Robustness Analysis of Gene Regulatory Networks

Claus T. Kadelka

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
T. M. Murali
Stanca Ciupe
Terry Herdman

March 30, 2015
Blacksburg, Virginia

Keywords: Discrete Modeling, Derrida Values, Nested Canalizing Functions, DNA Mismatch Repair, MicroRNAs, Functional Enrichment

Copyright 2015, Claus T. Kadelka
Robustness Analysis of Gene Regulatory Networks

Claus T. Kadelka

Cells generally manage to maintain stable phenotypes in the face of widely varying environmental conditions. This fact is particularly surprising since the key step of gene expression is fundamentally a stochastic process. Many hypotheses have been suggested to explain this robustness. First, the special topology of gene regulatory networks (GRNs) seems to be an important factor as they possess feedforward loops and certain other topological features much more frequently than expected. Second, genes often regulate each other in a canalizing fashion: there exists a dominance order amidst the regulators of a gene, which \textit{in silico} leads to very robust phenotypes. Lastly, an entirely novel gene regulatory mechanism, discovered and studied during the last two decades, which is believed to play an important role in cancer, is shedding some light on how canalization may in fact take place as part of a cell’s gene regulatory program. Short segments of single-stranded RNA, so-called microRNAs, which are embedded in several different types of feedforward loops, help smooth out noise and generate canalizing effects in gene regulation by overriding the effect of certain genes on others.

Boolean networks and their multi-state extensions have been successfully used to model GRNs for many years. In this dissertation, GRNs are represented in the time- and state-discrete framework of Stochastic Discrete Dynamical Systems (SDDS), which captures the cell-inherent stochasticity. Each gene has finitely many different concentration levels and its concentration at the next time step is determined by a gene-specific update rule that depends on the current concentration of the gene’s regulators. The update rules in published gene regulatory networks are often nested canalizing functions. In Chapter 2, this class of functions is introduced, generalized and analyzed with respect to its potential to confer robustness. Chapter 3 describes a simulation study, which supports the hypothesis that microRNA-mediated feedforward loops have a stabilizing effect on GRNs. Chapter 4 focuses on the cellular DNA mismatch repair machinery. A first regulatory network for this machinery is introduced, partly validated and analyzed with regard to the role of microRNAs and certain genes in conferring robustness to this particular network. Due to steady exposure to mutations, GRNs have evolved over time into their current form. In Chapter 5, a new framework for modeling the evolution of GRNs is developed and then used to identify topological features that seem to stabilize GRNs on an evolutionary time-scale. Chapter 6 addresses a completely separate project in Bioinformatics. A novel functional enrichment method is developed and compared to various popular methods.

Funding for this work was provided by NSF grant CMMI-0908201 and NSF grant 1062878.
To my parents.
Acknowledgments

I am thankful to many people for their contributions, encouragement and discussions. First, I would like to thank my advisor, Dr. Reinhard Laubenbacher, who has always been there as a guide and mentor throughout these last four years. Dr. Laubenbacher encouraged me to start working at the Virginia Bioinformatics Institute (VBI) and taught me the functioning of the academic system. Next, I would like to thank my committee members: Dr. T. M. Murali, whom I had the pleasure of working with on a very interesting side project in Bioinformatics, and Dr. Stanca Ciupe and Dr. Terry Herdman for their time and support.

I thank my colleagues in the Math Department and at VBI, especially the members of my research group, Madison Brandon, Seda Arat, Matt Oremland, David Murrugarra, Shernita Lee, Cory Brunson, Sook Ha, Ariane Hofmann, and Blake Treadway for making studying and researching an enjoyable and interesting experience. I would also like to thank senior members of my research group, Dr. Jennifer Galovich, Dr. Kathy O’Hara and Dr. Anael Verdugo for all their advice, editing and helpful discussions. Special thanks to Betsy Williams and Nicole Sutphin for taking care of all the bureaucratic hurdles that allowed me to focus on my studies, and to Dr. Peter Haskell for the Math Department’s financial support that allowed me to attend multiple conferences and to make many valuable professional connections.

