise comparisons, so here we used permutation testing, again with 10,000 samples.

Comparing individual modules

To see whether the trend of RL modules conferring greater “Booleanizability” persisted when the modules were isolated, we compare the unanimity score (metric 1) for each of the four isolated module types: AL1, RL1, AL2, and RL2. (Metrics 2 and 3 measure how well isolated Boolean representations of isolated modules predict the

Table A2.1. Mean values of “Booleanizability” estimators for classes of two-module networks. The value for each network is the mean metric across all combinations of representative parameter sets for the network. Means across network classes are presented here, e.g., the RL1/RL1 mean encompassing all RL1/RL1 two-module networks.

| | | | | | | |
|---|---------------------|-----------------------|----------------------|---------------------|-----------------------|----------------------|
| a) First metric: Unanimity of the trajectory box-transitions in the DE- simulated state table. | Network category | Number of networks | Mean metric value | Network category | Number of networks | Mean metric value |
| | AL1/AL1 | 5 | 0.902956 | AL1/RL2 | 12 | 0.862513 |
| | RL1/RL1 | 5 | 0.903043 | AL2/RL1 | 12 | 0.838095 |
| | AL2/AL2 | 14 | 0.738550 | | | |
| | RL2/RL2 | 14 | 0.824100 | | | |
| b) Second metric: Agreement between predicted and simulated state tables. | Network category | Number of networks | Mean metric value | Network category | Number of networks | Mean metric value |
| | AL1/AL1 | 5 | 0.804530 | AL1/RL2 | 12 | 0.774515 |
| | RL1/RL1 | 5 | 0.816009 | AL2/RL1 | 12 | 0.766349 |
| | AL2/AL2 | 14 | 0.737468 | | | |
| | RL2/RL2 | 14 | 0.748740 | | | |
| c) Third metric: Agreement between predicted and simulated estimates of initial/steady states. | Network category | Number of networks | Mean metric value | Network category | Number of networks | Mean metric value |
| | AL1/AL1 | 5 | 0.844660 | AL1/RL2 | 12 | 0.800168 |
| | RL1/RL1 | 5 | 0.848733 | AL2/RL1 | 12 | 0.783503 |
| | AL2/AL2 | 14 | 0.730644 | | | |
| | RL2/RL2 | 14 | 0.758127 | | | |

behavior of larger networks, and were therefore inappropriate for testing the isolated modules themselves.) For each isolated module, we measured the unanimity score for each of its four representative parameter sets, and took the mean value as the measurement for that module. The results are:

AL1 0.951
 RL1 0.954
 AL2 0.809
 RL2 0.836

Statistical comparisons are inappropriate here: We cannot assume that the values for the various parameter sets are independent, so we cannot apply conventional statistics to comparing the four parameter set measurements of (say) AL1 to those of AL2. The

four parameter sets of each module capture its repertoire of bistable behaviors (Figure 2.3) and serve as a basis for making representative parameter set combinations for two-module networks, but are insufficient to allow resampling methods. (Practically speaking, the sample size must be at least 6 so that the 10,000 random draws will not repeat themselves too often. There are N^N unique samples with replacement of size N , and 4^4 is only 256, much less than 10,000). We must confine ourselves, therefore, to examining the means above. We observe that while the two-gene modules have lower scores than the 1-gene modules (continuing the trend noted with 2-, 3-, and 4-gene networks reported in the main text), there is no clear trend between AL and RL modules, as AL1 and RL1 are nearly identical.

APPENDIX REFERENCES

- Gädke, A, Küpper, J, Schnizer, P. 2007. pygsl: Python interface for the GNU scientific library. <http://sourceforge.net/projects/pygsl/>
- Press, WH. 1992. Numerical recipes in C: the art of scientific computing. Cambridge; New York: Cambridge University Press. p. xxvi, 994.

REFERENCES

- Aguda, BD, Goryachev, AB. 2007. From pathways databases to network models of switching behavior. *PLoS Comput Biol* 3:1674–1678.
- Albert, R, Othmer, HG. 2003. The topology of the regulatory interactions predicts the expression pattern of the segment polarity genes in *Drosophila melanogaster*. *J Theor Biol* 223(1):1–18.
- Alon, U. 2007. An introduction to systems biology : design principles of biological circuits. Boca Raton, FL: Chapman & Hall/CRC. p. xvi, 301 p., [4] p. of plates.
- Brandman, O, Ferrell, JEJ, Li, R, Meyer, T. 2005. Interlinked fast and slow positive feedback loops drive reliable cell decisions. *Science* 310:496–498.
- Chaves, M, Sontag, ED, Albert, R. 2006. Methods of robustness analysis for Boolean models of gene control networks. *Syst Biol (Stevenage)* 153:154–167.
- Covert, MW, Palsson, BO. 2002. Transcriptional regulation in constraints-based metabolic models of *Escherichia coli*. *J Biol Chem* 277(31):28058–28064.
- Covert, MW, Xiao, N, Chen, TJ, Karr, JR. 2008. Integrating metabolic, transcriptional regulatory and signal transduction models in *Escherichia coli*. *Bioinformatics* 24:2044–2050.
- Davidson, EH. 2006. The regulatory genome: gene regulatory networks in development and evolution. Burlington, MA ; San Diego: Academic. p. xi, 289 p.
- Elowitz, MB, Levine, AJ, Siggia, ED, Swain, PS. 2002. Stochastic gene expression in a single cell. *Science* 297(5584):1183–1186.
- Ferrell, JEJ. 2002. Self-perpetuating states in signal transduction: positive feedback, double-negative feedback and bistability. *Curr Opin Cell Biol* 14(2):140–148.

- Glass, L, Kauffman, S. 1973. The logical analysis of continuous, non-linear biochemical control networks. *J Theor Biol* 39:103–129.
- Gutenkunst, RN, Atlas, JC, Casey, FP, Kuczynski, RS, Waterfall, JJ, Myers, CR, Sethna, JP. 2007. SloppyCell. <http://sloppycell.sourceforge.net>
- Husbands, P, Harvey, I. 1997. Fourth European Conference on Artificial Life. Cambridge, Mass: MIT Press. p. ix, 583 p.
- Istrail, S, Davidson, EH. 2005. Logic functions of the genomic *cis*-regulatory code. *Proc Natl Acad Sci U S A* 102:4954–4959.
- Istrail, S, De-Leon, SB, Davidson, EH. 2007. The regulatory genome and the computer. *Dev Biol* 310:187–195.
- Li, F, Long, T, Lu, Y, Ouyang, Q, Tang, C. 2004. The yeast cell-cycle network is robustly designed. *Proc Natl Acad Sci U S A* 101:4781–4786.
- Mayo, AE, Setty, Y, Shavit, S, Zaslaver, A, Alon, U. 2006. Plasticity of the *cis*-regulatory input function of a gene. *PLoS Biol* 4:e45.
- McAdams, HH, Arkin, A. 1999. It's a noisy business! Genetic regulation at the nanomolar scale. *Trends Genet* 15:65–69.
- McAdams, HH, Shapiro, L. 1995. Circuit simulation of genetic networks. *Science* 269(5224):650–656.
- Ozbudak, EM, Thattai, M, Kurtser, I, Grossman, AD, van Oudenaarden, A. 2002. Regulation of noise in the expression of a single gene. *Nat Genet* 31(1):69–73.
- Sanchez, L, Thieffry, D. 2001. A logical analysis of the *Drosophila* gap-gene system. *J Theor Biol* 211(2):115–141.
- Sanchez, L, Thieffry, D. 2003. Segmenting the fly embryo: a logical analysis of the pair-rule cross-regulatory module. *J Theor Biol* 224:517–537.

- Sanchez, L, van Helden, J, Thieffry, D. 1997. Establishment of the dorso-ventral pattern during embryonic development of *Drosophila melanogaster*: a logical analysis. *J Theor Biol* 189:377–389.
- Setty, Y, Mayo, AE, Surette, MG, Alon, U. 2003. Detailed map of a *cis*-regulatory input function. *Proc Natl Acad Sci U S A* 100(13):7702–7707.
- Shmulevich, I, Dougherty, ER, Kim, S, Zhang, W. 2002. Probabilistic Boolean Networks: a rule-based uncertainty model for gene regulatory networks. *Bioinformatics* 18:261–274.
- Smolen, P, Baxter, DA, Byrne, JH. 2000. Modeling Transcriptional Control in Gene Networks—Methods, Recent Results, and Future Directions. *Bulletin of Mathematical Biology* 62:247–292.
- Soulé, C. 2003. Graphic requirements for multistationarity. *ComplexUs* 1:123–133.
- Sudarsan, N, Hammond, MC, Block, KF, Welz, R, Barrick, JE, Roth, A, Breaker, RR. 2006. Tandem riboswitch architectures exhibit complex gene control functions. *Science* 314:300–304.
- Swain, PS, Elowitz, MB, Siggia, ED. 2002. Intrinsic and extrinsic contributions to stochasticity in gene expression. *Proc Natl Acad Sci U S A* 99:12795–12800.
- Szallasi, Z, Liang, S. 1998. Modeling the normal and neoplastic cell cycle with "realistic Boolean genetic networks": their application for understanding carcinogenesis and assessing therapeutic strategies. *Pac Symp Biocomput* 66–76.
- Thakar, J, Piloni, M, Kirimanjeswara, G, Harvill, ET, Albert, R. 2007. Modeling Systems-Level Regulation of Host Immune Responses. *PLoS Comput Biol* 3:e109.
- Thieffry, D, Sanchez, L. 2002. Alternative epigenetic states understood in terms of specific regulatory structures. *Ann N Y Acad Sci* 981:135–153.

- Thieffry, D, Thomas, R. 1998. Qualitative analysis of gene networks. Pac Symp Biocomput 77–88.
- Thomas, R, D'Ari, R. 1990. Biological feedback. Boca Raton: CRC Press. p. 316 p.
- Tomlin, CJ, Axelrod, JD. 2007. Biology by numbers: mathematical modelling in developmental biology. Nat Rev Genet 8:331–340.
- Yuh, CH, Bolouri, H, Davidson, EH. 1998. Genomic *cis*-regulatory logic: experimental and computational analysis of a sea urchin gene. Science 279:1896–1902.
- Yuh, C-H, Bolouri, H, Bower, JM, Davidson, EH. 2001. A logical model of *cis*-regulatory control in a eukaryotic system. In: Bower, JM, Bolouri, H, editors. Computational modeling of genetic and biochemical networks. Cambridge, MA: MIT Press. p. 73–100.

CHAPTER 3

MODELING THE *DROSOPHILA* GAP GENE NETWORK

Abstract

The past several years have seen a flowering of modeling techniques applied to gene regulatory networks. The network of gap genes that pattern the early *Drosophila* embryo has received special attention from modelers because of the especially abundant experimental data available. A well-studied system like the gap gene network is an ideal context in which to compare recent modeling techniques, examine the benefits and drawbacks of each, and explore where our understanding could benefit from alternative methods. We discuss what has been learned about fly development, and about developmental modeling, from the models, and we synthesize and categorize the approaches that have been applied. Finally, we apply a model that has proved successful with other *Drosophila* segmentation genes to the gap genes, and place our findings in the context of other gap gene models.

Introduction

The gap genes make up a small network of transcription factors that define the broad domains of the anterior-posterior axis of the *Drosophila melanogaster* embryo. They respond to regulation by maternal factors and, crucially, to regulatory interactions within the gap gene network itself. Relying only on the initial patterning information of two opposing diffusion-mediated gradients of maternal gene products, the gap genes must "decide" among themselves where to establish non-overlapping domains of expression. The borders between these domains serve as positional markers for the next set of patterning genes, the pair-rule genes, which divide the

embryo into smaller partitions to guide the segment polarity genes and later development. This early organization of the embryo must be robust to substantial variation in embryo length, which can vary by 10-20% (Lott et al. 2007) and, presumably, to genetic variation in regulatory elements and the genes themselves.

