Algebraic Methods for Modeling Gene Regulatory Networks

David M. Murrugarra Tomairo

Dissertation submitted to the Faculty of the
Virginia Polytechnic Institute and State University
in partial fulfillment of the requirements for the degree of

Doctor of Philosophy
in
Mathematics

Reinhard Laubenbacher, Chair
Henning Mortveit
Stanca Ciupe
Stefan Hoops

July 16, 2012
Blacksburg, Virginia

Keywords: Stochastic Discrete Modeling, Systems Biology, Nested Canalyzing Functions,
Gene Regulatory Networks, Robustness, Intrinsic Noise.

Algebraic Methods for Modeling Gene Regulatory Networks

David Murrugarra

(ABSTRACT)

So called discrete models have been successfully used in engineering and computational systems biology. This thesis discusses algebraic methods for modeling and analysis of gene regulatory networks within the discrete modeling context. The first chapter gives a background for discrete models and put in context some of the main research problems that have been pursued in this field for the last fifty years. It also outlines the content of each subsequent chapter. The second chapter focuses on the problem of inferring dynamics from the structure (topology) of the network. It also discusses the characterization of the attractor structure of a network when a particular class of functions control the nodes of the network. Chapters 3 and 4 focus on the study of multi-state nested canalyzing functions as biologically inspired functions and the characterization of their dynamics. Chapter 5 focuses on stochastic methods, specifically on the development of a stochastic modeling framework for discrete models. Stochastic discrete modeling is an alternative approach from the well-known mathematical formalizations such as stochastic differential equations and Gillespie algorithm simulations. Within the discrete setting, a framework that incorporates propensity probabilities for activation and degradation is presented. This approach allows a finer analysis of discrete models and provides a natural setup for cell population simulations. Finally, Chapter 6 discusses future research directions inspired by the work presented here.

This work received support from grants NSF CMMI-0908201 and ARO 56757-MA.

Dedication

A mis queridas abuelitas Luisa Geronimo Misari y Agripina Geronimo Misari.

Acknowledgments

I would like to thank Reinhard Laubenbacher, my advisor, for his constant support and patience during this research. I am also thankful to Peter Haskell for his guidance through the early years of my graduate studies. Many thanks to my committee members, Dr. Mortveit, Dr. Hoops and Dr. Cuipe for reading this thesis and making corrections. Thank to my collaborators for permission to include coauthored material, Henning Mortveit, Alan Veliz-Cuba, Yuan Li, John Adeyeye, Boris Aguilar, Seda Arat, Franziska Hinkelmann, and Abdul Jarrah.

From the Department of Mathematics, I would like to thank my professors, Dr. Green, Dr. Floyd, Dr. Haskell and many others. To old classmates and friends Quanlei, Bart, Arturo, Moises, Carlos, Alan, Sandra and many others. To my professors from other departments, Dr. Murali, Dr. Tyson and Dr. Grene. From VBI, to the Laubenbacher group, Shernita, Cory, Madison, Matt, Seda, Claus, Ariane, Kasia, Franziska, Greg, Brandy, and Betsy Williams. To REU students that worked with me, Matt Kirkham, Natalie Stanley, Madison Brown, Henderson Wallace, Kathryn Brewer.

I am also very thankful to Blacksburg friends who made life at this beautiful town more enjoyable. To Ed and Val for introducing me into the cider making experience. Thanks Val and Ed for letting Fely and I spend every thanksgiving at your house and for letting me be the supreme carver last year, such an honor. To Tatiana and Virgilio for many activities together, especially the huaylas dance and to Tatiana for pushing me into Saturday morning quechua sessions. To Nicholas and Tara for their friendship. To the Peruvian clan, Bart, Moises, Arturo, Cristina, Boris, Mario, Roxana, Edgardo and many others for some of the beer sessions.

I am very thankful to my beautiful wife, Fely, for her patience and love.

Finally, I am very grateful to my family for their constant support and love. To Esther for constant encouragement. To Edith, Nina, Brez, Kathy, Edward for their friendship. To my uncles Edilberto, Victor, Hector, Lucho, and Raul. To my aunts Juana, Justina, Betty. To my mom and my friend Victorio. To everyone else.

Blacksburg, Virginia, 24060, USA David Murrugarra

July 23, 2012.

Contents

1 Introduction **1**

1.1 Problems to Study Within the Discrete Modeling Strategy 3

1.1.1 On the Structure of Networks 3

1.1.2 On the Dynamics of Networks 3

1.2 Overview of content 4

1.2.1 Overview of Chapter 2 4

1.2.2 Overview of Chapter 3 4

1.2.3 Overview of Chapter 4 4

1.2.4 Overview of Chapter 5 4

1.2.5 Overview of Chapter 6 5

1.3 Papers discussed in this thesis 5

1.3.1 Papers that I coauthored and that are not discussed in this thesis . . 6

2 From Structure to Dynamics: Discrete Dynamical Systems **7**

2.1 Structure and Dynamics of Acyclic Networks 7

2.1.1 The algebra of dynamical systems 9

2.1.2 A property of $\mathcal{P} \longrightarrow \mathcal{M}$ 10

2.1.3 Networks with acyclic dependency graph 11

2.1.4 Application 11

2.1.5 Conclusions 15

2.2 Attractor Structure of Bi-Threshold Networks 15

2.2.1 Synchronous Bi-Threshold Networks 15

3 Multi-state Nested Canalyzing Functions **17**

3.1 Introduction 17

3.2 Nested Canalyzing Rules 19

3.3 The dynamics of nested canalyzing networks 19

3.4 Nested canalyzing rules are biologically meaningful 25

3.5 Examples 25

3.5.1 Regulation in the p53-Mdm2 network 27

3.6 Discussion 30

4 The Number of Multi-state Nested Canalyzing Functions **31**

4.1 Introduction 31

4.2 Nested Canalyzing Functions 33

4.3 Polynomial form of nested canalyzing functions 34

4.4 The algebraic variety of nested canalyzing functions 36

4.5 Number of nested canalyzing functions 40

4.6 Asymptotic properties of $NCF(n)$ 47

4.7 Discussion 50

5 Stochastic Discrete Dynamical Systems **52**

5.1 Introduction 52

5.1.1 Modeling stochasticity in gene regulatory networks 53

5.2 Method 55

5.2.1 Dynamics of SDDS 55

5.3 Applications 57

5.3.1 Regulation in the p53-Mdm2 network 57

5.3.2 Phage lambda infection of bacteria 58

5.4 Conclusions 68

6 Future Directions **70**

Appendices **73**

A **73**

A.1 Appendix for Chapter 4 73

A.2 Proof of Theorem 4.4.2 76

B **80**

B.1 Stochastic Discrete Dynamical Systems: (SDDS) 80

B.1.1 SDDS includes trajectories from the synchronous and asynchronous schemes 80

B.1.2 SDDS can be written as a PBN but needs more parameters 81

B.1.3 Pseudo-Codes 84

B.2 Regulation in the p53-Mdm2 network 84

B.3 Lambda phage infection of bacteria 88

B.4 Relating the propensity parameters with biological information 89

List of Figures

2.1 Dynamics of the constrained model 13

2.2 Adjacency graph for virus competition 13

2.3 Dynamics where cells evolve according to the rules. Each initialization of infected cells determines the final (unique) outcome. 14

3.1 Attractor distribution for networks with nested canalyzing functions (solid circles) and networks with random functions (open circles). The figures show the percentage of networks that returned the number of attractors specified in the x-axis. The parameters N, k and p correspond to the number of nodes, the range for the in-degree distribution, and the number of states for each node, respectively. The figures for $n = 5, 10$ and $p = 2, 5$ were generated for 1000000 networks, the figure for $n = 10$ and $p = 3$ was generated for 100000 networks, and the figure for $n = 20$ and $p = 2$ was generated for 10000 networks. 21

3.2 Cycle length for networks with nested canalyzing functions (solid circles) and networks with random functions (open circles). The parameters N, k and p correspond to the number of nodes, the range for the in-degree distribution, and the number of states for each node, respectively. The figures were generated for one million networks. The upper figures show the mean number of attractors of length specified in the x-axis (solid lines) and their standard deviations (dashed lines), the x-axis of these figures are in a logarithmic scale. The bottom left figure shows the mean number of attractors of lengths specified in the x-axis in a log-log plot (here rnd means random), and finally the bottom right figure shows the percentage of networks that returned a particular cycle of length specified on the x-axis. 22

3.3 Cycle length for networks with nested canalyzing functions (solid circles) and networks with random functions (open circles). The parameters N, k and p correspond to the number of nodes, the range for the in-degree distribution, and the number of states for each node, respectively. The figures were generated for 10000 networks. The upper figures show the mean number of attractors of length specified on the x-axis (solid lines) and their standard deviations (dashed lines), the x-axis of these figures is a logarithmic scale. The bottom left figure shows the mean number of attractors of lengths specified on the x-axis in a log-log plot (here rnd means random), and finally the bottom right figure shows the percentage of networks that returned a particular cycle of length specified on the x-axis. 23

3.4 Cycle length for networks with nested canalyzing functions (solid circles) and networks with random functions (open circles). The parameters N, k and p correspond to the number of nodes, the range for the in-degree distribution, and the number of states for each node, respectively. The figures were generated for 100000 networks. The upper figures show the mean number of attractors of length specified on the x-axis (solid lines) and their standard deviations (dashed lines), the x-axis of these figures are in a logarithmic scale. The bottom left figure shows the mean number of attractors of lengths specified on the x-axis in a log-log plot (here rnd means random), and finally the bottom right figure shows the percentage of networks that returned a particular cycle of length specified on the x-axis. 24

3.5 Cycle length for networks with nested canalyzing functions (solid circles) and networks with random functions (open circles). The parameters N, k and p correspond to the number of nodes, the range for the in-degree distribution, and the number of states for each node, respectively. The figures were generated for one million networks. The upper figures show the mean number of attractors of length specified on the x-axis (solid lines) and their standard deviations (dashed lines), the x-axis of these figures are in a logarithmic scale. The bottom left figure shows the mean number of attractors of lengths specified on the x-axis in a log-log plot (here rnd means random), and finally the bottom right figure shows the percentage of networks that returned a particular cycle of length specified on the x-axis. 26

3.6 Four-variable model for the lambda phage regulatory network. 29

3.7 Four-variable model for the p53-Mdm2 regulatory network. 29

5.1 **State Space**. There is a 9% chance that the system will transition from 01 to 10. There is an 81% chance that the system will transition from 01 to 00. The latter was expected because there is a high degradation propensity for f_2. Note that 00 is a fixed point, i.e., there is 100% chance of staying at this state. 57

5.2 **Four-variable model for the p53-Mdm2 regulatory network. P, Mc, and Mn stand for protein p53, cytoplasmic Mdm2, and nuclear Mdm2, respectively**. 59

5.3 **State space diagram for parameters described in Table 5.1**. The numbers next to the edges encode the transition probabilities. The order of the variables in each vector state is P, Mc, Mn, DNA damage. Self-loops are not depicted. States with darker background comprise the cycle with DNA damage. A second cycle with a lighter shaded background corresponds to the cycle with no DNA damage. The oval shaped state is a steady state. 60

5.4 **Individual cell simulations for parameters described in Table 5.1**. Each subfigure shows oscillations as long as the damage is present. This figure shows variability in the timing of damage repair and in the period of the oscillations. Each frame was generated from a single simulation with sixty time steps. The x-axis represents discrete time steps and the y-axis the expression level. The initial state for all simulations is 0011. 61

5.5 **Cell population simulations**. Each subfigure was generated from 100 simulations, each representing a single cell with sixty time steps. Starting from the top left frame to the right bottom frame the degradation propensity for DNA damage was increased by 5%, i.e. $p^{\downarrow}_{\text{dam}} = .05$ (top left), $p^{\downarrow}_{\text{dam}} = .10$ (top right), $p^{\downarrow}_{\text{dam}} = .15$ (bottom left), and $p^{\downarrow}_{\text{dam}} = .2$ (bottom right). The x-axis represents discrete time steps and the y-axis the average expression level. The initial state for all simulations was 0011. This figure shows that, if the intensity of the damage is increased more cells exhibit oscillations in the level of p53, in agreement with experimental observations [63]. 62

5.6 **Wiring diagram for phage lambda infection model**. 64

5.7 **State space for phage lambda model**. The order of variables in each vector state is CI, CRO, CII, N. The steady state 2000 represents lysogeny where CI is fully expressed while other genes are off. The cycle between 0200 and 0300 represents lysis where CRO is active and other genes are repressed. Self-loops are not depicted. 65

5.8 **Cell population simulations**. Both figures were generated from 100 simulations, each representing a single cell iteration of ten time steps. Top frame for parameters in Table 5.2 shows 93% lysis and 7% lysogeny while bottom frame for parameters in Table 5.3 shows 4% lysis and 96% lysogeny. The x-axis represents discrete time steps while the y-axis shows the average expression level. The initial state for all the simulations is 0000. Solid (circle) points correspond to the average of CI (CRO), and dashed lines represent standard deviations. 66

5.9 **Variation in developmental outcome as a function of the propensity parameters**. Star points indicate the percentage of networks that transition to lysogeny and circle shaped points indicate the percentage of networks that end up in lysis. Bottom axis represents the activation (and degradation) propensities for CI (CRO) in increasing order. Likewise, the top axis represents the activation (and degradation) propensities for CRO (CI) in decreasing order. 67

B.1 Four-variable model for the p53-Mdm2 regulatory network. P, Mc, and Mn stand for protein p53, nuclear Mdm2, and cytoplasmic Mdm2 respectively. . 86

B.2 Four-variable model for the lambda phage regulatory network. 88

B.3 SDDS vs Gillespie. Number of molecules for Gillespie is 1000 and degradation rate $k = 0.05$. Number of simulations for SDDS is 1000 and number of steps for each simulation was 6. The number of molecules for Gillespie was normalized so that it goes from 0 to 1. The scale parameter δ for SDDS was set equal to 20, i.e., 1 second in the scale of Gillespie is equivalent to 20 time steps in the scale of SDDS. Thus, $p_1^{\downarrow} = 0.6213$ (see text for better description). 92

List of Tables

3.1 Nested canalyzing functions for multi-state models 25

4.1 Number of nested canalyzing functions for $p = 3$ and $n = 1, \ldots, 8$. 46

4.2 Number of nested canalyzing functions for $p = 5$ and $n = 1, \ldots, 8$. 46

5.1 **Propensity probabilities for the p53-Mdm2 regulatory network** . . . 59

5.2 **Propensity parameters for Figure 5.8 (top frame)** 63

5.3 **Propensity parameters for Figure 5.8 (bottom frame)** 64

B.1 Truth table for P 86

B.2 Truth table for Mc 86

B.3 Truth table for Mn 87

B.4 Truth tables for *DNA-damage* 87

B.5 Truth table for CI 88

B.6 Truth table for CRO 89

B.7 Truth table for CII 90

B.8 Truth tables for N 90

Chapter 1

Introduction

Understanding the precise mechanisms that drive the functioning of organisms is one of the most important goals of biology [1]. Systems biology aims to understand a biological system as an integrated unit by focusing on the interactions of its constitute components . In doing so, it uses two complementary approaches: top-down systems biology and bottom-up systems biology. Top-down systems biology starts from the whole and aims to characterize the biological mechanisms of the parts while bottom-up systems biology starts from the components, formulating the interactive behavior and then integrates these formulations to predict the behavior of the system [1]. Mathematical modeling can be very helpful for both of these approaches. Different modeling strategies can be found in the literature. Dynamic mathematical models can be broadly divided into two classes: continuous, such as systems of differential equations (and their stochastic variants) and discrete, such as Boolean networks and their generalizations (and their stochastic variants). This thesis will focus on the later class of models.

Discrete models have a long tradition in engineering and computational biology, including finite state machines, Boolean networks, logical models [65], Petri nets [40, 66], and agent-based models [38]. Discrete models play an important role in modeling processes that can be viewed as evolving in discrete time, in which state variables have only finitely many possible states. Decision processes, electrical switching networks, or intra-cellular molecular networks represent examples. Discrete models do not require detailed information about kinetic rate constants and they tend to be more intuitive. In turn, they only provide qualitative information about the system. The most general setting is as follows. Network nodes represent genes, proteins, and other molecular components of gene regulation while network edges describe biological interactions among network nodes that are given as logical rules representing the biochemical mechanisms governing their interaction. Time in this framework is implicit and progresses in discrete steps. More formally, let $x_1, \ldots, x_n$ be variables, which can take values in finite sets $X_1, \ldots, X_n$, respectively. Let $X = X_1 \times \cdots \times X_n$ be the

Cartesian product. A discrete dynamical system in the variables $x_1, \dots, x_n$ is a function

$$f = (f_1, \dots, f_n) : X \to X$$

where each coordinate function $f_i : X \to X_i$ is a function in a subset of $\{x_1, \dots, x_n\}$. Dynamics is generated by iteration of f, and can use different update schemes, e.g. synchronous and asynchronous. As an example, if $X_i = \{0, 1\}$, then each f_i is a Boolean rule and f is a Boolean network where all the variables are updated simultaneously. We will assume that each set X_i comes with a natural total ordering of its elements (corresponding to the concentration levels of the associated molecular species).

As described, such systems are set functions without additional mathematical structure. It is therefore advantageous to impose additional mathematical structure on X, namely that of a finite field. This is of course utilized in the case of Boolean networks. Evaluation of Boolean functions is equivalent to carrying out arithmetic in the Boolean field with two elements. The translation between Boolean functions and polynomials is straight-forward, with the Boolean operator AND corresponding to multiplication, addition corresponding to XOR, and negation corresponding to addition of 1. Once we have a finite field structure on X, which we shall now denote by $\mathbb{K}$, it is well-known that any function $\mathbb{K}^n \longrightarrow \mathbb{K}$ can be represented as a polynomial function [53, p. 369]. Thus, we can focus on systems

$$f = (f_1, \dots, f_n) : \mathbb{K}^n \longrightarrow \mathbb{K}^n,$$

for which each f_i is a polynomial function in the variables $x_1, \dots, x_n$. We will call such a system a *polynomial dynamical system (PDS) over a finite field* $\mathbb{F}$. A powerful consequence is that one can use algorithms and theoretical results from computer algebra and computational algebraic geometry, such as the theory of Gröbner bases.

It has been shown [44, 9] that discrete models in several different frameworks can be translated into the rich and general mathematical framework of PDS, namely k-bounded Petri nets, so-called logical models used in molecular biology, and many agent-based models, which are becoming very prominent in biology. Thus, the framework of PDS provides a common theoretical formulation. Using algorithms from polynomial algebra, (Software to carry out this translation is available at `http://dvd.vbi.vt.edu/adam.html` [83] and is very efficient in analyzing various features of PDS, including their dynamics). Thus, any theoretical results about PDS apply to all these model types.

On the other hand, stochasticity is an inherent property of any biological system, therefore the development of a stochastic modeling framework for the discrete strategy is of especial importance. To account for stochasticity within the discrete setting several methods have been considered. Probabilistic Boolean networks [67, 68] introduce stochasticity in the update functions, allowing a different update function to be used at each iteration, chosen from a probability space of such functions for each network node. For other approaches see [69, 70, 71]. This thesis presents a model type (see Chapter 5) that introduces activation and degradation propensities. This approach allows a finer analysis of discrete models and provides a natural setup for cell population simulations to study cell-to-cell variability. It is worth to mention that these stochastic variants for discrete models can be studied in the context of Markov chains [67, 68].

1.1 Problems to Study Within the Discrete Modeling Strategy

Given a discrete dynamical system in the variables $x_1, \ldots, x_n$,

$$f = (f_1, \ldots, f_n) : X \to X$$

we will associate to it two directed graphs (see Chapter 2): its dependency graph or wiring diagram and its state space. The dependency graph is referred as the structure of the network and the state space is referred as the dynamics of the network. In this setting there are several interesting questions one can ask about the structural and dynamical properties of networks and how these two correlate to each other.

1.1.1 On the Structure of Networks

1. Identification of network motifs in the dependency graph that constrain network dynamics, e.g. feedback and feedforward loops. See Chapter 2.
2. Understanding the regulatory logic of molecular interaction networks.
   - Identification of patterns in the mechanisms that govern network dynamics, see Chapter 3.
   - Analysis of biologically inspired functions: nested canalyzing functions and threshold functions. See Chapters 2, 3, 4.

1.1.2 On the Dynamics of Networks

1. Characterization of dynamics of biologically inspired functions. See Chapter 3.
2. Stable dynamics: few attractors and large basins. See Chapter 3.
3. Probability of maintaining same dynamics under small perturbations. See Chapter 3.
4. Perturbation propagation analysis: probability of maintaining same dynamics under small perturbations. See Chapter 3.

5. Effect of different update schemes, e.g., Synchronuos versus Asynchronous. See Chapter 5.

6. Stochastic dynamics, see Chapter 5.

See Chapter 6 for other types of questions to study within the discrete modeling context.

1.2 Overview of content

Here we outline the content of each subsequent chapter.

1.2.1 Overview of Chapter 2

This chapter focuses on the problem of inferring dynamics from the structure of gene regulatory networks. Results are presented for two especial cases: Acyclic directed networks and undirected bithreshold networks. This chapter is based on published papers [2, 3, 4].

1.2.2 Overview of Chapter 3

This chapter focuses on the study of multi-state nested canalyzing functions as biologically inspired functions. It presents a computational analysis about the dynamics of networks controlled by multi-state nested canalyzing functions. It also presents statistics about how often this functions appear in published models. This chapter is based on published paper [16].

1.2.3 Overview of Chapter 4

This chapter focuses on the study of multi-state nested canalyzing functions from an algebraic point of view. It gives a parametrization for the class of multi-state nested canalyzing functions and a formula for the number of multi-state nested canalyzing functions. Asymptotic properties of such a formula are also given. This chapter is based on published paper [47].

1.2.4 Overview of Chapter 5

This chapter focuses on stochastic methods, specifically on the development of a stochastic modeling framework for discrete models. Stochastic discrete modeling is an alternative approach from the well-known mathematical formalizations such as stochastic differential equations and Gillespie algorithm simulations. Within the discrete setting, a framework called SDDS is presented. SDDS allows a finer analysis of discrete models and provides a natural set up for cell population simulations. SDDS incorporates trajectories from different update schemes, e.g., synchronuos versus asynchronous. This Chapter has an appendix B.4 where the relationship between SDDS and the Gillespie algorithm is discussed. This chapter is based on published paper [59].

1.2.5 Overview of Chapter 6

This chapter presents a list of open problems and research directions inspired by the work presented in this thesis.

1.3 Papers discussed in this thesis

Most of the material discussed in this thesis is published or is under review. Below is the list of papers coauthored by me and a description of my contribution to each of them.

1. Modeling Stochasticity and Variability in Gene Regulatory Networks. **David Murrugarra**, Alan Veliz-Cuba, Boris Aguilar, Seda Arat, Reinhard Laubenbacher. EURASIP Journal on Bioinformatics and Systems Biology, 2012: 2012:5. doi:10.1186/1687-4153-2012-5.
   Author's Contribution: My contribution to this paper were the following: 1. Played a major role on the design of the study. 2. Model construction and analysis of results. 3. Writing of the manuscript.

2. The Number of Multi-state Nested Canalyzing Functions. **David Murrugarra** and Reinhard Laubenbacher. Physica D: Nonlinear Phenomena, 241, 921-938, 2012.
   Author's Contribution: My contribution to this paper were the following: 1. Played a major role on the design of the study. 2. Formulation and proofs of theorems. 3. Part of the writing of the manuscript.

3. Regulatory Patterns in Molecular Interaction Networks. **David Murrugarra** and Reinhard Laubenbacher. Journal of Theoretical Biology, 288, 66-72, 2011.
   Author's Contribution: My contribution to this paper were the following: 1. Played a major role on the design of the study. 2. Model construction and analysis of results. 3. Part of the writing of the manuscript.

4. Bifurcations in Boolean Networks. Chris J. Kuhlman, Henning S. Mortveit, **David Murrugarra**, V. S. Anil Kumar. Discrete Mathematics and Theoretical Computer Science, proc, AP, 29-46, 2012.
   Author's Contribution: My contribution to this paper was the proof of one the main theorems.

5. Structure and Dynamics of Acyclic Networks. **David Murrugarra**, Alan Veliz-Cuba, Reinhard Laubenbacher. Under Review.
   Author's Contribution: My contribution to this paper were the following: 1. Played a major role on the design of the study. 2. Model construction and analysis of results. 3. Part of the writing of the manuscript.

6. Structure and Dynamics of Polynomial Dynamical Systems. Reinhard Laubenbacher, **David Murrugarra**, Alan Veliz-Cuba. Proc. 2011 NSF Engineering Research and Innovation Conference, Atlanta, GA, January 2011.
   Author's Contribution: My contribution to this paper were the following: 1. Played a major role on the design of the study. 2. Model construction and analysis of results. 3. Part of the writing of the manuscript.

1.3.1 Papers that I coauthored and that are not discussed in this thesis

1. A Mathematical Framework for Agent Based Models of Complex Biological Networks. Franziska Hinkelmann, **David Murrugarra**, Abdul Jarrah, Reinhard Laubenbacher. Bulletin of Mathematical Biology, 73(7):1583-1602, 2011.

2. Boolean nested canalizing functions: a comprehensive analysis. Yuan Li, John O. Adeyeye, **David Murrugarra**, Boris Aguilar, Reinhard Laubenbacher. Under Review.

3. The Characterization and Enumeration of Canalyzing Functions over Finite Fields. Yuan Li, **David Murrugarra**, John Adeyeye, Reinhard Laubenbacher. Submitted.

Chapter 2

From Structure to Dynamics: Discrete Dynamical Systems

Understanding how the model structure constrains its dynamics is of particular importance. This chapter addresses this problem for two particular classes of networks: acyclic directed networks and undirected bithreshold networks. In the first section, the systems are given by mappings on an affine space over a finite field whose coordinate functions are polynomials. They form a general class of models which can represent many discrete model types. Assigning to such a system its dependency graph, that is, the directed graph that specifies the variable dependencies, provides a mapping from systems to graphs. A basic property of this mapping is derived and used to prove that dynamical systems with an acyclic dependency graph can only have a unique fixed point in their phase space and no periodic orbits. This result is then applied to a published model of in vitro virus competition. In the second section, the attractor structure of undirected bithreshold networks is characterized, they can only have fixed points or 2-cycles as attractors.

This chapter is based on published papers [2, 3, 4]. My contribution to [4] was the proof of one the main theorems and my contribution to [2, 3] were the following: 1. Played a major role on the design of the study. 2. Model construction and analysis of results. 3. Part of the writing of the manuscript.

2.1 Structure and Dynamics of Acyclic Networks

In this section the attractor structure of directed acyclic networks is characterized.

Discrete models play an important role in modeling processes that can be viewed as evolving in discrete time, in which state variables have only finitely many possible states. Decision processes, electrical switching networks, or intra-cellular molecular networks represent examples. There is a long tradition in the engineering literature of work related to an understanding of how the structure of such models constrains their dynamics, see, e.g., [7, 48]. This problem is important both for the design of networks as well as for their analysis. In both theoretical and applied studies of dynamical systems, the problem of predicting dynamic features of the system from structural properties is very important, see, e.g., [11, 13]. An instantiation of this problem is the question in systems biology whether the connectivity structure of molecular networks, such as metabolic or gene regulatory networks, has special features that correlate with the type of dynamics these networks exhibit. (This type of application motivated the present study.) Some progress has been made on this question. For instance, in [5] it was shown that certain small network motifs appear much more often in such networks than would be expected in a random graph. The work in this section was motivated by the desire to provide a theoretical framework within which the relationship between structure and dynamics of certain families of dynamical systems can be studied. We briefly describe this framework and show one example of a result and its implications.

The type of dynamical system studied here is discrete in time as well as in variable states. That is, we consider a collection $x_1, \ldots, x_n$ of variables, each of which can take on values in a finite set X. We consider time-discrete dynamical systems

$$f : X^n \longrightarrow X^n.$$

A well-known instantiation of such systems are Boolean networks, which have many applications in molecular biology as well as engineering. In this case, $X = \{0, 1\}$. As described, such systems are set functions without additional mathematical structure. It is therefore advantageous to impose additional mathematical structure on X, namely that of a finite field. This is of course utilized in the case of Boolean networks. Evaluation of Boolean functions is equivalent to carrying out arithmetic in the Boolean field with two elements. The translation between Boolean functions and polynomials is straight-forward, with the Boolean operator AND corresponding to multiplication, addition corresponding to XOR, and negation corresponding to addition of 1. Once we have a finite field structure on X, which we shall now denote by $\mathbb{K}$, it is well-known that any function $\mathbb{K}^n \longrightarrow \mathbb{K}$ can be represented as a polynomial function [53, p. 369]. Thus, we can focus on systems

$$f = (f_1, \ldots, f_n) : \mathbb{K}^n \longrightarrow \mathbb{K}^n,$$

for which each f_i is a polynomial function in the variables $x_1, \ldots, x_n$. We will call such a system a polynomial dynamical system (PDS) over $\mathbb{K}$. A powerful consequence is that one can use algorithms and theoretical results from computer algebra and computational algebraic geometry, such as the theory of Gröbner bases.

In this section, we are concerned with the family $\mathcal{P}$ of all PDS of a given dimension n over a finite field $\mathbb{K}$. The structural information for a PDS includes a directed graph that indicates the dependency relationships between variables. In the context of a molecular network model, this graph would represent the wiring diagram of the network. Graphs can be represented by their adjacency matrices. The set of $n \times n$ matrices carries an algebraic structure using max and min operations, which has been studied previously over the real field, see, e.g., [8]. Here we prove a result about the relationship between properties of dynamical systems in $\mathcal{P}$ under composition and properties of the corresponding adjacency matrices in the max-min algebra. As a consequence of this result, we derive a dynamic property of PDS with acyclic dependency graphs. Finally, we give an application of this consequence to a published model of *in vitro* virus competition.