Almost all the research presented in this dissertation is based on collaborations. I would like to thank all my collaborators for their countless hours of discussion, Dr. Chris Heinen, Dr. T. M. Murali, Dr. David Murrugarra, Dr. Yuan Li, Dr. John Adeyeye, Marschall Furman, Ross Donatelli and Madeleine Weinstein. Special thanks to my collaborators Madison Brandon and Seda Arat without whom researching would not have been half the fun.

I am thankful to all the friends I met in Blacksburg who made my time so enjoyable: Stephen, Jonas, Trevor, Bradley, Natasha, Madison, Matt, and many more. Thanks also to all my friends in Germany who still talk to me although I stayed much longer in the United States than I originally ever imagined: Jonas, Pascal, Julia, and all the others.

Big thanks to my parents for everything they provided for me, especially for supporting my decision to study mathematics and to go to Virginia Tech - two very important decisions, neither of which I ever regretted.

Finally, I wholeheartedly thank my wife, Christina, for all her love and support!
Attribution

Besides my advisor Reinhard Laubenbacher, PhD, who is currently the Director of the Center for Quantitative Medicine at the University of Connecticut Health Center, several colleagues aided in the writing and research behind several chapters of my dissertation. A brief description of their contributions is included here.

Chapter 2

Yuan Li, PhD, is currently an Associate Professor in the Department of Mathematics at Winston-Salem State University. Dr. Li drafted most proofs in Section 2.4 & Section 2.5.

John O. Adeyeye, PhD, is currently a Professor in the Department of Mathematics at Winston-Salem State University. Dr. Adeyeye participated in discussions about Section 2.4 & Section 2.5.

Chapter 3

David Murrugarra, PhD, is currently an Assistant Professor in the Department of Mathematics at the University of Kentucky. Dr. Murrugarra participated in initial discussions about the study presented in Chapter 3 and proof-read the manuscript.

Chapter 4

The research project described in Chapter 4 started as a summer Research for Undergraduates (REU) program. I mentored three undergraduates: Ross Donatelli, currently at Florida Institute of Technology, Marschall Furman, currently at North Carolina State University, and Madeleine Weinstein, currently at Harvey Mudd College. Under my guidance, the students conducted the large literature search, which served as a basis of the developed regulatory network. In addition, the undergraduate researchers wrote code for the robustness analysis of the network and performed initial computer experiments that I extended after the end of the summer program.
Chris Heinen, PhD, is currently an Associate Professor in the Department of Medicine at the University of Connecticut Health Center. Dr. Heinen’s knowledge in biology helped in the set-up of the study. Besides, Dr. Heinen mentors Seda Arat in conducting biological experiments in order to validate the developed regulatory network.

Seda Arat is a doctoral student in the Department of Mathematics at Virginia Tech and a predoctoral fellow at the Center for Quantitative Medicine at the University of Connecticut Health Center. Guided by Dr. Heinen, Seda is currently conducting biological experiments in order to validate the developed regulatory network.

Chapter 5

The research described in this chapter was supported by biweekly discussions with Seda Arat, Dr. Heinen and Dr. Laubenbacher.

Chapter 6

The research of this final chapter started as a class project in Computational Systems Biology, taught by T. M. Murali who is currently an Associate Professor in the Department of Computer Science at Virginia Tech. Dr. Murali guided the set-up of the study through many discussions.