Understanding of the gap gene network has benefited from a number of reviews of the experimental literature (Akam 1987; St Johnston and Nusslein-Volhard 1992; Pankratz and Jackle 1993; Rivera-Pomar and Jackle 1996; Niessing et al. 1997; Sanchez and Thieffry 2001). In brief: Bicoid and Nanos maternal gradients establish *hunchback* and *caudal* mRNA gradients, which in turn establish the expression domains of the other trunk gap genes (*Krüppel*, *knirps*, *giant*, *zygotic hunchback*) and indirectly modify those of the terminal gap genes including *tailless* and *huckebein*. The gap genes are transcription factors, and those in the trunk adjust the position and sharpness of their domains largely by mutual repression. Each gap gene has multiple transcriptional regulators that affect the position of its band(s) of expression along the anterior-posterior axis of the embryo. Their expression pattern serves as the template for subsequent stages of segmentation patterning.

Thousands of person-hours of experimental work have gone into establishing what we know about regulation of and by gap genes (beginning with the Nobel-Prize winning initial mutant screen (Nusslein-Volhard and Wieschaus 1980)), and this is one of the best-studied gene regulatory networks known in eukaryotes. However, dynamical patterns of any complexity are not easy to intuit from the individual interactions among the genes. The gap gene expression bands arise, adjust their positions, and then subside, over the course of just a few hours. The quest to understand how the genes cross-regulate to produce this dynamic result has given rise to many spatial models that abstract the gap genes in a variety of ways. The rich literature of gap gene models gives us an opportunity to explore what modeling

approaches can teach us about the dynamics of these genes in particular, and about gene regulatory networks in general.

Modeling the gap gene network

"The best material model for a cat is another [cat], or preferably the same cat," (Rosenblueth and Wiener 1945). This comment sums up many biologists' views of modeling genetic interactions. In order to make a model credible, we should include in it all the experimental detail we know. This "kitchen sink" approach is intuitively appealing, since it ties the model as firmly as possible to data, but it introduces practical problems. On a pragmatic level, we often have data for multiple temporal and spatial scales (for example, binding interactions occupy milliseconds, diffusion takes seconds, transcription and translation last minutes), and integrating these can be numerically difficult. On a heuristic level, a model that is almost as complex as the cat is nearly as hard to understand as the cat itself. On the other hand, an oversimplified model that leaves out the messy biological details risks ignoring critical elements. A useful model captures the essential features of a system, simplifying it enough to render it interpretable without sacrificing important interactions. The choice of simplifications is crucial to the success of the model.

Modelers have been attempting to achieve this balance of detail and abstraction with the *Drosophila* segmentation gene network (Thieffry and Sanchez 2003; Perkins 2007). In the process, they have found a number of important insights into the gap gene system. Because there is so much data on the interactions among these genes, and because the data are so varied (living and fixed-embryo expression patterns, time-series data, knockouts and other genetic perturbation experiments, computational and experimental discovery of binding sites, etc.), there have been many different

modeling approaches used for this network. Each approach employs a different choice of abstractions of the biology, and the choices are driven in part by the particular questions the researchers wished to ask about the system. We compare the techniques employed in gap gene models and the insights gained thereby. To complete the array of techniques, we also present a new model that uses some previously-successful modeling approaches that have not yet been employed for the gap gene network. We discuss which abstractions are most useful for which questions.

The models we consider here are those of the Reinitz group ((Jaeger et al. 2004a; Jaeger et al. 2004b; Perkins et al. 2006) and its 2009 gap gene papers in particular (Manu et al. 2009a; Manu et al. 2009b)), Alves & Dilão (Alves and Dilao 2006), and Sanchez & Thieffry (Sanchez and Thieffry 2001; Thieffry and Sanchez 2002), as well as our own model (Stockwell), described below.

Representations of mRNA and protein concentrations

The first choice a modeler must make is how realistically to treat the numbers of each kind of molecule in the system. Real cells contain finite (and often small) numbers of each species of mRNA and transcription factor, so one approach is a stochastic model of individual molecules and their interactions, perhaps using the Gillespie algorithm (1977). This is computationally expensive and probably provides more detail than the gap gene system requires. As a result, most modelers have chosen either a continuous approximation, where concentrations are represented on a real-number scale (Reinitz, Alves & Dilão, Stockwell) or a much simpler discrete one, where concentrations may assume one of a small number of integer values (Sanchez & Thieffry). Discrete models allow faster simulations, which means modelers can explore more parameter combinations and more mutant phenotypes. This tradeoff between realism and speed is generally resolved in favor of speed when modelers are

interested in the coarse-scale expression patterns and are using low-resolution data (Thieffry), and in favor of more realistic continuous models when they are fitting nucleus-scale expression data (Reinitz), attempting to estimate measurable parameters (Alves & Dilão, Stockwell), or measuring the effect of weak regulation (Reinitz, Stockwell).

The Sanchez & Thieffry model follows the formalism developed by René Thomas (Thomas and D'Ari 1990), which adapts a Boolean approach to accommodate more biological detail than the traditional single-threshold, on/off Boolean method. Gene products may have as many integer values as are required to fit the empirical data; in Sanchez & Thieffry, for example, Caudal activates the transcription of *knirps* if its concentration is greater than 1, and that of *giant* as well if its concentration is greater than 2. A task that arises with discrete-value models is deciding where to set the thresholds that determine what concentration of regulator A provokes a step up (or down) to the next concentration level for target gene B. These thresholds are considered parameters in the models, and are fitted to data. A given target gene may have multiple different thresholds with respect to different regulators. The gradual gradients of the morphogens Bicoid and Nanos are among the most important guides for where the gap gene bands arise according to their differing sensitivities, so this regulatory network offers particular challenges for discrete-value modelers. Perhaps for this reason, most gap gene modelers have elected to use real-valued molecular concentrations. The Reinitz group uses a neural-network approach, adding weighted real-valued concentrations of regulatory inputs and then applying a sigmoid function to determine whether the target will be transcribed (1) or not (0).

Boolean models, where molecules can be only present or absent (1/0), present a special set of problems. While being by far the fastest to solve with a computer, they contain no way to express the idea of different thresholds of activation for different

targets of a given regulator. That is, there is no way to capture, "A is present with respect to target B (sensitive target), but absent with respect to target C (less sensitive target)." The morphogen gradients of the gap gene system reveal a fundamental limitation of the Boolean approach that worked reasonably well for segment polarity genes (Albert and Othmer 2003; Irons and Monk 2007). Even the multi-level discrete models (Sanchez & Thieffry) require special finessing of which regions are considered to have Bicoid (for example) at "high," "medium," and "low" concentrations, but in the gap genes we encounter a system where the Boolean abstraction fails entirely.

Nonlinear interactions

The simplest way to represent how regulators combine to guide the expression of one gene is to add up their effects. One can weight the influence of particular regulators more strongly than that of others, and in fact these weights are the dominant parameters in models that use this summing-up, or linear, approach (Reinitz). Linear equations are dramatically faster for a computer to solve and much more tractable to analyze, even if the weighted sums of regulatory inputs is subsequently put through a sigmoid function akin to those used for modeling neurons (Reinitz). Decomposing the regulators that cause a particular behavior in the model is far more straightforward if the regulators are combined linearly (Manu et al. 2009b). It is not surprising, then, that models of gene regulatory networks often use linear equations (Reinitz, Alves & Dilão).

However, the speed and tractability come with some tradeoffs. There is no way to express how regulators interact with each other in the regulation of their target. The weighted-sum approach means that repressors have negative coefficients and activators have positive ones, and the sum of their effects is the regulation experience by the target. As a result, there is no convenient way to express, "A activates C, but

any amount of repressor B above a certain threshold will negate this effect, no matter how high the concentration of activator A," because the regulation of C is just the A minus B, perhaps with weight on each concentration. Similarly, "A up-regulates C, but only to the extent that cofactor B is also present" requires a nonlinear term for the interaction of A and B. Finally, a linear system cannot capture "A activates C's expression, and B interferes with this activation without affecting the concentration of A (e.g., allosterically)." There are linear approximations of these situations, but they are incomplete (Figure 3.1).

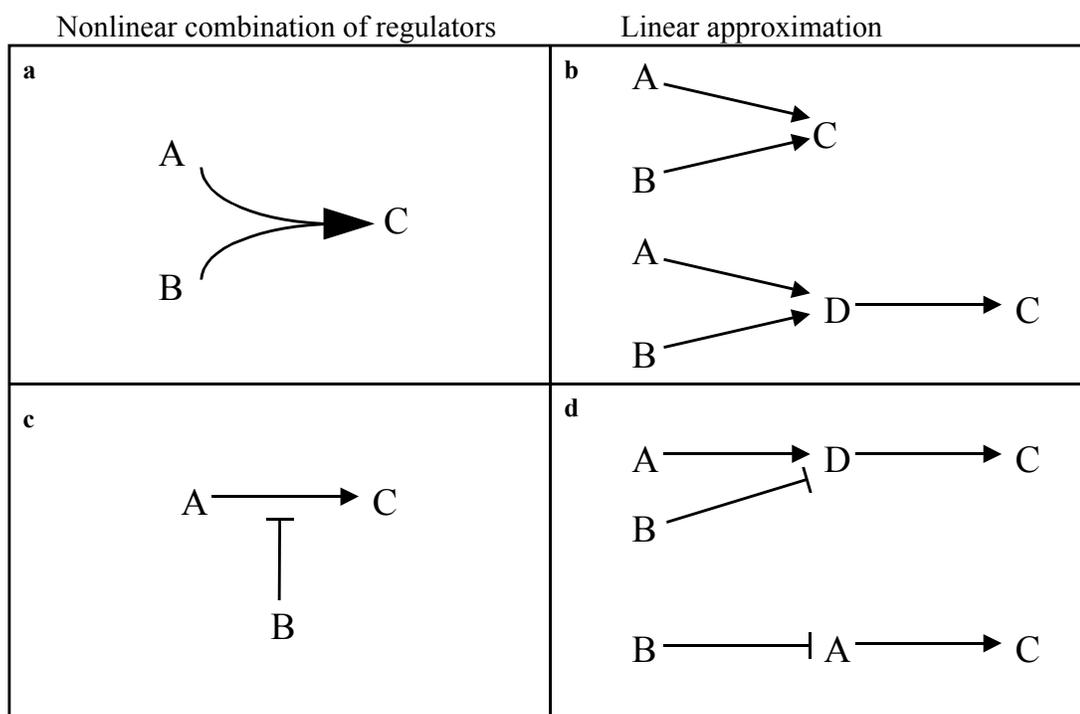


Figure 3.1. Regulators combining to regulate a target in ways that cannot be adequately represented by a linear summation of their effects. **a**, B acting as A's required cofactor. In the linear approximations of this scenario (**b**), A and B have independent effects on C, and a larger amount of A can compensate for absent B. **c**, a repressor (B) which negates the effect of activator A regardless of the concentration of A. **d**, introducing an intermediate node (D) distorts the timescale of the system; alternately, having B directly repress A changes the concentration of A, which was not the case in **c**.

One solution is the compromise adopted by Alves & Dilão, in which transcription (at a constant rate) takes place only when at least one activator is bound to the gene's promoter or enhancer, and no repressors are. This is a variant of the logical equations used by Sanchez & Thieffry. The Alves & Dilão model combines this logical rule with more traditional mass-action kinetics. The cis-regulatory regulatory region for each gene moves into and out of the active-transcription state at rates determined by binding coefficients for the regulatory proteins. The result is that while transcription is an all-or-nothing proposition, the *time* its regulators spend bound is real-valued, and the model produced is thus a set of coupled, linear differential equations (DEs). While the binding model is simple – one binding site for each regulator, independent of the others, with no cooperativity or interaction between co-regulators other than the Boolean rule stated above – the continuous dynamics yield smooth expression bands domains that are easier to compare to experimental data than the discrete sections of embryo employed by Sanchez & Thieffry.