2.1.1 The algebra of dynamical systems

Let $\mathbb{K}$ be a finite field. We consider dynamical systems

$$F = (f_1, \ldots, f_n) : \mathbb{K}^n \to \mathbb{K}^n,$$

where $f_i : \mathbb{K}^n \to \mathbb{K}$ for $i = 1, \ldots, n$. As observed in the previous section, any such F is a PDS. Let $\mathcal{P}$ denote the family of all such systems. It is well-known that $\mathcal{P}$ has the structure of an associative algebra under coordinate-wise addition and composition of functions.

To F we can associate a directed graph with the n nodes $x_1, \ldots, x_n$. Let $[F]$ be the $n \times n$ adjacency matrix of this graph. That is, $[F] = (a_{ij})$ is defined as follows:

$$a_{ij} = \begin{cases} 1, & f_i \text{ depends on } x_j \\ 0, & \text{otherwise} \end{cases}$$

Equivalently, $a_{ij} = 1$ if and only if there are $p \neq q \in \mathbb{K}$ such that

$$f_i(x^{(j,p)}) \neq f_i(x^{(j,q)})$$

where $x^{(j,p)} = (x_1, \ldots, x_{j-1}, p, x_{j+1}, \ldots, x_n) \in \mathbb{K}^n$.

The adjacency matrix has binary entries, and we now define two operations on such matrices. Let $\mathbb{B}$ denote the Boolean field with two elements, with the natural order $0 < 1$.

Definition 2.1.1. *Given $a, b \in \mathbb{B}$, we let:*

1) $a \oplus b = \max\{a, b\}$ *and*

2) $a \star b = \min\{a, b\}$

Definition 2.1.2. *Given $A = (a_{ij})$ and $B = (b_{ij})$ matrices with entries in $\mathbb{B}$, we define the following operation:*

$$A \star B = C = (c_{ij}), \text{ where } c_{ij} = \bigoplus_{k=1}^{n} (a_{ik} \star b_{kj})$$

Remark 2.1.1. $c_{ij} = 1$ if and only if there is k such that $a_{ik} \star b_{kj} = 1$

Definition 2.1.3. *We define* $A \leqslant B$ *if and only if* $a_{ij} \leqslant b_{ij}$ *for all i,j.*

Forming the adjacency matrix of the dependency graph of a PDS then gives a mapping

$$\mathcal{P} \longrightarrow \mathcal{M}.$$

where $\mathcal{M} = \mathbb{B}^{n \times n}$. And we also have a mapping that associates to an element in $\mathcal{P}$ its phase space, a directed graph on the vertex set $\mathbb{K}^n$ and edges $x \to y$ whenever $F(x) = y$, which encodes the dynamics of the system. We will denote by $\mathcal{S}$ the set of all directed graphs on $\mathbb{K}^n$ with the property that each vertex has a unique outgoing edge (the requirement for being the phase space of a deterministic PDS). Hence, we have mappings

$$\begin{tikzcd} \mathcal{P} \arrow[r] \arrow[d] & \mathcal{M} \arrow[dl, dashed] \\ \mathcal{S} & \end{tikzcd}$$

The result in the next section relates information in $\mathcal{M}$ to information in $\mathcal{S}$.

2.1.2 A property of $\mathcal{P} \longrightarrow \mathcal{M}$

The main result of this section is the following theorem. It describes a basic property of the mapping that extracts the adjacency matrix from the system.

Theorem 2.1.4. *We have that*

$$[F \circ G] \leqslant [F] \star [G]$$

in $\mathcal{M}$ *for all* $F, G \in \mathcal{P}$.

Proof. Let A, B, and C be the adjacency matrices for F, G, and $F \circ G$, respectively. It suffices to prove that, if $c_{ij} = 1$, then $a_{ik} b_{kj} = 1$ for some k. Let us assume that $c_{ij} = 1$, then there are $p \neq q \in \mathbb{K}$ such that

$$(F \circ G)_i(x^{(j,p)}) \neq (F \circ G)_i(x^{(j,q)}),$$

where $x^{(j,p)} = (x_1, \ldots, x_{j-1}, p, x_{j+1}, \ldots, x_n) \in \mathbb{K}^n$. Now, since $(F \circ G)_i = f_i \circ G$ we have that

$$f_i(y_1, \ldots, y_n) \neq f_i(\hat{y}_1, \ldots, \hat{y}_n),$$

where $y_s = g_s(x^{(j,p)})$ and $\hat{y}_s = g_s(x^{(j,q)})$ for $s = 1, \ldots, n$. Then there is an index k such that f_i depends on x_k and

$$y_k \neq \hat{y}_k,$$

or

$$g_k(x^{(j,p)}) \neq g_k(x^{(j,q)}).$$

Therefore, $a_{ik} = 1$ and $b_{kj} = 1$ and, consequently, $a_{ik} b_{kj} = 1$. $\square$

Let $[F]^r = [F] \star \cdots \star [F]$ (r times). Then we have the following corollary.

Corollary 2.1.5. *$[F^r] \leqslant [F]^r$ for all $f \in \mathcal{P}$, and for any $r \geq 1$.*

Proof. Replace G by F in Theorem 2.1.4. □

2.1.3 Networks with acyclic dependency graph

As an easy application of our main result we now consider networks that have an acyclic dependency graph, that is, no feedback loops in their structure. For any $F : \mathbb{K}^n \to \mathbb{K}^n$ with acyclic dependency graph we can see easily that its adjacency matrix $[f]$ can be written (re-indexing variables if necessary) as a strictly triangular matrix, i.e., with zeros in the diagonal (otherwise, there would be loops in the graph).

$$\begin{bmatrix} 0 & 0 & 0 & \dots & 0 \\ * & 0 & 0 & \dots & 0 \\ * & * & 0 & \dots & 0 \\ \vdots & & & \ddots & \\ * & * & \dots & * & 0 \end{bmatrix}$$

Corollary 2.1.6. *Any discrete dynamical system $F : \mathbb{K}^n \to \mathbb{K}^n$ with acyclic dependency graph has a unique fixed point.*

Proof. Let r be the nilpotency index of $[f]$. Then by Corollary 2.1.5, $[f^r] \leqslant [f]^r = 0$ so f^r is constant: $f^r \equiv x_0$. Therefore, x_0 is the unique fixed point of f.

□

A similar result to Corollary 2.1.6 is also true for discrete time continuous space dynamical systems and for ODE systems with acyclic dependency graphs, see [2].

2.1.4 Application

In this section we will present an application of Corollary 2.1.6 to a published model of an *in vitro* competition between two strains of a murine coronavirus studied in Jarrah et. al. [28]. The model is presented as a hexagonal grid of cells, with color coding of cells to indicate their infection status. Normal cells are represented as white. Infected cells are represented as red or green, or yellow, in the case of dual infection. The infection spreads from the center outwards. At each time step one ring of new cells is infected. The outcome of a cell in the new ring depends on the infection status of its two neighbors in the previous infected ring. The local update function for each cell is constructed using the following rules:

- If a cell has only one infected neighbor, it will get the same type of infection.
- If a cell has two infected neighbors, then we use the following table to determine the type of infection of that cell.

Rules for the update function				
	Green	White	Red	Yellow
Green	Green	Green	Yellow	Green
White	Green	White	Red	Yellow
Red	Yellow	Red	Red	Red
Yellow	Green	Yellow	Red	Yellow

The dynamics of this system is represented in Figure 2.1. In this example we will use 169 cells. In order to simulate experimental conditions described in [28] we initialize the infection using the 37 center cells. Each cell can have only one of four colors at a time. We use the field with four elements, $\mathbb{F}_4 = \{0, 1, 2, 3\}$, to represent the set of different colors. The color assignment is as follows.

Color assignment	
Color	Field element
Green	0
Red	1
White	2
Yellow	3

We represent the 169 cells by the variables $x_1, \ldots, x_{169}$, with $x_1, \ldots, x_{37}$ representing the center cells. The variables $x_{128}, \ldots, x_{169}$ represents the cells in the outermost ring.

In [28] the system was studied from the point of view of the experimental system. There, a collection of cells was infected in the center of the dish, and the infection was then observed to spread to the rest of the Petri dish in a pattern that showed distinct segmentation, matching the experimental results. In order to mimic the biological system, it was assumed that the cells initially infected did not change their infection state subsequently, so that the model is constrained and heterogeneous with respect to the rules assigned to all the nodes. This, in effect, changes the dynamical system since those cells are now assigned constant functions rather than the rules described above. The outcome is a steady state that shows a characteristic segmentation which coincides with experimental observations. See Figure 2.1 for an example. In this case Corollary 2.1.6 applies, since the dependency graph of the network is acyclic. See Figure 2.2. If we now assume that a larger number of cells in the center is infected, that is, is assigned a color that does not change subsequently, as in the initial experiments, then the color distribution along the edge of the infected region propagates and produces a steady state that show the segmentation patterns observed in [28].

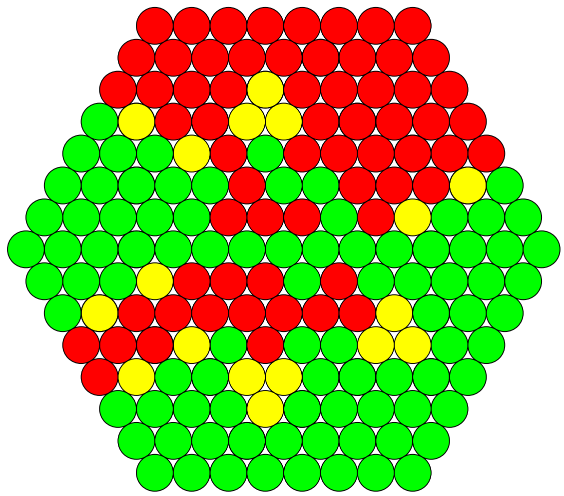


Figure 2.1: Dynamics of the constrained model

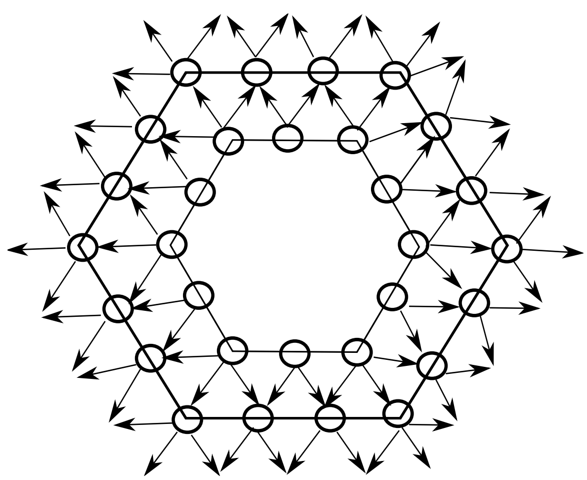
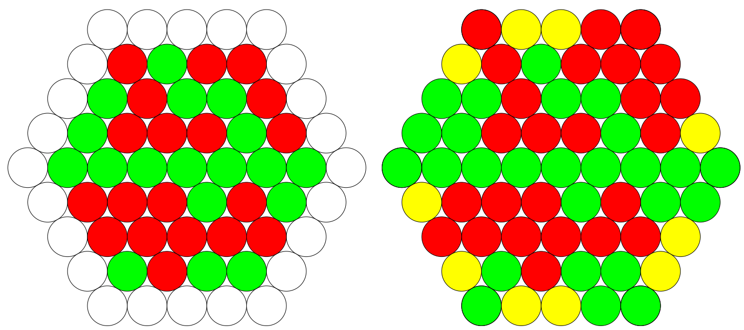
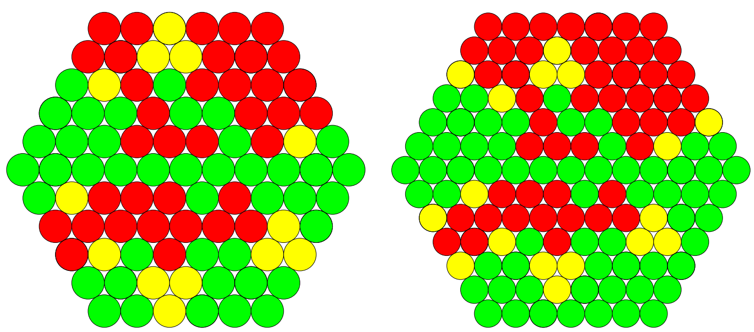


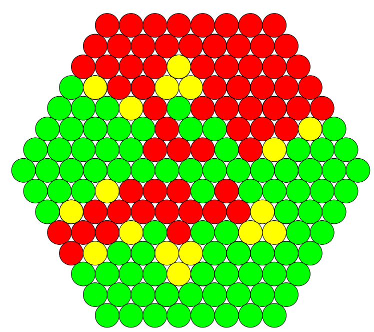
Figure 2.2: Adjacency graph for virus competition



(a) Initial State and first iteration.



(b) Subsequent iterations.



(c) Final outcome.

Figure 2.3: Dynamics where cells evolve according to the rules. Each initialization of infected cells determines the final (unique) outcome.

2.1.5 Conclusions

In this paper we have shown that it is fruitful to study the relationship between the structure and the dynamics of discrete dynamical systems by looking at the algebraic properties of the mapping $\mathcal{P} \longrightarrow \mathcal{M}$ from the algebra of PDS to the algebra of adjacency matrices. With a very straightforward proof we have shown that dynamics is very simple in the absence of feedback loops. It is worth observing that this result, Corollary 2.1.6, could also have been obtained as a consequence of a more general result that says that for the existence of more than one fixed point, a positive feedback loop is required in the dependency graph, and for periodic orbits to exist a negative feedback is necessary (but not sufficient). This result implies that in order to obtain more than one fixed point or periodic orbits, it is necessary that the dependency graph of the system have feedback loops [13]. However, the proof we have given here of this same result is very simple and stems from a basic property of the mapping $\mathcal{P} \longrightarrow \mathcal{M}$ rather than complicated phase space arguments. It emphasizes our belief that the proper framework for studying the relationship between structure and dynamics of PDS is the algebra inherent in this mapping.

Furthermore, we have shown that one can use Corollary 2.1.6 to draw non-obvious conclusions about a system of interest. This conclusion would be very difficult to arrive at through simulations, due to the combinatorial complexity of the dynamics on a large grid.

2.2 Attractor Structure of Bi-Threshold Networks

This section contains results about dynamics of synchronous bi-threshold networks. Further analysis of bithreshold networks and results for asynchronous bithreshold networks can be found in [4]

2.2.1 Synchronous Bi-Threshold Networks

Let $\mathbb{K} = \{0, 1\}$ as before, let $A = (a_{ij})$ be a real-valued symmetric matrix, let $(k_i^{\uparrow})_{i=1}^n$ and $(k_i^{\downarrow})_{i=1}^n$ be vertex-indexed sequences of up- and down-thresholds, and define the function $F = (f_1, \ldots, f_n)\colon \mathbb{K}^n \longrightarrow \mathbb{K}^n$ by

$$f_i(x_1, \ldots, x_n) = \begin{cases} 1 & \text{if } x_i = 0 \text{ and } \sum_{j=1}^n a_{ij}x_j \geq k_i^{\uparrow} \\ 0 & \text{if } x_i = 1 \text{ and } \sum_{j=1}^n a_{ij}x_j < k_i^{\downarrow} \\ x_i & \text{otherwise.} \end{cases} \tag{2.2.1}$$

The following theorem is a generalization of a result for standard threshold functions (main result of [15]) to the case of bi-threshold functions.

Theorem 2.2.1. *If $F : \mathbb{K}^n \to \mathbb{K}^n$ is a synchronous bithreshold network, i.e., each coordinate function is given as in Equation* (2.2.1)*, then for all $x \in \mathbb{K}^n$, there exists $s \in \mathbb{N}$ such that $F^{s+2}(x) = F^s(x)$.*

Proof. See the proof of this theorem in [4]. □

An immediate consequence of Theorem 2.2.1 is the following:

Corollary 2.2.2. *A synchronous bi-threshold network may only have fixed points and 2-cycles as attractors.*

Chapter 3

Multi-state Nested Canalyzing Functions

Understanding design principles of molecular interaction networks is an important goal of molecular systems biology. Some insights have been gained into features of their network topology through the discovery of graph theoretic patterns that constrain network dynamics. This chapter discusses the identification of patterns in the mechanisms that govern network dynamics. The control of nodes in gene regulatory, signaling, and metabolic networks is governed by a variety of biochemical mechanisms, with inputs from other network nodes that act additively or synergistically. This chapter focuses on a certain type of logical rule that appears frequently as a regulatory pattern. Within the context of the multi-state discrete model paradigm, a rule type is introduced that reduces to the concept of nested canalyzing function in the Boolean network case. It is shown that networks that employ this type of multivalued logic exhibit more robust dynamics than random networks, with few attractors and short limit cycles. It is also shown that the majority of regulatory functions in many published models of gene regulatory and signaling networks are nested canalyzing.

This chapter is based on published paper [16]. My contribution to this paper were: 1. Played a major role on the design of the study. 2. Model construction and analysis of results. 3. Part of the writing of the manuscript.

3.1 Introduction

Elucidating the large-scale graph structure of complex molecular interaction networks, from transcriptional networks [18] to protein-protein interaction networks [46] and metabolic network [23] is an important step toward an understanding of design principles of the cellular architecture. For instance, it has been shown that certain graph theoretic network patterns are much more prevalent in such networks than could be expected. See, e.g., [35]. The next step is to understand cells as complex nonlinear dynamical systems. There now exist many dynamic models of gene regulatory, signaling, and metabolic pathways that provide snapshots of cellular dynamics using a range of modeling platforms. Many of these models represent the interactions of different molecular species as logical rules of some type that describe the combinatorics of how the species combine to regulate others; see, e.g., [36, 33]. The logical rules of Boolean network models are an example of such a description, in which network states are reduced to binary states, with a species either present or absent. It was shown in [51, 52] that a special type of Boolean logical rule which appears frequently in published Boolean network models [27] exhibits robustness properties characteristic of molecular networks. These rules, so-called *nested canalyzing functions*, capture the spirit of Waddington's concept of canalization in gene regulation [57]. Several other classes of Boolean functions have also been investigated in the search for biologically meaningful rules to describe molecular interactions, including random functions [29], hierarchical canalyzing function [41, 55], chain functions [24], and unate functions [26].

In many cases the regulatory relationships are too complicated to be captured with Boolean logic, and more general models have been developed to represent these. Common other discrete model types, in addition to Boolean networks, are so-called logical models [65], Petri nets [40], and agent-based models [38]. In [44] and [9] it was shown that many of these models can be translated into the rich and general mathematical framework of *polynomial dynamical systems over a finite field* $\mathbb{F}$. (Software to carry out this translation is available at `http://dvd.vbi.vt.edu/adam.html` [83]). Since the algebraic structure of $\mathbb{F}$ is not relevant for our purposes, we will consider a slightly more general setup. Let $x_1, \ldots, x_n$ be variables, which can take values in finite sets $X_1, \ldots, X_n$, respectively. Let $X = X_1 \times \cdots \times X_n$ be the Cartesian product. A dynamical system in the variables $x_1, \ldots, x_n$ is a function

$$f = (f_1, \ldots, f_n) : X \to X$$

where each coordinate function f_i is a function on a subset of $\{x_1, \ldots, x_n\}$, and takes on values in X_i. Dynamics is generated by iteration of f. As an example, if $X_i = \{0, 1\}$, then each f_i is a Boolean rule and f is a Boolean network.

Here, we use this very general framework to give a definition of the notion of *nested canalyzing rule*, which then applies to all different model types simultaneously. We show through extensive simulations that dynamical systems constructed from such rules as coordinate functions have important dynamic properties characteristic of molecular networks, namely very short limit cycles and very few attractors, compared with the set of all possible functions. Furthermore, we show that many published models use logical interaction rules whose polynomial form is nested canalyzing, thereby providing evidence that general nested canalyzing rules represent a frequently occurring pattern in molecular network regulation.

3.2 Nested Canalyzing Rules

Here we present the general definition of a nested canalyzing rule in variables $x_1, \ldots, x_n$ with state space $X = X_1 \times \cdots \times X_n$.

Definition. Assume that each X_i is totally ordered, that is, its elements can be arranged in linear increasing order. In the Boolean case this could be $X_i = \{0 < 1\}$. Let $S_i \subset X_i, i = 1, \ldots, n$, be subsets that satisfy the property that each S_i is a proper, nonempty subinterval of X_i, that is, every element of X_i that lies between two elements of S_i in the chosen order is also in S_i. Furthermore, we assume that the complement of each S_i is also a subinterval, that is, each S_i can be described by a threshold s_i, with all elements of S_i either larger or smaller than s_i.

- The function $f_i : X \to X_i$ is a *nested canalyzing rule* in the variable order $x_{\sigma(1)}, \ldots, x_{\sigma(n)}$ with *canalyzing input sets* $S_1, \ldots, S_n \subset X$ and *canalyzing output values* $b_1, \ldots, b_n, b_{n+1} \in X_i$ with $b_n \neq b_{n+1}$ if it can be represented in the form:

$$f(x_1, \ldots, x_n) = \begin{cases} b_1 \text{ if } x_{\sigma(1)} \in S_1 \\ b_2 \text{ if } x_{\sigma(1)} \notin S_1, x_{\sigma(2)} \in S_2 \\ b_3 \text{ if } x_{\sigma(1)} \notin S_1, x_{\sigma(2)} \notin S_2, x_{\sigma(3)} \in S_3 \\ \vdots \\ b_n \text{ if } x_{\sigma(1)} \notin S_1, \ldots, x_{\sigma(n)} \in S_n \\ b_{n+1} \text{ if } x_{\sigma(1)} \notin S_1, \ldots, x_{\sigma(n)} \notin S_n \end{cases}$$

- The function $f_i : X \to X_i$ is a nested canalyzing function if it is a nested canalyzing function in some variable order $x_{\sigma(1)}, \ldots, x_{\sigma(n)}$ for some permutation σ on $\{1, \ldots, n\}$.

It is straightforward to verify that, if $X_i = \{0, 1\}$ for all i, then we recover the definition in [51] of a Boolean nested canalyzing rule. As mentioned above, several important classes of multi-state discrete models can be represented in the form of a dynamical system $f : X \longrightarrow X$, so that the concept of a nested canalyzing rule defined in this way has broad applicability.

3.3 The dynamics of nested canalyzing networks

Aside from incorporating the biological concept of canalyzation, networks whose nodes are controlled by combinatorial logic expressed by nested canalyzing rules have dynamic properties resembling those of biological networks. In particular, they are robust, due to the fact that they have a small number of attractors, which are therefore large. That is, perturbations are more likely to remain in the same attractor. In addition, limit cycles tend to be very short, compared to random networks, which implies that these networks have very regular behavior. We have performed extensive simulation experiments for this purpose, whose results we report here.

Computer experiments

We have generated random network topologies with in-degree distribution k ranging between 2 and 5, i.e., each node depends on at least two inputs and at most on five inputs. This assumption is not unrealistic and is based on the observation that gene regulation networks are sparse [32]. For each network topology we have generated two discrete dynamical systems: one where the update rules are all nested canalyzing and the other where the update rules are randomly chosen (and non-degenerate, in the sense that all inputs indicated in the network topology are realized).

Attractor distribution for nested canalyzing networks versus random networks

We present our results concerning attractor distributions in Figure 3.1. For each histogram, on the x-axis we represent the number of attractors, and on the y-axis we represent the percentage of networks that show the given number of attractors specified on the x-axis. The parameters N, k and p correspond to the number of nodes, the range for the in-degree distribution, and the number of states for each node respectively. For the experiments performed here we have generated the in-degree distribution from a uniform distribution, independently for each node and network realization. Figure 3.1 shows very clearly that the number of attractors in nested canalyzing networks is dramatically smaller than for general networks. Thus, attractor sizes are larger on average than in general networks, leading to more robust behavior under perturbations.

Cycle length distribution for nested canalyzing networks versus random networks

We present the results concerning cycle lengths in Figures 3.2 - 3.5. For each figure, the upper subfigures show the mean number of attractors of length specified on the x-axis (solid lines) and their standard deviations (dashed lines), the x-axis of these subfigures has a logarithmic scale. The bottom left subfigures show the mean number of attractors of lengths specified on the x-axis in a log-log plot, and finally the bottom right subfigures show the percentage of networks that returned a particular cycle of length specified on the x-axis. The parameters N, k and p correspond to the number of nodes, the range for the in-degree distribution, and the number of states for each node, respectively. For all of the experiments, we have generated the in-degree distribution from a uniform distribution, independently for each node and network realization. Figures 3.2 - 3.5 show clearly that networks with nested canalyzing rules exhibit significantly smaller cycle lengths, leading to more regular behavior.

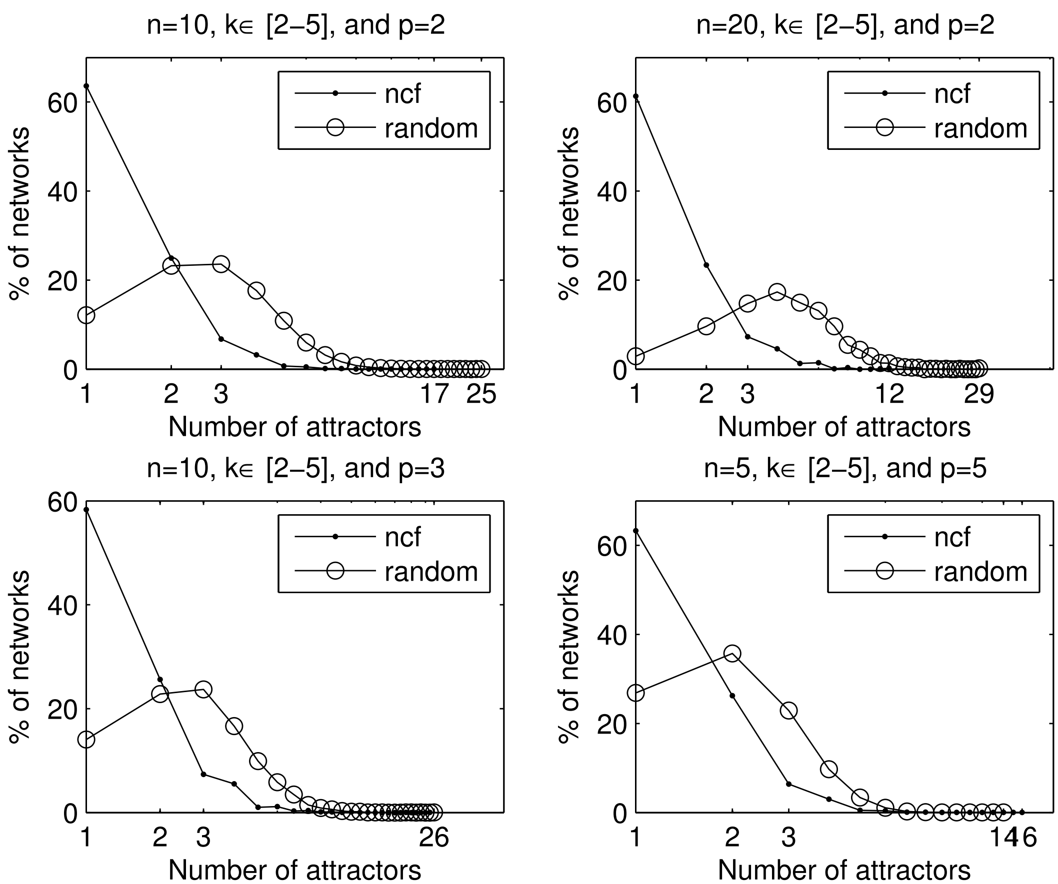


Figure 3.1: Attractor distribution for networks with nested canalyzing functions (solid circles) and networks with random functions (open circles). The figures show the percentage of networks that returned the number of attractors specified in the x-axis. The parameters N, k and p correspond to the number of nodes, the range for the in-degree distribution, and the number of states for each node, respectively. The figures for $n = 5, 10$ and $p = 2, 5$ were generated for 1000000 networks, the figure for $n = 10$ and $p = 3$ was generated for 100000 networks, and the figure for $n = 20$ and $p = 2$ was generated for 10000 networks.