Madison Brandon is currently a doctoral student in Computational Biology at the University of Connecticut Health Center. Through countless discussions Madison Brandon and I developed the key ideas of this chapter together. I wrote almost all the code, created the web application for online use of our method, and was the primary author of Subsections 6.2.2, 6.2.3 & 6.3.1 and Section 6.4. Subsection 6.3.2 is mainly the work of Madison Brandon. I included it for conclusiveness.
# Contents

<table>
<thead>
<tr>
<th>List of Figures</th>
<th>xi</th>
</tr>
</thead>
<tbody>
<tr>
<td>List of Tables</td>
<td>xiii</td>
</tr>
</tbody>
</table>

## 1 Introduction

1.1 Gene Regulation .................................................. 1

1.2 Modeling Gene Regulatory Networks ................................. 4

1.2.1 Three Different Modeling Frameworks ......................... 4

1.2.2 Stochastic Discrete Dynamical Systems (SDDS) ............... 5

1.3 Measuring Robustness ............................................... 6

1.3.1 Derrida Values .................................................. 7

1.3.2 Average Maximal Transition Probability ...................... 8

1.4 Overview of Main Results .......................................... 10

## 2 Nested Canalizing Functions and the Robustness of Their Networks 11

2.1 Introduction ......................................................... 12

2.2 Definitions and Notation ........................................... 13

2.3 Derrida Values of Networks Governed by NCFs .................. 14

2.4 Characterization of Nested Canalizing Functions ............... 23

2.5 Generalization of NCFs ............................................. 28
## 5 Evolutionary Robustness of Gene Regulatory Networks

### 5.1 Introduction

### 5.2 Methods

#### 5.2.1 Modeling Framework

#### 5.2.2 Modeling Mutations

#### 5.2.3 Refinements of the Model

#### 5.2.4 Selective Advantage Constant

#### 5.2.5 Quantifying Evolutionary Robustness

### 5.3 Results

### 5.4 Discussion

## 6 Concise Functional Enrichment of Ranked Gene Lists (CRFE)

### 6.1 Introduction

### 6.2 Methods

#### 6.2.1 Generative Model

#### 6.2.2 Algorithm

#### 6.2.3 Quality measures

#### 6.2.4 Data sets

### 6.3 Results

#### 6.3.1 Comparison of CRFE to other Functional Enrichment Algorithms

#### 6.3.2 Interpretation of the Enriched Set of Processes

### 6.4 Discussion

### 6.5 Conclusion

## Bibliography
## List of Figures

<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.1</td>
<td>Wiring diagram of the gene regulatory network of Lambda phage infection</td>
<td>2</td>
</tr>
<tr>
<td>1.2</td>
<td>State space of the SDDS in Example 1.2.2</td>
<td>7</td>
</tr>
<tr>
<td>2.1</td>
<td>Derrida plot for networks of $n = 100$ genes governed by NCFs with $k = 5$</td>
<td>18</td>
</tr>
<tr>
<td>2.2</td>
<td>Derrida plot for networks of $n = 100$ genes governed by NCFs with $k = 8$</td>
<td>18</td>
</tr>
<tr>
<td>3.1</td>
<td>The effect of microRNAs on the phenotypical robustness</td>
<td>37</td>
</tr>
<tr>
<td>3.2</td>
<td>Correlation of the number of attractors in the basic and extended network</td>
<td>41</td>
</tr>
<tr>
<td>4.1</td>
<td>Wiring Diagram of the DNA Mismatch Repair regulatory network</td>
<td>49</td>
</tr>
<tr>
<td>4.2</td>
<td>Steady states of the DNA Mismatch Repair model for all external conditions</td>
<td>50</td>
</tr>
<tr>
<td>4.3</td>
<td>Model predictions: Effect of an individual node on the network phenotype by splitting up the state space</td>
<td>52</td>
</tr>
<tr>
<td>4.4</td>
<td>Model predictions: Effect of an individual node on the network phenotype by knocking out and over-expressing the node</td>
<td>53</td>
</tr>
<tr>
<td>4.5</td>
<td>AMTP values for different variants of the DNA Mismatch Repair network</td>
<td>54</td>
</tr>
<tr>
<td>4.6</td>
<td>Derrida plot for different initial perturbations and different variants of the DNA Mismatch Repair network</td>
<td>55</td>
</tr>
<tr>
<td>4.7</td>
<td>Derrida plot for multiple time steps and different variants of the DNA Mismatch Repair network</td>
<td>56</td>
</tr>
</tbody>
</table>
List of Tables