If modelers are interested in the details of *how* regulators act upon their targets, rather than *whether* and *how much* they do, then more complicated rules are necessary for combining regulators that co-regulate a single target. Our model below follows the example of a mechanistic segment polarity network model (von Dassow et al. 2000) in using nonlinear DEs to encapsulate what is known about molecular interactions. For example, Bicoid and Krüppel often compete for binding to overlapping sites in the *giant* cis-regulatory region (Makeev et al. 2003), so we encoded Krüppel's influence as reducing Bicoid's effective concentration at the cis-regulatory sites of *giant*. We also included such details as the known differential regulation of the two Giant expression domains, and since *hunchback* is known to have two promoters with different transcriptional activators for each, we modeled the regulation of each promoter separately.

However, these kinds of nonlinear DEs are much slower to simulate than the neural-network style weighted sums of the Reinitz models or the even simpler first-order kinetics of Alves & Dilão. This fact limited the number of parameter sets we were able to explore, as well as the spatial and temporal detail of the data we were able to fit the model to. The Reinitz model is fitted to concentration measurements for each nucleus at nine 6.5-minute time intervals, and ours was fitted to continuous, averaged concentrations at a single timepoint (early cycle 14, when the gap genes reach their final pattern).

Steady state assumption

Some models require that the expression patterns come to a stable equilibrium, and throw out parameter sets where the equilibrium pattern fails to match this stable pattern (Sanchez & Thieffry, Alves & Dilão). This is potentially problematic because gap gene expression domains do not ever stop changing in the fly embryo. Unlike the expression stripes of later segment polarity genes, these subside and fade away as soon as their downstream targets have achieved the proper pattern. Of course, looking only at steady states offers a substantial increase in efficiency in the parameter-fitting process; modelers can automatically discard unstable parameter sets, and examine only those that reach equilibrium for a match to the correct expression pattern (Alves & Dilão, (Siegal and Bergman 2002)). However, it introduces a substantial artifact into the model when the network being represented is as dynamic as that of the gap genes. An alternate solution is to compare the model to a snapshot of the expression patterns at the most important stage of gap gene development, when their domains set the positions for the pair rule genes that follow (Stockwell) or at a series of such snapshots (Reinitz).

Biologically meaningful parameters

All of the models contain parameters that allow the equations to be fit to experimental data. Some mathematical representations include parameters that have measurable, biological analogs, such as binding affinity, binding cooperativity, or half-lives of molecules. This is useful because such a model, once fitted, offers measurable predictions. It has the additional benefit that parameter space sampling can be limited to known biological ranges of the parameters (von Dassow et al. 2000). Other models' parameters are more abstract, but allow more convenient or faster fitting (Reinitz). The choice of variable representation constrains the choice of parameters; for example, since the Reinitz models conflate the mRNA and protein products of a gene into a single variable, the parameter representing the half-life of the hybrid molecular concentration is more difficult to disentangle once fitted. The discrete thresholds in the Sanchez & Thieffry model are even more difficult to relate to measurable characteristics of real molecules. The Alves & Dilão model, like Reinitz and Sanchez & Thieffry, reduces mRNA and protein molecules into single variables except where required by the maternal mRNA contributions. This reduces the interpretability of the binding strength parameters they use in their kinetic equations, but such parameters do have biological meaning and can be compared to empirical measurements. For our model, we chose to separate mRNA and proteins into separate variables, and selected a formalism that yielded parameters grounded in the mechanics of molecular interactions.

An alternate model

The gap gene expression domains provide positional information for the pair-rule genes, which in turn demarcate the expression boundaries for the segment polarity

gene network. Von Dassow et al. (von Dassow et al. 2000; Meir et al. 2002; Von Dassow and Odell 2002) created a very successful model of the segment polarity network, in which they discovered that a surprisingly large proportion of parameter space produced a wild-type expression pattern in the model. They chose a continuous-variable, continuous-time mathematical representation with measurable, biologically meaningful parameters.

We applied this technique to the gap gene network, adapting the original ordinary DEs to the partial DEs required for a spatial model. The segment polarity genes settle into a steady expression pattern; their task is to record the earlier, transient signals in order to guide downstream developmental differentiation. The gap genes, however, do not achieve a steady state, so we eliminated the requirement for the model to arrive at an equilibrium expression level. We incorporated experimental findings about gap gene interactions to create a model that would complement the existing catalog of techniques (Table 3.1) and investigate whether alternate methods could shed light on the gap gene dynamic patterns.

We used nonlinear continuous partial differential equations to represent the regulation of and by the gap genes. These formalisms, as detailed in the segment polarity network model (von Dassow et al. 2000; Meir et al. 2002), offer a convenient way of encapsulating the kinds of complex cross-regulatory relationships detailed in Figure 3.1, while using parameters that correspond to measurable quantities: half-lives of mRNA and protein molecules (H), binding strengths (k), and binding cooperativity (n). We represented mRNA and protein concentrations as separate, continuous variables. Transcriptional regulation was modeled as sigmoid dose-response curves (composed Hill functions); translation rates were linear functions of mRNA concentrations. All molecular species had first-order decay, and diffusion through the syncytial blastoderm was held constant throughout the embryo. We fitted the model

to data published by the Reinitz group (Poustelnikova et al. 2004). More detail on the model is provided in the supporting information.

Table 3.1. A summary of gap gene models.

| Model | Representation of gap gene concentrations | Combining regulatory effects | Steady state assumption? | Experimentally measurable parameters? |
|--------------------|---|--|--|---|
| Sanchez & Thieffry | Discrete (mRNA and protein conflated; concentrations are 0, 1, 2, or 3) | Boolean logic functions, modified for discrete real-valued concentrations and thresholds | Yes | No (thresholds are fitted, but integer concentrations prevent relating the thresholds to measurable quantities) |
| Alves & Dilão | Continuous (mRNA and protein conflated) | Repressors trump activators completely; in absence of bound repressors, any bound activator suffices for maximal transcription | Yes | Yes (binding constants) |
| Reinitz | Continuous (mRNA and protein conflated) | Weighted sum of regulator concentrations (linear) | No; compared model to data at multiple time points | No (weight matrix) |
| Stockwell | Continuous (mRNA and protein separate variables) | Composed Hill functions approximating logical functions (AND, OR, NOT) and competition for binding sites | No; compared model at specified time point | Yes (decay rates, binding affinities, rates of reaction) |

We found a number of insights as we fitted the model to expression data from the FlyEx database (Kosman et al. 1998a; Kosman et al. 1998b; Surkova et al. 2008) (Figure 3.2). First, we discovered that realistically steep curves at the boundaries of expression domains required short half-lives for both the mRNA and protein molecules. With longer half-lives, diffusion spreads the protein out in a shallower gradient. Gap genes whose half-lives have been measured experimentally have turned out to agree with our model in this regard. Hunchback protein is estimated to have a brief half-life of 30-40 minutes (Hülkamp et al. 1994), and its mRNA is short-lived as

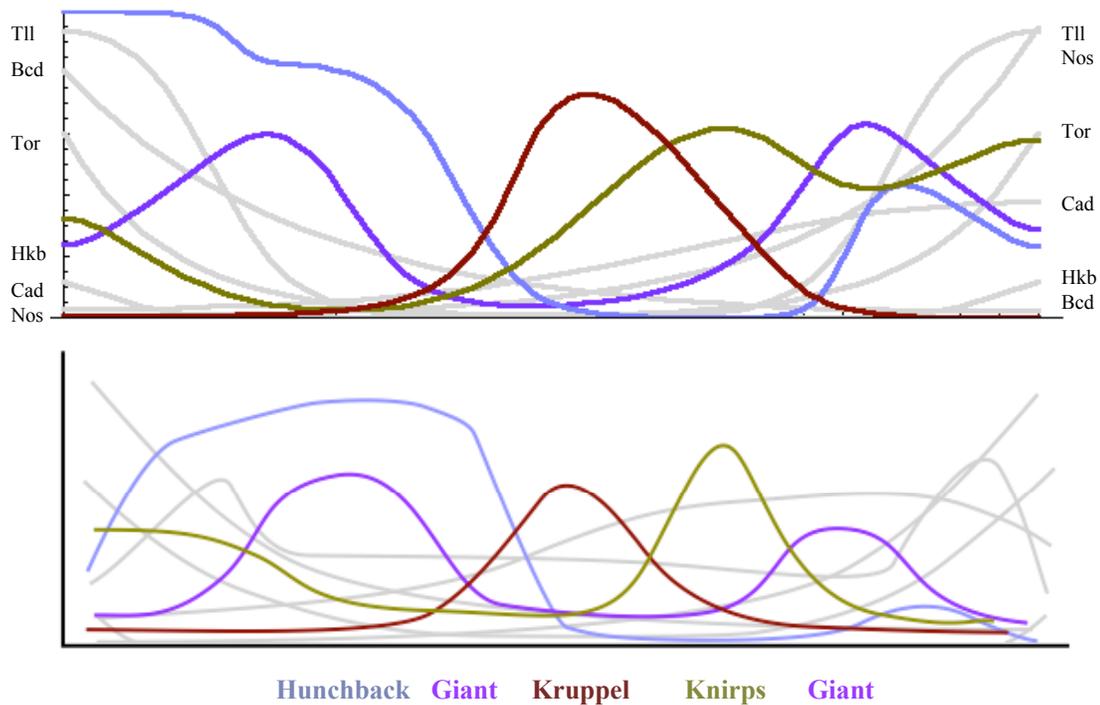


Figure 3.2. Comparing protein concentrations in our model (above) with smoothed protein concentrations measured in fixed embryos (below). Trunk gap gene proteins (fitted variables) are in color. Maternal morphogens and terminal gap genes (inputs to the model) are in gray, labeled at right and left. Anterior end is at left. Fixed embryo data is adapted from plots in the FlyEx database (Kosman et al. 1998a; Kosman et al. 1998b; Surkova et al. 2008).

well (Hülkamp et al. 1994; Grosskortenhaus et al. 2005), with a half-life as short as 6-9 minutes (Weir et al. 1988). The protein Krüppel also decays quickly and, like Hunchback, seems to be actively degraded (Jacob et al. 1991; Grosskortenhaus et al. 2005) – perhaps in order to achieve the much-discussed steep posterior border of its anterior expression domain. *Krüppel* mRNA has a half-life of less than 10 minutes (Weir et al. 1988). Downstream segmentation gene products also tend to have short half-lives, perhaps for similar reasons. For example, the pair-rule gene *fushi tarazu* is well-known for having the impressively small half-life of about 10 minutes for both its mRNA and protein molecules (Edgar et al. 1986; Kellerman et al. 1990) and

engrailed, *even-skipped*, *hairy*, and *runt* mRNAs are similar (Weir et al. 1988). The decay rates of the other gap gene products remain to be measured.

It is, perhaps, unsurprising that the half-lives of the RNAs and proteins were one of the primary parameters we needed to tune in our model to achieve a fit with experimental data. It appears that evolution has been doing something similar: Gregor et al. (Gregor et al. 2005) finds that the half-life of Bicoid has changed to achieve the proper morphogen gradient in eggs of different lengths in different Dipteran species.

A slower diffusion rate would also have helped sharpen the boundaries of the expression domains in our model. However, the global diffusion constant is constrained to a certain value because it is the dominant determinant of Bicoid and Nanos gradients; if we fit those to experimental measurements, we thereby fix the diffusion constant. However, since the gap genes are transcription factors, they may spend substantial time sequestered in the nucleus and not diffusing substantially. Also, recent findings have suggested that *bicoid* mRNA plays a role in forming the exponential Bicoid protein gradient. The exact nature of Bicoid gradient formation is an area of active research (Gregor et al. 2005; Coppey et al. 2007; Gregor et al. 2007a; Gregor et al. 2007b; Lipshitz 2009).

We also found that the only way to allow the gap genes to effectively "negotiate" the boundaries between their expression bands, without damping each other's expression entirely, was to keep the mutual repression fairly weak. This is especially true for genes like *Krüppel*, which express bands in the middle of the embryo where Bicoid, its main activator, is at a relatively low concentration. Even nearer the anterior end of the embryo, however, repression between gap genes had to be mild (large "*k*" parameters) to generate the correct stripe pattern. In general, the mutual repression worked best when its primary role was to sharpen the edges of the

bands, whose position was mainly determined by how strongly they responded to levels of transcriptional activators.