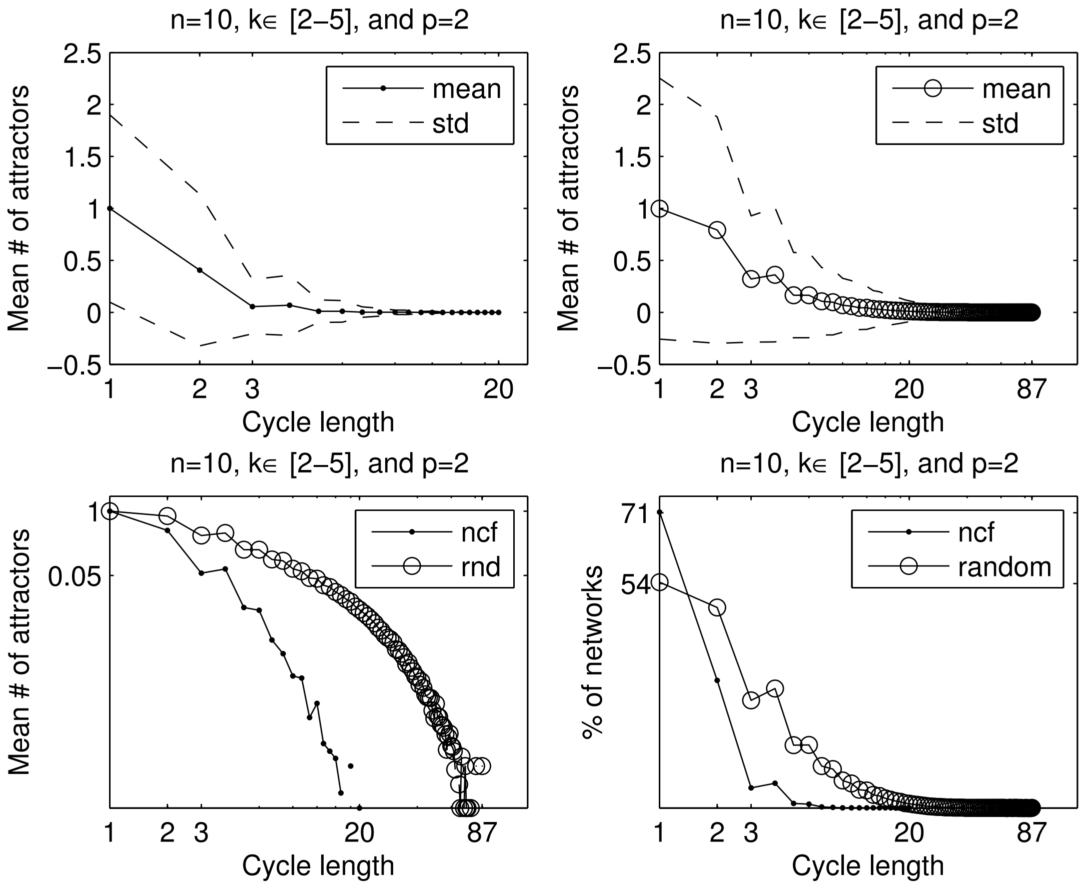


Figure 3.2: Cycle length for networks with nested canalyzing functions (solid circles) and networks with random functions (open circles). The parameters N, k and p correspond to the number of nodes, the range for the in-degree distribution, and the number of states for each node, respectively. The figures were generated for one million networks. The upper figures show the mean number of attractors of length specified in the x-axis (solid lines) and their standard deviations (dashed lines), the x-axis of these figures are in a logarithmic scale. The bottom left figure shows the mean number of attractors of lengths specified in the x-axis in a log-log plot (here rnd means random), and finally the bottom right figure shows the percentage of networks that returned a particular cycle of length specified on the x-axis.

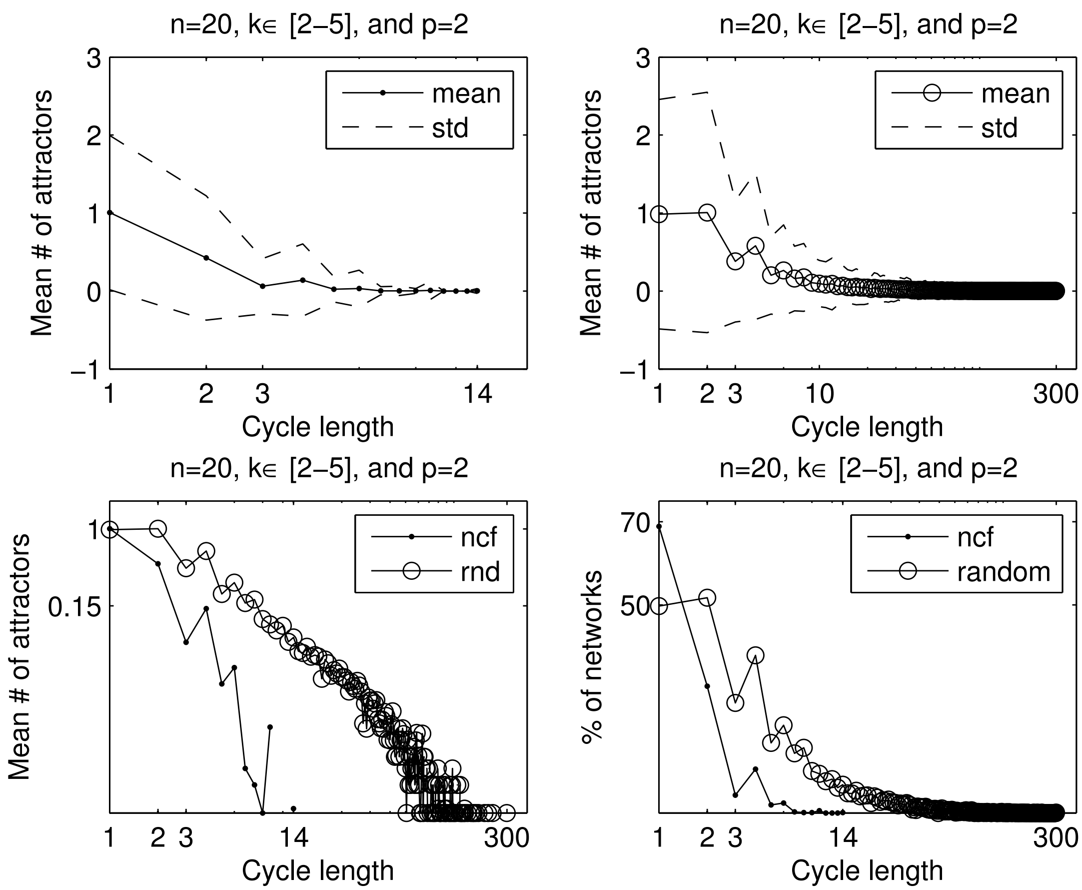


Figure 3.3: Cycle length for networks with nested canalyzing functions (solid circles) and networks with random functions (open circles). The parameters N, k and p correspond to the number of nodes, the range for the in-degree distribution, and the number of states for each node, respectively. The figures were generated for 10000 networks. The upper figures show the mean number of attractors of length specified on the x-axis (solid lines) and their standard deviations (dashed lines), the x-axis of these figures is a logarithmic scale. The bottom left figure shows the mean number of attractors of lengths specified on the x-axis in a log-log plot (here rnd means random), and finally the bottom right figure shows the percentage of networks that returned a particular cycle of length specified on the x-axis.

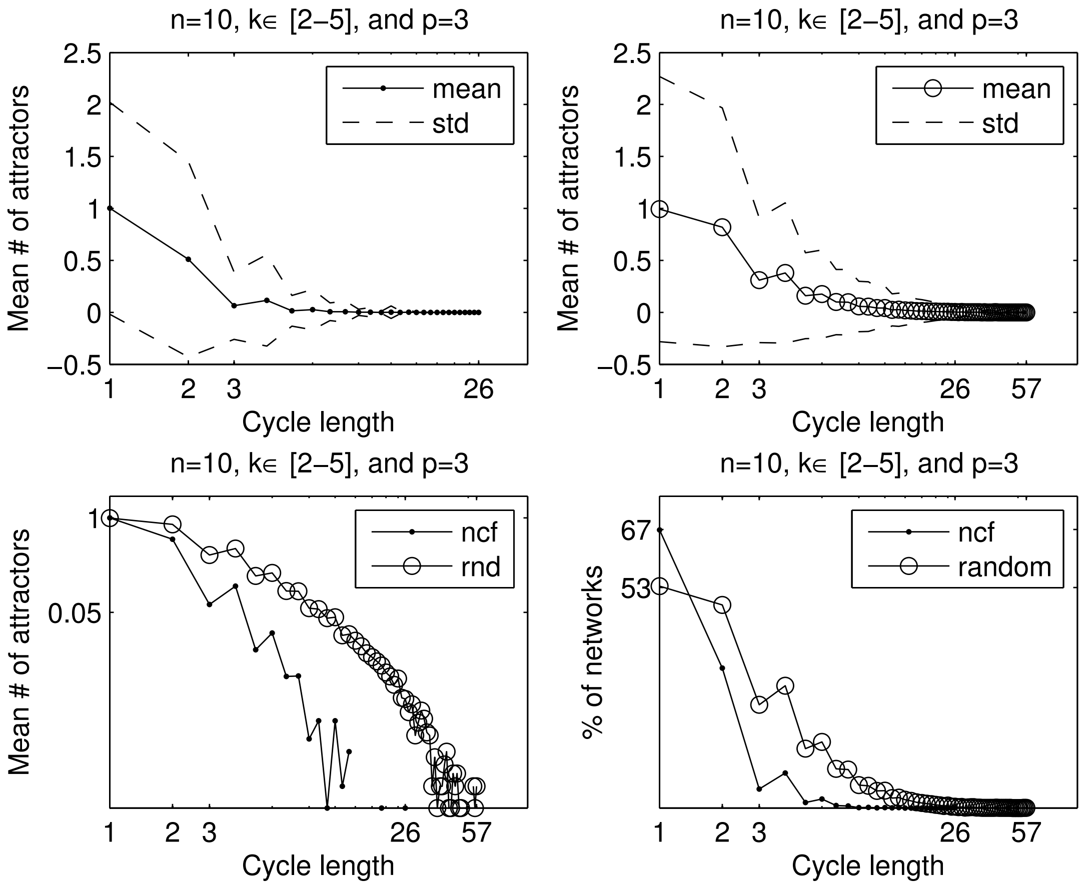


Figure 3.4: Cycle length for networks with nested canalyzing functions (solid circles) and networks with random functions (open circles). The parameters N, k and p correspond to the number of nodes, the range for the in-degree distribution, and the number of states for each node, respectively. The figures were generated for 100000 networks. The upper figures show the mean number of attractors of length specified on the x-axis (solid lines) and their standard deviations (dashed lines), the x-axis of these figures are in a logarithmic scale. The bottom left figure shows the mean number of attractors of lengths specified on the x-axis in a log-log plot (here rnd means random), and finally the bottom right figure shows the percentage of networks that returned a particular cycle of length specified on the x-axis.

Model	References	n [a]	% NCF[b]
lysis-lysogeny decision in the lambda phage	[87]	4	100%
p53-Mdm2 regulation	[84]	4	100%
Signalling pathways controlling Th cell differentiation	[36]	42	92.8%
Budding yeast exit module	[22, 39]	9	77.7%
Dorsal-ventral boundary formation of the Drosophila wing imaginal disc	[25]	24	75%
Control of Th1/Th2 cell differentiation	[34, 20]	14	71.4%
Yeast morphogenetic checkpoint	[22, 21]	8	50%

Table 3.1: Nested canalyzing functions for multi-state models

[a] Number of nodes. Only nodes with in-degree ≥ 1 are considered, i.e. non-constant nodes.

[b] Percentage of nodes regulated by nested canalyzing functions.

3.4 Nested canalyzing rules are biologically meaningful

We hypothesize that nested canalyzing rules are biologically meaningful. To test this hypothesis we have explored a range of published multi-state models as to their frequency of appearance. Table 3.1 shows that they are indeed very prevalent, providing evidence that the nested canalyzation is indeed a common pattern for the regulatory logic in molecular interaction networks. To illustrate this phenomenon we discuss specific examples. For a complete list of models we have studied see the supporting materials.

3.5 Examples

Lambda Phage Regulation

Thieffry and Thomas [87] built a multi-state logical model for the core lambda phage regulatory network. This model encompasses the roles of the regulatory genes *CI*, *CRO*, *CII*, and *N*, see Figure B.2.

The state space for this model is specified by $[0, 2] \times [0, 3] \times [0, 1] \times [0, 1]$, that is, the first variable has three levels $\{0, 1, 2\}$, the second variable has four levels $\{0, 1, 2, 3\}$, and the third and fourth variables are Boolean.

The update rule for *CI*, f_{CI}, has inputs *CRO* and *CII* and is nested canalyzing in the variable order *CII*, *CRO*, with canalyzing input sets $S_1 = \{1\}$, and $S_2 = \{1, 2, 3\}$ and canalyzed

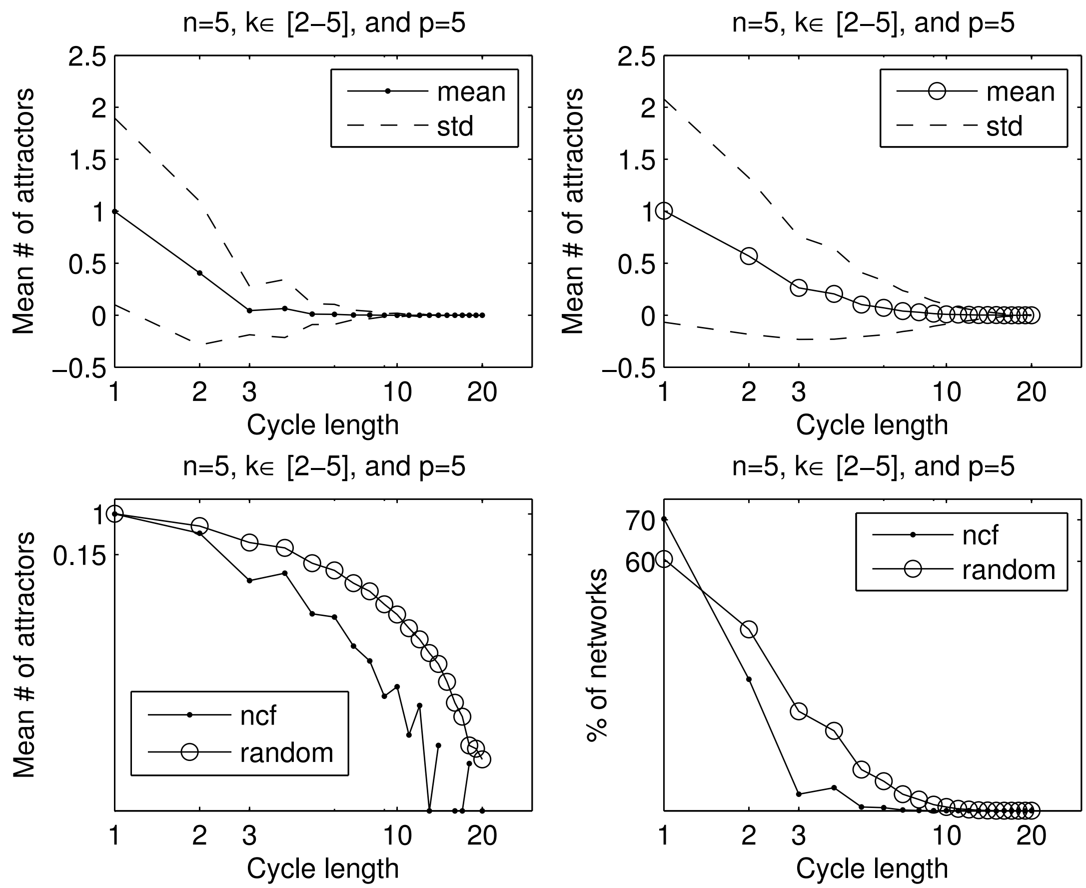


Figure 3.5: Cycle length for networks with nested canalyzing functions (solid circles) and networks with random functions (open circles). The parameters N, k and p correspond to the number of nodes, the range for the in-degree distribution, and the number of states for each node, respectively. The figures were generated for one million networks. The upper figures show the mean number of attractors of length specified on the x-axis (solid lines) and their standard deviations (dashed lines), the x-axis of these figures are in a logarithmic scale. The bottom left figure shows the mean number of attractors of lengths specified on the x-axis in a log-log plot (here rnd means random), and finally the bottom right figure shows the percentage of networks that returned a particular cycle of length specified on the x-axis.

output values $2, 0, 2$, i.e., (see the supporting materials for complete truth tables)

$$f_{CI}(CRO, CII) = \begin{cases} 2 \text{ if } CII \in S_1 \\ 0 \text{ if } CII \notin S_1, CRO \in S_2 \\ 2 \text{ if } CII \notin S_1, CRO \notin S_2. \end{cases}$$

The update rule for *CRO*, f_{CRO}, is nested canalyzing in the variable order *CI*, *CRO*, with canalyzing input sets $S_1 = \{2\}$, and $S_2 = \{0, 1, 2\}$, and canalyzed output values $0, 3, 2$, i.e.,

$$f_{CRO}(CI, CRO) = \begin{cases} 0 \text{ if } CI \in S_1 \\ 3 \text{ if } CI \notin S_1, CRO \in S_2 \\ 2 \text{ if } CI \notin S_1, CRO \notin S_2. \end{cases}$$

The update rule for *CII*, f_{CII}, is nested canalyzing in the variable order *CI*, *CRO*, *N*, with canalyzing input sets $S_1 = \{2\}$, $S_2 = \{3\}$, and $S_3 = \{1\}$, and canalyzed output values $0, 0, 1, 0$, i.e.,

$$f_{CII}(CI, CRO, N) = \begin{cases} 0 \text{ if } CI \in S_1 \\ 0 \text{ if } CI \notin S_1, CRO \in S_2 \\ 1 \text{ if } CI \notin S_1, CRO \notin S_2, N \in S_3 \\ 0 \text{ if } CI \notin S_1, CRO \notin S_2, N \notin S_3. \end{cases}$$

Finally, the update rule for *N*, f_N, is nested canalyzing in the variable order *CI*, *CRO*, with canalyzing input sets $S_1 = \{1, 2\}$, and $S_2 = \{2, 3\}$, and canalyzed output values $0, 0, 1$, i.e.,

$$f_N(CI, CRO) = \begin{cases} 0 \text{ if } CI \in S_1 \\ 0 \text{ if } CI \notin S_1, CRO \in S_2 \\ 1 \text{ if } CI \notin S_1, CRO \notin S_2. \end{cases}$$

3.5.1 Regulation in the p53-Mdm2 network

The following model comes from Abou-Jaude W., Ouattara A., Kauffman M.(2009) [84]. The model represents the interactions of the tumor supressor protein p53 and its negative regulator Mdm2. Here, P, Mn, Mc, and *Dam* stand for protein p53, nuclear Mdm2, cytoplasmic Mdm2, and DNA damage, respectively.

The state space for this model is specified by $[0, 2] \times [0, 1] \times [0, 1] \times [0, 1]$, that is, except for the first variable P that has three levels $\{0, 1, 2\}$, all the other variables are still Boolean.

As shown in Figure B.1, Mn acts negatively on P. The update rule of P, f_P, is nested canalyzing with canalyzing input set $S_1 = \{0\}$ and canalyzed output values $K_P, K_{P,\{Mn\}}$, i.e., we can represent f_P as,

$$f_P(Mn) = \begin{cases} K_p \text{ if } Mn \in S_1 \\ K_{p,\{Mn\}} \text{ if } Mn \notin S_1. \end{cases}$$

Here K_P is the basal value and $K_{P,\{Mn\}}$ is the parameter value under the influence of Mn.

Similarly, for Mc, its update rule, f_{Mc}, is nested canalyzing with canalyzing input set $S_1 = \{0, 1\}$ and canalyzed output values $K_{Mc}, K_{Mc,\{P\}}$, i.e., we can represent f_{Mc} as,

$$f_{Mc}(P) = \begin{cases} K_{Mc} \text{ if } P \in S_1 \\ K_{Mc,\{P\}} \text{ if } P \notin S_1. \end{cases}$$

Here K_{Mc} is the basal value and $K_{Mc,\{P\}}$ is the parameter value under the influence of P.

For Mn, a set of possible parameters for its truth table is given in [84]. We have checked all these cases and found that for each case we either get a nested canalyzing function or a constant function. For example, for the second column of Figure 3 (a) in [84] we get that the update rule for Mn, f_{Mn}, is nested canalyzing in the variable order Mc, P with canalyzing input sets $S_1 = \{1\}$, $S_2 = \{1, 2\}$ and canalyzed output values $1, 0, 1$, i.e.,

$$f_{Mn}(P, Mc) = \begin{cases} 1 \text{ if } Mc \in S_1 \\ 0 \text{ if } Mc \notin S_1, P \in S_2 \\ 1 \text{ if } Mc \notin S_1, P \notin S_2. \end{cases}$$

When DNA damage is introduced, it has a negative effect on Mn. From the set of all possible parameters for its truth table given in inequalities (3)-(5) at [84], we check that we can always find a nested canalyzing function for its truth table. For example, for the third column of Figure 3 (a) in [84] we get that the update rule for Mn (under DNA damage) is nested canalyzing in the variable order Mc, Dam, P, with canalyzing input sets $S_1 = \{1\}$, $S_2 = \{1\}$, $S_3 = \{1, 2\}$ and canalyzed output values $1, 0, 0, 1$, i.e., we can represent f_{Mn} as,

$$f_{Mn}(P, Mc, Dam) = \begin{cases} 1 \text{ if } Mc \in S_1 \\ 0 \text{ if } Mc \notin S_1, Dam \in S_2 \\ 0 \text{ if } Mc \notin S_1, Dam \notin S_2, P \in S_3 \\ 1 \text{ if } Mc \notin S_1, Dam \notin S_2, P \notin S_3. \end{cases}$$

Finally, the update rule for DNA damage, f_{Dam}, is nested canalyzing in the variable order P, Dam, with canalyzing input sets $S_1 = \{2\}$, $S_2 = \{1\}$ and canalyzed output values $0, 1, 0$, i.e.,

$$f_{Dam}(P, Dam) = \begin{cases} 1 \text{ if } P \in S_1 \\ 1 \text{ if } P \notin S_1, Dam \in S_2 \\ 0 \text{ if } P \notin S_1, Dam \notin S_2. \end{cases}$$

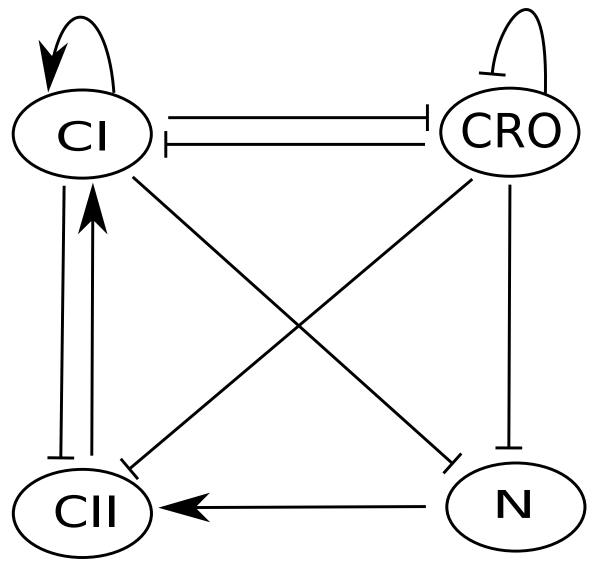


Figure 3.6: Four-variable model for the lambda phage regulatory network.

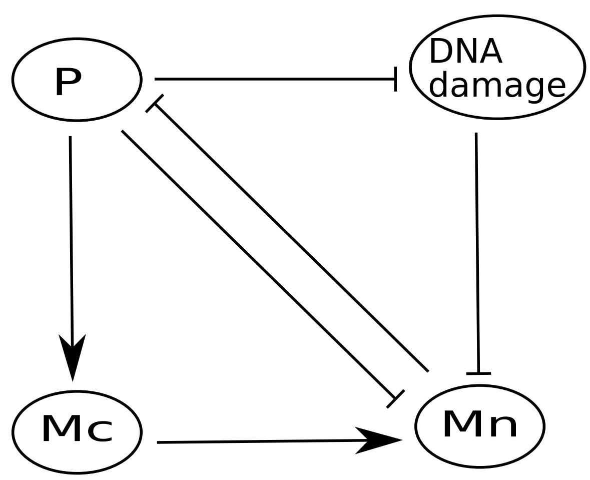


Figure 3.7: Four-variable model for the p53-Mdm2 regulatory network.

3.6 Discussion

In this chapter we have given a definition of a nested canalyzing rule, inspired by the special case of Boolean networks, and we have shown that it appears as a frequent pattern for the regulatory logic of many molecular interaction networks. We have shown that this regulatory pattern leads to networks that have robust and regular dynamics, as a result of having very small numbers of attractors and very short limit cycles, compared to random networks. This behavior is also characteristic of many molecular interaction networks. An important application of this result is to the construction of discrete models, both via a bottom-up or a top-down approach. For both approaches, the possibility of restricting the choice of rules to the family of nested canalyzing rules is a significant reduction in the possible model space that is available.

Another interesting aspect of our results is suggested by [28]. There is was shown that in the Boolean case, the class of nested canalyzing Boolean rules is in fact identical to the class of unate cascade functions. These functions have been studied extensively in computer engineering and have been shown [48] to have the important property that they comprise exactly the class of Boolean functions that lead to binary decision diagrams with shortest average path length. Thus, they make good candidates for the representation of efficient information processing. It would be interesting to study this property for general nested canalyzing rules.

The results in [28] were derived by using a special representation of Boolean functions, namely as polynomial functions over the Boolean number field, with arithmetic given by addition and multiplication "tables," with the key rule that $1+1=0$. Using a parametrization of the family of all nested canalyzing Boolean polynomials, it was shown in [28] that the family of all nested canalyzing polynomials in a given number of variables is in fact identical to the class of unate cascade functions, which, in turn, is equal to the class of Boolean functions that result in binary decision diagrams of shortest average path length [48]. In a paper in preparation we have shown that one can give a similar parameterization of the variety of general nested canalyzing rules, and we use this parameterization to derive a formula for the number of nested canalyzing rules for a given number of variables. It would be interesting to investigate whether this general class of nested canalyzing rules leads to N-ary decision diagrams that have similar properties to those of Boolean nested canalyzing rules.

Chapter 4

The Number of Multi-state Nested Canalyzing Functions

Identifying features of molecular regulatory networks is an important problem in systems biology. It has been shown that the combinatorial logic of such networks can be captured in many cases by special functions called nested canalyzing in the context of discrete dynamic network models. It was also shown that the dynamics of networks constructed from such functions has very special properties that are consistent with what is known about molecular networks, and that simplify analysis. It is important to know how restrictive this class of functions is, for instance for the purpose of network reverse-engineering. This chapter contains a formula for the number of such functions and a comparison to the class of all functions. In particular, it is shown that, as the number of variables becomes large, the ratio of the number of nested canalyzing functions to the number of all functions converges to zero. This shows that the class of nested canalyzing functions is indeed very restrictive. The principal tool used for this investigation is a description of these functions as polynomials and a parameterization of the class of all such polynomials in terms of relations on their coefficients. This parametrization can also be used for the purpose of network reverse-engineering using only nested canalyzing functions.

This chapter is based on published paper [47]. My contribution to this paper were: 1. Played a major role on the design of the study. 2. Formulation and proofs of theorems. 3. Part of the writing of the manuscript.

4.1 Introduction

A central problem of molecular systems biology is to understand the structure and dynamics of molecular networks, such as gene regulatory, signaling, or metabolic networks. Some progress has been made in elucidating general design principles of such networks. For instance, in [54] it was shown that certain graph theoretic motifs appear far more often in the topology of regulatory network graphs than would be expected at random. In [51, 52] it was shown that a certain type of Boolean regulatory logic, encoded by so-called *nested canalyzing* Boolean functions, has the kind of dynamic properties one would expect from molecular networks. In [16] we showed that the Boolean functions studied there do have a multi-state generalization that shows similar dynamic properties. Furthermore, we showed that the large majority of regulatory rules that appear in published models of molecular networks, whether Boolean or multi-state, do indeed have this form. Thus, there is evidence that multi-state nested canalyzing rules capture key features of molecular regulation and deserve further study.

These rules, so-called *nested canalyzing* rules, are a special case of *canalyzing* rules, which are reminiscent of Waddington's concept of canalyzation in gene regulation [57]. Nested canalyzing Boolean rules were shown in [28] to be identical with the class of unate cascade functions, which have been studied extensively in computer engineering. They represent exactly the class of Boolean functions that result in binary decision diagrams of shortest average path length [48]. This in itself has interesting implications for information processing in molecular networks. One consequence of this result is that a recursive formula derived earlier for the number of unate cascade functions of a given number of variables [56] applies to give a formula for the number of nested canalyzing Boolean functions, described in [28]. A formula for the number of canalyzing Boolean functions had been given in [49].

Many molecular networks cannot be described using the Boolean framework, since more than one threshold for a molecular species might be required to represent different modes of action. There are several frameworks available for multi-state discrete models, such as so-called logical models, Petri nets, and agent-based models. It has been shown in [44] and [9] that all these model types can be translated into the general and mathematically well-founded framework of polynomial dynamical systems over a finite number system. In [16] the concept of nested canalyzing logical rule has been generalized to such polynomial systems. It has been shown there, furthermore, that a large proportion of rules in multi-state discrete models are indeed nested canalyzing, showing that this concept captures an important feature of the regulatory logic of molecular networks.

As was pointed out in [49] and [28], knowing the number of nested canalyzing rules for a given number of input variables and for a given number of possible variable states is important because on the one hand it provides an estimate of how plausible it is that such rules have evolved as regulatory principles and, on the other hand, provides an estimate of how restrictive the set of rules is. The latter is important, for instance, for the reverse-engineering of networks [50]. If the set of rules is sufficiently restrictive, then the reverse-engineering problem, which is almost always underdetermined due to limited data, becomes more tractable when restricted to a smaller model space. Knowing the proportion of regulatory rules that are nested canalyzing gives an estimate of how much one can additionally constrain the problem by limiting the search space of reverse-engineering algorithms to these rules. That this is indeed the case is an important consequence of the results in this paper. We present a formula for the number of nested canalyzing functions in a given number of variables and show that the ratio of nested canalyzing functions and all multi-state functions converges to zero as the number of variables increases. We follow the approach in [28] and solve the problem within the framework of polynomial dynamical systems, which makes it possible to frame it as a problem of counting solutions to a system of polynomial equations.