<table>
<thead>
<tr>
<th>Table</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.1</td>
<td>Hamming distance of all possible pairs of state space configurations</td>
<td>8</td>
</tr>
<tr>
<td></td>
<td>in the example system</td>
<td></td>
</tr>
<tr>
<td>3.1</td>
<td>Average number of attractors in the SDDS framework for different</td>
<td>33</td>
</tr>
<tr>
<td></td>
<td>network sizes</td>
<td></td>
</tr>
<tr>
<td>3.2</td>
<td>Example of how an additional microRNA is embedded as an input variable</td>
<td>35</td>
</tr>
<tr>
<td></td>
<td>in a target gene’s update function</td>
<td></td>
</tr>
<tr>
<td>3.3</td>
<td>The effect of microRNAs on the phenotypical robustness in a 50-gene</td>
<td>38</td>
</tr>
<tr>
<td></td>
<td>network</td>
<td></td>
</tr>
<tr>
<td>3.4</td>
<td>Truth table that shows a case where a network can be theoretically</td>
<td>39</td>
</tr>
<tr>
<td></td>
<td>fully stabilized by the introduction of enough microRNAs</td>
<td></td>
</tr>
<tr>
<td>3.5</td>
<td>Derrida values for a basic 50-gene network and for the same network</td>
<td>40</td>
</tr>
<tr>
<td></td>
<td>with 30 additional microRNAs</td>
<td></td>
</tr>
<tr>
<td>4.1</td>
<td>Default update rule in the DNA Mismatch Repair network</td>
<td>45</td>
</tr>
<tr>
<td>4.2</td>
<td>AMTP value for various one-node knockout or over-expression variants</td>
<td>57</td>
</tr>
<tr>
<td></td>
<td>of the DNA Mismatch Repair network</td>
<td></td>
</tr>
<tr>
<td>5.1</td>
<td>Truth table of the three possible update functions for connectivity</td>
<td>63</td>
</tr>
<tr>
<td></td>
<td>k = 2</td>
<td></td>
</tr>
<tr>
<td>6.1</td>
<td>List of the top 15 returned processes</td>
<td>88</td>
</tr>
</tbody>
</table>
Chapter 1

Introduction

1.1 Gene Regulation

A gene is a hereditary DNA segment at a specific chromosomal location, which codes for an RNA or protein sequence that has a biological function. Every human cell contains more than 20,000 genes, and the process of converting the information that is stored in a gene sequence into RNA via transcription and then into protein via translation is called gene expression. Any living organism uses this process to generate vital macromolecules. In 1958, Francis Crick enunciated: "Once information has got into protein, it can’t get out again" [33], which became known as the central dogma of molecular biology. It means that there exists only a one-directional flow of information as DNA can be transcribed into RNA, and RNA can be translated into protein, but transfer of information from protein to RNA or DNA is impossible.

Information from a gene can however only be transferred if that gene is expressed. To properly function, a cell therefore possesses many different mechanisms that regulate different steps of the expression of its genes. The collection of all these methods is termed gene regulation. The most important regulation happens at the level of transcription initiation. Certain proteins, so-called transcription factors, contain one or more DNA-binding domains, which allow them to attach to specific DNA sequences. By binding to a region upstream of the coding region of a gene (the promoter region), transcription factors can promote or inhibit the recruitment of RNA polymerase, the enzyme that transcribes DNA into mRNA (messenger RNA). Transcription factors can function alone or in complexes leading to a wide range of combinatorial control mechanisms.