We were surprised by this conclusion. Since adjacent gap gene expression domains have limited overlap (Kraut and Levine 1991), we had hypothesized that they repressed each other strongly. However, diffusion in the syncytial blastoderm means that small amounts of each protein constantly seep out from its expression domain into that of its neighbors. If mutual repression is made strong enough that "jostling" among domains is the main way they establish their position, then the seepage from each domain damps transcription in the neighboring domains to the point that all the gap genes quickly subside into non-expression. Such non-intuitive insights illustrate the value of constructing a dynamic model.

We fitted our model to the data by iterative testing and simulation, as did Sanchez & Thieffry and Alves & Dilão. The Reinitz model was fitted via nonlinear optimization methods (Jaeger et al. 2004a) but even that automated process produced a large number of good fits, which the researchers had to hand-check for biological plausibility (Manu et al. 2009b). Exploring parameter space by hand is a useful process for the modeler, however, because it provides an intuition for the parameter landscape and how it affects the model. In our case, we found that identifying a parameter set that provided a good fit with the data was not difficult, and that nearby parameter choices produced nearly-as-good fits. This suggests that, as with the earlier version of this modeling approach (von Dassow et al. 2000), a relatively large proportion of parameter sets produced the correct pattern. The gap gene network, like the segment polarity network, appears to be relatively robust to parameter variation. Experimental evidence supports this finding; indeed, the positional patterning system is robust to even extreme insults like maintaining the two halves of the embryo at significantly different temperatures (Lucchetta et al. 2005).

We undoubtedly left details of the transcriptional regulation out of our model; some that we were unaware of, and some that are not yet known. Despite this, however, the model achieved a respectable approximation of the data. This suggests that the biological network itself may be able to accommodate a fair amount of regulatory “sloppiness” in its interactions. Indeed, some tolerance is required for the variation that allows evolvability; and in fact one of the cornerstones of the patterning system, Bicoid, is a recent invention, confined to higher Dipterans (Sommer and Tautz 1991; Schröder and Sander 1993; Stauber et al. 1999; Wimmer et al. 2000). If the gap gene network is flexible and robust enough accommodate the replacement of one of its most fundamental steps, perhaps that helps explain why so many different modeling incarnations of the network can all achieve an approximation of its behavior.

Questions and insights

Any model is useful only insofar as it allows us to ask questions and generate testable hypotheses about the system it represents. Different choices of abstractions enable the investigation of different questions. For our model, we began with the regulatory relationships that we could infer from the experimental literature, and then chose a formalism that encoded these mechanistic details in terms of parameters that can be tested empirically. The Reinitz model took the reverse approach, fitting a less mechanistic but more computationally tractable model to infer the regulatory rules. By fitting the model to much more detailed data (generated in their own lab), they were able to ask questions that depended on very fine-scale qualities of gap gene regulation: how far the precision of the Bicoid gradient could specify the position of the gap gene domains, and what other mechanisms were involved in reducing the embryo-to-embryo variation of the expression band positions. By doing so, they were

able to help resolve longstanding questions about the causes of developmental canalization. The Sanchez & Thieffry model was one of the first models of gap gene dynamics, and it showed that the published interactions were largely sufficient to generate the patterns observed in wild-type and mutant embryos. Their approach has the advantage of requiring the least detailed knowledge of the system, and could offer an approach for researchers interested in modeling gene networks where our understanding is less complete than for the *Drosophila* segmentation genes. For such networks, the Alves & Dilão method is also a viable alternative. Like the discrete logical method, it requires only the network topology (A represses B, C activates D) and can test the sufficiency of known interactions to create the expression patterns observed. Unlike that approach, it offers fitted parameters that can be tested via experiments.

All models, except the most automated, prompt questions of those who construct them: What did we notice about this system as we wrote down explicit equations that force us to specify how we think it works, and when we chose and fit its parameters? What assumptions did we find we had to make? What simplifications were necessary, and what gaps in our knowledge did we identify? The process of making the model and comparing it to experimental data is generally as informative as the answers to any hypotheses the modeler specifically set out to address. For less well-known gene networks, simpler models firmly based on experimental observations, with a set of parameters small enough to be fit by trial and error, can lend more easily interpretable insights into what we think we know (and what we find we don't know) about the network being modeled.

Much of the research on the gap gene system in the past few years has focused, like the most recent Reinitz models (Manu et al. 2009a; Manu et al. 2009b), on the details of variability and precision in the patterning mechanism (Houchmandzadeh et

al. 2002; Houchmandzadeh et al. 2005; Gregor et al. 2007a; Gregor et al. 2007b).

This kind of work is only possible with a gene network where nearly all the genes are known and their behavior characterized, and (equally importantly) where experimental techniques and mutant lines have been well-developed. Most gene networks are much less completely known, and for these, simpler modeling approaches with more general questions are appropriate. With the array of techniques that have been developed to deepen our understanding of the gap gene system, future modelers will have a well-equipped tool-chest with which to examine the inner workings of the networks we do not yet understand.

APPENDIX

A3.1. A new model

Introduction

Much work has been done to reveal individual interactions within the gap gene network. While some of the regulatory mechanisms are still unclear, the gap genes represent one of the best-characterized eukaryotic transcription factor networks. However, dynamical patterns of any complexity are not easy to intuit from known rules. A well-known network is not guaranteed to include all the interactions required to reliably produce a dynamic phenotype. When von Dassow et al. (2000) modeled the segment polarity network, they found that robustly recreating the correct pattern required two missing interactions. To investigate the completeness of our understanding of the gap genes in a similar way, we created a mathematical model that captured what is known about gap gene regulation, and asked whether it was sufficient to generate the patterns we see in real embryos.

We formalized the known interactions in modular partial differential equations (PDEs) representing the regulatory pressures on each gap gene, and fitted parameters in the model to recreate the evolution of the gap gene bands as they arise over the course of several mitotic cycles. We fitted the model to experimental data from the FlyEx database (Kosman et al. 1998a; Kosman et al. 1998b; Surkova et al. 2008) at late cycle 13/early cycle 14, when the pair rule genes solidify their pattern based on the gap gene domains (Foe et al. 1993). We did not fit the transient expression levels before this time, but we found that the time period between the beginning of gap gene transcription and the end of their influence is short enough that the transients cannot vary too much if the early cycle-14 expression pattern is pinned to experimental data.

The gap gene bands must arise, reposition, and sharpen in time for the pair-rule genes to "read out" their position, and not much time is left for other changes in expression.

Methods

Drosophila melanogaster embryos develop with an unusually simple geometry in the early stages, providing an especially suitable system for spatial modeling. Until gastrulation begins in the 14th cell cycle, the embryo is a roughly radially symmetric ovoid, with bands of gene expression dividing it up along the anterior-posterior axis. Because of the symmetry, we can capture the banding patterns by modeling different protein concentrations along a single line running the length of the embryo, from future head to future abdomen. (All models reviewed here make this simplification.) No cell membranes form between the dividing nuclei while the gap gene bands are forming, so fluids and proteins may be presumed to diffuse relatively freely within the embryo. We allowed proteins to diffuse, but not mRNAs, because they were presumed to be confined to the nuclei.

We used the mathematical formalism presented in a segment polarity model (von Dassow et al. 2000; Meir et al. 2002), adapted via PDEs for a spatially distributed system. When we lacked specific experimental information about how different proteins interacted to regulate a single gap gene, we used these default rules:

- Two or more activators are "OR'd" together. That is: Either is sufficient to bring about the maximal transcription its efficiency parameter permits (see below). We included a parameter for weighting the activators relative to each other, but did not need to use it to fit the model to the data. See function "phiphiTwoGlobalActivators" in the Mathematica code in A3.2, adapted from the equation in Figures A4-C in (Meir et al. 2002).

- Two or more repressors are "OR'd" together in a similar way; either is sufficient to bring about complete transcriptional repression. This is achieved by multiplying psi (repressor) terms together; see Mathematica code in A3.2.
- An activator and repressor are assumed to interact by the repressor competing the activator off its binding site. See function “*phipsiCompet*” in the Mathematica code in A3.2; adapted from eq. A17 in (Meir et al. 2002).

Gap genes included in the model:

hunchback (hb) (zygotic)

Krüppel (Kr)

knirps (kni)

giant (gt)

tailless (tll) (as an input to the other zygotic gap genes)

huckebein (hkb) (as an input to the other zygotic gap genes)

Maternal effect genes included in the model, as inputs:

bicoid (bcd)

hunchback (hb) (maternal)

caudal (cad)

nanos (nos)

torso (tor)

Some model details:

State variables represented the concentrations of the various molecular species in the model. We modeled mRNA and protein as separate variables because of previous work showing that conflating the two products of a single gene reduces the

robustness of the network (Odell and von Dassow 2003). Proteins were synthesized at a rate proportional to the concentration of the relevant mRNA. mRNA synthesis depended on the concentrations of known transcriptional regulators of the gene (details below), as well as on protein products that specifically target the mRNA for degradation where there was experimental evidence of this. All molecular species also had first-order decay terms.

We chose Hill functions as the building block of regulatory terms in our model because they saturate in a biologically realistic way, and because (unlike some mathematical representations of similar sigmoid curves) the parameters have readily understandable and measurable biological analogs.

Parameters

The Hill coefficient, n , is the cooperativity of the regulatory molecule. An example of cooperativity is if, when an activator binds, the binding increases the chances of a second molecule of that activator binding; that is, it helps recruit others of its kind. The extent of this kind of self-reinforcing effect corresponds to the steepness of the curve of the Hill function. Cooperativity can also be increased by indirect molecular interactions that cause regulators to bind at a rate that is more than a linear function of their concentration.

The half-maximal value, k , represents the binding affinity of a regulator to an enhancer. A Hill function is at half its maximal value (1, in these normalized and nondimensionalized equations) when the regulator variable has the value k .

The half-life of a molecule, whether mRNA or protein, was denoted by the parameter H .

All three of these parameters provide different ways of changing the sensitivity of transcription to the concentration of the regulator.

Regulatory interactions

The background conditions (Akam 1987; Gaul and Jackle 1990; Pankratz and Jackle 1993; Rivera-Pomar and Jackle 1996; Thieffry and Sanchez 2003): The egg's initial polarity derives from maternally deposited *bicoid* mRNA anchored at the anterior tip of the embryo, and *nanos* mRNA deposited at the posterior. There are also maternally provided uniform distributions of *caudal* and *hunchback* mRNA throughout the embryo. The egg begins translation of the maternal-effect mRNAs at fertilization. The stationary source of Bicoid protein diffuses through the embryo toward the posterior end, producing an exponential gradient of Bicoid with its maximum at the anterior end. (Modifications to this pure-diffusion model have appeared recently (Gregor et al. 2007a; Gregor et al. 2007b) but the final answer is not clear (Reinitz 2007; Lipshitz 2009) so we have used the simple diffusion approach.) Similarly, Nanos protein diffuses to form an opposing gradient. Since these two gradients hold their positions while the gap gene bands arise, we modeled them as establishing quickly and then remaining stationary during the gap gene band development.

Bicoid protein, once present, inhibits *caudal* mRNA translation. The Bicoid gradient thus produces a Caudal protein gradient with its maximum at the posterior end, somewhat mimicking the Nanos gradient. Newly translated Nanos protein (with help from Pumilio, implicit in our model) prevents translation of *hunchback* mRNA, creating an anterior region of Hunchback protein that ends, fairly abruptly, at the middle of the embryo. Bicoid contributes to the shape of this Hunchback expression pattern later in development by up-regulating zygotic *hunchback* transcription in the anterior.

Together, these four maternal effect genes establish the initial anterior-posterior polarity of the egg. The gap genes, including zygotic *hunchback*, first have detectable transcripts at mitotic cycle 10 (Foe et al. 1993) and "read out" the positional information contained in these gradients. The regulation from the maternal effect genes, and the inter-regulation between the gap genes, gives rise to the dynamic patterns that determine the patterning of the embryo.