4.2 Nested Canalyzing Functions

As mentioned in the previous section, it is possible to view most discrete models within the framework of dynamical systems over a finite number system, or finite field. For our purposes we will use the finite fields $\mathbb{F}_p = \{0, 1, \dots, p-1\}$, p an arbitrary prime number, otherwise known as $\mathbb{Z}/p$, the integers modulo p. Furthermore, we will assume that $\mathbb{F}_p$ is totally ordered under the canonical order, that is, its elements are arranged in linear increasing order, $\mathbb{F}_p = \{0 < 1 < \cdots < p-1\}$. Let $\mathbb{F} = \mathbb{F}_p$ for some prime p. We first recall the general definition of a nested canalyzing function in variables $x_1, \dots, x_n$ from [**?**]. The underlying idea is as follows: A rule is nested canalyzing, if there exists a variable x such that, if x receives certain inputs, then it by itself determines the value of the function. If x does not receive these certain inputs, then there exists another variable y such that, if y receives certain other inputs, then it by itself determines the value of the function; and so on, until all variables are exhausted.

Definition 4.2.1. *Let $S_i \subset \mathbb{F}$, for $i = 1, \dots, n$, be subsets that satisfy the property that each S_i is a proper, nonempty subinterval of $\mathbb{F}$; that is, every element of $\mathbb{F}$ that lies between two elements of S_i in the chosen order is also in S_i. Furthermore, we assume that the complement of each S_i is also a subinterval, that is, each S_i can be described by a threshold s_i, with all elements of S_i either larger or smaller than s_i. Let σ be a permutation on $\{1, \dots, n\}$.*

- *The function $f : \mathbb{F}^n \to \mathbb{F}$ is a nested canalyzing function in the variable order $x_{\sigma(1)}, \dots, x_{\sigma(n)}$ with canalyzing input sets $S_1, \dots, S_n \subset \mathbb{F}$ and canalyzing output values $b_1, \dots, b_n, b_{n+1} \in \mathbb{F}$, with $b_n \neq b_{n+1}$, if it can be represented in the form*

$$f(x_1, \dots, x_n) =$$

$$\begin{cases} b_1 \ \textit{if}\ x_{\sigma(1)} \in S_1, \\ b_2 \ \textit{if}\ x_{\sigma(1)} \notin S_1, x_{\sigma(2)} \in S_2, \\ \vdots \\ b_n \ \textit{if}\ x_{\sigma(1)} \notin S_1, \dots, x_{\sigma(n)} \in S_n, \\ b_{n+1} \ \textit{if}\ x_{\sigma(1)} \notin S_1, \dots, x_{\sigma(n)} \notin S_n. \end{cases}$$

- *The function $f : \mathbb{F}^n \to \mathbb{F}$ is a nested canalyzing function if it is a nested canalyzing function in some variable order $x_{\sigma(1)}, \dots, x_{\sigma(n)}$ for some permutation σ on $\{1, \dots, n\}$.*

It is straightforward to verify that, if $p = 2$, that is $\mathbb{F} = \{0, 1\}$, then we recover the definition in [51] of a Boolean nested canalyzing rule. We emphasize that several important classes of multi-state discrete models can be represented in the form of a dynamical system $f : \mathbb{F}^n \longrightarrow \mathbb{F}^n$, as mentioned above, so that the concept of a nested canalyzing rule defined in this way has broad applicability.

Example 4.2.2. *Let $\mathbb{F}$ be the field with three elements, i.e. $\mathbb{F} = \{0, 1, 2\}$. The function $f : \mathbb{F}^2 \to \mathbb{F}$ given by*

$$f(x_1, x_2) = \begin{cases} 1 & \text{if } x_1 \in \{0,1\}, \\ 2 & \text{if } x_1 \notin \{0,1\}, x_2 \in \{2\}, \\ 0 & \text{if } x_1 \notin \{0,1\}, x_2 \notin \{2\}, \end{cases}$$

is nested canalyzing in the variable order x_1, x_2, with canalyzing input sets $S_1 = \{0, 1\}, S_2 = \{2\} \subset \mathbb{F}$, and canalyzing output values $b_1 = 1, b_2 = 2, b_3 = 0 \in \mathbb{F}$.

4.3 Polynomial form of nested canalyzing functions

We now use the fact that any function $f : \mathbb{F}^n \to \mathbb{F}$ can be expressed as a polynomial in n variables [53, p. 369]. In this section we determine the polynomial form of nested canalyzing functions. That is, we will determine relationships among the coefficients of a polynomial that make it nested canalyzing. We follow the approach in [28]. Let B_n be the set of functions from $\mathbb{F}^n$ to $\mathbb{F}$, i.e., $B_n = \{f : \mathbb{F}^n \longrightarrow \mathbb{F}\}$. The set B_n is endowed with an addition and multiplication that is induced from that of $\mathbb{F}$, which makes it into a ring. Let I be the ideal of the ring of polynomials $\mathbb{F}[x_1, \ldots, x_n]$ generated by the polynomials $\{x_i^p - x_i\}$ for all $i = 1, \ldots, n$, where p is the number of elements in $\mathbb{F}$. There is an isomorphism between B_n and the quotient ring $\mathbb{F}[x_1, \ldots, x_n]/I$ which is also isomorphic to

$$R = \left\{ \sum_{\substack{(i_1,\ldots,i_n) \\ i_t \in \mathbb{F} \\ t=1,\ldots,n}} C_{i_1 \ldots i_n} x_1^{i_1} x_2^{i_2} \cdots x_n^{i_n} \right\}$$

Now we use this identification to study nested canalyzing functions as elements of R.

Given a subset S of $\mathbb{F}$, we will denote by Q_S the indicator function of the complement of S, i.e., for $x_0 \in \mathbb{F}$, let

$$Q_S(x_0) = \begin{cases} 0 & \text{if } x_0 \in S, \\ 1 & \text{if } x_0 \notin S. \end{cases}$$

We will derive the polynomial form for $Q_S(x)$ in Lemma A.1.2. The following theorem gives the polynomial form of a nested canalyzing function.

Theorem 4.3.1. *Let f be a function in R. Then the function f is nested canalyzing in the variable order $x_1, \ldots, x_n$ with canalyzing input sets $S_1, \ldots, S_n$ and canalyzing output values $b_1, \ldots, b_n, b_{n+1}$, with $b_n \neq b_{n+1}$, if and only if it has the polynomial form*

$$f(x_1, \ldots, x_n) = \sum_{j=0}^{n-1} \left\{ B_{n-j} \prod_{i=1}^{n-j} Q_{S_i}(x_i) \right\} + b_1, \tag{4.3.1}$$

where Q_{S_i} is defined as in Lemma A.1.2 and $B_{n-j} = (b_{n-j+1} - b_{n-j})$ for $j = 0, \ldots, n-1$.

Proof. Let f be a nested canalyzing function as in Definition 4.2.1, and let

$$g(x_1, \ldots, x_n) = \sum_{j=0}^{n-1} \left\{ B_{n-j} \prod_{i=1}^{n-j} Q_{S_i}(x_i) \right\} + b_1.$$

Since g has the right form to be in R, we can use the isomorphism between B_n and R, to reduce the proof to showing that

$$g(a_1, \ldots, a_n) = f(a_1, \ldots, a_n)$$

for all $(a_1, \ldots, a_n) \in \mathbb{F}^n$.

If $a_1 \in S_1$, then $Q_{S_1}(a_1) = 0$, therefore

$$g(a_1, \ldots, a_n) = b_1 \text{ whenever } a_1 \in S_1.$$

If $a_1 \notin S_1$ and $a_2 \in S_2$, then $Q_{S_1}(a_1) = 1$ and $Q_{S_2}(a_2) = 0$, therefore

$$g(a_1, \ldots, a_n) = (b_2 - b_1) + b_1 = b_2.$$

Iterating this process, if $a_1 \notin S_1, a_2 \notin S_2, \ldots, a_n \in S_n$, then $Q_{S_1}(a_1) = 1$, $Q_{S_2}(a_2) = 1, \ldots,$ and $Q_{S_n}(a_n) = 0$, therefore

$$g(a_1, \ldots, a_n) = (b_n - b_{n-1}) + \cdots + (b_2 - b_1) + b_1 = b_n.$$

Finally, if $a_1 \notin S_1, \ldots, a_n \notin S_n$, then $Q_{S_1}(a_1) = 1$, $Q_{S_2}(a_2) = 1, \ldots,$ and $Q_{S_n}(a_n) = 1$. Therefore,

$$g(a_1, \ldots, a_n) = \\ (b_{n+1} - b_n) + \cdots + (b_2 - b_1) + b_1 = b_{n+1}$$

This completes the proof. ∎

Example 4.3.2. *The polynomial form of the nested canalyzing function in Example 4.2.2 is*

$$f(x_1, x_2) = Q_{S_1}(x_1) Q_{S_2}(x_2) + Q_{S_1}(x_1) + 1,$$

where

$$\begin{aligned} Q_{S_1}(x_1) &= 2(x_1^2 + 2x_1), \\ Q_{S_2}(x_2) &= x_2^2 + 2x_2 + 1. \end{aligned}$$

The expanded form of f is

$$\begin{aligned} f(x_1, x_2) &= 2x_1^2x_2^2 + x_1x_2^2 + x_1^2x_2 \\ &\quad +2x_1x_2 + x_1^2 + 2x_1 + 1. \end{aligned}$$

Theorem 4.3.3. *Let f be a function in R. Then the function f is nested canalyzing in the variable order $x_{\sigma(1)}, \dots, x_{\sigma(n)}$ with canalyzing input sets $S_1, \dots, S_n$ and canalyzing output values $b_1, \dots, b_n, b_{n+1}$, with $b_n \neq b_{n+1}$, if and only if it has the polynomial form*

$$f(x_1, \cdots, x_n) = \sum_{j=0}^{n-1} \left\{ B_{n-j} \prod_{i=1}^{n-j} Q_{s_{\sigma(i)}}(x_i) \right\} + b_1,$$

where $Q_{S_{\sigma(i)}}$ is defined as in Lemma A.1.2, and $B_{n-j} = (b_{n-j+1} - b_{n-j})$ for $j = 0, \dots, n-1$.

Proof. The proof is very similar to the proof of Theorem 4.3.1.

□

4.4 The algebraic variety of nested canalyzing functions

Here we derive a parametrization for the coefficients of any nested canalyzing function. We will use this parametrization to derive a formula to compute the number of nested canalyzing functions for a given number of variables and a given finite field in the next section.

Recall that elements of $B_n = \{f : \mathbb{F}^n \to \mathbb{F}\}$ can be seen as elements of

$$R = \left\{ \sum_{\substack{(i_1,\dots,i_n) \\ i_t \in \mathbb{F} \\ t=1,\dots,n}} C_{i_1 \dots i_n} x_1^{i_1} x_2^{i_2} \cdots x_n^{i_n} \right\}.$$

Now, as a vector space over $\mathbb{F}$, R is isomorphic to $\mathbb{F}^{p^n}$ via the correspondence

$$\sum_{\substack{(i_1,\dots,i_n) \\ i_t \in \mathbb{F} \\ t=1,\dots,n}} C_{i_1 \dots i_n} x_1^{i_1} x_2^{i_2} \cdots x_n^{i_n} \leftrightarrow (\dots, C_{i_1 \dots i_n}, \dots).$$

We will identify the set of nested canalyzing functions in R with a subset V^{ncf} of $\mathbb{F}^{p^n}$ by imposing relations on the coordinates of its elements. We are going to use the following notation:

Notation 4.4.1. *For $r \in \mathbb{F}$ and for $S \subset \mathbb{F}$,*

- $C_{[r]} = C_{r\dots r}$.
- $C^{j}_{[r]\setminus\{i\}} = C_{r\dots j\dots r}$ *where j goes in the i-th position.*
- $C^{i_1,\dots,i_j}_{[r]\setminus\{1,\dots,j\}} = C_{i_1,\dots,i_j,r\dots r}$ *where $i_1,\dots,i_j$ go in the $1,\dots,j$ positions, respectively.*
- $C_{i_1\dots i_{n-j}} = C_{i_1\dots i_{n-j}0\dots 0}$, *i.e., $i_{n-j} \neq 0$ and $i_s = 0$ for all $s > n-j$.*
- $S^c = \mathbb{F}\setminus S$, *i.e., S^c denote the complement of S.*

Theorem 4.4.2. *Let $f \in B_n$ be given by*

$$f(x_1,\dots,x_n) = \sum_{\substack{(i_1,\dots,i_n)\\ i_t\in\mathbb{F}\\ t=1,\dots,n}} C_{i_1\dots i_n} x_1^{i_1} x_2^{i_2}\cdots x_n^{i_n}. \tag{4.4.1}$$

The polynomial $f(x_1,\dots,x_n)$ is a nested canalyzing function in the variable order $x_1,\dots,x_n$ with canalyzing input sets $S_1,\dots,S_n$ and canalyzing output values $b_1,\dots,b_{n+1}$ if and only if its coefficients satisfy the following equations:

$$C_{i_1\dots i_{n-\mu}} = C^{p-1,\dots,p-1}_{[0]\setminus\{1,\dots,n-\mu\}} \prod_{j=1}^{n-\mu} C^{-1}_{[p-1]} C^{i_j}_{[p-1]\setminus\{j\}} \tag{4.4.2}$$

for $\mu = 0,\dots,n-1$, where

$$C_{[p-1]} = (b_{n+1}-b_n)\prod_{i=1}^{n} a^{i}_{p-1}, \tag{4.4.3}$$

$$C^{i_j}_{[p-1]\setminus\{j\}} = a^{j}_{i_j}(a^{j}_{p-1})^{-1} C_{[p-1]}, \tag{4.4.4}$$

for $j = 1,\dots,n-1$.

$$C^{p-1,\dots,p-1}_{[0]\setminus\{1,\dots,n-\mu\}} = \sum_{j=0}^{\mu}\Big\{B_{n-j}a^{1}_{p-1}\dots a^{n-\mu}_{p-1}a^{n-\mu+1}_{0}\dots a^{n-j}_{0}\Big\}, \tag{4.4.5}$$

$$C_{[0]} = C^{p-1}_{[0]\setminus\{1\}} C^{0}_{[p-1]\setminus\{1\}} C^{-1}_{[p-1]} + b_1 \tag{4.4.6}$$

where a^{j}_{p-1} and $a^{j}_{i_j}$ is defined as in Lemma A.1.2 and $B_{n-j} = (b_{n-j+1} - b_{n-j})$ for $j = 1,\dots,n-1$.

Example 4.4.3. *The parametrization for the polynomial in Example 4.3.2 is given by*

$$f(x_1, x_2) = \sum_{\substack{(i_1,i_2)\\ i_t \in \mathbb{F}\\ t=1,2}} C_{i_1 i_2} x_1^{i_1} x_2^{i_2} \leftrightarrow (C_{00}, C_{01} \dots, C_{22}),$$

where from Lemma A.1.2 we get

$$a_2^1 = (p-1) \mid S_1^c \mid = 2, \; a_2^2 = (p-1) \mid S_2^c \mid = 1,$$
$$a_1^1 = (2)(2^{2-1}) = 1, \; a_1^2 = (2)(0^{2-1} + 1^{2-1}) = 2,$$
$$a_0^1 = Q_{S_1}(0) = 0, \; a_0^2 = Q_{S_2}(0) = 1,$$

and from Equation 4.4.3 we get

$$\begin{aligned} C_{[2]} = C_{22} &= (b_3 - b_2) a_2^1 a_2^2 \\ &= 1(2)(1) \mod 3 \\ &= 2. \end{aligned}$$

Similarly, from Equation 4.4.4, we get

$$C_{21} = C^1_{[2]\setminus\{2\}} = 1, \; C_{12} = C^1_{[2]\setminus\{1\}} = 1,$$
$$\textit{and } C_{02} = C^0_{[2]\setminus\{1\}} = 0.$$

From Equation 4.4.5 we get

$$C_{20} = C^2_{[0]\setminus\{1\}} = B_2 a_2^1 a_0^2 + B_1 a_2^1 = 1.$$

From Equation 4.4.2 we get

$$\begin{aligned} C_{11} &= C^{22}_{[0]\setminus\{1,2\}}(C^{-1}_{[2]} C^1_{[2]\setminus\{1\}})(C^{-1}_{[2]} C^1_{[2]\setminus\{1\}}) \\ &= C_{12} C_{21} C_{22}^{-1} = 2, \end{aligned}$$

$$\begin{aligned} C_{01} &= C^{22}_{[0]\setminus\{1,2\}}(C^{-1}_{[2]} C^0_{[2]\setminus\{1\}})(C^{-1}_{[2]} C^1_{[2]\setminus\{2\}}) \\ &= C_{02} C_{21} C_{22}^{-1} = 0, \end{aligned}$$

$$\begin{aligned} C_{10} &= C^2_{[0]\setminus\{1\}}(C^{-1}_{[2]} C^1_{[2]\setminus\{1\}}) \\ &= C_{20} C_{12} C_{22}^{-1} = 2. \end{aligned}$$

Finally, from Equation 4.4.6 we get

$$\begin{aligned} C_{00} &= C^2_{[0]\setminus\{1\}} C^0_{[2]\setminus\{1\}} C^{-1}_{[2]} + b_1 \\ &= C_{20} C_{02} C_{22}^{-1} + b_1 = 0 \end{aligned}$$

Thus, the function $f(x_1, x_2)$ corresponds to the point $(0, 0, 0, 2, 2, 1, 1, 1, 2) \in \mathbb{F}^{3^2}$.

The proof of Theorem 4.4.2 is given in Appendix A.2. We now need to provide a similar parametrization for functions that are nested canalyzing with respect to an arbitrary variable ordering.

Theorem 4.4.4. *Let $f \in B_n$ be given by*

$$f(x_1, \ldots, x_n) = \sum_{\substack{(i_1,\ldots,i_n) \\ i_t \in \mathbb{F} \\ t=1,\ldots,n}} C_{i_1 \ldots i_n} x_1^{i_1} x_2^{i_2} \cdots x_n^{i_n}. \tag{4.4.7}$$

The polynomial $f(x_1, \ldots, x_n)$ is a nested canalyzing function in the variable order $x_{\sigma(1)}, \ldots, x_{\sigma(n)}$ with canalyzing input sets $S_1, \ldots, S_n$ and canalyzing output values $b_1, \ldots, b_{n+1}$ if and only if

$$\begin{aligned} &C_{i_1 \ldots i_{n-\mu}} = \\ &C^{p-1,\ldots,p-1}_{[0]\setminus\{\sigma(i_1)\ldots\sigma(i_{n-\mu})\}} \prod_{j=1}^{n-\mu} C^{-1}_{[p-1]} C^{i_j}_{[p-1]\setminus\{\sigma(j)\}} \end{aligned} \tag{4.4.8}$$

for $\mu = 0, \ldots, n-1$, where

$$C_{[p-1]} = (b_{n+1} - b_n) \prod_{i=1}^{n} a^{\sigma(i)}_{p-1}, \tag{4.4.9}$$

$$C^{i_j}_{[p-1]\setminus\{\sigma(j)\}} = a^{\sigma(j)}_{i_j} (a^{\sigma(j)}_{p-1})^{-1} C_{[p-1]}, \tag{4.4.10}$$

for $j = 1, \ldots, n-1$.

$$\begin{aligned} &C^{p-1,\ldots,p-1}_{[0]\setminus\{\sigma(i_1)\ldots\sigma(i_{n-\mu})\}} = \\ &\sum_{j=0}^{\mu} \left\{ B_{n-j} a^{\sigma(1)}_{p-1} \ldots a^{\sigma(n-\mu)}_{p-1} a^{\sigma(n-\mu+1)}_{0} \ldots a^{\sigma(n-j)}_{0} \right\}, \end{aligned} \tag{4.4.11}$$

$$C_{[0]} = C^{p-1}_{[0]\setminus\{\sigma(1)\}} C^{0}_{[p-1]\setminus\{\sigma(1)\}} C^{-1}_{[p-1]} + b_1 \tag{4.4.12}$$

where $a^{\sigma(j)}_{p-1}$ and $a^{\sigma(j)}_{i_j}$ is defined as in Lemma A.1.2 and $B_{n-j} = (b_{n-j+1} - b_{n-j})$ for $j = 1, \ldots, n-1$.

Proof. The proof follows the same line of reasoning used for the proof of Theorem 4.4.2. □

Remark 4.4.1. Notice from Equation 4.4.9 that a nested canalyzing function $f(x_1, \ldots, x_n)$ with canalyzing input sets $S_1, \ldots, S_n$ and canalyzing output values $b_1, \ldots, b_n, b_{n+1}$ is also a nested canalyzing function in the same variable order with canalyzing input sets $S_1, \ldots, S_n^c$ and canalyzing output values $b_1, \ldots, b_{n+1}, b_n$. In fact, Equation 4.4.9 implies that

$$\begin{aligned} C_{[p-1]} &= (b_{n+1} - b_n)(p-1)^n \prod_{i=1}^{n} \mid S^c_{\sigma(i)} \mid \\ &= (b_n - b_{n+1}) \mid S_{\sigma(n)} \mid (p-1)^n \prod_{i=1}^{n-1} \mid S^c_{\sigma(i)} \mid. \end{aligned}$$

Note that $- \mid S^c_{\sigma(n)} \mid = \mid S_{\sigma(n)} \mid \pmod p$.

Remark 4.4.2. For every nonzero $b \in \mathbb{F}$ and for every nested canalyzing function $f(x_1, \ldots, x_n)$ in the variable order $x_{\sigma(1)}, \ldots, x_{\sigma(n)}$ with canalyzing input sets $S_1, \ldots, S_n$ and canalyzing output values $b_1, \ldots, b_{n+1}$, $f(x_1, \ldots, x_n) + b$ is also a nested canalyzing function in the same variable order and with the same canalyzing input sets $S_1, \ldots, S_n$ and canalyzing output values $b_1 + b, \ldots, b_{n+1} + b$. In fact, for $S_1, \ldots, S_n$ and $b_1 + b, \ldots, b_{n+1} + b$, Equations 4.4.8 - 4.4.11 stay the same and Equation 4.4.12 becomes

$$C_{[0]} = C^{p-1}_{[0]\setminus\{\sigma(1)\}} C^{0}_{[p-1]\setminus\{\sigma(1)\}} C^{-1}_{[p-1]} + b_1 + b.$$

4.5 Number of nested canalyzing functions

Here we derive a formula to compute the number of nested canalyzing functions in a given number of variables n and a given finite field $\mathbb{F}$ with p elements.

Let us denote the number of distinct nested canalyzing functions in n variables by $NCF(n)$ and the number of distinct nested canalyzing functions that can be written as a product of r nested canalyzing functions by $RNCF(n, r)$ ('R' for reducible). It is clear from Formula 4.3.1 that any nested canalyzing function can be written as a product of at most n nested canalyzing functions. Hence $RNCF(n, r) = 0$ for all $r > n$. We will denote the number of distinct nested canalyzing functions in n variables that cannot be written as a product of two or more nested canalyzing functions by $INCF(n)$ ('I' for irreducible) and the number of distinct nested canalyzing functions in n variables that can be written as a product of two or more nested canalyzing functions by $RNCF(n)$. Then

$$RNCF(n) = \sum_{r=2}^{n} RNCF(n, r).$$

Example 4.5.1. *Let $n = 3$. From formula 4.3.1,*

$$\begin{aligned} f(x_1, x_2, x_3) = \quad & B_3 Q_{S_1}(x_1) Q_{S_2}(x_2) Q_{S_3}(x_3) \\ & + B_2 Q_{S_1}(x_1) Q_{S_2}(x_2) \\ & + B_1 Q_{S_1}(x_1) + b_1. \end{aligned}$$

If $b_1 = 0$, then $f(x_1, x_2, x_3) = Q_{S_1}(x_1) g(x_2, x_3)$, where

$$g(x_2, x_3) = B_3 Q_{S_2}(x_2) Q_{S_3}(x_3) + B_2 Q_{S_2}(x_2) + b_2.$$

Note that Q_{S_1} and $g(x_2, x_3)$ are nested canalyzing functions. Therefore, $f \in RNCF(3)$ whenever $b_1 = 0$.

The following lemma relates $RNCF(n)$ and $INCF(n)$.

Lemma 4.5.2. *For each natural number n, we have $INCF(n) = (p-1)RNCF(n)$.*

Proof. Taking $b = -b_1$ in Remark 4.4.2, one can see that for each $f \in RNCF(n)$, there are $p-1$ functions in $INCF(n)$ (see Example 4.5.1). Conversely, let $f \in INCF(n)$. From formula 4.3.1,

$$f(x_1, \ldots, x_n) = \sum_{j=0}^{n-1} \left\{ B_{n-j} \prod_{i=1}^{n-j} Q_{S_i}(x_i) \right\} + b_1.$$

Let $b = p - b_1$. Then

$$f(x_1, \ldots, x_n) + b = \sum_{j=0}^{n-1} \left\{ B_{n-j} \prod_{i=1}^{n-j} Q_{S_i}(x_i) \right\}$$

$$= Q_{S_i}(x_i) \left[\sum_{j=0}^{n-2} \left\{ (b_{n-j+1} - b_{n-j}) \prod_{i=2}^{n-j} Q_{S_i}(x_i) \right\} + (b_2 - b_1). \right]$$

Therefore, any element of $INCF(n)$ can be obtained from an element of $RNCF(n)$. $\square$

The following theorem gives us a formula to compute the number of nested canalyzing functions for a given number of variables n.

Theorem 4.5.3. *The number of nested canalyzing functions in n variables, denoted by $NCF(n)$, is given by*

$$NCF(n) = pRNCF(n),$$

where

$$RNCF(1) = (p-1)^2,$$

$$RNCF(2) = 4(p-1)^4,$$

and, for $n \geq 3$,

$$RNCF(n) = \sum_{r=2}^{n-1} \binom{n}{r-1} 2^{r-1}(p-1)^r RNCF(n-r+1) + 2^{n-1}(p-1)^{n+1}(2 + n(p-2)).$$

Proof. From Remark 4.4.2, in order to calculate $NCF(1)$ it is enough to calculate the number of nested canalyzing functions with canalyzing output values $(0, b_2)$, with $b_2 \neq 0$, and then multiply by p, because of the isomorphism between the sets of vectors $\{(0, b_2)\}$ and $\{(b_1, b_2)\}$ with $b_2 \neq b_1$. This isomorphism is given by

$$(0, b_2) \xrightarrow{+(b_1,b_1)} (b_1, b_1 + b_2)$$

for $b_1 = 1, \ldots, p-1$. For output values of the form $(0, b_2)$, Formula 4.3.1 gives us

$$f(x_1) = b_2 Q_{S_1}(x_1).$$

There are $p-1$ choices for b_2 and $p-1$ choices for $Q_{S_1}(x_1)$. Note that we do not consider $2(p-1)$ choices for $Q_{S_1}(x_1)$ because a nested canalyzing function with canalyzing input set S_1 and canalyzing output values $(0, b_2)$ is also a nested canalyzing function with canalyzing input set S_1^c and canalyzing output values $(b_2, 0)$ (see remark 4.4.1). Therefore,

$$RNCF(1) = (p-1)^2,$$

Similarly, in order to calculate $NCF(2)$, it is enough to calculate the number of nested canalyzing functions with canalyzing output values $(0, b_2, b_3)$, with $b_3 \neq b_2$, and then multiply by p, because of the isomorphism between the sets of vectors $\{(0, b_2, b_3)\}$ and $\{(b_1, b_2, b_3)\}$ given by

$$(0, b_2, b_3) \xrightarrow{+(b_1,b_1,b_1)} (b_1, b_1 + b_2, b_1 + b_3)$$

for $b_1 = 1, \ldots, p-1$. For output values of the form $(0, 0, b_3)$ Formula 4.3.1 gives us

$$f(x_1) = b_3 Q_{S_1}(x_1) Q_{S_2}(x_2).$$

There are $p-1$ choices for b_3 and $2(p-1)$ choices for each of the $Q_{S_i}(x_i)$ for $i = 1, 2$. Therefore, for $(0, 0, b_3)$ there are $4(p-1)^3$ functions.

For $(0, b_2, b_3)$, where $b_2 \neq 0$, $b_3 \neq 0$, and $b_3 \neq b_2$, Formula 4.3.1 give us

$$f(x_1) = Q_{S_1}(x_1)\Big((b_3 - b_2) Q_{S_2}(x_2) + b_2\Big)$$

There are $2(p-1)$ choices for $Q_{S_1}(x_1)$, $p-1$ choices for b_2, $p-2$ choices for b_3, and $p-1$ choices for $Q_{S_2}(x_2)$. Note that we do not consider $2(p-1)$ choices for $Q_{S_2}(x_2)$ because a nested canalyzing function with canalyzing input sets S_1, S_2 and canalyzing output values $(0, 0, b_3)$ is also a nested canalyzing function with canalyzing input sets S_1, S_2^c and canalyzing output values $(0, b_3, 0)$ (see Remark 4.4.1). Therefore, there are $2(p-1)^3(p-2)$ functions. If we count the number of functions after permuting the variables, we get

$$RNCF(2) = 4(p-1)^3 + \binom{2}{1} 2(p-1)^3(p-2),$$

Simplifying the formula above we get

$$RNCF(2) = 4(p-1)^4.$$

For $n \geq 3$ let us compute the number of distinct nested canalyzing functions that can be written as a product of n nested canalyzing functions, $RNCF(n,n)$. From Formula 4.3.1 it is clear that a nested canalyzing function can be written as a product of n nested canalyzing functions if and only if the output values must have the form $(0,\dots,0,b_n,b_{n+1})$, where $b_{n+1} \neq b_n$. First consider the case where $b_n = 0$, i.e. the output values have the form $(0,\dots,0,b_{n+1})$, with $b_{n+1} \neq 0$, for which we have:

$$f(x_1,\cdots,x_n) = b_{n+1}Q_{S_1}(x_1)\dots Q_{S_n}(x_n).$$

There are $p-1$ choices for b_{n+1} and $2(p-1)$ choices for each of the $Q_{S_i}(x_i)$ for $i = 1,\dots,n$. Therefore, there are $2^n(p-1)^{n+1}$ functions. Note that if we permute the variables, we will still get the same functions.