Another important regulation of gene expression occurs after the transcription. Short segments of single-stranded RNA molecules, called microRNAs, which have been discovered only within the last two decades, modulate the amount of protein that is produced. MicroRNAs are only about 22 nucleotides long and have been found in animals, plants and even
some viruses [6, 13, 84]. They act as post-transcriptional regulators by employing different ways to silence mRNA molecules [131]. Mammalian microRNAs are fairly unspecific; they can often regulate multiple hundred different mRNAs [45]. By attaching to the three prime untranslated region (3′-UTR) of a mRNA molecule, mammalian microRNAs can prevent their target from being translated by the ribosome [14]. Plant microRNAs, on the other hand, are very specific; they are almost perfectly complementary to a part of the coding region of their target mRNAs, which allows them to cleave the mRNA molecule into two pieces. MicroRNAs are involved in a diverse set of regulatory pathways, e.g. in apoptosis in human glioblastoma cells [23], tissue development in C. elegans [68], hepatitis C infection [70], stem cell division [25], skeletal muscle proliferation and differentiation [25], or mammalian myoblast differentiation [106]. MicroRNAs are produced from introns or from specific microRNA genes, and just like protein-encoding genes, these microRNA-encoding genes themselves are subject to transcriptional control [90].

All these different regulators and interactions form a complex network, the gene regulatory network (GRN). A node in this network represents a gene and its direct product (protein or microRNA), while an edge from node A to node B means that the gene product of gene A regulates gene B, or (in short) gene A regulates gene B. There are positive regulations (e.g., a transcription factor that promotes RNA polymerase recruitment) and negative regulations (e.g., a microRNA that prevents mRNA from being translated into protein, or a transcription factor that inhibits RNA polymerase recruitment). One of the best understood regulatory systems is the infection of E. coli with the Lambda phage virus [7, 128]. Figure 1.1 shows the four main genes CI, CII, CRO and N and their regulations as an example of a wiring diagram (or dependency graph) of a GRN.

At any given instant, each gene in a GRN is expressed to a certain extent, and without external perturbations, the concentration levels usually start to follow a regular pattern after some time. Different patterns can be directly linked to different cellular phenotypes. In Lambda phage, CI and CRO inhibit each other so that, at any time, only one of the
two can be significantly expressed. When the virus first enters a host cell, all four genes are unexpressed but the expression pattern starts to change quickly, and two outcomes are possible [7, 128]. On the one hand, CRO may become highly expressed, which implies that CI is not expressed. This pattern is observed whenever the virus multiplies rapidly, eventually leading to the death (lysis) of the cell and the release of many new viruses. If, on the other hand, CI becomes highly expressed (through activation by CII), the viral DNA is integrated into the bacterial chromosome, and the cell and the virus coexist (lysogeny). One question that naturally arises is how some cells can behave lysogenic while others behave lytic, or in more general terms, how can phenotypic differences exist among genetically identical cells [7].

Around the turn of the millennium, it has been shown that gene expression is fundamentally a stochastic process [12, 100]; even two genetically identical cells produce different amounts of a certain protein, which can be attributed to intrinsic and extrinsic noise. Transcription depends on many microscopic biochemical reactions and the order and timing of these reactions introduces a cell-inherent stochasticity, or intrinsic noise [127]. Extrinsic noise is caused by cell-to-cell differences in the specific concentration, state or location of regulatory molecules [42].

These sources of stochasticity may cause a cell to switch between different phenotypes. However, cells generally manage to maintain stable phenotypes even in the face of widely varying environmental conditions. A lysogenic cell after Lambda phage infection, for instance, hardly ever starts to behave lytic. Given that gene expression is a stochastic process, this robust behavior is very surprising. Several hypotheses have been suggested as a possible explanation of the observed robustness. First, it has been discovered that the wiring diagrams of complex networks ranging from the transcriptional networks in yeast and E. coli to engineered systems seem to be enriched for feedforward loops and certain other topological features, which may lead to a stabilization of the network phenotype [99]. Second, many genes seem to be regulated in a so-called canalizing fashion, which means that a dominance order exists among the regulators of a gene. This special kind of regulatory logic seems to cause particularly robust dynamics. Lastly, microRNAs seem to play a large role in conferring robustness to gene regulatory programs. They help smooth out noise and generate canalizing effects in gene regulation by overriding the effects of certain genes on others [97].