We modeled only the main section of the embryo where the gap gene bands form, neglecting the anterior and posterior tips where a number of other "terminal" genes are expressed to begin the patterning of the head (*orthodenticle*, *buttonhead*, and *empty spiracles*) and extreme posterior. (All models reviewed here also restricted their analysis to the trunk gap genes.) We included the terminal genes *tailless* and *huckebein* as independent variables, helping regulate the other gap genes but not being regulated themselves, to provide a simplified but reasonably realistic environment for the gap genes near the tips. Since Torso's signaling cascade results in the activation of both these genes (Lu et al. 1993), we set up Tailless and Huckebein gradients by placing artificial Torso gradients at the anterior and posterior ends of the embryo, and having this protein alone control *tailless* and *huckebein* transcription (and nothing else).

We assume that all regulation is direct unless there is experimental evidence otherwise. We assumed that the following quantities were constant throughout the time covered by the model (through cycle 14A): temperature, diffusion constants, volume and length of the embryo, physical configuration (i.e., no gastrulation or other rearranging of cells), absence of cell walls, distance between nuclei, and the availability of general transcription factors.

Most of the regulation in the model was transcriptional, because the gap genes and most maternal effect genes are transcription factors. For each mRNA's PDE, we

included terms to represent its known transcriptional regulators in as mechanistically faithful a way as possible. We gathered the facts we could find in the literature to compile a list of each gap gene's regulators. For example, for *Krüppel*, we began with activation by Bicoid, the main activator of anterior and central gap genes. Then for each of *Krüppel*'s other regulators, we decided which was the most biologically realistic of the parameters for the regulator to affect. For instance, Knirps competes with Bicoid to reduce the number of Bicoid binding sites, so we modeled Knirps as reducing n , the Hill coefficient of Bicoid's activation of *Krüppel* transcription. In this function, greater Knirps concentration lowers n .

We combined the transcriptional regulators for each gap gene, added the diffusion process and (unless otherwise regulated) linear functions for translation and decay, and tested the resulting equations for each individual gene to make sure the behavior was biologically reasonable. Then we assembled all the genes into one network (a system of 19 PDEs) and tuned the parameters to get lifelike behavior.

Since mitotic cycle 13/early cycle 14 is when the pair-rule genes form stripes based on gap gene expression patterns (and cycle 14 is when cellularization begins, and the embryo departs from the free diffusion we assumed for the syncytial blastoderm) (Foe et al. 1993), we fitted our model to the gap gene expression measurements from that stage of development.

Details of the transcriptional regulation of each gene

The regulatory details of these interactions, while well studied, are not all completely understood. We applied the general rule that all gap genes repress one another generically, except where we had more detailed information. We gathered the facts we could find about how each gene is regulated, from genetic experiments (mutant phenotypes), molecular data about regulatory binding sequences (enhancers

and promoters) and computational models extrapolating binding predictions from regulatory binding sequences.

Giant

Giant has different regulatory influences for its anterior and posterior expression domains. To represent this, we modeled *giant* as being capable of maximal transcription when activated by either Caudal (the posterior activator, (Rivera-Pomar et al. 1995)) or Bicoid, the anterior activator (Eldon and Pirrotta 1991). Bicoid's effective concentration was mediated by competition from Krüppel (Makeev et al. 2003) and Hunchback, both repressors of *giant* (Eldon and Pirrotta 1991; Kraut and Levine 1991b). Knirps, Krüppel (Kraut and Levine 1991a), Hucklebein (Eldon and Pirrotta 1991), and Tailless (Kraut and Levine 1991a) are general repressors.

Knirps

The activators modeled for *knirps* were Bicoid (Bate and Martinez 1993) and Caudal (Rivera-Pomar et al. 1995). We modeled the repressors Krüppel and Hunchback (Hülkamp et al. 1990; Kraut and Levine 1991a) as competing for Bicoid with binding sites (Makeev et al. 2003). General repressors in the model were Giant (Eldon and Pirrotta 1991), Hucklebein, and Tailless.

Krüppel

The gene *Krüppel* has especially complicated transcriptional regulation. Maternal Bicoid activates its transcription (Bate and Martinez 1993), but Knirps acts to slightly reduce Bicoid's ability to recruit other Bicoid proteins to the binding site (Hoch et al. 1992). We therefore modeled Knirps as acting to lower the Hill coefficient of Bicoid's activation of *Krüppel* transcription. Additionally, Hunchback and Giant are

repressors that can interact to regulate *Krüppel*. Intermediate levels of Hunchback can amplify *Krüppel* transcription, perhaps indirectly, while high levels of Hunchback repress *Krüppel*. We modeled this in the following way: Hunchback is a repressor of *Krüppel*, but a less effective one when bound than Giant is. Hunchback competes with Giant for binding sites, and thereby reduces the effective concentration of Giant at the enhancer. This approach allows Hunchback to retain its identity as a repressor, but still have the mild activator effect on *Krüppel* that has been documented. Tailless is an additional repressor for *Krüppel*; like Knirps, it acts by competing with the activator Bicoid for binding sites (Hoch et al. 1992).

Hunchback

Hunchback protein in the embryo comes from translation of both maternal *hunchback* RNA, deposited in the egg, and of newly transcribed zygotic *hunchback* RNA. We treat these two protein sources separately in the model. Maternal *hunchback* RNA begins uniformly distributed throughout the embryo, and begins to be degraded by Nanos (Pumilio (Murata and Wharton 1995) is implicit) as the model run begins. Zygotic *hunchback* RNA begins the model run with a concentration of 0, like other zygotic gap gene products.

Zygotic *hunchback* has two well-studied promoters, governed by different combinatorial regulatory logic. We treat these separately.

Promoter 1 transcription:

This promoter does not respond to Bicoid protein, but does respond to Hunchback protein itself (Hülkamp et al. 1994; Wimmer et al. 2000). The parasegment 4 stripe, the posterior cap, and to some extent the anterior part of the posterior cap are the

product of this promoter (Margolis et al. 1995). We modeled Hunchback (Wimmer et al. 2000) and Tailless (Margolis et al. 1995) as independent activators of P1 zygotic *hunchback*. Repressors were the global inhibitors Krüppel (Clyde et al. 2003), Knirps (Clyde et al. 2003), Giant, and Hucklebein (Margolis et al. 1995).

Promoter 2 transcription:

This promoter, and its associated enhancer, respond to Bicoid protein as an activator (Margolis et al. 1995), (Driever and Nusslein-Volhard 1989), and correspondingly the expression domains of P2 *hunchback* (the anterior cap) form earlier than those of P1. We modeled the activator Tailless (Margolis et al. 1995) and Bicoid as activators competing for the same binding sites. Since Bicoid can activate *hunchback* P2 in the absence of Hunchback and Tailless, we chose a mathematical function encapsulating the idea that Hunchback helps Bicoid's transcriptional activation of the P2 promoter (by lowering the k parameter), but that Bicoid can also function as a lower-efficiency independent activator when Hunchback is absent. Repressors are Krüppel (Clyde et al. 2003), Knirps (Clyde et al. 2003), Giant, and Hucklebein (Margolis et al. 1995). The repressor Krüppel and Bicoid share binding sites (Makeev et al. 2003), so we modeled Krüppel's repression mechanism as competing the activator Bicoid off its binding sites.

Results

We were able to achieve a good fit with published experimental data (Figure 3.2, main text). We report the fitted half-life parameter values (H) in Table A3.1. See Mathematica code for k parameter values ("thresh" variables).

Table A3.1. Fitted half-life parameters from the model.

| Molecule | Fitted half-life parameter |
|------------------------------------|----------------------------|
| <i>giant</i> mRNA | 20 |
| Giant protein | 20 |
| <i>hunchback</i> mRNA (promoter 1) | 10 |
| <i>hunchback</i> mRNA (promoter 2) | 10 |
| Hunchback protein | 2 |
| <i>knirps</i> mRNA | 20 |
| Knirps protein | 20 |
| <i>Krüppel</i> mRNA | 20 |
| Krüppel protein | 20 |

Conclusions

A careful mathematization of what is known about genetic and molecular data can produce a model that produces lifelike behavior. Unlike the segment polarity network (von Dassow et al. 2000; Von Dassow and Odell 2002), the published interactions between maternal morphogens and gap genes appear sufficient to produce observed embryonic patterns, at least at the stage (early cycle 14) that we examined.

See the main text for observations from the parameter fitting.

A3.2. Mathematica code

```
Off[General::spell];
Off[General::spell1];

(* General-use functions *)
Clear[phi, psi, phipsiCompet, phipsipsiCompet, phiphiTwoGlobalActivators];

(*Order of arguments: regulator molecule, threshold (K), nu *)
(* Basic "activator" function *)
phi[x_, k_, n_] := x^n / (k^n + x^n);

(* Basic "inhibitor" function *)
psi[x_, k_, n_] := 1 - (x^n / (k^n + x^n));

(* Simplest case of an activator and an inhibitor competing for
the same binding site. From eq. A17 in appendix of Ingeneue paper. *)
phipsiCompet[activ_, threshA_, nuA_, inhib_, threshI_, nuI_] := (phi[activ, threshA, nuA])
(1 - phi[inhib, threshI, nuI]) / (1 - (phi[activ, threshA, nuA]) (phi[inhib, threshI, nuI]));

phipsipsiCompet[activ_, threshA_, nuA_, inhib_, threshI_, nuI_, inhib2_, threshI2_, nuI2_] :=
(phi[activ, threshA, nuA]) (1 - phi[inhib, threshI, nuI]) (1 - phi[inhib2, threshI2, nuI2]) /
(1 - (phi[activ, threshA, nuA]) (phi[inhib, threshI, nuI]) (phi[inhib2, threshI2, nuI2]));

(* This is how to "OR" together two global activators,
a la Fig.A4-C. Good default option when you don't know how they
interact with each other or anything else. activA is the stronger one,
and weightB is the maximum transcription that B can manage on
its own (0 < weightB ≤ 1). To weight them equally,
make weightB=1. You could add a third activator by adding another term like B's. *)
phiphiTwoGlobalActivators[activA_, threshA_, nuA_, activB_, threshB_, nuB_, weightB_] :=
(1 - (1 - phi[activA, threshA, nuA]) (1 - weightB * phi[activB, threshB, nuB]));

(*Parameters, globally defined*)

Clear[elen, tend];
elen = 500;
tend = 200;

(*Bicoid protein:*)
Clear[hBCD, BCDSynthRate];
hBCD = 50;
BCDSynthRate = 2;

(*Nanos protein:*)
Clear[hNOS, NOSSynthRate];
hNOS = 30;
NOSSynthRate = 3;

(* For maternal hunchback mRNA *)
Clear[hhbMat, maternalhbrnaconc];
maternalhbrnaconc = .3;
```

```

hhbMat = 30;

(* caudal mRNA *)
cadrnaconc = .8;
hcad = 300;
ThreshBCDcad = .05;
nuBCDcad = 3;
hCAD = 30;

(* torso protein *)
Clear [TORASynthRate, hTORA]
TORASynthRate = 1.1;
hTORA = 30;

(* Tailless protein *)
Clear[hTLL];
hTLL = 2;

(* Huckebein protein *)
Clear [hHCK];
hHCK = 2;

(* tailless transcription *)
Clear [threshTORAt11, nuTORAt11, ht11]
threshTORAt11 = .15;
nuTORAt11 = 5;
ht11 = 20;

(* huckebein transcription *)
Clear [threshTORAhck, nuTORAhck, hhck]
threshTORAhck = .7;
nuTORAhck = 5;
hhck = 2;

(* Used in Giant transcription *)
Clear [threshHBgt, nuHBgt, threshBCDgt, nuBCDgt, threshKRgt, nuKRgt, threshKNgt,
      nuKNgt, threshTLLgt, nuTLLgt, threshHCKgt, nuHCKgt, threshCADgt, nuCADgt];
threshHBgt = 4;
nuHBgt = 1;
threshBCDgt = .2;
nuBCDgt = 5;
threshKRgt = .04;
nuKRgt = 5;
threshKNgt = 30;
nuKNgt = 1;
threshTLLgt = .7;