For $(0,\dots,0,b_n,b_{n+1})$, where $b_{n+1} \neq 0$, $b_n \neq 0$, and $b_{n+1} \neq b_n$, Formula 4.3.1 gives us

$$\begin{aligned} &f(x_1,\cdots,x_n) = \\ &Q_{S_1}(x_1)\dots Q_{S_{n-1}}(x_{n-1})\Big\{(b_{n+1}-b_n)Q_{S_n}(x_n) \\ &\qquad\qquad +b_n\Big\}. \end{aligned}$$

There are $2(p-1)$ choices for each $Q_{S_i}(x_i)$ for $i = 1,\dots,n-1$, $p-1$ choices for b_n, $p-2$ choices for b_{n+1}, and $p-1$ choices for $Q_{S_n}(x_n)$. Note that we do not consider $2(p-1)$ choices for $Q_{S_n}(x_n)$ because a nested canalyzing function with canalyzing input sets $S_1,\dots,S_n$ and canalyzing output values $(0,\dots,0,b_n,b_{n+1})$ is also a nested canalyzing function with canalyzing input set $S_1,\dots,S_{n-1},S_n^c$ and canalyzing output values $(0,\dots,0,b_{n+1},b_n)$ (see Remark 4.4.1). Therefore, there are $2^{n-1}(p-1)^{n+1}(p-2)$ such functions. If we count the number of functions after permuting the variables, we get

$$\begin{aligned} &RNCF(n,n) = \\ &2^n(p-1)^{n+1} + \binom{n}{1}2^{n-1}(p-1)^{n+1}(p-2) \\ &= 2^{n-1}(p-1)^{n+1}(2+n(p-2)). \end{aligned}$$

Let us now compute the number of distinct nested canalyzing functions that can be written as a product of $n-r$ nested canalyzing functions, $RNCF(n,n-r)$ for $r = 1,\dots,n-2$. From Formula 4.3.1, it is clear that a nested canalyzing function can be written as a product of $n-1$ nested canalyzing functions if and only if the output values have the form $(0,\dots,0,b_{n-r},\dots,b_n,b_{n+1})$, where $b_{n-r} \neq 0$ and $b_{n+1} \neq b_n$. In this case we have:

$$\begin{aligned} &f(x_1,\dots,x_n) = \\ &\prod_{i=1}^{n-r-1} Q_{S_i}(x_i) \\ &\sum_{j=0}^{r}\Big\{(b_{n-j+1}-b_{n-j})\prod_{i=n-r}^{n-j} Q_{S_i}(x_i) + b_{n-r}\Big\}. \end{aligned}$$

There are $2(p-1)$ choices for each $Q_{S_i}(x_i)$ for $i = 1, \ldots, n-r-1$ and $INCF(r+1)$ functions for the summation part. Note that here we do have repetition coming from Remark 4.4.1 because $b_{n-r} \neq 0$. Therefore, there are $2^{n-r-1}(p-1)^{n-r-1}INCF(r+1)$ such functions. If we count the number of functions after permuting the variables, we get

$$\begin{aligned} &RNCF(n, n-r) = \\ &\binom{n}{r+1} 2^{n-r-1}(p-1)^{n-r-1} INCF(r+1). \end{aligned} \tag{4.5.1}$$

Now, since

$$\begin{aligned} &RNCF(n) = \\ &\sum_{r=2}^{n} RNCF(n, r) = \sum_{r=0}^{n-2} RNCF(n, n-r), \end{aligned}$$

we have

$$RNCF(n) = \sum_{r=1}^{n-2} RNCF(n, n-r) + RNCF(n, n).$$

Replacing Equation 4.5.1 in the previous formula, we have

$$\begin{aligned} &RNCF(n) = \\ &\sum_{r=1}^{n-2} \binom{n}{r+1} 2^{n-r-1}(p-1)^{n-r-1} INCF(r+1) \\ &\qquad +RNCF(n, n). \end{aligned}$$

If we make the change of variables $\mu = n - r$, then $\mu - 1 = n - r - 1$ and $r + 1 = n - \mu + 1$. Therefore,

$$\begin{aligned} &RNCF(n) = \\ &\sum_{\mu=2}^{n-1} \binom{n}{n-\mu+1} 2^{\mu-1}(p-1)^{\mu-1} INCF(n-\mu+1) \\ &\qquad +RNCF(n, n). \end{aligned}$$

But since,

$$\binom{n}{n-(\mu-1)} = \binom{n}{\mu-1},$$

we have

$$RNCF(n) =$$

$$\sum_{\mu=2}^{n-1} \binom{n}{\mu-1} 2^{\mu-1}(p-1)^{\mu-1} INCF(n-\mu+1)$$

$$+RNCF(n,n).$$

From Lemma 4.5.2,

$$RNCF(n) =$$

$$\sum_{\mu=2}^{n-1} \binom{n}{\mu-1} 2^{\mu-1}(p-1)^{\mu} RNCF(n-\mu+1)$$

$$+RNCF(n,n),$$

where

$$RNCF(n,n) = 2^{n-1}(p-1)^{n+1}(2+n(p-2)). \tag{4.5.2}$$

Finally, again using Lemma 4.5.2,

$$NCF(n) = RNCF(n) + INCF(n) = pRCF(n).$$

This completes the proof. □

Example 4.5.4 (Boolean case)**.** *Jarrah et. al. [28] show that the class of Boolean nested canalyzing functions is identical to the class of unate cascade functions. Sasao and Kinoshita [56] found a recursive formula for the number of unate cascade functions. Therefore, the same formula can be used to compute the number of Boolean nested canalyzing functions. Below is the formula originally given by Sasao and Kinoshita [56] which is a particular case of our formula in Theorem 4.5.3, namely when $p = 2$:*

$$NCF(n) = 2E(n),$$

where

$$E(1) = 2, \;\; E(2) = 4,$$

and

$$E(n) = \sum_{r=2}^{n-1} \binom{n}{r-1} 2^{r-1} E(n-r+1) + 2^n.$$

Example 4.5.5. *The number of nested canalyzing function for $n = 3$ and $p = 3$ is given by*

$$NCF(3) = 3RNCF(3),$$

Table 4.1: Number of nested canalyzing functions for $p = 3$ and $n = 1, \ldots, 8$.

n	NCF(n)
1	12
2	192
3	5568
4	219648
5	10834944
6	641335296
7	44288360448
8	3495313145856

Table 4.2: Number of nested canalyzing functions for $p = 5$ and $n = 1, \ldots, 8$.

n	NCF(n)
1	80
2	5120
3	547840
4	78561280
5	14082703360
6	3029304606720
7	760232846295040
8	218043057365319680

where

$$RNCF(3) = \binom{3}{1} 2^1 2^1 INCF(2,1) + RNCF(3,3),$$

$$RNCF(3,3) = 2^3 \times 2^4 + \binom{3}{1} 2^2 2^4 = 128 + 192 = 320,$$

and

$$INCF(2,1) = 2RNCF(2) = 2 \times (4 \times 2^4) = 2 \times 64 = 128.$$

Therefore

$$RNCF(3) = 3 \times 4 \times 128 + 320 = 1536 + 320 = 1856.$$

Finally,

$$NCF(3) = 3 \times 1856 = 5568.$$

Tables 4.1 - 4.2 show the number of nested canalyzing functions for $p = 3, 5$ and $n = 1, \ldots, 8$.

4.6 Asymptotic properties of $NCF(n)$

In this section we examine the asymptotic properties of the formula given in Theorem 4.5.3, i.e. we want to know the behavior of $NCF(n)$ as n becomes large. First we derive the following inequalities:

Lemma 4.6.1. *For fixed n and $r = 2, \ldots, n-1$, we have*

$$2^{r-1}(p-1)^r RNCF(n-r+1) \leq RNC(n)$$

Proof. It is a direct consequence of the formula given at Theorem 4.5.3. □

Lemma 4.6.2. *For all natural numbers n, we have*

$$RNCF(n,n) \leq 2^{2n}(p-1)^{n+2}.$$

Proof. From Equation 4.5.2,

$$\begin{aligned} RNCF(n,n) &= 2^{n-1}(p-1)^{n+1}(2+n(p-2)) \\ &\leqslant 2^{n-1}(p-1)^{n+1}(2+2^n(p-1)) \\ &\leqslant 2^{n-1}(p-1)^{n+1}(2^{n+1}(p-1)) \\ &\leqslant 2^{2n}(p-1)^{n+2}. \end{aligned}$$

□

Lemma 4.6.3. *For all natural numbers $n \geq 3$, we have*

$$RNCF(n) \leq 2^{n(n-1)}(p-1)^{2n}.$$

Proof. We prove this by induction over n. First note that for $n = 3$,

$$RNCF(3) = \binom{3}{1} 2^1 (p-1)^2 RNCF(2)$$

$$+2^2(p-1)^4(2+3p-6)$$

$$= 24(p-1)^6 + 4(p-1)^4(3p-4)$$

$$\leq 2^5(p-1)^6 + 4(p-1)^4(3(p-1)-1)$$

$$\leq 2^5(p-1)^6 + 2^5(p-1)^6$$

$$= 2^6(p-1)^6 = 2^{3(3-1)}(p-1)^{2(3)}.$$

Now, assume that

$$RNCF(n) \leq 2^{n(n-1)}(p-1)^{2n}.$$

Then

$$\begin{aligned}
&RNCF(n+1) = \\
&\sum_{r=2}^{n} \binom{n+1}{r-1} 2^{r-1}(p-1)^r RNCF(n-r+2) \\
&\qquad\qquad +RNCF(n+1, n+1) \\
&= \binom{n+1}{1} 2(p-1)^2 RNCF(n) \\
&+\sum_{r=3}^{n} \binom{n+1}{r-1} 2^{r-1}(p-1)^r RNCF(n-r+2) \\
&\qquad\qquad +RNCF(n+1, n+1).
\end{aligned}$$

From Lemma 4.6.1 we have

$$\begin{aligned}
&\sum_{r=3}^{n} \binom{n+1}{r-1} 2^{r-1}(p-1)^r RNCF(n-r+2) \\
&= \sum_{r=2}^{n-1} \binom{n+1}{r} 2^{r}(p-1)^{r+1} RNCF(n-r+1) \\
&\leqslant \sum_{r=2}^{n-1} \binom{n+1}{r} 2(p-1) RNCF(n).
\end{aligned}$$

Therefore,

$$RNCF(n+1) =$$

$$\leqslant \binom{n+1}{1} 2(p-1)^2 RNCF(n)$$

$$+\sum_{r=2}^{n-1} \binom{n+1}{r} 2(p-1) RNCF(n)$$

$$+RNCF(n+1, n+1)$$

$$\leqslant 2(p-1)^2 RNCF(n) \sum_{r=1}^{n-1} \binom{n+1}{r}$$

$$+RNCF(n+1, n+1).$$

Using the inductive hypothesis and Lemma 4.6.2 we have

$$RNCF(n+1) =$$

$$\leqslant 2(p-1)^2 2^{n(n-1)} (p-1)^{2n} \sum_{r=1}^{n-1} \binom{n+1}{r}$$

$$+2^{2(n+1)}(p-1)^{n+3}$$

$$= 2^{n(n-1)+1}(p-1)^{2(n+1)} \sum_{r=1}^{n-1} \binom{n+1}{r}$$

$$+2^{2(n+1)}(p-1)^{n+3}.$$

Now, since

$$\sum_{r=1}^{n-1} \binom{n+1}{r} = 2^{n+1} - n - 3$$

we have

$$\begin{aligned}
&RNCF(n+1) \leqslant \\
&2^{n(n-1)+1}(p-1)^{2(n+1)}(2^{n+1}) \\
&\qquad +2^{2(n+1)}(p-1)^{n+3} \\
&\leqslant 2^{n^2+2}(p-1)^{2(n+1)} \\
&\qquad +2^{n^2+n-1}(p-1)^{2(n+1)} \\
&\leqslant 2^{(n+1)n}(p-1)^{2(n+1)}.
\end{aligned}$$

Note that $n-1 \geq 2$. This completes our proof.

□

Let us denote the number of all possible functions on n variables by $\psi(n)$. The following theorem show that the set of all nested canalyzing functions is an increasingly smaller subset of the set of all functions.

Theorem 4.6.4. *The ratio $NCF(n)/\psi(n)$ converges to 0 as n becomes large.*

Proof.

$$\begin{aligned}
\frac{NCF(n)}{\psi(n)} &= \frac{NCF(n)}{p^{p^n}} = \frac{pRNCF(n)}{p^{p^n}} \\
&\leq \frac{p(2^{n(n-1)}(p-1)^{2n})}{p^{p^n}} \\
&\leq \frac{2^{n(n-1)}p^{2n+1}}{p^{p^n}} \\
&\leq \frac{p^{n(n-1)}p^{2n+1}}{p^{p^n}} \\
&= \frac{p^{n^2+n+1}}{p^{p^n}} \to 0 \text{ as } n \to \infty,
\end{aligned}$$

because the exponential function p^n grows much faster than the quadratic function n^2+n+1 as n becomes large. □

4.7 Discussion

The concept of a nested canalyzing rule has been shown to be a useful approach to elucidating design principles for molecular regulatory networks, and such rules appear very frequently in published network models. But how much of a restriction do such rules impose on the regulatory logic of the network, that is, how "special" are such rules? It was shown in [16] that networks with nested canalyzing rules have very special dynamic properties. In this paper we have shown that nested canalyzing rules do indeed make up a very small subset of all possible rules. In particular, we provide an explicit formula for the number of such rules for a given number of variables.

This was done by translating the problem into the mathematical context of polynomial functions over finite fields and the language of algebraic geometry. The formula we provide uses a parametric description of the class of all nested canalyzing polynomials as an algebraic variety. In particular, this provides a very easy way to generate such polynomials through particular parameter choices, which is very useful, for instance, for large-scale simulation studies.

Another interesting aspect of the formula derived in this chapter is as a future discovery tool. The formula for Boolean nested canalyzing functions in [28] was obtained essentially through serendipity. Once the parameterization of this class was obtained it was possible to explicitly compute the number of solutions of the parametric equations for small numbers of variables. The resulting integer sequence, giving the number of Boolean nested canalyzing rules for small numbers of variables was matched to the number of Boolean unate cascade functions, for which a formula is known. It was shown in [28] that the two classes of functions are in fact identical. This is of independent interest, since Boolean unate cascade functions have been shown to lead to binary decision diagrams with smallest average path length, suggesting that they are very efficient in processing information. The formula for multi-state nested canalyzing rules provides a similar opportunity. While there is no obvious match to other function classes for small fields, it is worth, in our opinion, to pursue this discovery approach further for larger fields.

Chapter 5

Stochastic Discrete Dynamical Systems

Modeling stochasticity in gene regulatory networks is an important and complex problem in molecular systems biology. To elucidate intrinsic noise, several modeling strategies such as the Gillespie algorithm have been used successfully. This chapter discusses an approach as an alternative to these classical settings. Within the discrete paradigm, where genes, proteins, and other molecular components of gene regulatory networks are modeled as discrete variables and are assigned as logical rules describing their regulation through interactions with other components. Stochasticity is modeled at the biological function level under the assumption that even if the expression levels of the input nodes of an update rule guarantee activation or degradation there is a probability that the process will not occur due to stochastic effects. This approach allows a finer analysis of discrete models and provides a natural setup for cell population simulations to study cell-to-cell variability. Applications are presented using two of the most studied regulatory networks, the outcome of lambda phage infection of bacteria and the p53-mdm2 complex.

This chapter is based on published paper [59]. My contribution to this paper were the following: 1. Played a major role on the design of the study. 2. Model construction and analysis of results. 3. Writing of the manuscript.

5.1 Introduction

Variability at the molecular level, defined as the phenotypic differences within a genetically identical population of cells exposed to the same environmental conditions, has been observed experimentally [60, 61, 62, 63]. Understanding mechanisms that drive variability in molecular networks is an important goal of molecular systems biology, for which mathematical modeling can be very helpful. Different modeling strategies have been used for this purpose and, depending on the level of abstraction of the mathematical models, there are several ways to introduce stochasticity. Dynamic mathematical models can be broadly divided into two classes: continuous, such as systems of differential equations (and their stochastic variants) and discrete, such as Boolean networks and their generalizations (and their stochastic variants). This article will focus on stochasticity and discrete models.

Discrete models do not require detailed information about kinetic rate constants and they tend to be more intuitive. In turn, they only provide qualitative information about the system. The most general setting is as follows. Network nodes represent genes, proteins, and other molecular components of gene regulation, while network edges describe biological interactions among network nodes that are given as logical rules representing their interactions. Time in this framework is implicit and progresses in discrete steps. More formally, let $x_1, \ldots, x_n$ be variables, which can take values in finite sets $X_1, \ldots, X_n$, respectively. Let $X = X_1 \times \cdots \times X_n$ be the Cartesian product. A discrete dynamical system (DDS) in the variables $x_1, \ldots, x_n$ is a function

$$f = (f_1, \ldots, f_n) : X \to X$$

where each coordinate function $f_i : X \to X_i$ is a function in a subset of $\{x_1, \ldots, x_n\}$. Dynamics is generated by iteration of f, and different update schemes can be used for this purpose. As an example, if $X_i = \{0, 1\}$ for all i, then each f_i is a Boolean rule and f is a Boolean network where all the variables are updated simultaneously. We will assume that each X_i comes with a natural total ordering of its elements (corresponding to the concentration levels of the associated molecular species). Examples of this type of dynamical system representation are Boolean networks, logical models and Petri nets [65, 66, 64].

To account for stochasticity in this setting several methods have been considered. Probabilistic Boolean networks (PBNs) [67, 68] introduce stochasticity in the update functions, allowing a different update function to be used at each iteration, chosen from a probability space of such functions for each network node. For other approaches, see [69, 70, 71]. These models will be discussed in more detail in the next section. In this article we present a model type related to PBNs, with additional features. We show that this model type is natural and a useful way to simulate gene regulation as a stochastic process, and is very useful to simulate experiments with cell populations.

5.1.1 Modeling stochasticity in gene regulatory networks

Gene regulation processes are inherently stochastic. Accurately modeling this stochasticity is a complex and important goal in molecular system biology. Depending on the level of knowledge of the biological system and the availability of data for it one could follow different approaches. For instance, viewing a gene regulatory network as a biochemical reaction network, the Gillespie algorithm can be applied to simulate each biochemical reaction separately generating a random walk corresponding to a solution of the chemical master equation of the system [72, 73]. At an even more detailed level one could introduce time delays into the Gillespie simulations to account for realistic time delays in activation or degradation such as in circadian rhythms [74, 75, 76]. At a higher level of abstraction, stochastic differential equations [77] contain a deterministic approximation of the system and an additional random white noise term. However, all these schemes require that all the kinetic rate constants to be known which could represent a strong constraint due to the difficulty of measuring kinetic parameters, limiting these approaches to small systems.

As mentioned in the introduction, discrete models are an alternative to continuous models, which do not depend on rate constants. In this setting, several approaches to introduce stochasticity have been proposed. Specially for Boolean networks, stochasticity has been introduced by flipping node states from 0 to 1 or vice versa with some flip probability [71, 78, 79, 80]. However, it has been argued that this way of introducing stochasticity into the system usually leads to over-representation of noise [70]. The main criticism of this approach is that it does not take into consideration the correlation between the expression values of input nodes and the probability of flipping the expression of a node due to noise. In fact, this approach models the stochasticity at a node regardless of the susceptibility to noise of the underlying biological function [70].

Probabilistic Boolean networks [67, 68, 81] is another stochastic method proposed within the discrete strategy. PBNs model the choice among alternate biological functions during the iteration process, rather than modeling the stochasticity of the function failure itself. We have adopted a special case of this setting, in which every node has associated to it two functions: the function that governs its evolution over time and the identity function. If the first is chosen, then the node is updated based on its logical rule. When the identity function is chosen, then the state of the node is not updated. The key difference to a PBN is the assignment of probabilities that govern which update is chosen. In our setting, each function gets assigned two probabilities. Precisely, let x_i be a variable. We assign to it a probability $p_i^{\uparrow}$, which determines the likelihood that x_i will be updated based on its logical rule, if this update leads to an increase/activation of the variable. Likewise, a probability $p_i^{\downarrow}$ determines this probability in case the variable is decreased/inhibited. The necessity for considering two different probabilities is that activation and degradation represent different biochemical processes and even if these two are encoded by the same function, their propensities in general are different. This is very similar to what is considered in differential equations modeling, where, for instance, the kinetic rate parameters for activation and for degradation/decay are, in principle, different.

Note that all these approaches only take account of intrinsic noise which is generated from small fluctuations in concentration levels, small number of reactant molecules, and fast and slow reactions. Another source of stochasticity is related to extrinsic noise such as a noisy cellular environment and temperature. For more about intrinsic vs extrinsic noise see [82, 62].

5.2 Method

Our aim is to model stochasticity at the biological function level under the main assumption that even if the expression levels of the input nodes of an update function guarantee activation or degradation there is a probability that the process will not occur due to stochasticity, for instance, if some of the chemical reactions encoded by the update function may fail to occur. This is similar to models based on the chemical master equation. This model type introduces activation and degradation propensities. More formally, let $x_1, \ldots, x_n$ be variables which can take values in finite sets $X_1, \ldots, X_n$, respectively. Let $X = X_1 \times \cdots \times X_n$ be the Cartesian product. Thus, the formal definition of a *stochastic discrete dynamical system (SDDS)* in the variables $x_1, \ldots, x_n$ is a collection of n triplets

$$F = \{f_i, p_i^{\uparrow}, p_i^{\downarrow}\}_{i=1}^{n}$$

where

- $f_i : X \to X_i$ is the update function for x_i, for all $i = 1, \ldots, n$.
- $p_i^{\uparrow}$ is the activation propensity.
- $p_i^{\downarrow}$ is the degradation propensity.
- $p_i^{\uparrow}, p_i^{\downarrow} \in [0, 1]$.

We now proceed to study the dynamics of such systems and two specific models as illustration.

5.2.1 Dynamics of SDDS

Let $F = \{f_i, p_i^{\uparrow}, p_i^{\downarrow}\}_{i=1}^{n}$ be a SDDS and consider $x \in X$. For all i we define $\pi_{i,x}(x_i \to f_i(x))$ and $\pi_{i,x}(x_i \to x_i)$ by

$$\pi_{i,x}(x_i \to f_i(x)) = \begin{cases} p_i^{\uparrow}, & \text{if } x_i < f_i(x), \\ p_i^{\downarrow}, & \text{if } x_i > f_i(x), \\ 1, & \text{if } x_i = f_i(x). \end{cases}$$

$$\pi_{i,x}(x_i \to x_i) = \begin{cases} 1 - p_i^{\uparrow}, & \text{if } x_i < f_i(x), \\ 1 - p_i^{\downarrow}, & \text{if } x_i > f_i(x), \\ 1, & \text{if } x_i = f_i(x). \end{cases}$$

That is, if the possible future value of the i-th coordinate is larger (smaller, resp.) than the current value, then the activation (degradation) propensity determines the probability that the i-th coordinate will increase (decrease) its current value. If the i-th coordinate and its possible future value are the same, then the i-th coordinate of the system will maintain its current value with probability 1. Notice that $\pi_{i,x}(x_i \to y_i) = 0$ for all $y_i \notin \{x_i, f_i(x)\}$.

The dynamics of F is given by the weighted graph X which has an edge from $x \in X$ to $y \in X$ if and only if $y_i \in \{x_i, f_i(x)\}$ for all i. The weight of an edge $x \to y$ is equal to the product

$$w_{x\to y} = \prod_{i=1}^{n} \pi_{i,x}(x_i \to y_i)$$

By convention we omit edges with weight zero. See Appendix B for pseudocodes of algorithms to compute dynamics of SDDS. Software to test examples is available at `http://dvd.vbi.vt.edu/adam.html` [83] as a web tool (choose SDDS in the model type).

Given $F = \{f_i, p_i^{\uparrow}, p_i^{\downarrow}\}_{i=1}^{n}$ a SDDS, it is straightforward to verify that F has the same steady states (fixed points) as the deterministic system $G = \{f_i\}_{i=1}^{n}$ (see Appendix B). It is also important to note that the dynamics of F includes the different trajectories that can be generated from G using other common update mechanisms such as the synchronous and asynchronous schemes (see Appendix B).

Example

Let $n = 2$, $X = \{0,1\} \times \{0,1\}$, $F = (f_1, f_2) : X \to X$, where

x_1	x_2	f_1	f_2
0	0	0	0
0	1	1	0
1	0	0	1
1	1	1	0

and

	x_1	x_2
Activation	.1	.5
Degradation	.2	.9

$$prob(01 \to 10) = (.1)(.9) = .09,\ prob(01 \to 00) = (1 - .1)(.9) = .81$$
$$prob(01 \to 01) = (1 - .1)(1 - .9) = .09,\ prob(01 \to 11) = (.1)(1 - .9) = .01$$
$$prob(10 \to 10) = (1 - .2)(1 - .5) = .4,\ prob(10 \to 01) = (.2)(.5) = .1$$
$$prob(10 \to 00) = (.2)(1 - .5) = .1,\ prob(10 \to 11) = (1 - .2)(.5) = .4$$
$$prob(11 \to 11) = (1)(1 - .9) = .1,\ prob(11 \to 10) = (1)(.9) = .9$$
$$prob(00 \to 00) = (1)(1) = 1.$$

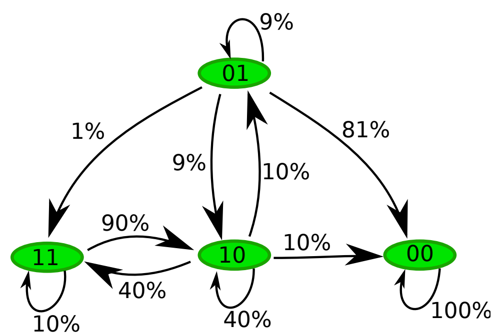


Figure 5.1: **State Space**. There is a 9% chance that the system will transition from 01 to 10. There is an 81% chance that the system will transition from 01 to 00. The latter was expected because there is a high degradation propensity for f_2. Note that 00 is a fixed point, i.e., there is 100% chance of staying at this state.

5.3 Applications

We illustrate the advantages of this model type by applying it to two widely studied biological systems, the regulation of the p53-mdm2 network and the control of the outcome of phage lambda infection of bacteria. These regulatory networks were selected because stochasticity plays a key role in their dynamics.

5.3.1 Regulation in the p53-Mdm2 network

The p53-Mdm2 network is one of the most widely studied gene regulatory networks. Abou-Jaude et al. [84] proposed a logical four-variable model to describe the dynamics of the tumor suppressor protein p53 and its negative regulator Mdm2 when DNA damage occurs. The wiring diagram of this model is represented in Figure B.1, where P denotes cytoplasmic p53, nucleic p53, and the gene *p53*. Mc and Mn stand for cytoplasmic Mdm2 and nuclear Mdm2, respectively. DNA damage caused by ionic irradiation decreases the level of nucleic Mdm2 which enables p53 to accumulate and to remain active, playing a key role in reducing the effect of the damage. There is a negative feedback loop involving three components: p53 increases the level of cytoplasmic Mdm2 which, in turn, increases the level of nuclear Mdm2. Nucleic Mdm2 reduces p53 activity. This model also contains a positive feedback loop involving two components where p53 inhibits its negative regulator nucleic Mdm2. Note the dual role of P, as it positively regulates nucleic Mdm2 through cytoplasmic Mdm2. On the other hand, P negatively regulates nucleic Mdm2 by inhibiting Mdm2 nuclear translocation [84]. For more about the p53-Mdm2 system (see [84, 85, 63]).

The dynamic behavior of the system is represented in a network of transitions called its state space (see Figure 5.3). This specifies the different paths to follow and the probabilities of following a specific trajectory from a given state. Dynamics here is not deterministic, i.e., most of the state vectors have different trajectories they can follow. The propensity parameters in Table 5.1 determine the likelihood of following certain paths. The state 0010 is a steady state, which is differentiated from the others by its oval shape.

The state space for this model is specified by $[0, 2] \times [0, 1] \times [0, 1] \times [0, 1]$, that is, except for the first variable P which has three levels $\{0, 1, 2\}$, all other variables are Boolean. The update functions for this model are provided in the supporting material and also in the model repository of our web tool at http://dvd.vbi.vt.edu/adam.html.

Individual cell simulations render plots similar to the ones shown in Figure 5.4. Each subfigure shows oscillations as long as the damage is present with a variability in the timing of damage repair. On the other hand, cell population simulations, Figure 5.5, exhibit damped oscillations of the expression level of p53 as the degradation propensities of the damage increases. This is correlated with the fact that, if the intensity of the damage is increased, more cells exhibit oscillations in the level of p53 which was experimentally observed in [63]. The initial state for all simulations was 0011 which represents the state when DNA damage is introduced (0010 is the steady state without perturbation).