```

```

nuTLLgt = 2;
threshHCKgt = .1;
nuHCKgt = 5;
threshCADgt = .1;
nuCADgt = 5;

(*Used in zygotic P1 Hunchback transcription:*)
Clear[hhbP1Zyg, threshHBhb1, nuHBhb1, threshHCKhb1, nuHCKhb1, threshGThb1,
      nuGThb1, threshKNhb1, nuKNhb1, threshKRhb1, nuKRhb1, threshTLLhb1, nuTLLhb1];
hhbP1Zyg = 10; (*half-life of P1 mRNA*)
threshHBhb1 = 2;
nuHBhb1 = 1;
threshHCKhb1 = 20;
nuHCKhb1 = 3;
threshTLLhb1 = .3;
nuTLLhb1 = 5;
threshKNhb1 = 20;
nuKNhb1 = 3;
threshKRhb1 = 20;
nuKRhb1 = 2;
threshGThb1 = 20;
nuGThb1 = 2;
(*hhbP1Zyg = 10; (*half-life of P1 mRNA*)
threshHBhb1=.2;
nuHBhb1=5;
threshHCKhb1=.8;
nuHCKhb1=3;
threshTLLhb1=.3;
nuTLLhb1=5;
threshKNhb1=.7;
nuKNhb1=3;
threshKRhb1=.6;
nuKRhb1=2;
threshGThb1=.9;
nuGThb1=2;*)

(*Used in zygotic P2 Hunchback transcription:*)
(* We'll set some of these to the same values as for P1 *)
Clear[hhbP2Zyg, epsilonHBBCDhb2, threshBCDhb2, nuBCDhb2,
      threshGThb2, nuGThb2, threshHBhb2, nuHBhb2, threshHCKhb2, nuHCKhb2,
      threshKNhb2, nuKNhb2, threshKRhb2, nuKRhb2, threshTLLhb2, nuTLLhb2];
hhbP2Zyg = 10; (* half-life of P2 hunchback mRNA*)
epsilonHBBCDhb2 = .75;
(* max extent to which hb can lower bcd's K for binding to activation site*)
threshBCDhb2 = .14;
nuBCDhb2 = 6;
threshGThb2 = threshGThb1;
nuGThb2 = nuGThb1;
threshHBhb2 = 3;
nuHBhb2 = 4;
threshHCKhb2 = threshHCKhb1;

```

```

nuHCKhb2 = nuHCKhb1;
threshKNhb2 = threshKNhb1;
nuKNhb2 = nuKNhb1;
threshKRhb2 = .5;
nuKRhb2 = 1;
threshTLLhb2 = .5;
nuTLLhb2 = 3;

(* Used in hunchback translation *)
Clear[threshNOSHB, nuNOSHB];
threshNOSHB = .5;
nuNOSHB = 2;

(* Used in Knirps transcription *)
Clear[threshHBkn, nuHBkn, threshBCDkn, nuBCDkn, threshKRkn, nuKRkn, threshGTkn,
nuGTkn, threshTLLkn, nuTLLkn, threshHCKkn, nuHCKkn, threshCADkn, nuCADkn];
threshHBkn = 3;
nuHBkn = 1;
threshBCDkn = .6;
nuBCDkn = 4;
threshKRkn = .1;
nuKRkn = 6;
threshGTkn = .6;
nuGTkn = 2;
threshTLLkn = 3;
nuTLLkn = 2;
threshHCKkn = 3;
nuHCKkn = 2;
threshCADkn = .2;
nuCADkn = 4;

(* Used in Kruppel transcription affector *)
Clear[threshBCDkr, nuBCDkr, threshGTkr, nuGTkr, threshHBkr,
nuHBkr, threshKNkr, nuKNkr, threshTLLkr, nuTLLkr, limitHBrepressionKr];
threshBCDkr = .05;
nuBCDkr = 5;
threshGTkr = 2;
nuGTkr = 4;
threshHBonGTkr = .1;
nuHBonGTkr = 5;
threshHBkr = .1;
nuHBkr = 6;
threshKNkr = 30;
nuKNkr = 1; (* because there's only 1 Knirps binding site *)
threshTLLkr = .08;
nuTLLkr = 5;
limitHBrepressionKr = .99; (*this should be <1*)
(* Make sure threshGTkr>threshHBkr so Giant is a weaker binder than Hunchback *)

```

```

(* Used in giant initial/boundary conditions and governing equations *)
Clear[tranGT, hGT, hgt];
tranGT = 1;
hGT = 20;
hgt = 20;

(* Used in hunchback initial/boundary conditions and governing equations*)
Clear[hHB];
hHB = 2;

(* Used in knirps initial/boundary conditions and governing equations *)
Clear[tranKN, hKN, hkn];
tranKN = 1;
hKN = 20;
hkn = 20;

(* Used in kruppel protein initial/boundary conditions and governing equations *)
Clear[tranKR, hKR, hkr];
tranKR = 1;
hKR = 20;
hkr = 20;

(* Diffusivity constants *)
Clear[stdPROTdiffusivity, DifKR, DifBCD, DifHB, DifKN, DifGT, DifNOS];
stdPROTdiffusivity = 30;
stdHIPROTdiffusivity = 300;
DifKR = stdPROTdiffusivity;
DifBCD = 300;
DifHB = stdPROTdiffusivity;
DifKN = stdPROTdiffusivity;
DifGT = stdPROTdiffusivity;
DifNOS = stdHIPROTdiffusivity;
DifCAD = stdHIPROTdiffusivity;
DifHCK = 250;
DifTLL = stdHIPROTdiffusivity;
DifTORA = 100;

txnGtl[Kr_, Bcd_, Kn_, Hb_, Tll_, Hck_, Cad_] :=
Block[
{BcdKrHbComp, GlobalKnRep, GlobalKrRep, GlobalHckRep, GlobalTllRep, BcdCadActivation},
BcdKrHbComp = phipsipsiCompet[Bcd, threshBCDgt, nuBCDgt,
Kr, threshKRgt, nuKRgt, Hb, threshHBgt, nuHBgt];
BcdCadActivation = 1 - (1 - BcdKrHbComp) * (1 - phi[Cad, threshCADgt, nuCADgt]);
(* two independent global activators, OR'd together a la Fig. A4-C *)
GlobalKnRep = psi[Kn, threshKNgt, nuKNgt];
GlobalKrRep = psi[Kr, threshKRgt, nuKRgt]; GlobalHckRep = psi[Hck, threshHCKgt, nuHCKgt];
GlobalTllRep = psi[Tll, threshTLLgt, nuTLLgt];
BcdCadActivation * GlobalKnRep * GlobalKrRep * GlobalHckRep * GlobalTllRep
];

```

```

(* Hunchback promoter 1 transcription *)
txnhbP1Zyg[Hb_, Kn_, Gt_, Kr_, Tll_, Hck_] :=
Block[{activationHBandTLL, HckRepress, KnRepress, KrRepress, GtRepress},
activationHBandTLL =
  phiPhiTwoGlobalActivators[Hb, threshHBhb1, nuHBhb1, Tll, threshTLLhb1, nuTLLhb1, 1];
HckRepress = psi[Hck, threshHCKhb1, nuHCKhb1];
KnRepress = psi[Kn, threshKNhb1, nuKNhb1];
KrRepress = psi[Kr, threshKRhb1, nuKRhb1];
GtRepress = psi[Gt, threshGThb1, nuGThb1];
activationHBandTLL * HckRepress * KnRepress * KrRepress * GtRepress
];

(* Hunchback promoter 2 transcription *)
txnhbP2Zyg[Bcd_, Hb_, Kn_, Gt_, Kr_, Tll_, Hck_] :=
Block[{KrInterference, HbModificationOfBcdThreshold, BcdKrHbTerm, TllTerm,
BcdTllActivatorCompetition, KnRepression, GtRepression, HckRepression},
KrInterference = psi[Kr, threshKRhb2, nuKRhb2];
HbModificationOfBcdThreshold = (1 - epsilonHBBCDhb2 * phi[Hb, threshHBhb2, nuHBhb2]);
BcdKrHbTerm =
  Bcd^nuBCDhb2 * KrInterference / (threshBCDhb2 + HbModificationOfBcdThreshold)^nuBCDhb2;
TllTerm = Tll^nuTLLhb2 / threshTLLhb2^nuTLLhb2;
BcdTllActivatorCompetition = (BcdKrHbTerm + TllTerm) / (1 + BcdKrHbTerm + TllTerm);
(* Do other repressors in simpleminded way since we don't know much about them*)
KnRepression = psi[Kn, threshKNhb2, nuKNhb2];
GtRepression = psi[Gt, threshGThb2, nuGThb2];
HckRepression = psi[Hck, threshHCKhb2, nuHCKhb2];
BcdTllActivatorCompetition * KnRepression * GtRepression * HckRepression
];

(*Hunchback translation, combining 3 kinds of Hunchback mRNA*)
tranHb1[Hbmat_, HbzygP1_, HbzygP2_, Nos_] :=
Block[{blending},
(*we could have had different constants that we multiplied, within this equation,
for translation maximum for the different kinds of mRNA. e.g.,
use tranMaxZygP1*HbzygP1 instead of HbzygP1.*)
blending = (HbzygP1^2 + HbzygP2^2 + Hbmat (HbzygP1 + HbzygP2)) /
  ((HbzygP1 + HbzygP2) (1 + Hbmat));
blending * psi[Nos, threshNOSHB, nuNOSHB]
];

txnkrl[Bcd_, Kn_, Gt_, Hb_, Tll_] :=
Block[{BCDn, BCDonRate, hbGtterm, HbmitigatedGtRepression, HbRepression},
BCDn = nuBCDkr (1 - 1 / 6 (phi[Kn, threshKNkr, nuKNkr]));
BCDonRate = Bcd^BCDn * psi[Tll, threshTLLkr, nuTLLkr] /
  (threshBCDkr^BCDn + Bcd^BCDn * psi[Tll, threshTLLkr, nuTLLkr]);
hbGtterm = Gt^nuGTkr * psi[Hb, threshHBonGTkr, nuHBonGTkr];
HbmitigatedGtRepression = 1 - (hbGtterm / (hbGtterm + threshGTkr^nuGTkr));
HbRepression = 1 - limitHBrepressionKr (phi[Hb, threshHBkr, nuHBkr]);
BCDonRate * HbmitigatedGtRepression * HbRepression
];

txnkn1[Bcd_, Cad_, Hb_, Gt_, Kr_, Tll_, Hck_] :=
Block[
{ BcdKrHbComp, GlobalGtRep, GlobalKrRep, GlobalHckRep, GlobalTllRep, BcdCadActivation},
BcdKrHbComp = phipsipsiCompet[Bcd, threshBCDkn, nuBCDkn,
  Kr, threshKRkn, nuKRkn, Hb, threshHBkn, nuHBkn];
BcdCadActivation = 1 - (1 - BcdKrHbComp) * (1 - phi[Cad, threshCADkn, nuCADkn]);
(* two independent global activators, OR'd together a la Fig. A4-C *)
GlobalGtRep = psi[Gt, threshGTkn, nuGTkn];
GlobalHckRep = psi[Hck, threshHCKkn, nuHCKkn];
GlobalTllRep = psi[Tll, threshTLLkn, nuTLLkn];
BcdCadActivation * GlobalGtRep * GlobalHckRep * GlobalTllRep
];