To highlight the features of our approach we compare our model with the one presented in [84] in which variability has been analyzed. The main difference between these two models is in the way the simulations are performed. In [84], the transition from one state to the next is determined by parameters called "on" and "off" time delays. For instance, to transition from 2001 to 2101 it is required that $t_{Mc} < t_{\overline{\text{dam}}}$ which means that the "on" delay for Mc (time for activating) is less than the "off" delay (time for degrading) of the damage. Otherwise, if $t_{Mc} > t_{\overline{\text{dam}}}$ the system will transition from 2001 to 2000. In this article, transitions from one state to others are given as probabilities which are determined from the propensity probabilities. Therefore, the complexity of the model presented here is at the level of the wiring diagram (i.e. the number of variables) while the complexity of the model in [84] is at the level of the state space (i.e. number of possible states) which is exponential in the number of variables. Another key difference is the way DNA damage repair is modeled. In [84], a delay parameter $t_{\overline{\text{dam}}}$ is associated with the disappearance of the damage, and this is decreased by a certain amount τ at each iteration so that $t^{(n)}_{\overline{\text{dam}}} = t^{(0)}_{\overline{\text{dam}}} - n\tau \geq 0$ where n is the number of iterations. In order to simulate DNA damage with this approach it is required to estimate τ, n, and $t^{(0)}_{\overline{\text{dam}}}$. Within our model framework a single parameter, the degradation propensity, is used to model the damage repair which is a more natural setup.

5.3.2 Phage lambda infection of bacteria

Control of the outcome of phage lambda infection is one of the best understood regulatory systems [86, 87, 62]. Figure 5.6 depicts its core regulatory network that was first modeled by Thieffry and Thomas [87] using a logical approach. This model encompasses the roles of the regulatory genes *CI*, *CRO*, *CII*, and *N*. From experimental reports [87, 88, 89, 62] it

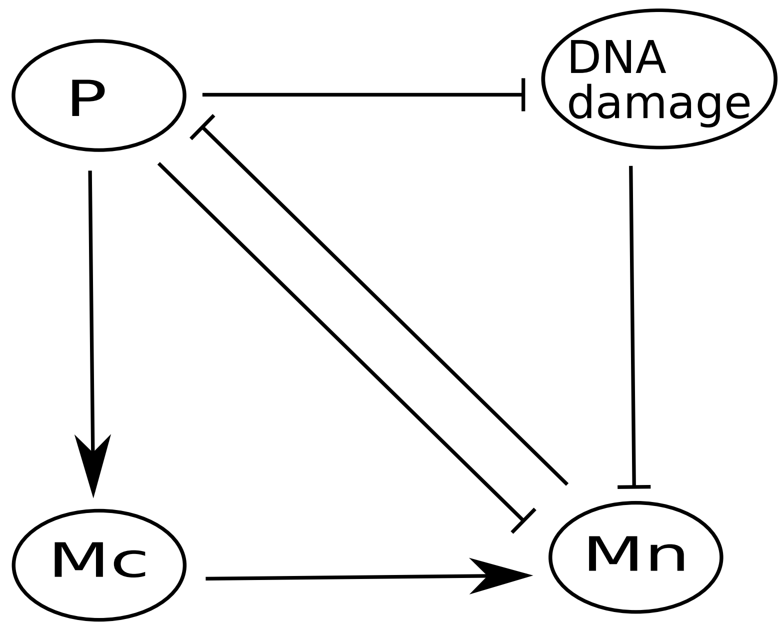


Figure 5.2: **Four-variable model for the p53-Mdm2 regulatory network. P, Mc, and Mn stand for protein p53, cytoplasmic Mdm2, and nuclear Mdm2, respectively**.

Table 5.1: **Propensity probabilities for the p53-Mdm2 regulatory network**

	P	**Mc**	**Mn**	**Dam**
Activation	.9	.9	.9	1
Degradation	.9	.9	.9	.05

Note that there is a low degradation propensity for DNA damage.

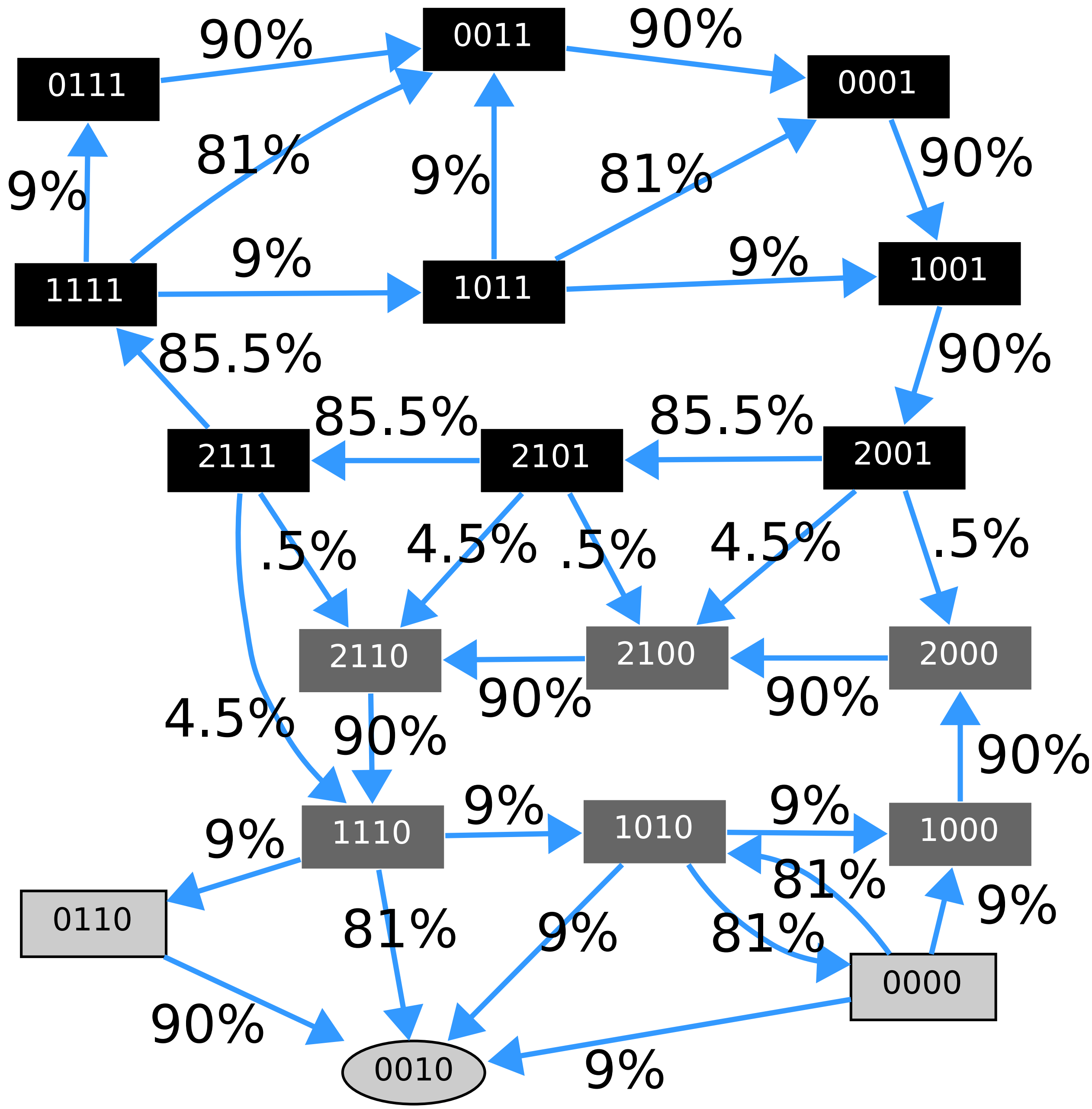


Figure 5.3: **State space diagram for parameters described in Table 5.1**. The numbers next to the edges encode the transition probabilities. The order of the variables in each vector state is P, Mc, Mn, DNA damage. Self-loops are not depicted. States with darker background comprise the cycle with DNA damage. A second cycle with a lighter shaded background corresponds to the cycle with no DNA damage. The oval shaped state is a steady state.

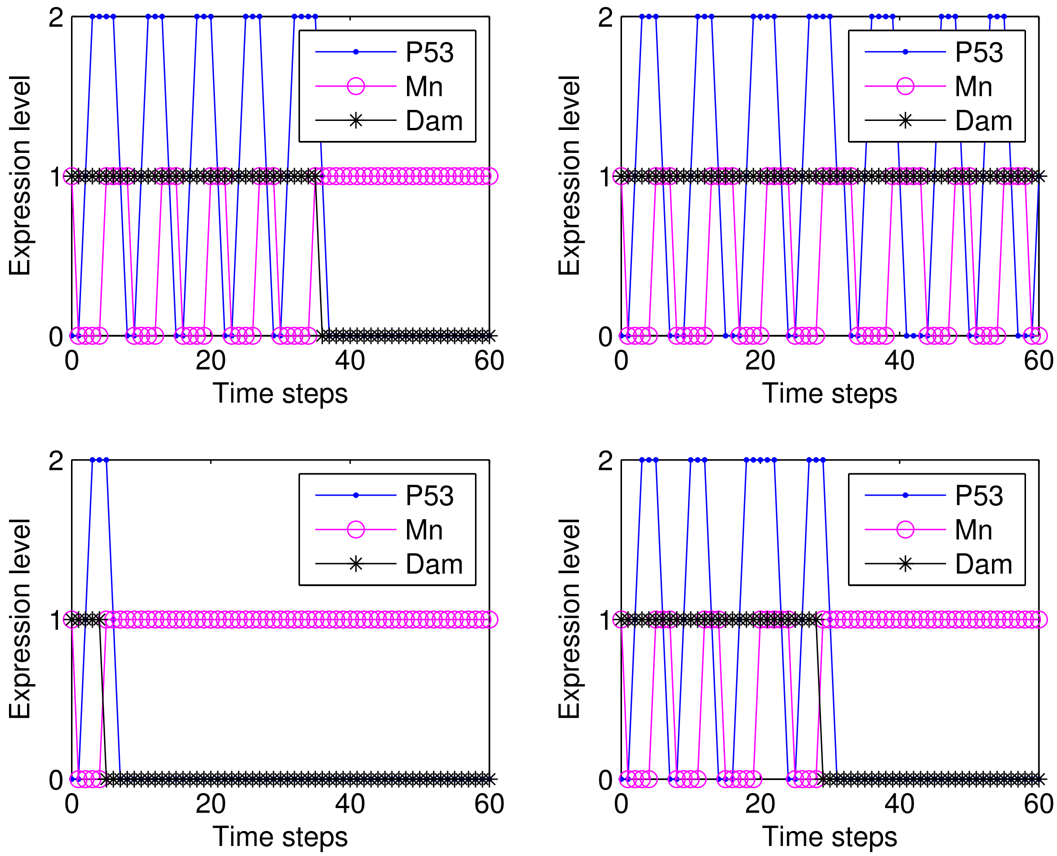


Figure 5.4: **Individual cell simulations for parameters described in Table 5.1**. Each subfigure shows oscillations as long as the damage is present. This figure shows variability in the timing of damage repair and in the period of the oscilations. Each frame was generated from a single simulation with sixty time steps. The x-axis represents discrete time steps and the y-axis the expression level. The initial state for all simulations is 0011.

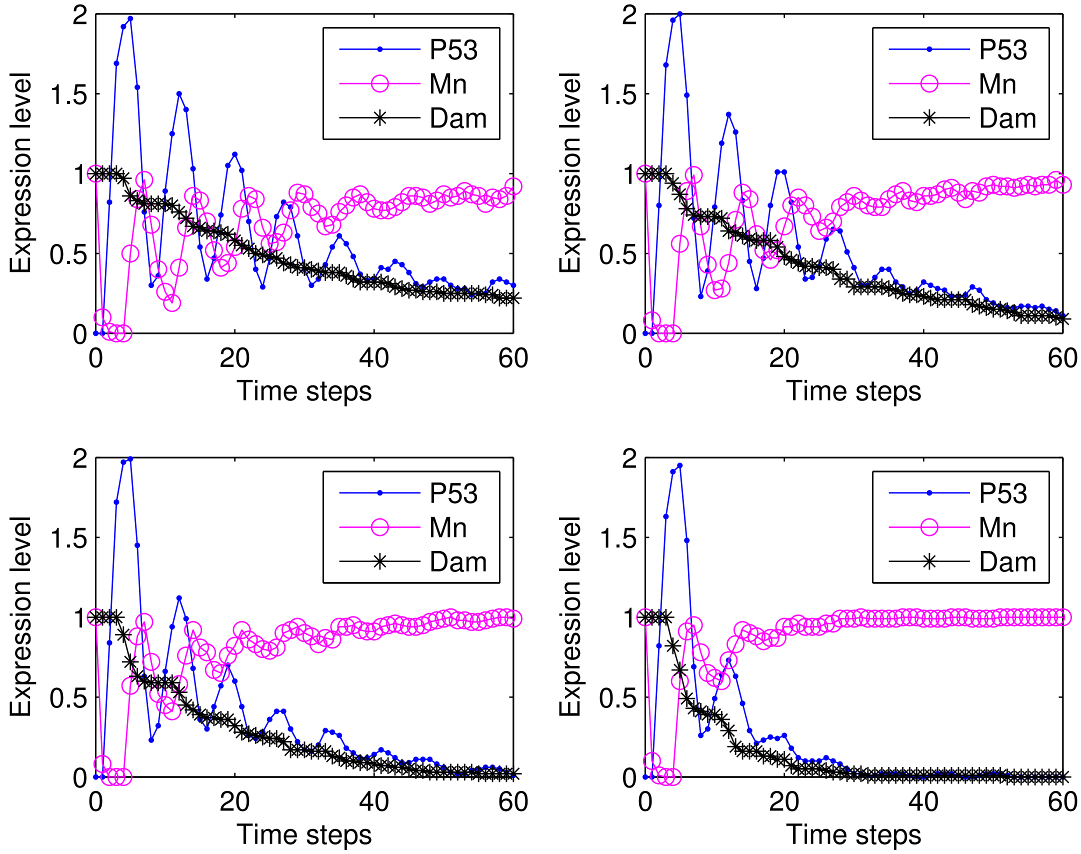


Figure 5.5: **Cell population simulations**. Each subfigure was generated from 100 simulations, each representing a single cell with sixty time steps. Starting from the top left frame to the right bottom frame the degradation propensity for DNA damage was increased by 5%, i.e. $p^{\downarrow}_{\text{dam}} = .05$ (top left), $p^{\downarrow}_{\text{dam}} = .10$ (top right), $p^{\downarrow}_{\text{dam}} = .15$ (bottom left), and $p^{\downarrow}_{\text{dam}} = .2$ (bottom right). The x-axis represents discrete time steps and the y-axis the average expression level. The initial state for all simulations was 0011. This figure shows that, if the intensity of the damage is increased more cells exhibit oscillations in the level of p53, in agreement with experimental observations [63].

Table 5.2: **Propensity parameters for Figure 5.8 (top frame)**

	CI	**CRO**	**CII**	***N***
Activation	.8	.2	.9	.9
Degradation	.2	.8	.9	.9

There is a high activation propensity for CI while a low activation propensity for CRO.

is known that, if the gene CI is fully expressed, all other genes are off. In the absence of CRO protein, CI is fully expressed (even in the absence of N and CII). CI is fully repressed provided that CRO is active and CII is absent.

The dynamics of this network is a bistable switch between lysis and lysogeny, Figure 5.7. Lysis is the state where the phage will be replicated, killing the host. Otherwise, the network will transition to a state called lysogeny where the phage will incorporate its DNA into the bacterium and become dormant. It has been suggested [90, 87] that these cell fate differences are due to spontaneous changes in the timing of individual biochemical reaction events.

The state space for this model is specified by $[0, 2] \times [0, 3] \times [0, 1] \times [0, 1]$, that is, the first variable, CI, has three levels $\{0, 1, 2\}$, the second variable, CRO, has four levels $\{0, 1, 2, 3\}$, and the third and fourth variables, CII and N, are Boolean. Update functions for this model are available in our supporting material. This model has a steady state, 2000, and a 2-cycle involving 0200 and 0300. The steady state 2000 represents lysogeny where CI is fully expressed while the other genes are off. The cycle between 0200 and 0300 represents lysis where CRO is active and other genes are repressed.

Cell population simulations were performed to measure the cell-to-cell variability. Figure 5.8 was generated using the probabilities given in Tables 5.2 (top frame) and 5.3 (bottom frame). The x-axis in both subfigures represents discrete time steps while the y-axis captures the average expression level. The initial state for all simulations was 0000 which represents the state of the bacterium at the moment of phage infection. Figure 5.8 shows variability in developmental outcome, some of the networks transition to lysis while others transition to lysogeny. To measure how sensitive the dynamics of the network is to changes in the propensity probabilities, we have plotted the outcome of lysis-lysogeny percentages for different choices of these parameters. Figure 5.9 shows the variation in developmental outcome as a function of the propensity parameters of CI and CRO. Star points indicate the percentage of networks that transition to lysogeny and circle shaped points indicate the percentage of networks that end up in lysis. The bottom x-axis contains activation propensities for CI and degradation propensities for CRO while the top x-axis contains activation propensities for CRO and degradation propensities for CI. The activation and degradation propensities for CII and N were all set equal to .9. Although the probability distributions for CI and CRO are very symmetric in Figure 5.9, it gives a good idea of how the variability in developmental outcome will change as the propensity parameters change.

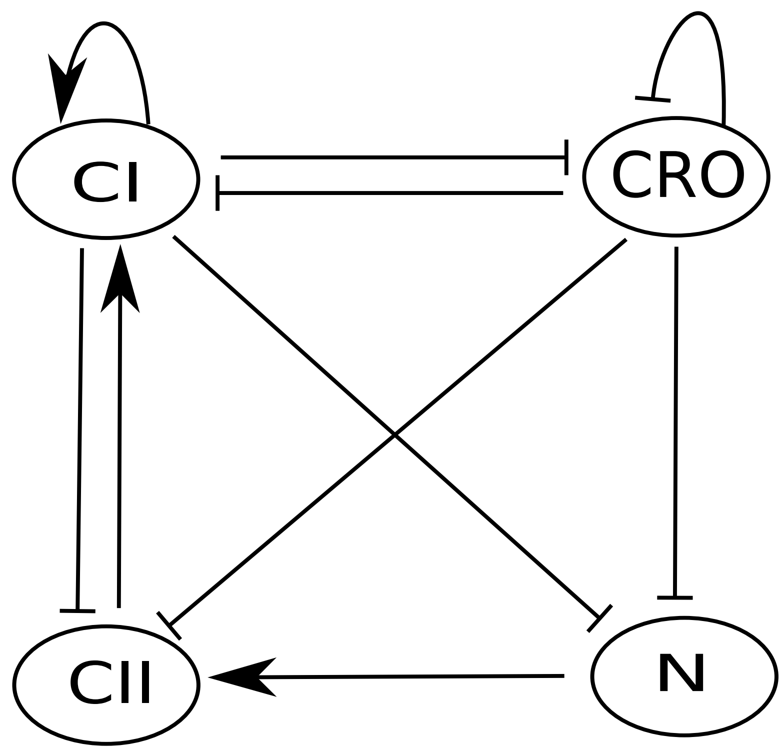


Figure 5.6: **Wiring diagram for phage lambda infection model**.

Table 5.3: **Propensity parameters for Figure 5.8 (bottom frame)**

	CI	**CRO**	**CII**	***N***
Activation	.3	.7	.9	.9
Degradation	.7	.3	.9	.9

There is a high activation propensity for *CRO* while a low activation propensity for *CI*.

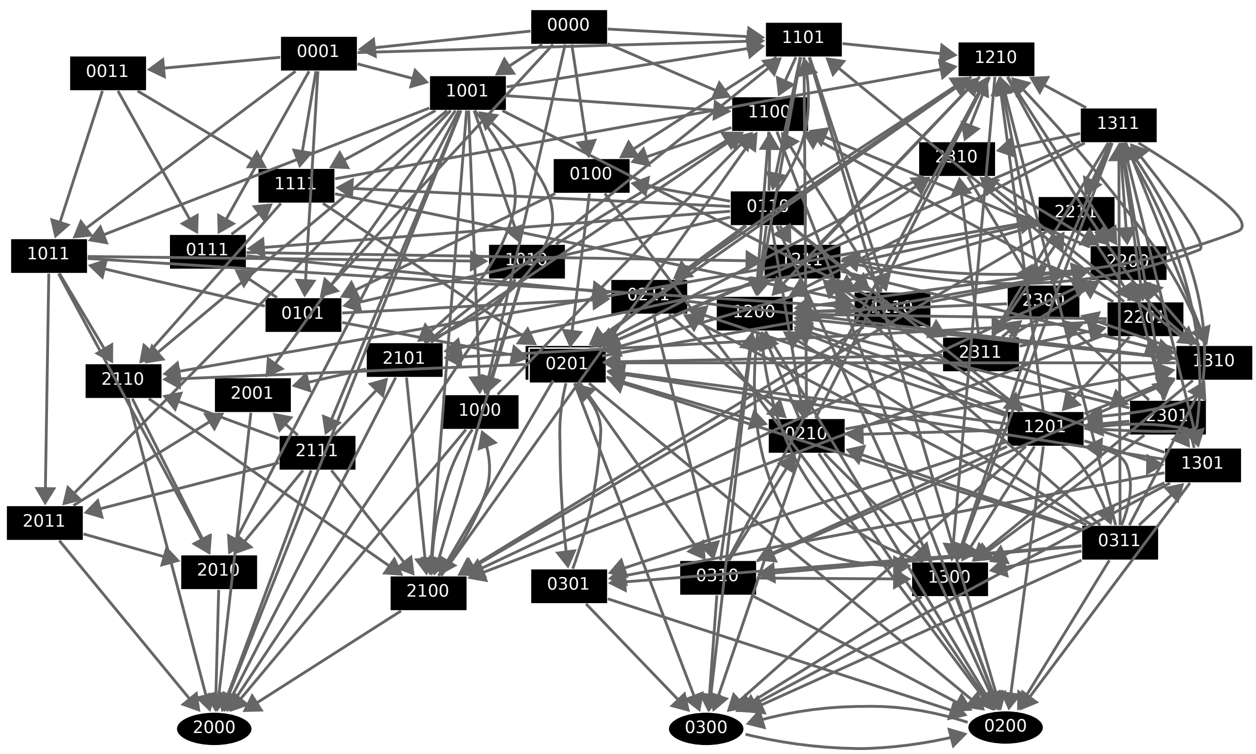


Figure 5.7: **State space for phage lambda model**. The order of variables in each vector state is CI, CRO, CII, N. The steady state 2000 represents lysogeny where CI is fully expressed while other genes are off. The cycle between 0200 and 0300 represents lysis where CRO is active and other genes are repressed. Self-loops are not depicted.

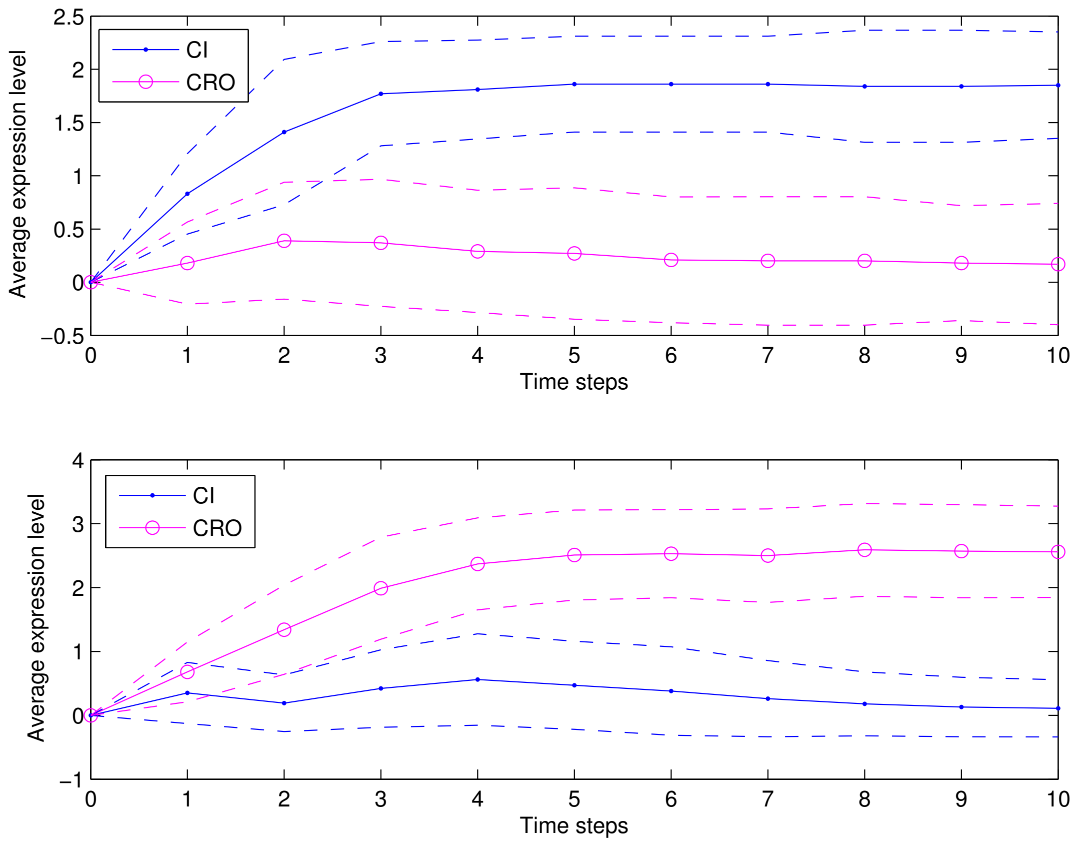


Figure 5.8: **Cell population simulations**. Both figures were generated from 100 simulations, each representing a single cell iteration of ten time steps. Top frame for parameters in Table 5.2 shows 93% lysis and 7% lysogeny while bottom frame for parameters in Table 5.3 shows 4% lysis and 96% lysogeny. The x-axis represents discrete time steps while the y-axis shows the average expression level. The initial state for all the simulations is 0000. Solid (circle) points correspond to the average of CI (CRO), and dashed lines represent standard deviations.

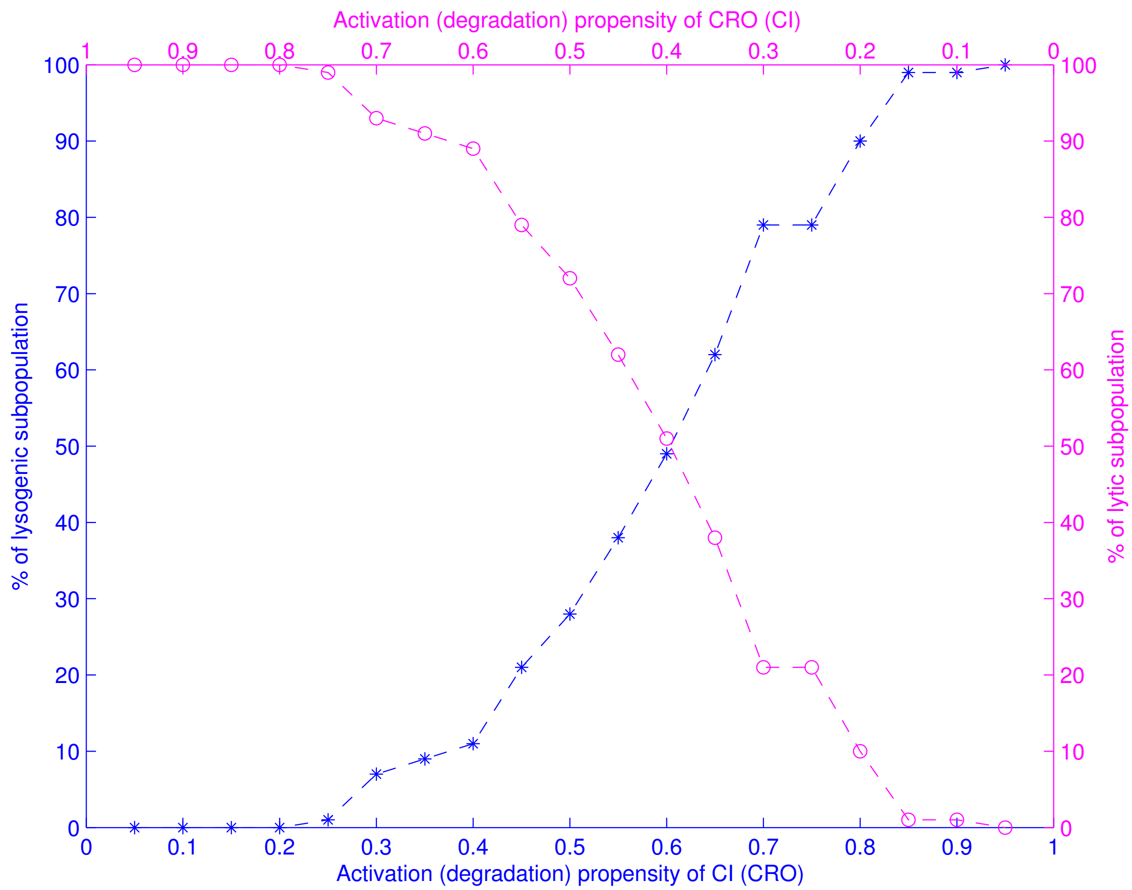


Figure 5.9: **Variation in developmental outcome as a function of the propensity parameters**. Star points indicate the percentage of networks that transition to lysogeny and circle shaped points indicate the percentage of networks that end up in lysis. Bottom axis represents the activation (and degradation) propensities for CI (CRO) in increasing order. Likewise, the top axis represents the activation (and degradation) propensities for CRO (CI) in decreasing order.