```

```

(* Calling all the blocks, and assembling all the relationships between nodes *)
Clear[f, clock];
f = {

  (* Call the transcription clock once, to get a function for how it
  varies with time. Then multiply that function by each transcription term,
  so each will be scaled by how much total transcription the embryo produces in
  a given cell cycle (which increases as the number of nuclei increases). Max
  value of this (which it hits at cycle 14) is 1. *)
  (*clock[t]=txnclock[t],*)

  cad[x, 0] == cadrnaconc,
  D[cad[x, t], t] == -cad[x, t] / hcad,

  D[CAD[x, t], t] == DifCAD D[CAD[x, t], x, x] +
    1 / hCAD (cad[x, t] (psi[BCD[x, t], ThreshBCDcad, nuBCDcad]) - CAD[x, t]),
  CAD[x, 0] == 0,
  (* initial condition for caudal protein *)
  (* Boundary conditions for caudal protein *)
  Derivative[1, 0][CAD][0, t] == 0, (* no flux of caudal protein at anterior end *)
  Derivative[1, 0][CAD][elen, t] == 0, (* no flux of caudal protein at posterior end *)

  D[BCD[x, t], t] == DifBCD D[BCD[x, t], x, x] - BCD[x, t] / hBCD,
  BCD[x, 0] == 0,
  (* Boundary conditions for bicoid protein *)
  Derivative[1, 0][BCD][0, t] == (-BCDSynthRate / DifBCD) (t / (t + .01)),
  (* flux at anterior end is where bicoid comes from *)
  Derivative[1, 0][BCD][elen, t] == 0, (* no flux at posterior end *)

  D[NOS[x, t], t] == DifNOS D[NOS[x, t], x, x] - NOS[x, t] / hNOS,
  NOS[x, 0] == 0,
  (* Boundary conditions for nanos protein *)
  Derivative[1, 0][NOS][elen, t] == (NOSSynthRate / DifNOS) (t / (t + .01)),
  (* flux at posterior end is where nanos comes from *)
  Derivative[1, 0][NOS][0, t] == 0,
  (* no flux at anterior end *)

  D[tll[x, t], t] == (phi[TORA[x, t], threshTORAtll, nuTORAtll] - tll[x, t]) / htll,
  tll[x, 0] == 0,
  D[TLL[x, t], t] == DifTLL D[TLL[x, t], x, x] + 1 / hTLL (tll[x, t] - TLL[x, t]),
  D[hck[x, t], t] == (phi[TORA[x, t], threshTORAhck, nuTORAhck] - hck[x, t]) / hhck,
  hck[x, 0] == 0,
  D[HCK[x, t], t] == DifHCK D[HCK[x, t], x, x] + 1 / hHCK (hck[x, t] - HCK[x, t]),

```

```

TLL[x, 0] == 0,
(* initial condition for tailless protein *)
(* Boundary conditions for tailless protein *)
Derivative[1, 0][TLL][0, t] == 0, (* no flux of Tailless protein at anterior end *)
Derivative[1, 0][TLL][elen, t] == 0, (* no flux of tailless protein at posterior end *)

HCK[x, 0] == 0,
(* initial condition for huckbein protein *)
(* Boundary conditions for huckbein protein *)

Derivative[1, 0][HCK][0, t] == 0, (* no flux of Tailless protein at anterior end *)
Derivative[1, 0][HCK][elen, t] == 0, (* no flux of tailless protein at posterior end *)

D[TORA[x, t], t] == DifTORA D[TORA[x, t], x, x] - TORA[x, t] / hTORA,
TORA[x, 0] == 0,
(* Boundary conditions for TORA protein *)
Derivative[1, 0][TORA][0, t] == (-TORASynthRate / DifTORA) (t / (t + .01)),
(* flux at anterior end is where TORA comes from *)
Derivative[1, 0][TORA][elen, t] == (TORASynthRate / DifTORA) (t / (t + .01)),
(* flux at posterior end where TORA comes from*)

(* Maternal hunchback mRNA *)
hbMat[x, 0] == maternalhbrnaconc, (* initial condition for maternal hunchback mRNA *)
D[hbMat[x, t], t] == (-hbMat[x, t]) / hhbMat,

(*Zygotic stuff: *)

(* Giant mRNA *)
D[gt[x, t], t] == (-gt[x, t] +
    txnGt1[KR[x, t], BCD[x, t], KN[x, t], HB[x, t], TLL[x, t], HCK[x, t], CAD[x, t]]) /
    hgt,
gt[x, 0] == 0, (* initial condition for giant mRNA *)

(* Giant protein *)
D[GT[x, t], t] == DifGT D[GT[x, t], x, x] + (-GT[x, t] + tranGT * gt[x, t]) / hGT,
GT[x, 0] == 0, (* initial condition for giant protein *)
Derivative[1, 0][GT][0, t] == 0, (* no flux of giant protein at anterior end *)
Derivative[1, 0][GT][elen, t] == 0,
(* no flux of giant protein at posterior end *)

(* Hunchback P1 mRNA *)
D[hbP1Zyg[x, t], t] == (-hbP1Zyg[x, t] +
    txnHbP1Zyg[HB[x, t], KN[x, t], GT[x, t], KR[x, t], TLL[x, t], HCK[x, t]]) / hhbP1Zyg,
hbP1Zyg[x, 0] == 0.01,

(* Hunchback P2 mRNA *)
D[hbP2Zyg[x, t], t] == (-hbP2Zyg[x, t] +
    txnHbP2Zyg[BCD[x, t], HB[x, t], KN[x, t], GT[x, t], KR[x, t], TLL[x, t], HCK[x, t]]) /
    hhbP2Zyg,
hbP2Zyg[x, 0] == 0.01,

(* Hunchback translation, combining all the hunchback mRNAs *)
D[HB[x, t], t] == DifHB D[HB[x, t], x, x] +
    (- HB[x, t] + tranHb1[hbMat[x, t], hbP1Zyg[x, t], hbP2Zyg[x, t], NOS[x, t]]) / hHB,
HB[x, 0] == 0, (* initial condition for hunchback protein *)
Derivative[1, 0][HB][0, t] == 0, (* no flux of hunchback protein at anterior end *)
Derivative[1, 0][HB][elen, t] == 0,
(* no flux of hunchback protein at posterior end *)

```

```

(* Knirps mRNA *)
D[kn[x, t], t] == (-kn[x, t] +
  txnKn1[BCD[x, t], CAD[x, t], HB[x, t], GT[x, t], KR[x, t], TLL[x, t], HCK[x, t]]) /
  hkn,
kn[x, 0] == 0, (* initial condition for knirps mRNA *)
(* Knirps protein *)
D[KN[x, t], t] == DifKN D[KN[x, t], x, x] + (-KN[x, t] + tranKN * kn[x, t]) / hKN,
KN[x, 0] == 0, (* initial condition for knirps protein *)
Derivative[1, 0][KN][0, t] == 0, (* no flux of knirps protein at anterior end *)
Derivative[1, 0][KN][elen, t] == 0,
(* no flux of knirps protein at posterior end *)

(* Kruppel mRNA *)
D[kr[x, t], t] == (-kr[x, t] +
  txnKr1[BCD[x, t], KN[x, t], GT[x, t], HB[x, t], TLL[x, t]]) / hkr,
kr[x, 0] == 0, (* initial condition for kruppel mRNA *)

(* Kruppel protein *)
D[KR[x, t], t] == DifKR D[KR[x, t], x, x] + (-KR[x, t] + tranKR * kr[x, t]) / hKR,
KR[x, 0] == 0, (* initial condition for kruppel protein *)
Derivative[1, 0][KR][0, t] == 0, (* no flux of kruppel protein at anterior end *)
Derivative[1, 0][KR][elen, t] == 0
(* no flux of kruppel protein at posterior end *)

};

(*soln=NDSolve[f, {cad, CAD, BCD, NOS, hck, HCK, t11, TLL, TORA},
  {x, 0, elen}, {t, 0, tend}, MaxStepFraction->0.02]*
soln = NDSolve[f, {cad, CAD, BCD, NOS, hck, HCK, t11, TLL, TORA, hbMat, gt, GT, hbP1Zyg,
  hbP2Zyg, HB, kn, KN, kr, KR}, {x, 0, elen}, {t, 0, tend}, MaxStepFraction->0.01];

```

APPENDIX REFERENCES

- Akam, M. 1987. The molecular basis for metameric pattern in the *Drosophila* embryo. *Development* 101:1–22.
- Bate, M, Martinez, A, Alfonso. 1993. The Development of *Drosophila melanogaster*. Plainview, N.Y: Cold Spring Harbor Laboratory Press. p. 2 v. (x, 1558 p.).
- Clyde, DE, Corado, MS, Wu, X, Pare, A, Papatsenko, D, Small, S. 2003. A self-organizing system of repressor gradients establishes segmental complexity in *Drosophila*. *Nature* 426:849–853.
- Driever, W, Nusslein-Volhard, C. 1989. The bicoid protein is a positive regulator of *hunchback* transcription in the early *Drosophila* embryo. *Nature* 337:138–143.
- Eldon, ED, Pirrotta, V. 1991. Interactions of the *Drosophila* gap gene *giant* with maternal and zygotic pattern-forming genes. *Development* 111:367–378.
- Foe, VE, Odell, GM, Edgar, BA. 1993. Mitosis and morphogenesis in the *Drosophila* embryo: Point and counterpoint. In: Bate, MA, Martinez Arias, A, editors. The Development of *Drosophila melanogaster*. Plainview, N.Y: Cold Spring Harbor Laboratory Press. p. 149–300.
- Gaul, U, Jackle, H. 1990. Role of gap genes in early *Drosophila* development. *Adv Genet* 27:239–275.
- Gregor, T, Tank, DW, Wieschaus, EF, Bialek, W. 2007a. Probing the limits to positional information. *Cell* 130:153–164.
- Gregor, T, Wieschaus, EF, McGregor, AP, Bialek, W, Tank, DW. 2007b. Stability and nuclear dynamics of the Bicoid morphogen gradient. *Cell* 130:141–152.
- Hoch, M, Gerwin, N, Taubert, H, Jackle, H. 1992. Competition for overlapping sites in the regulatory region of the *Drosophila* gene *Kruppel*. *Science* 256:94–97.

- Hülskamp, M, Lukowitz, W, Beermann, A, Glaser, G, Tautz, D. 1994. Differential regulation of target genes by different alleles of the segmentation gene hunchback in *Drosophila*. *Genetics* 138:125–134.
- Hülskamp, M, Pfeifle, C, Tautz, D. 1990. A morphogenetic gradient of hunchback protein organizes the expression of the gap genes Kruppel and knirps in the early *Drosophila* embryo. *Nature* 346:577–580.
- Kosman, D, Reinitz, J, Sharp, DH. 1998a. Automated assay of gene expression at cellular resolution. *Pac Symp Biocomput* 6–17.
- Kosman, D, Small, S, Reinitz, J. 1998b. Rapid preparation of a panel of polyclonal antibodies to *Drosophila* segmentation proteins. *Dev Genes Evol* 208:290–294.
- Kraut, R, Levine, M. 1991a. Mutually repressive interactions between the gap genes giant and Kruppel define middle body regions of the *Drosophila* embryo. *Development* 111:611–621.
- Kraut, R, Levine, M. 1991b. Spatial regulation of the gap gene giant during *Drosophila* development. *Development* 111:601–609.
- Lipshitz, HD. 2009. Follow the mRNA: a new model for Bicoid gradient formation. *Nat Rev Mol Cell Biol* 10:509–512.
- Lu, X, Chou, TB, Williams, NG, Roberts, T, Perrimon, N. 1993. Control of cell fate determination by p21ras/Ras1, an essential component of torso signaling in *Drosophila*. *Genes Dev* 7:621–632.
- Makeev, VJ, Lifanov, AP, Nazina, AG, Papatsenko, DA. 2003. Distance preferences in the arrangement of binding motifs and hierarchical levels in organization of transcription regulatory information. *Nucleic Acids Res* 31:6016–6026.
- Margolis, JS, Borowsky, ML, Steingrimsson, E, Shim, CW, Lengyel, JA, Posakony, JW. 1995. Posterior stripe expression of hunchback is driven from two promoters by a common enhancer element. *Development* 121:3067–3077.

- Meir, E, Munro, EM, Odell, GM, Von Dassow, G. 2002. Ingeneue: a versatile tool for reconstituting genetic networks, with examples from the segment polarity network. *J Exp Zool* 294:216–251.
- Murata, Y, Wharton, RP. 1995. Binding of pumilio to maternal hunchback mRNA is required for posterior patterning in *Drosophila* embryos. *Cell* 80:747–756.
- Odell, GM, von Dassow, G. 2003.
- Pankratz, MJ, Jackle, H. 1993. Blastoderm segmentation. In: Bate, MA, Martinez Arias, A, editors. *The Development of Drosophila melanogaster*. Cold Spring Harbor, NY: Cold Spring Harbor Laboratory Press. p. 467–516.
- Reinitz, J. 2007. Developmental biology: a ten per cent solution. *Nature* 448:420–421.
- Rivera-Pomar, R, Jackle, H. 1996. From gradients to stripes in *Drosophila* embryogenesis: filling in the gaps. *Trends Genet* 12:478–483.
- Rivera-Pomar, R, Lu, X, Perrimon, N, Taubert, H, Jackle, H. 1995. Activation of posterior gap gene expression in the *Drosophila* blastoderm. *Nature* 376:253–256.
- Surkova, S, Kosman, D, Kozlov, K, Manu, Myasnikova, E, Samsonova, AA, Spirov, A, Vanario-Alonso, CE, Samsonova, M, Reinitz, J. 2008. Characterization of the *Drosophila* segment determination morphome. *Dev Biol* 313:844–862.
- Thieffry, D, Sanchez, L. 2003. Dynamical modelling of pattern formation during embryonic development. *Curr Opin Genet Dev* 13:326–330.
- von Dassow, G, Meir, E, Munro, EM, Odell, GM. 2000. The segment polarity network is a robust developmental module. *Nature* 406:188–192.
- Von Dassow, G, Odell, GM. 2002. Design and constraints of the *Drosophila* segment polarity module: robust spatial patterning emerges from intertwined cell state switches. *J Exp Zool* 294:179–215.

Wimmer, EA, Carleton, A, Harjes, P, Turner, T, Desplan, C. 2000. Bicoid-independent formation of thoracic segments in *Drosophila*. *Science* 287(5462):2476–2479.

REFERENCES

- Akam, M. 1987. The molecular basis for metameric pattern in the *Drosophila* embryo. *Development* 101:1–22.
- Albert, R, Othmer, HG. 2003. The topology of the regulatory interactions predicts the expression pattern of the segment polarity genes in *Drosophila melanogaster*. *J Theor Biol* 223(1):1–18.
- Alves, F, Dilao, R. 2006. Modeling segmental patterning in *Drosophila*: Maternal and gap genes. *J Theor Biol* 241:342–359.
- Coppey, M, Berezhkovskii, AM, Kim, Y, Boettiger, AN, Shvartsman, SY. 2007. Modeling the bicoid gradient: diffusion and reversible nuclear trapping of a stable protein. *Dev Biol* 312:623–630.
- Edgar, BA, Weir, MP, Schubiger, G, Kornberg, T. 1986. Repression and turnover pattern *fushi tarazu* RNA in the early *Drosophila* embryo. *Cell* 47:747–754.
- Gillespie, DT. 1977. Exact stochastic simulation of coupled chemical reactions. *J Phys Chem* 81:2340–2361.
- Gregor, T, Bialek, W, de Ruyter van Steveninck, RR, Tank, DW, Wieschaus, EF. 2005. Diffusion and scaling during early embryonic pattern formation. *Proc Natl Acad Sci U S A* 102:18403–18407.
- Gregor, T, Tank, DW, Wieschaus, EF, Bialek, W. 2007a. Probing the limits to positional information. *Cell* 130:153–164.
- Gregor, T, Wieschaus, EF, McGregor, AP, Bialek, W, Tank, DW. 2007b. Stability and nuclear dynamics of the Bicoid morphogen gradient. *Cell* 130:141–152.
- Grosskortenhaus, R, Pearson, BJ, Marusich, A, Doe, CQ. 2005. Regulation of temporal identity transitions in *Drosophila* neuroblasts. *Dev Cell* 8:193–202.

- Houchmandzadeh, B, Wieschaus, E, Leibler, S. 2002. Establishment of developmental precision and proportions in the early *Drosophila* embryo. *Nature* 415:798–802.
- Houchmandzadeh, B, Wieschaus, E, Leibler, S. 2005. Precise domain specification in the developing *Drosophila* embryo. *Phys Rev E Stat Nonlin Soft Matter Phys* 72:061920.
- Hülkamp, M, Lukowitz, W, Beermann, A, Glaser, G, Tautz, D. 1994. Differential regulation of target genes by different alleles of the segmentation gene *hunchback* in *Drosophila*. *Genetics* 138:125–134.
- Irons, DJ, Monk, NA. 2007. Identifying dynamical modules from genetic regulatory systems: applications to the segment polarity network. *BMC Bioinformatics* 8:413.
- Jacob, Y, Sather, S, Martin, JR, Ollo, R. 1991. Analysis of *Kruppel* control elements reveals that localized expression results from the interaction of multiple subelements. *Proc Natl Acad Sci U S A* 88:5912–5916.
- Jaeger, J, Blagov, M, Kosman, D, Kozlov, KN, Manu, Myasnikova, E, Surkova, S, Vanario-Alonso, CE, Samsonova, M, Sharp, DH, Reinitz, J. 2004a. Dynamical analysis of regulatory interactions in the gap gene system of *Drosophila melanogaster*. *Genetics* 167:1721–1737.
- Jaeger, J, Surkova, S, Blagov, M, Janssens, H, Kosman, D, Kozlov, KN, Manu, Myasnikova, E, Vanario-Alonso, CE, Samsonova, M, Sharp, DH, Reinitz, J. 2004b. Dynamic control of positional information in the early *Drosophila* embryo. *Nature* 430:368–371.
- Kellerman, KA, Mattson, DM, Duncan, I. 1990. Mutations affecting the stability of the *fushi tarazu* protein of *Drosophila*. *Genes Dev* 4:1936–1950.
- Kosman, D, Reinitz, J, Sharp, DH. 1998a. Automated assay of gene expression at cellular resolution. *Pac Symp Biocomput* 6–17.

- Kosman, D, Small, S, Reinitz, J. 1998b. Rapid preparation of a panel of polyclonal antibodies to *Drosophila* segmentation proteins. *Dev Genes Evol* 208:290–294.
- Kraut, R, Levine, M. 1991. Mutually repressive interactions between the gap genes giant and Kruppel define middle body regions of the *Drosophila* embryo. *Development* 111:611–621.
- Lipshitz, HD. 2009. Follow the mRNA: a new model for Bicoid gradient formation. *Nat Rev Mol Cell Biol* 10:509–512.
- Lott, SE, Kreitman, M, Palsson, A, Alekseeva, E, Ludwig, MZ. 2007. Canalization of segmentation and its evolution in *Drosophila*. *Proc Natl Acad Sci U S A* 104:10926–10931.
- Lucchetta, EM, Lee, JH, Fu, LA, Patel, NH, Ismagilov, RF. 2005. Dynamics of *Drosophila* embryonic patterning network perturbed in space and time using microfluidics. *Nature* 434:1134–1138.
- Makeev, VJ, Lifanov, AP, Nazina, AG, Papatsenko, DA. 2003. Distance preferences in the arrangement of binding motifs and hierarchical levels in organization of transcription regulatory information. *Nucleic Acids Res* 31:6016–6026.
- Manu, Surkova, S, Spirov, AV, Gursky, VV, Janssens, H, Kim, AR, Radulescu, O, Vanario-Alonso, CE, Sharp, DH, Samsonova, M, Reinitz, J. 2009a. Canalization of gene expression and domain shifts in the *Drosophila* blastoderm by dynamical attractors. *PLoS Comput Biol* 5:e1000303.
- Manu, Surkova, S, Spirov, AV, Gursky, VV, Janssens, H, Kim, AR, Radulescu, O, Vanario-Alonso, CE, Sharp, DH, Samsonova, M, Reinitz, J. 2009b. Canalization of gene expression in the *Drosophila* blastoderm by gap gene cross regulation. *PLoS Biol* 7:e1000049.

- Meir, E, Munro, EM, Odell, GM, Von Dassow, G. 2002. Ingeneue: a versatile tool for reconstituting genetic networks, with examples from the segment polarity network. *J Exp Zool* 294:216–251.
- Niessing, D, Rivera-Pomar, R, La Rosee, A, Hader, T, Schock, F, Purnell, BA, Jackle, H. 1997. A cascade of transcriptional control leading to axis determination in *Drosophila*. *J Cell Physiol* 173:162–167.
- Nusslein-Volhard, C, Wieschaus, E. 1980. Mutations affecting segment number and polarity in *Drosophila*. *Nature* 287:795–801.
- Pankratz, MJ, Jackle, H. 1993. Blastoderm segmentation. In: Bate, MA, Martinez Arias, A, editors. *The Development of Drosophila melanogaster*. Cold Spring Harbor, NY: Cold Spring Harbor Laboratory Press. p. 467–516.
- Perkins, TJ. 2007. The gap gene system of *Drosophila melanogaster*: model-fitting and validation. *Ann N Y Acad Sci* 1115:116–131.
- Perkins, TJ, Jaeger, J, Reinitz, J, Glass, L. 2006. Reverse engineering the gap gene network of *Drosophila melanogaster*. *PLoS Comput Biol* 2:e51.
- Poustelnikova, E, Pisarev, A, Blagov, M, Samsonova, M, Reinitz, J. 2004. A database for management of gene expression data in situ. *Bioinformatics* 20:2212–2221.
- Rivera-Pomar, R, Jackle, H. 1996. From gradients to stripes in *Drosophila* embryogenesis: filling in the gaps. *Trends Genet* 12:478–483.
- Rosenblueth, A, Wiener, N. 1945. The Role of Models in Science. *Philosophy of Science* 12:316–321.
- Sanchez, L, Thieffry, D. 2001. A logical analysis of the *Drosophila* gap-gene system. *J Theor Biol* 211(2):115–141.
- Schröder, R, Sander, K. 1993. A comparison of transplantable bicoid activity and partial bicoid homeobox sequences in several *Drosophila* and blowfly species (*Calliphoridae*). *Roux's Arch Dev Biol* 203:34–43.

- Siegal, ML, Bergman, A. 2002. Waddington's canalization revisited: developmental stability and evolution. *Proc Natl Acad Sci U S A* 99:10528–10532.
- Sommer, R, Tautz, D. 1991. Segmentation gene expression in the housefly *Musca domestica*. *Development* 113:419–430.
- St Johnston, D, Nusslein-Volhard, C. 1992. The origin of pattern and polarity in the *Drosophila* embryo. *Cell* 68:201–219.
- Stauber, M, Jackle, H, Schmidt-Ott, U. 1999. The anterior determinant *bicoid* of *Drosophila* is a derived *Hox* class 3 gene. *Proc Natl Acad Sci U S A* 96:3786–3789.
- Surkova, S, Kosman, D, Kozlov, K, Manu, Myasnikova, E, Samsonova, AA, Spirov, A, Vanario-Alonso, CE, Samsonova, M, Reinitz, J. 2008. Characterization of the *Drosophila* segment determination morphome. *Dev Biol* 313:844–862.
- Thieffry, D, Sanchez, L. 2002. Alternative epigenetic states understood in terms of specific regulatory structures. *Ann N Y Acad Sci* 981:135–153.
- Thieffry, D, Sanchez, L. 2003. Dynamical modelling of pattern formation during embryonic development. *Curr Opin Genet Dev* 13:326–330.
- Thomas, R, D'Ari, R. 1990. *Biological feedback*. Boca Raton: CRC Press. p. 316 p.
- von Dassow, G, Meir, E, Munro, EM, Odell, GM. 2000. The segment polarity network is a robust developmental module. *Nature* 406:188–192.
- Von Dassow, G, Odell, GM. 2002. Design and constraints of the *Drosophila* segment polarity module: robust spatial patterning emerges from intertwined cell state switches. *J Exp Zool* 294:179–215.
- Weir, MP, Edgar, BA, Kornberg, T, Schubiger, G. 1988. Spatial regulation of engrailed expression in the *Drosophila* embryo. *Genes Dev* 2:1194–1203.

Wimmer, EA, Carleton, A, Harjes, P, Turner, T, Desplan, C. 2000. Bicoid-independent formation of thoracic segments in *Drosophila*. *Science* 287(5462):2476–2479.