5.4 Conclusions

Using a discrete modeling strategy, this article introduces a framework to simulate stochasticity in gene regulatory networks at the function level, based on the general concept of PBNs. It accounts for intrinsic noise due to spontaneous differences in timing, small fluctuations in concentration levels, small numbers of reactant molecules, and fast and slow reactions. This framework was tested using two widely studied regulatory networks, the regulation of the $p53$-$Mdm2$ network and the control of phage lambda infection of bacteria. It is shown that in both of these examples the use of propensity probabilities for activation and degradation of network nodes provides a natural setup for cell population simulations to study cell-to-cell variability. The new features of this framework are the introduction of activation and degradation propensities that determine how fast or slow the discrete variables are being updated. This provides the ability to generate more realistic simulations of both single cell and cell population dynamics. In the example of the $p53$-$Mdm2$ system, one can see that individual simulations show sustained oscillations when DNA damage is present, while at the cell population level these individual oscillations average to a damped oscillation. This agrees with experimental observations [63]. In the second example, λ-phage infection of bacteria, it is observed that differences in developmental outcome due to intrinsic noise can be captured with this framework. Due to the lack of experimental data we are unable to calibrate the model so that it reproduces the correct difference in percentages due to intrinsic noise. So instead we present a plot of the difference in developmental outcome as a function of the propensity parameters.

It is worth noting that this article addresses only intrinsic noise generated from small fluctuations in concentration levels, small numbers of reactant molecules, and fast and slow reactions. Extrinsic noise is another source of stochasticity in gene regulation [82, 62], and it would be interesting to see if this framework or a similar setup can be adapted to account for extrinsic stochasticity under the discrete approach. This framework also lends itself to the study of intrinsic noise and it is useful for the study of developmental robustness. For instance, one could ask what the effect of this type of noise is on the dynamics of networks controlled by biologically inspired functions.

Relating the propensity parameters to biologically meaningful information or having a systematic way for estimating them is very important. A preliminary analysis shows that it is possible to relate the propensity parameters in this framework with the propensity functions in the Gillespie algorithm under some conditions (see Appendix B where for a simple degradation model, the degradation propensity is correlated by a linear equation with the decay rate of the species being degraded). More precisely, in the Gillespie algorithm [72, 73], if one discretizes the number of molecules of a chemical species into discrete expression levels such that within these levels the propensity functions for this species do not change significantly, then one obtains the setup of the framework presented here as a discrete model. That is, simulation within the framework presented here can be viewed as a further discretization of the Gillespie algorithm, in a setting that does not require exact knowledge of model parameters.

For a similar approach see [69].

Chapter 6

Future Directions

Building off of the research presented in this thesis, I have compiled a list of research problems that one could pursue.

1. Characterize more finely the dynamics of multi-state nested canalyzing functions [58], taking into account the level of influence of individual variables.
2. The framework presented in Chapter 5 is ideal for the study of developmental robustness, e.g. intrinsic noise. Precisely, changes in the propensity parameters account for different degrees of intrinsic noise. In this setting, understanding how dynamics of a network is affected by this kind of noise is of particular importance. One could start testing the robustness of nested canalyzing functions against this type of noise.
3. Expand the framework in Chapter 5 so that it can account for extrinsic noise as well.
4. Comparing the framework in Chapter 5 with the Gillespie algorithm and with stochastic differential equations to correlate the propensity probabilities with biologically meaningful information such as reaction rates and concentrations.
5. For Chapter 3, design a project to experimentally test some of the theoretical results about nested canalyzing functions.
6. For Chapter 2, the study bithreshold system for directed graphs, specially to find out how the loop number of the network would constrain its dynamics.

Other research directions to follow within this context include,

1. Study the relationship of mutational robustness (structure) and developmental robustness (dynamics).

2. Characterizing minimal bistable systems within the discrete setting, i.e. determining necessary and sufficient conditions under which the system undergoes bistability.

3. Development of bifurcation theory for discrete dynamical systems. See [4] for a preliminary approach.

Appendices

Appendix A

A.1 Appendix for Chapter 4

In this appendix we derive the polynomial form for the indicator functions Q_S. First, for each $r \in \mathbb{F}$ we denote the indicator function of the singleton set $\{r\}$ by P_r, i.e., for $x_0 \in \mathbb{F}$,

$$P_r(x_0) = \begin{cases} 1 & \text{if } x_0 = r, \\ 0 & \text{if } x_0 \neq r. \end{cases}$$

The polynomial form of P_r is given in the following lemma.

Lemma A.1.1. *For $r \in \mathbb{F}$, we have*

$$P_r(x) = (p-1)\prod_{\substack{a\in\mathbb{F}\\ a\neq r}}(x-a),$$

which has the expanded form

$$P_r(x) = \\ (p-1)\left[x^{p-1} + rx^{p-2} + r^2x^{p-3} + \cdots + r^{p-2}x + \prod_{\substack{a\in\mathbb{F}\\ a\neq r}} a\right].$$

Proof. Let $g(x) = (p-1)\prod_{\substack{a\in\mathbb{F}\\ a\neq r}}(x-a)$. We want to prove that $g(x) = P_r(x)$ for all $x \in \mathbb{F}$. Clearly, $g(x_0) = 0$ if $x_0 \neq r$. It remains to prove that $g(r) = 1$. From the definition of g, it can be expanded as

$$\begin{aligned} g(x) &= (p-1)(x(x-1)\cdots(x-(r-1)) \\ &\quad \cdots(x-(r+1))\cdots(x-(p-1))). \end{aligned}$$

Then

$$\begin{aligned} g(r) &= (p-1)r!(-1)^{p-1-r}(p-1-r)! \\ &= (p-1)(-1)^r(-1)(-2)\cdots(-r) \\ &\quad (-1)^{p-1-r}(p-1-r)!. \end{aligned}$$

Now, since $p - i = -i$ (mod p) for $i = 1, \ldots, r$, we get

$$\begin{aligned} g(r) &= (p-1)(-1)^{p-1}(p-1)(p-2)\cdots \\ &\quad (p-r)(p-1-r)! \\ &= (p-1)(p-1)! \\ &= (p-1)(p-1) \text{ (from Wilson's Theorem)} \\ &= 1. \end{aligned}$$

This proves the first assertion. For the second claim, from the previous formula for P_r, we see that P_r is a polynomial of degree $p-1$, so P_r can be written in the form

$$\begin{aligned} P_r(x) &= (p-1)[x^{p-1} + a_{p-2}x^{p-2} \\ &\quad +a_{p-3}x^{p-3} + \cdots + a_1x + a_0], \end{aligned}$$

where

$$a_{p-j} = (-1)^j \sum_{\substack{b_1,\ldots,b_j \in \mathbb{F} \\ b_1,\ldots,b_j \neq r}} b_1 \ldots b_j,$$

for $j \in \{2, \ldots, p-1\}$. Then

$$a_{p-j} =$$

$$(-1)^j \sum_{\substack{b_1=0 \\ b_1\neq r}}^{p-1} \cdots \sum_{\substack{b_{j-1}=0 \\ b_{j-1}\neq r}}^{p-1} \left[\sum_{\substack{b_j\in\mathbb{F} \\ b_j\neq r}} b_1 \ldots b_{j-1} b_j \right]$$

$$= (-1)^j \sum_{\substack{b_1=0 \\ b_1\neq r}}^{p-1} \cdots \sum_{\substack{b_{j-1}=0 \\ b_{j-1}\neq r}}^{p-1} \left[b_1 \ldots b_{j-1} \sum_{\substack{b_j\in\mathbb{F} \\ b_j\neq r}} b_j \right]$$

$$= (-1)^j \sum_{\substack{b_1=0 \\ b_1\neq r}}^{p-1} \cdots \sum_{\substack{b_{j-1}=0 \\ b_{j-1}\neq r}}^{p-1} \left[b_1 \ldots b_{j-1}(-r) \right]$$

$$= (-1)^j \sum_{\substack{b_1=0 \\ b_1\neq r}}^{p-1} \cdots \sum_{\substack{b_{j-1}=0 \\ b_{j-1}\neq r}}^{p-1} \left[b_1 \ldots b_{j-2}(-r) \sum_{\substack{b_{j-1}\in\mathbb{F} \\ b_{j-1}\neq r}} b_{j-1} \right]$$

$$= (-1)^j \sum_{\substack{b_1=0 \\ b_1\neq r}}^{p-1} \cdots \sum_{\substack{b_{j-1}=0 \\ b_{j-1}\neq r}}^{p-1} \Big[b_1 \ldots b_{j-2}(-r)(-r) \Big]$$

$$= (-1)^j(-r)^j$$

$$= r^j.$$

Finally,

$$a_0 = (-1)^{p-1} \prod_{\substack{a\in\mathbb{F} \\ a\neq r}} a = \prod_{\substack{a\in\mathbb{F} \\ a\neq r}} a.$$

Remark A.1.1. Note that $p-1$ is even for $p>2$ and $1=-1$ for $p=2$.

This completes the proof. □

Finally, the polynomial form of Q_S is given in the following lemma.

Lemma A.1.2. *For $S \subset \mathbb{F}$, we have*

$$Q_S(x) = \sum_{r\in\mathbb{F}\setminus S} P_r(x)$$

$$= a_{p-1}x^{p-1} + a_{p-2}x^{p-2} + \cdots + a_1 x + a_0,$$

where $a_i = (p-1)\sum_{r\in\mathbb{F}\setminus S} r^{p-1-i}$ for $i = 1,\ldots,p-1$ and $a_0 = (p-1)\sum_{r\in\mathbb{F}\setminus S}\Bigg(\prod_{\substack{a\in\mathbb{F} \\ a\neq r}} a\Bigg)$.

Proof. Clearly, if $x_0 \in S$, then $P_r(x_0) = 0$ for all $r \in \mathbb{F}\setminus S$. Therefore

$$Q_S(x_0) = \sum_{r\in\mathbb{F}\setminus S} P_r(x_0) = 0.$$

Similarly, if $x_0 \notin S$, then $P_{x_0}(x_0) = 1$ and $P_r(x_0) = 0$ for all $r \neq x_0$ in $\mathbb{F}\setminus S$. Therefore

$$Q_S(x_0) = \sum_{r\in\mathbb{F}_p\setminus S} P_r(x_0) = 1.$$

The second equality follows from Lemma A.1.1. This completes the proof.

□

A.2 Proof of Theorem 4.4.2

Proof. Let us first assume that the polynomial f is a nested canalyzing function with canalyzing input sets $S_1, \ldots, S_n$ and canalyzing output values $b_1, \ldots, b_{n+1}$. Then, by Theorem 4.3.1, f can be expanded as

$$f(x_1, \ldots, x_n) = \sum_{j=0}^{n-1} \left\{ B_{n-j} \prod_{i=1}^{n-j} Q_{S_i}(x_i) \right\} + b_1$$

$$= \sum_{j=0}^{n-1} \left\{ B_{n-j} \sum_{\substack{(i_1,\ldots,i_{n-j}) \\ i_t \in \mathbb{F} \\ t=1,\ldots,n-j}} a^1_{i_1} \ldots a^{n-j}_{i_{n-j}} x_1^{i_1} x_2^{i_2} \ldots x_{n-j}^{i_{n-j}} \right\}$$

$$+ \sum_{j=0}^{n-1} \left\{ (b_{n-j+1} - b_{n-j}) a^1_0 \ldots a^{n-j}_0 \right\} + b_1,$$

where Q_{S_i} is defined as in Lemma A.1.2, i.e.,

$$Q_{S_i}(x_i) = a^i_{p-1} x_i^{p-1} + a^i_{p-2} x_i^{p-2} + \cdots + a^i_0.$$

Then we have that

$$C_{[p-1]} = (b_{n+1} - b_n) a^1_{p-1} \cdots a^n_{p-1}$$

Now, if $j \neq n$, then

$$C^{i_j}_{[p-1]\setminus\{j\}} =$$
$$(b_{n+1} - b_n) a^1_{p-1} \ldots a^{j-1}_{p-1} a^j_{i_j} a^{j+1}_{p-1} \ldots a^n_{p-1}$$

Then

$$C^{i_j}_{[p-1]\setminus\{j\}} = C_{[p-1]} (a^j_{p-1})^{-1} a^j_{i_j}$$

So

$$a^j_{i_j} = a^j_{p-1} C^{-1}_{[p-1]} C^{i_j}_{[p-1]\setminus\{j\}} \tag{A.2.1}$$

for $j = 1, \ldots, n-1$. Now,

$$C_{i_1 \ldots i_{n-\mu}} =$$
$$\sum_{j=0}^{\mu} \left\{ B_{n-j} a^1_{i_1} \ldots a^{n-\mu}_{i_{n-\mu}} a^{n-\mu+1}_0 \ldots a^{n-j}_0 \right\}$$
$$= a^1_{i_1} \ldots a^{n-\mu}_{i_{n-\mu}} \sum_{j=0}^{\mu} \left\{ B_{n-j} a^{n-\mu+1}_0 \ldots a^{n-j}_0 \right\}$$

and

$$C^{p-1,\ldots,p-1}_{[0]\setminus\{1,\ldots,n-\mu\}} =$$

$$\sum_{j=0}^{\mu} \left\{ B_{n-j} a^1_{p-1} \ldots a^{n-\mu}_{p-1} a^{n-\mu+1}_0 \ldots a^{n-j}_0 \right\}$$

$$= a^1_{p-1} \dots a^{n-\mu}_{p-1} \sum_{j=0}^{\mu} \left\{ B_{n-j} a_0^{n-\mu+1} \dots a_0^{n-j} \right\}$$

$$= \left[\prod_{i=1}^{n-\mu} a^i_{p-1} \right] \sum_{j=0}^{\mu} \left\{ B_{n-j} a_0^{n-\mu+1} \dots a_0^{n-j} \right\}$$

Therefore

$$\sum_{j=0}^{\mu} \left\{ B_{n-j} a_0^{n-\mu+1} \dots a_0^{n-j} \right\} = \left[\prod_{i=1}^{n-\mu} a^i_{p-1} \right]^{-1} C^{p-1,\dots,p-1}_{[0]\setminus\{1,\dots,n-\mu\}}. \tag{A.2.2}$$

Now, using Equations A.2.1 and A.2.2

$$C_{i_1 \dots i_{n-\mu}} = \prod_{j=1}^{n-\mu} a^j_{i_j} \left[\sum_{j=0}^{\mu} \left\{ B_{n-j} a_0^{n-\mu+1} \dots a_0^{n-j} \right\} \right]$$

$$= \prod_{j=1}^{n-\mu} \left[a^j_{p-1} C^{-1}_{[p-1]} C^{i_j}_{[p-1]\setminus\{j\}} \right] \left[\prod_{j=1}^{n-\mu} a^j_{p-1} \right]^{-1} C^{p-1,\dots,p-1}_{[0]\setminus\{1,\dots,n-\mu\}}$$

$$= C^{p-1,\dots,p-1}_{[0]\setminus\{1,\dots,n-\mu\}} \prod_{j=1}^{n-\mu} C^{-1}_{[p-1]} C^{i_j}_{[p-1]\setminus\{j\}}.$$

Conversely, suppose Equations 4.4.2 - 4.4.6 hold for the coefficients of the polynomial $f(x_1, \dots, x_n)$ in Equation 4.4.7. We need to show that $f(x_1, \dots, x_n)$ is a nested canalyzing function. Let $a^j_{p-1} = (p-1) \mid S_j^c \mid$,
$a^j_{i_j} = (p-1) \sum_{r \in S_j^c} r^{p-1-i_j}$, and $a^j_0 = Q_{S_j}(0)$ for $j = 1, \dots, n$. Then

$$C_{[p-1]} = (b_{n+1} - b_n) \prod_{i=1}^{n} a^i_{p-1},$$

and

$$C^{-1}_{[p-1]} C^{i_j}_{[p-1]\setminus\{j\}} = a^j_{i_j} [a^j_{p-1}]^{-1}, \tag{A.2.3}$$

as well as

$$C^{p-1,\dots,p-1}_{[0]\setminus\{1,\dots,n-\mu\}} = \left[\prod_{i=1}^{n-\mu} a^i_{p-1}\right]\sum_{j=0}^{\mu}\left\{B_{n-j}a_0^{n-\mu+1}\dots a_0^{n-j}\right\}. \tag{A.2.4}$$

From Equation 4.4.2 we get

$$C_{i_1\dots i_{n-\mu}} = C^{p-1,\dots,p-1}_{[0]\setminus\{1,\dots,n-\mu\}}\prod_{j=1}^{n-\mu} C^{-1}_{[p-1]}C^{i_j}_{[p-1]\setminus\{j\}},$$

and from Equation A.2.3 we get

$$C_{i_1\dots i_{n-\mu}} = C^{p-1,\dots,p-1}_{[0]\setminus\{1,\dots,n-\mu\}}\prod_{j=1}^{n-\mu} a^j_{i_j}[a^j_{[p-1]}]^{-1}.$$

Then, from Equation A.2.4 we get

$$C_{i_1\dots i_{n-\mu}} = \left[\prod_{i=1}^{n-\mu} a^i_{i_j}\right]\sum_{j=0}^{\mu}\left\{B_{n-j}a_0^{n-\mu+1}\dots a_0^{n-j}\right\}.$$

From Equation 4.4.6 we get

$$C_{[0]} - b_1 = C^{p-1}_{[0]\setminus\{1\}}(C^0_{[p-1]\setminus\{1\}}C^{-1}_{[p-1]}).$$

From Equation A.2.4 we get

$$C^{p-1}_{[0]\setminus\{1\}} = a^1_{p-1}\sum_{j=0}^{n-1}\left\{B_{n-j}a_0^2\dots a_0^{n-j}\right\}, \tag{A.2.5}$$

and from Equation A.2.3 we get

$$(C^0_{[p-1]\setminus\{1\}}C^{-1}_{[p-1]}) = a^1_0(a^1_{p-1})^{-1}. \tag{A.2.6}$$

Then, from Equations A.2.5- A.2.6 we get

$$C_{[0]} = b_1 + \sum_{j=0}^{n-1}\left\{B_{n-j}a^1_0\dots a_0^{n-j}\right\}.$$

Now,

$$f(x_1, \cdots, x_n) =$$

$$\sum_{\mu=0}^{n-1} \Bigg\{ \sum_{\substack{(i_1,\ldots,i_{n-\mu}) \\ i_t \in \mathbb{F}_p \\ t=1,\ldots,n-\mu}} C_{i_1 \ldots i_{n-\mu}} x_1^{i_1} \ldots x_{n-j}^{i_{n-j}} \Bigg\}$$

$$+C_{[0]}.$$

We have

$$f(x_1, \cdots, x_n) =$$

$$\sum_{j=0}^{n-1} \Bigg\{ B_{n-j} \sum_{\substack{(i_1,\ldots,i_{n-j}) \\ i_t \in \mathbb{F} \\ t=1,\ldots,n-j}} a_{i_1}^1 \ldots a_{i_{n-j}}^{n-j} x_1^{i_1} x_2^{i_2} \ldots x_{n-j}^{i_{n-j}} \Bigg\}$$

$$+b_1 + \sum_{j=0}^{n-1} \Bigg\{ (b_{n-j+1} - b_{n-j}) a_0^1 \ldots a_0^{n-j} \Bigg\}.$$

Therefore

$$f(x_1, \ldots, x_n) = \sum_{j=0}^{n-1} \Bigg\{ B_{n-j} \prod_{i=1}^{n-j} Q_{S_i}(x_i) \Bigg\} + b_1,$$

which is nested canalyzing by Theorem 4.3.1. $\square$

Appendix B

B.1 Stochastic Discrete Dynamical Systems: (SDDS)

Let $x_1, \ldots, x_n$ be variables which can take values in finite sets $X_1, \ldots, X_n$, respectively. Let $X = X_1 \times \cdots \times X_n$ be the Cartesian product. A *stochastic discrete dynamical system* in the variables $x_1, \ldots, x_n$ is a collection of n triplets

$$F = \{f_i, p_i^{\uparrow}, p_i^{\downarrow}\}_{i=1}^n$$

Let $m_1, \ldots, m_n$ be the number of elements of finite sets $X_1, \ldots, X_n$, respectively. Let $m = m_1 \times \cdots \times m_n$. The state space of $F = \{f_i, p_i^{\uparrow}, p_i^{\downarrow}\}_{i=1}^n$ over X consists of m states. The network may transition from one state to other possible states. Algorithm B.1.1 describes how to calculate a next state starting from an initialization. The probability of transition from state $x = (x_1, \ldots, x_n)$ to state $y = (y_1, \ldots, y_n)$ is calculated using Algorithm B.1.2.

B.1.1 SDDS includes trajectories from the synchronous and asynchronous schemes

If $F = \{f_i, p_i^{\uparrow}, p_i^{\downarrow}\}_{i=1}^n$ a stochastic discrete dynamical system, then F has the same steady states as the deterministic system $G = \{f_i\}_{i=1}^n$. This follows from the fact that

$$prob(x \to x) = 1 \iff prob(x_i \to x_i) = 1 \; \forall i \iff x_i = f_i(x) \iff f(x) = x.$$

Note that $prob(x \to y) = prob(x_1 \to y_1) \cdots prob(x_n \to y_n)$. Above, ‘ $\iff$ ’ means ‘if and only if’.

It is also important to note that the dynamics of F includes all the different dynamics that can be generated from G using different update schemes. For instance, for the synchronous case, consider x, y such that $f(x) = y$, that is $f_i(x) = y_i$ for all $i = 1, \ldots, n$. There are two cases to consider:

- If $x_i = y_i = f_i(x)$, then $prob(x_i \to y_i) = 1$.
- If $x_i \neq y_i = f_i(x)$, then $prob(x_i \to y_i) = (p_i^{\uparrow} \text{ or } p_i^{\downarrow}) > 0$.

In any case, $prob(x_i \to y_i) > 0$. Then,

$$prob(x \to y) = \text{product(positive numbers)} > 0.$$

Then $x \to y$ is an edge in the state space of the SDDS. Note that if $p_i^{\uparrow} = p_i^{\downarrow} = 1$ for all $i = 1, \ldots, n$, then F is a synchronous deterministic system.

For the asynchronous case: assume all propensities are in $(0, 1)$. Consider x, y such that $x_i = y_i$ for all $i \neq k$ and $f_k(x) = y_k \neq x_k$. For $i \neq k$: two cases:

- if $x_i = y_i = f_i(x)$, then $prob(x_i \to y_i) = 1$.
- if $x_i = y_i \neq f_i(x)$, then $prob(x_i \to y_i) = (1 - p_i^{\uparrow} \text{ or } 1 - p_i^{\downarrow}) > 0$.

In any case $prob((x_i \to y_i) > 0$. For $i = k$: then $prob((x_i \to y_i) = (p_i^{\uparrow} \text{ or } p_i^{\downarrow}) > 0$. Then, for all $i = 1, \ldots, n$ we have $prob(x_i \to y_i) > 0$. Then,

$$prob(x \to y) = \text{product(positive numbers)} > 0.$$

Therefore $x \to y$ is an edge in the state space of the SDDS.

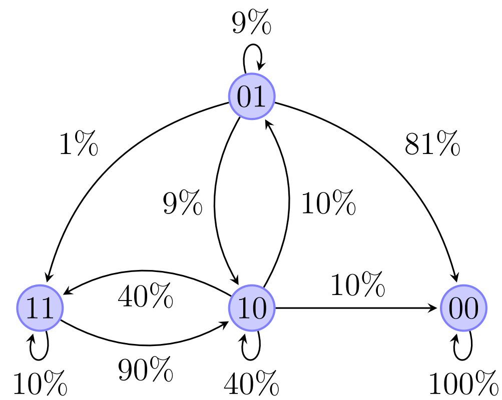
B.1.2 SDDS can be written as a PBN but needs more parameters

For $n = 2$, $X = \{0, 1\} \times \{0, 1\}$, $F = (f_1, f_2) : X \to X$, where

x_1	x_2	f_1	f_2
0	0	0	0
0	1	1	0
1	0	0	1
1	1	1	0

and

	x_1	x_2
Activation	.1	.5
Degradation	.2	.9



Now, suppose that there is a PBN that has the same state space. Let us focus on the Boolean functions for the first variable, which we denote $F_1 = \{h_1, h_2, \dots, h_r\}$. Notice that for the first variable we have the following transitions:

$$prob(00 \to 0) = 1.$$

$$prob(01 \to 0) = .9 = 1 - p_1^{\uparrow}$$
$$prob(01 \to 1) = .1 = p_1^{\uparrow}$$
$$prob(10 \to 0) = .2 = p_1^{\downarrow}$$
$$prob(10 \to 1) = .8 = 1 - p_1^{\downarrow}$$
$$prob(11 \to 1) = 1$$

From these equalities it follows that the number of Boolean functions for the first variable is 4 (all other Boolean functions would have probability 0), namely:

x_1	x_2	h_1	h_2	h_3	h_4
0	0	0	0	0	0
0	1	0	0	1	1
1	0	0	1	0	1
1	1	1	1	1	1

Now, denote with p_1, p_2, p_3, p_4 the probabilities corresponding to h_1, h_2, h_3, h_4, resp. Then, we have the following equations:

$$.1 = p_3 + p_4,$$

$$.8 = p_2 + p_4,$$

and

$$p_1 + p_2 + p_3 + p_4 = 1.$$

It follows that

$$p_2 = .9 - p_1, p_3 = .2 - p_1, p_4 = p_1 - .1$$

If we want to keep the number of functions in the PBN as small as possible, we can consider $p_3 = 0$ or $p_4 = 0$ (but not both). Hence, the PBN needs at least 3 Boolean functions for the first variable.

Now, suppose that a PBN has the same state space and let us focus on the Boolean functions for the second variable, which we denote $F_2 = \{g1, g2, \dots, gm\}$. Notice that for the second variable we have the following transitions:

$$prob(00- > 0) = 1$$

$$prob(01- > 1) = .1 = 1 - p_2^{\downarrow}$$
$$prob(01- > 0) = .9 = p_2^{\downarrow}$$
$$prob(10- > 1) = .5 = p_2^{\uparrow}$$
$$prob(10- > 0) = .5 = 1 - p_2^{\uparrow}$$
$$prob(11- > 1) = .1 = 1 - p_2^{\downarrow}$$
$$prob(11- > 0) = .9 = p_2^{\downarrow}$$

From these equalities it follows that the number of Boolean functions for the first variable is 3 (all other Boolean functions would have probability 0), namely:

x_1	x_2	g_1	g_2	g_3
0	0	0	0	0
0	1	0	0	1
1	0	0	1	1
1	1	1	1	0

$$p_2 = .1,$$
$$p_2 + p_3 = .5,$$
$$p_2 = .1,$$
$$p_1 + p_2 + p_3 = 1,$$

Then,

$$p_1 = .5, p_2 = .1, p_3 = .4$$

This shows that the PBN needs at least 3 Boolean functions for the second variable. Therefore, the PBN needs a total of at least 6 Boolean functions and 6 parameters. Notice that the SDDS has only 2 functions and 4 parameters.

B.1.3 Pseudo-Codes

Algorithm B.1.1 describes how to calculate a next state starting from an initialization. The probability of transition from state $x = (x_1, \ldots, x_n)$ to state $y = (y_1, \ldots, y_n)$ is calculated using Algorithm B.1.2.

Algorithm B.1.1 Next state for Stochastic Discrete Dynamical Systems

---

input : Initial state x_0, and a SDDS $F = \{f_i, p_i^{\uparrow}, p_i^{\downarrow}\}_{i=1}^{n}$.
output: y = one of the next states of x.

```
Let z = F(x_0) = (f_1(x_0), ..., f_n(x_0)).
for i = 1 to n do
    Let r be a random number in [0,1].
    if x_i < z_i then
        if r < p_i^↑ then
            y_i = z_i with probability p_i^↑
        else
            y_i = x_i with probability 1 − p_i^↑
        end
    else if x_i > z_i then
        if r < p_i^↓ then
            y_i = z_i with probability p_i^↓
        else
            y_i = x_i with probability 1 − p_i^↓
        end
    else
        y_i = x_i with probability 100%
    end
end
```

B.2 Regulation in the p53-Mdm2 network

The p53-Mdm2 network is one of the most widely studied gene regulatory networks. W. Abou-Jaude, D. Ouattara, M. Kauffman [84] proposed a logical four-variable model to describe the dynamics of the tumor suppressor protein p53 and its negative regulator Mdm2 in the presence and absence of DNA damage. The wiring diagram of this model is represented in Figure B.1, where P, Mc, Mn, and Dam stand for protein p53, nuclear Mdm2, cytoplasmic Mdm2, and DNA damage, respectively.

The state space for this model is specified by $[0, 2] \times [0, 1] \times [0, 1] \times [0, 1]$, that is, except for

Algorithm B.1.2 Transition probability between two states for SDDS.

input : Two states x, y, and a SDDS $F = \{f_i, p_i^{\uparrow}, p_i^{\downarrow}\}_{i=1}^{n}$.
output: $P_{x,y}$ = The probability of transitioning from x to y.

```
Let z = F(x) = (f_1(x), ..., f_n(x)).
P_{x,y} = 1.
for i = 1 to n do
    Let c = 0.
    if x_i < z_i then
        if y_i = z_i then
            c = p_i^↑
        end
        if y_i = x_i then
            c = 1 − p_i^↑
        end
    else if x_i > z_i then
        if y_i = z_i then
            c = p_i^↓
        end
        if y_i = x_i then
            c = 1 − p_i^↓
        end
    else
        if y_i = x_i then
            c = 1
        end
    end
    P_{x,y} = P_{x,y} ∗ c
end
```

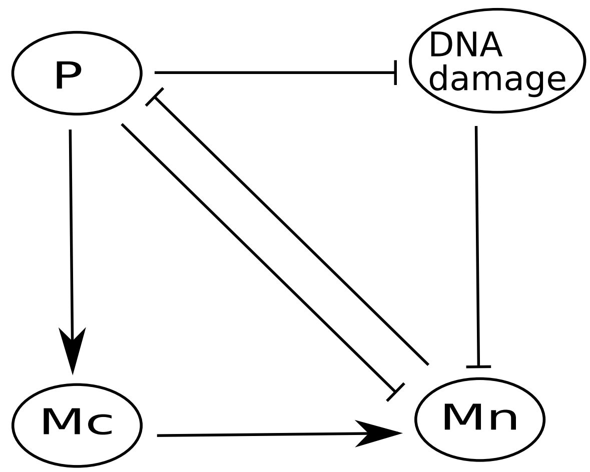


Figure B.1: Four-variable model for the p53-Mdm2 regulatory network. P, Mc, and Mn stand for protein p53, nuclear Mdm2, and cytoplasmic Mdm2 respectively.

Table B.1: Truth table for P

Mn P	P
0 0	1
0 1	2
0 2	2
1 0	0
1 1	0
1 2	1

the first variable P that has three levels $\{0, 1, 2\}$, all the other variables are Boolean.

As shown in Figure B.1, Mn acts negatively on P. The update rule for P, f_P, is specified by the truth table given at Table B.1

The update rule for Mc, f_{Mc}, which is specified by the truth table given at Table B.2.

The update rule for Mn, f_{Mn}, which is specified by the truth table given at Table B.3.

Finally, the update rule for DNA-damage, f_{Dam}, which is specified by the truth table given at Table B.4.

Table B.2: Truth table for Mc

P	Mc
0	0
1	0
2	1

Table B.3: Truth table for Mn

P Mc Dam	Mn
0 0 0	1
0 0 1	0
0 1 0	1
0 1 1	1
1 0 0	0
1 0 1	0
1 1 0	1
1 1 1	1
2 0 0	0
2 0 1	0
2 1 0	1
2 1 1	1

Table B.4: Truth tables for *DNA-damage*

P Dam	Dam
0 0	0
0 1	1
1 0	0
1 1	1
2 0	0
2 1	0

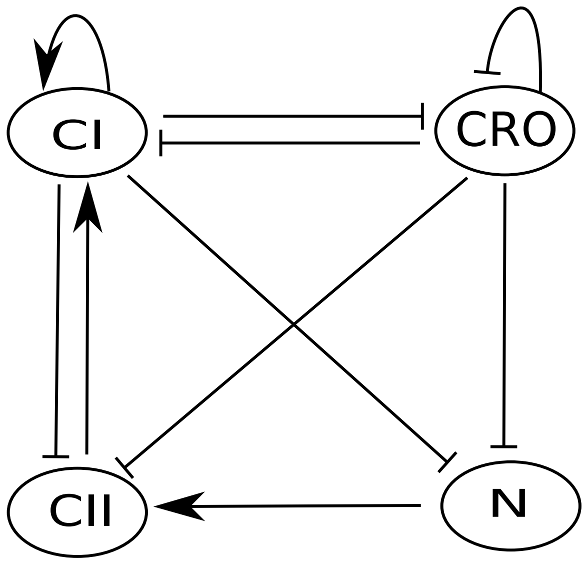


Figure B.2: Four-variable model for the lambda phage regulatory network.

Table B.5: Truth table for CI

CRO CII	CI
0 0	2
0 1	2
1 0	0
1 1	2
2 0	0
2 1	2
3 0	0
3 1	2

B.3 Lambda phage infection of bacteria

Thieffry and Thomas [87] built a multi-state logical model for the core lambda phage regulatory network. This model encompasses the roles of the regulatory genes *CI*, *Cro*, *CII*, and *N*. See Figure B.2 .

The state space for this model is specified by $[0,2] \times [0,3] \times [0,1] \times [0,1]$, that is, the first variable has three levels $\{0,1,2\}$, the second variable has four levels $\{0,1,2,3\}$, and the third and fourth variables are still Boolean.

The update rule for CI, f_{CI}, has inputs CRO and CII which is specified by the truth table given at Table B.5

Table B.6: Truth table for CRO

CI CRO	CRO
0 0	3
0 1	3
0 2	3
0 3	2
1 0	3
1 1	3
1 2	3
1 3	2
2 0	0
2 1	0
2 2	0
2 3	0

The update rule for CRO, f_{CRO}, which is specified by the truth table given at Table B.6.

The update rule for CII, f_{CII}, which is specified by the truth table given at Table B.7.

Finally, the update rule for N, f_N, which is specified by the truth table given at Table B.8.

B.4 Relating the propensity parameters with biological information

Consider the following simple degradation model from [73], page 40,

$$S \xrightarrow{k} \varnothing$$

The propensity function (in the Gillespie context) is $a(x) = kx$ and the state vector change is $\nu = -1$. The expected value of the solution of this stochastic model is given by $X(t) = x_0 \exp(-kt)$, where $X(t)$ represent the state of the system at time t.

Let us discretize the number of states even further, into two states 0 and 1.

$$\tilde{X}(t) = \begin{cases} 0 & X(t) < m \\ 1 & X(t) \geq m \end{cases}$$

where m is a fraction of the initial number of molecules, i.e. $m = \frac{x_0}{\eta}$. Now, from the expected value estimate we have

$$m = \frac{x_0}{\eta} = x_0 \exp(-k\tilde{t}) \text{ for some } \tilde{t}.$$

then

$$\ln(\frac{1}{\eta}) = -k\tilde{t}.$$

Table B.7: Truth table for CII

CI CRO N	CII
0 0 0	0
0 0 1	1
0 1 0	0
0 1 1	1
0 2 0	0
0 2 1	1
0 3 0	0
0 3 1	0
1 0 0	0
1 0 1	1
1 1 0	0
1 1 1	1
1 2 0	0
1 2 1	1
1 3 0	0
1 3 1	0
2 0 0	0
2 0 1	0
2 1 0	0
2 1 1	0
2 2 0	0
2 2 1	0
2 3 0	0
2 3 1	0

Table B.8: Truth tables for N

CI CRO	N
0 0	1
0 1	1
0 2	0
0 3	0
1 0	0
1 1	0
1 2	0
1 3	0
2 0	0
2 1	0
2 2	0
2 3	0

then

$$\frac{\ln(\eta)}{k} = \tilde{t}.$$

The probability distribution for the time to go from 1 to 0 in SDDS is a geometric distribution [91] (Pg. 65) with expected value $1/p_1^{\downarrow}$, i.e. the average 'waiting time' to go from 1 to 0 in SDDS is $1/p_1^{\downarrow}$. Therefore

$$\frac{\ln(\eta)}{k} \approx \delta \frac{1}{p_1^{\downarrow}}.$$

Then

$$p_1^{\downarrow} \approx \frac{\delta}{\ln(\eta)} k$$

Therefore

$$p_1^{\downarrow} \approx ck.$$

where $c = \frac{\delta}{\ln(\eta)}$.

In Figure B.3 we compare SDDS simulations with Gillespie simulations. In order to fit both plots in a single figure, we have normalized the number of molecules so that it goes from 0 to 1. Number of simulations for SDDS is 1000 and number of steps for each simulation was 6. The number of molecules for Gillespie was normalized so that it goes from 0 to 1. The scale parameter δ for SDDS was set equal to 20, i.e., 1 second in the scale of Gillespie is equivalent to 20 time steps in the scale of SDDS. Thus

$$c = \frac{\delta}{\ln(\eta)} = \frac{20}{\ln(5)} = 12.4267$$

Therefore

$$p_1^{\downarrow} \approx (12.4267)(0.05) = 0.6213.$$

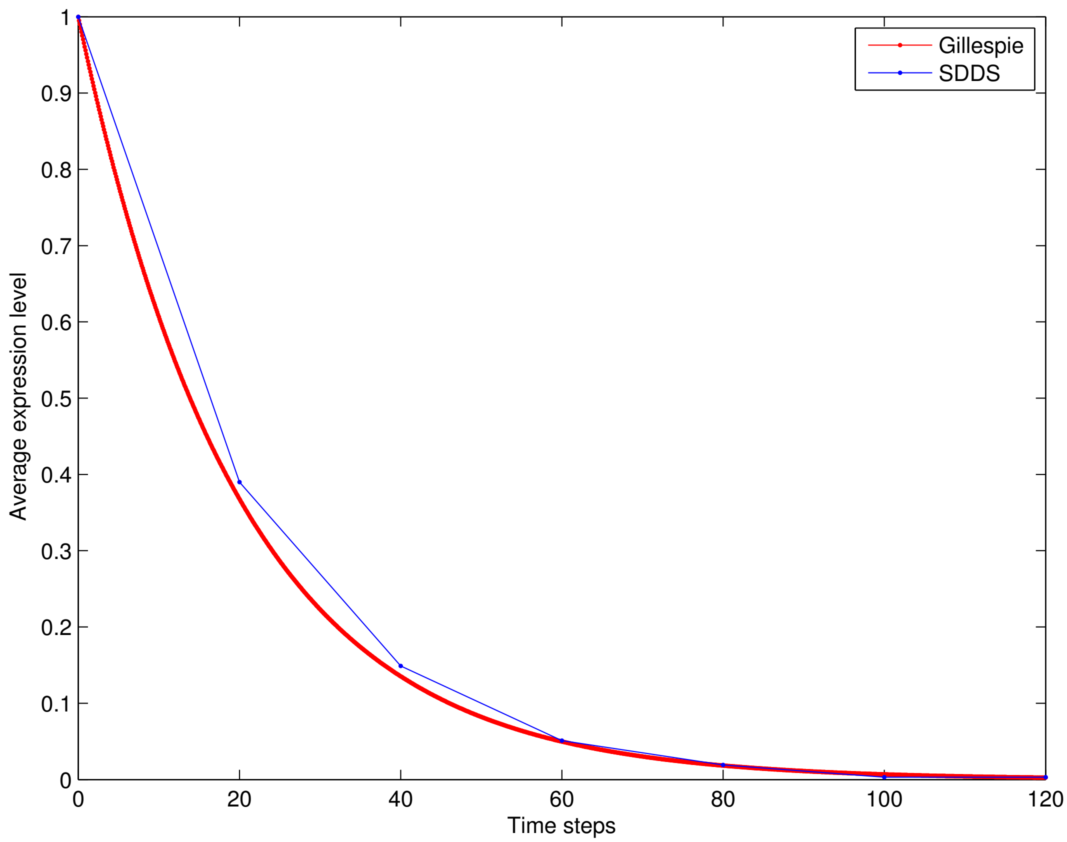


Figure B.3: SDDS vs Gillespie. Number of molecules for Gillespie is 1000 and degradation rate $k = 0.05$. Number of simulations for SDDS is 1000 and number of steps for each simulation was 6. The number of molecules for Gillespie was normalized so that it goes from 0 to 1. The scale parameter δ for SDDS was set equal to 20, i.e., 1 second in the scale of Gillespie is equivalent to 20 time steps in the scale of SDDS. Thus, $p_1^{\downarrow} = 0.6213$ (see text for better description).

Bibliography

[1] Frank J. Bruggeman and Hans V. Westerhoff. (2007) The nature of systems biology, *Trends in Microbiology,* 15: 1, 45 - 50.

[2] David Murrugarra, Alan Veliz-Cuba, Reinhard Laubenbacher. (2012) Structure and Dynamics of Acyclic Networks. *Under review.*

[3] Reinhard Laubenbacher, David Murrugarra, Alan Veliz-Cuba. (2011) Structure and Dynamics of Polynomial Dynamical Systems. *Proc. 2011 NSF Engineering Research and Innovation Conference, Atlanta, GA, January 2011.*

[4] Chris J. Kuhlman, Henning S. Mortveit, David Murrugarra, V. S. Anil Kumar. (2012) Bifurcations in Boolean Networks. *Discrete Mathematics and Theoretical Computer Science, proc, AP, 29-46, 2012.*

[5] U. Alon (2006) *An introduction to systems biology: design principles of biological circuits*. Chapman&Hall.

[6] J. T. Butler, T. Sasao, and M. Matsuura (2005), Average path length of binary decision diagrams. *IEEE Trans. Comp.* 27: 1041–1053.

[7] B. Elspas (1959) The theory of autonomous linear sequential networks. *IRE Trans. Circuit Theory* 6: 45–60.

[8] M. Gavalec (2001) Monotone eigenspace structure in max-min algebra. *Linear Algebra and its Applications* 345: 149–167.

[9] F. Hinkelmann, D. Murrugarra, A. Jarrah, and R. Laubenbacher. (2011) A mathematical framework for agent based models of complex biological networks. *Bulletin of Mathematical Biology*, pages 1-20. 10.1007/s11538-010-9582-8.

[10] B. De Schutter and B. De Moor. (1999) On the sequence of consecutive powers of a matrix in a Boolean algebra. *SIAM Journal on Matrix Analysis and Applications*, 21, 328 - 354.

[11] M. Golubitsky and I. Stewart (2002) Patterns of oscillation in coupled cell systems. Geometry, Mechanics, and Dynamics, 243 - 286, Springer Verlag, New York.

[12] R. Lidl and H. Niederreiter (1997) *Finite Fields*. Encyclopedia of Mathematics and its Applications 20, 2nd ed.. Cambridge University Press, Cambridge.

[13] C. Soulé (2003) Graphic requirements for multistationarity. ComPlexUs 1: 123–133.

[14] A. Jarrah, et al. An optimal Control Problem For in vitro Virus Competition. *43rd IEEE* Conference on Decision and Control, Bahamas, 2004.

[15] Goles E. and Olivos J. (1981) Comportement Periodique Des Fonctions a Seuil Binaires et Applications. *Discrete Applied Mathematics,* 3: 93-105.

[16] David Murrugarra and Reinhard Laubenbacher. (2011) Regulatory Patterns in Molecular Interaction Networks. *Journal of Theoretical Biology.* 288, 66-72, 2011.

[17] Wassim Abou-Jaoudé, Djomangan A. Ouattara, and Marcelle Kaufman. From structure to dynamics: Frequency tuning in the p53-mdm2 network: I. logical approach. *Journal of Theoretical Biology*, 258(4):561 – 577, 2009.

[18] Richard Bonneau, Marc T. Facciotti, David J. Reiss, Amy K. Schmid, Min Pan, Amardeep Kaur, Vesteinn Thorsson, Paul Shannon, Michael H. Johnson, J. Christopher Bare, William Longabaugh, Madhavi Vuthoori, Kenia Whitehead, Aviv Madar, Lena Suzuki, Tetsuya Mori, Dong-Eun Chang, Jocelyne DiRuggiero, Carl H. Johnson, Leroy Hood, and Nitin S. Baliga. A predictive model for transcriptional control of physiology in a free living cell. *Cell*, 131(7):1354 – 1365, 2007.

[19] Jon T. Butler, Tsutomu Sasao, and Munehiro Matsuura. Average path length of binary decision diagrams. *IEEE Transactions on Computers*, 54(9):1041–1053, 2005.

[20] Claudine Chaouiya, Elisabeth Remy, and Denis Thieffry. Qualitative petri net modelling of genetic networks, 2006.

[21] Andrea Ciliberto, Bela Novak, and John J. Tyson. Mathematical model of the morphogenesis checkpoint in budding yeast. *The Journal of Cell Biology*, 163(6):1243–54, Dec 2003.

[22] Adrien Faure, Aurelien Naldi, Fabrice Lopez, Claudine Chaouiya, Andrea Ciliberto, and Denis Thieffry. Modular logical modelling of the budding yeast cell cycle. *Mol. BioSyst.*, 5:1787–1796, 2009.

[23] Adam M Feist, Christopher S Henry, Jennifer L Reed, Markus Krummenacker, Andrew R Joyce, Peter D Karp, Linda J Broadbelt, Vassily Hatzimanikatis, and Bernhard O Palsson. A genome-scale metabolic reconstruction for escherichia coli k-12 mg1655 that accounts for 1260 orfs and thermodynamic information. *Mol Syst Biol*, 3, 06 2007.

[24] I. Gat-Viks and R. Shamir. Chain functions and scoring functions in genetic networks. *Bioinformatics*, 19(suppl 1):i108–i117, 2003.

[25] Aitor Gonzalez, Claudine Chaouiya, and Denis Thieffry. Dynamical analysis of the regulatory network defining the dorsal-ventral boundary of the drosophila wing imaginal disc. *Genetics*, 174(3):1625–1634, 2006.

[26] John Grefenstette, Sohyoung Kim, and Stuart Kauffman. An analysis of the class of gene regulatory functions implied by a biochemical model. *Biosystems*, 84(2):81 – 90, 2006. Dynamical Modeling of Biological Regulatory Networks.

[27] Stephen E. Harris, Bruce K. Sawhill, Andrew Wuensche, and Stuart Kauffman. A model of transcriptional regulatory networks based on biases in the observed regulation rules. *Complex.*, 7:23–40, March 2002.

[28] Abdul Salam Jarrah, Blessilda Raposa, and Reinhard Laubenbacher. Nested canalyzing, unate cascade, and polynomial functions. *Physica D: Nonlinear Phenomena*, 233(2):167 – 174, 2007.

[29] S Kauffman. Metabolic stability and epigenesis in randomly constructed genetic nets. *Journal of Theoretical Biology*, 22(3):437–467, March 1969.

[30] Stuart Kauffman, Carsten Peterson, Björn Samuelsson, and Carl Troein. Random boolean network models and the yeast transcriptional network. *Proceedings of the National Academy of Sciences of the United States of America*, 100(25):14796–14799, 2003.

[31] Stuart Kauffman, Carsten Peterson, Björn Samuelsson, and Carl Troein. Genetic networks with canalyzing boolean rules are always stable. *Proceedings of the National Academy of Sciences of the United States of America*, 101(49):17102–17107, 2004.

[32] Robert D Leclerc. Survival of the sparsest: robust gene networks are parsimonious. *Mol Syst Biol*, 4, 08 2008.

[33] Fangting Li, Tao Long, Ying Lu, Qi Ouyang, and Chao Tang. The yeast cell-cycle network is robustly designed. *Proceedings of the National Academy of Sciences of the United States of America*, 101(14):4781–4786, 2004.

[34] Luis Mendoza. A network model for the control of the differentiation process in th cells. *Biosystems*, 84(2):101 – 114, 2006. Dynamical Modeling of Biological Regulatory Networks.

[35] Ron Milo, Shalev Itzkovitz, Nadav Kashtan, Reuven Levitt, Shai Shen-Orr, Inbal Ayzenshtat, Michal Sheffer, and Uri Alon. Superfamilies of evolved and designed networks. *Science*, 303(5663):1538–1542, 2004.

[36] Aurélien Naldi, Jorge Carneiro, Claudine Chaouiya, and Denis Thieffry. Diversity and plasticity of th cell types predicted from regulatory network modelling. *PLoS Comput Biol*, 6(9):e1000912, 09 2010.

[37] S. Nikolajewa, M. Friedel, and T. Wilhelm. Boolean networks with biologically relevant rules show ordered behavior. *Biosystems*, 90(1):40 – 47, 2007.

[38] Mark Pogson, Rod Smallwood, Eva Qwarnstrom, and Mike Holcombe. Formal agent-based modelling of intracellular chemical interactions. *Biosystems*, 85(1):37 – 45, 2006.

[39] Ethel Queralt, Chris Lehane, Bela Novak, and Frank Uhlmann. Downregulation of pp2acdc55 phosphatase by separase initiates mitotic exit in budding yeast. *Cell*, 125(4):719 – 732, 2006.

[40] L. Jason Steggles, Richard Banks, Oliver Shaw, and Anil Wipat. Qualitatively modelling and analysing genetic regulatory networks: a Petri net approach. *Bioinformatics*, 23(3):336–343, 2007.

[41] Z. Szallasi and S. Liang. Modelling the normal and neoplastic cell cycle with "realistic genetic networks": their application for understanding carcinogenesis and assessing therapeutic strategies. *Pac. Symp. Biocomput.*, pages 66–76., 1998.

[42] Denis Thieffry and René Thomas. Dynamical behaviour of biological regulatory networks-II. immunity control in bacteriophage lambda. *Bulletin of Mathematical Biology*, 57:277–297, 1995. 10.1007/BF02460619.

[43] René Thomas and Richard D'Ari. *Biological Feedback.* CRC Press, Boca Raton, Florida., first edition, 1989.

[44] Alan Veliz-Cuba, Abdul Salam Jarrah, and Reinhard Laubenbacher. Polynomial algebra of discrete models in systems biology. *Bioinformatics*, 26(13):1637–1643, 2010.

[45] C.H. Waddington. Canalization of development and the inheritance of acquired characters. *Nature*, 150:563–565, November 1942.

[46] Haiyuan Yu, Pascal Braun, Muhammed A. Yildirim, Irma Lemmens, Kavitha Venkatesan, Julie Sahalie, Tomoko Hirozane-Kishikawa, Fana Gebreab, Na Li, Nicolas Simonis, Tong Hao, Jean-Franois Rual, Amlie Dricot, Alexei Vazquez, Ryan R. Murray, Christophe Simon, Leah Tardivo, Stanley Tam, Nenad Svrzikapa, Changyu Fan, Anne-Sophie de Smet, Adriana Motyl, Michael E. Hudson, Juyong Park, Xiaofeng Xin, Michael E. Cusick, Troy Moore, Charlie Boone, Michael Snyder, Frederick P. Roth, Albert-Laszlo Barabasi, Jan Tavernier, David E. Hill, and Marc Vidal. High-quality binary protein interaction map of the yeast interactome network. *Science*, 322(5898):104–110, 2008.

[47] David Murrugarra and Reinhard Laubenbacher. (2012) The Number of Multistate Nested Canalyzing Functions. *Physica D: Nonlinear Phenomena,* 241, 921-938, 2012.

[48] J.T. Butler et al. (2005) Average path length of binary decision diagrams. *IEEE Trans. Comput.* **54**:1041–1053.

[49] Winfried Just, Ilya Shmulevich, John Konvalina. (2004) The number and probability of canalyzing functions, *Physica D: Nonlinear Phenomena* 197, 211–221.

[50] Inferring Biologically Relevant Models: Nested Canalyzing Functions. (2011) Franziska Hinkelmann and Abdul S. Jarrah. *under review*. Available on arXiv.org as arXiv:1011.6064.

[51] Stuart Kauffman, Carsten Peterson, B. Samuelsson, and Carl Troein. (2003) Random boolean network models and the yeast transcriptional network. *Proceedings of the National Academy of Sciences of the United States of America*, 100(25):14796–14799.

[52] Stuart Kauffman, Carsten Peterson, Björn Samuelsson, and Carl Troein. (2004) Genetic networks with canalyzing boolean rules are always stable. *Proceedings of the National Academy of Sciences of the United States of America*, 101(49):17102–17107.

[53] R. Lidl and H. Niederreiter. (1997). *Finite Fields*, Cambridge University Press, New York.

[54] R. Milo et al. (2004) Superfamilies of evolved and designed networks, *Science* **303**:1538–1542.

[55] S. Nikolajewa, et al. (2007) Boolean networks with biologically relevant rules show ordered behavior. *BioSystems* 90, 40–47.

[56] T. Sasao, et. al. (1979) On the Number of Fanout-Free Functions and Unate Cascade Functions. *IEEE trans. Comput.* 28 (1), 66–72.

[57] C.H. Waddington. (1942) Canalization of development and the inheritance of acquired characters. *Nature*, 150:563–565.

[58] Yuan Li, John O. Adeyeye, David Murrugarra, Boris Aguilar, Reinhard Laubenbacher. (2012) Boolean nested canalizing functions: a comprehensive analysis. *Under Review.*

[59] David Murrugarra, Alan Veliz-Cuba, Boris Aguilar, Seda Arat, Reinhard Laubenbacher. (2012) Modeling Stochasticity and Variability in Gene Regulatory Networks. *EURASIP Journal on Bioinformatics and Systems Biology,* 2012: 2012:5. doi:10.1186/1687-4153-2012-5.

[60] E Avigdor, M Elowitz, Functional roles for noise in genetic circuits. Nature **467**, 167–173 (2010)

[61] M Acar, J Mettetal, A van Oudenaarden, Stochastic switching as a survival strategy in fluctuating environments. Nat. Gen. **40**, 471–475 (2008)

[62] F St-Pierre, D Endy, Determination of cell fate selection during phage lambda infection. PNAS **105**, 20705–20710 (2008)

[63] N Geva-Zatorsky, N Rosenfeld, S Itzkovitz, R Milo, A Sigal, E Dekel, T Yarnitzky, Y Liron, P Polak, G Lahav, U Alon, Oscillations and variability in the p53 system. Mol. Syst. Biol. **2**, 2006.0033 (2006) doi: 10.1038/msb4100068

[64] D Irons, Logical analysis of the budding yeast cell cycle. J. Theor. Biol. **257**, 543–559 (2009)

[65] R Thomas, R D'Ari, *Biological Feedback* (CRC Press, Boca Raton, 1990)

[66] C Chaouiya, E Remy, B Mossé, D Thieffry, *Qualitative Analysis of Regulatory Graphs: A Computational Tool Based on a Discrete Formal Framework*. Lecture Notes in Control and Information Sciences, vol. 294, pp. 830–832 (2003)

[67] I Shmulevich, E Dougherty, S Kim, W Zhang, Probabilistic Boolean networks: a rule based uncertainty model for gene regulatory networks. Bioinformatics **18**(2), 261–274 (2002)

[68] I Shmulevich, and E Dougherty, Probabilistic Boolean Networks: The Modeling and Control of Gene Regulatory Networks. (SIAM, Philadelphia, 2010)

[69] S Teraguchi, Y Kumagai, A Vandenbon, S Akira,D Standley, Stochastic binary modeling of cells in continuous time as an alternative to biochemical reaction equations. Phys. Rev. E **84**(4), 062903 (2011)

[70] A Garg, K Mohanram, A Di Cara, G De Micheli, I Xenarios, Modeling stochasticity and robustness in gene regulatory networks. Bioinformatics **15;25**(12), i101-i109 (2010)

[71] AS Ribeiro, SA Kauffman, Noisy attractors and ergodic sets in models of gene regulatory networks. J. Theor. Biol. **247**, 743–755 (2007)

[72] D Gillespie, Exact stochastic simulation of coupled chemical reactions. J. Phys. Chem. **81**(25), 2340–2361 (1977)

[73] D Gillespie, Stochastic simulation of chemical kinetics. Annu. Rev. Phys. Chem. **58**, 35–55 (2007)

[74] D Bratsun, D Volfson, LS Tsimring, J Hasty, delay-induced stochastic oscillations gene regulation. PNAS **102**(41), 14593–14598 (2005)

[75] AS Ribeiro, Stochastic and delayed stochastic models of gene expression and regulation. Math. Biosci. **223**(1), 1–11 (2010)

[76] AS Ribeiro, R Zhu, SA Kauffman, A general modeling strategy for gene regulatory networks with stochastic dynamics. J. Comput. Biol. **13**(9), 1630-1639 (2006)

[77] T Toulouse, P Ao, I Shmulevich, S Kauffman, Noise in a small genetic circuit that undergoes bifurcation. Complexity **11**(1), 45–51 (2005)

[78] ER Álvarez-Buylla, A Chaos, M Aldana, M Benítez, Y Cortes-Poza, C Espinosa-Soto, DA Hartasánchez, RB Lotto, D Malkin, GJ Escalera Santos, P Padilla-Longoria, Floral morphogenesis: stochastic explorations of a gene network epigenetic landscape. *PLoS ONE* 3(11), e3626 (2008). doi:10.1371/journal.pone.0003626

[79] MI Davidich, S Bornholdt, Boolean network model predicts cell cycle sequence of fission yeast. PLoS ONE 3(2), e1672 (2008). doi:10.1371/journal.pone.0001672

[80] K Willadsen, J Wiles, Robustness and state-space structure of Boolean gene regulatory models. J. Theor. Biol. **249**(4), 749–765 (2008)

[81] R Layek, A Datta, R Pal, ER Dougherty, Adaptive intervention in probabilistic Boolean networks. Bioinformatics **25**(16), 2042–2048 (2009)

[82] SS Peter, BE Michael, DS Eric, Intrinsic and extrinsic contributions to stochasticity in gene expression. PNAS **99**(20), 12795–12800 (2002)

[83] F Hinkelmann, M Brandon, B Guang, R McNeill, G Blekherman, A Veliz-Cuba, R Laubenbacher, ADAM: analysis of the dynamics of algebraic models of biological systems using computer algebra. BMC Bioinf. **12**, 295 (2011)

[84] W Abou-Jaoudé, D Ouattara, M Kaufman, From structure to dynamics: frequency tuning in the p53-mdm2 network: I. logical approach. J. Theor. Biol. **258**(4), 561–577 (2009)

[85] E Batchelor, A Loewer, G Lahav, The ups and downs of p53: understanding protein dynamics in single cells. Nat. Rev. Cancer **9**, 371–377 (2009)

[86] M Ptashne, A Genetic Switch: Phage λ and Higher Organisms (Cell Press and Blackwell Scientific Publications, Cambridge, 1992)

[87] D Thieffry, R Thomas, Dynamical behaviour of biological regulatory networks–II. Immunity control in bacteriophage lambda. Bull. Math. Biol. **57**, 277–295 (1995)

[88] L Reichardt, D Kaiser, Control of λ repressor synthesis. PNAS **68**, 2185–2189 (1971)

[89] P Kourilsky, Lysogenization by bacteriophage lambda I. Multiple infection and the lysogenic response. Mol. Gen. Gen. **122**, 183–195 (1973)

[90] A Arkin, J Ross, H McAdams, Stochastic kinetic analysis of developmental pathway bifurcation in Phage λ-infected *Escherichia coli* cells. Genetics **149**, 1633–1648 (1998)

[91] Darren J. Wilkinson. (2006) Stochastic Modelling for Systems Biology *Chapman and Hall/CRC.*