Using mathematical modeling techniques, this thesis tests different mechanisms that potentially confer robustness to gene regulation. GRNs can be modeled in several ways and the next section describes three of them.
1.2 Modeling Gene Regulatory Networks

1.2.1 Three Different Modeling Frameworks

GRNs are comprised of genes and their interactions among each other. Mathematical models aim to capture the key dynamics of these networks, and can be used to generate various kinds of predictions, which can then be tested in experiments. GRNs can be successfully modeled in several ways.

1. The reaction kinetics can be described as a system of - possibly stochastic - coupled ordinary differential equations. The concentration level of each gene as well as the time in these models is continuous. If there are $n$ genes and $x_i(t), i = 1, 2, \ldots, n$ represents the expression level of gene $i$ at time $t$, then the temporal development of a GRN can be modeled by

$$\frac{dx_i}{dt} = f_i(x_1, x_2, \ldots, x_n),$$

where the functions $f_i$ are often low-order polynomials or Hill functions, derived for instance from Michaelis-Menten enzymatic kinetics.

2. GRNs can be viewed as biochemical reaction networks, and since the number of regulatory molecules in a cell is generally low, the Gillespie algorithm can be used to separately simulate each individual biochemical reaction between two molecules. The trajectories of a Gillespie simulation constitute exact samples from the solution function of the chemical master equation [48, 49].

The Gillespie algorithm starts with a certain number of each molecule, and only one reaction may happen at a time. After each reaction, the next reaction as well as the time that passes until the next reaction is randomly chosen, based on the number of molecules and the kinetic parameters.

3. The key reactions can be modeled as a discrete dynamical system. The concentration level of each gene as well as the time steps in these models are discrete, meaning that there exists only a finite number of different expression levels for each gene. In the simplest (Boolean) case, each gene can only be unexpressed or expressed, corresponding to 0 or 1. If there are $n$ genes and $x_i(t) \in X_i, \left(|X_i| < \infty, i = 1, 2, \ldots, n\right)$ represents the expression level of gene $i$ at time $t$, then the temporal development of a GRN can be modeled by

$$x_i(t + 1) = f_i(x_1(t), x_2(t), \ldots, x_n(t)),$$

where $f_i : X_1 \times X_2 \times \cdots \times X_n \rightarrow X_i$ is called the update function of gene $i$. In classical dynamical systems, all genes are updated at the same time, which implies that the expression level of a gene at the next time step only depends on the current expression level of its regulators.

The so-called state space of a discrete dynamical system consists of all possible configurations of concentration levels. Because of the finite size of the state space any
trajectory will eventually fall into a cycle, which are called attractors of the system. Cycles of size one are also referred to as steady states and any larger cycle is called a limit cycle. The set of all state space configurations that lead to an attractor is called the basin of this attractor or basin of attraction. Already in 1969, Stuart Kauffman introduced the idea that different attractors may correspond to different cellular phenotypes \[76, 78\]. Since then, various research projects have created strong support for this hypothesis, e.g., by showing that gene expression patterns of real organisms can be identified in carefully assembled GRN models \[4, 17, 60\]. A popular dynamical model of Lambda phage infection with the same wiring diagram as in Figure 1.1 has for instance two attractors: one with high expression of CI and low expression of CRO, CII, and N, which corresponds to the immune, lysogenic state; and another with high levels of CRO and low levels of the three other genes, which corresponds to the lytic state \[128\].

All three modeling techniques have certain advantages and disadvantages. The Gillespie algorithm is computationally very expensive and therefore not feasible for larger systems although there are many modifications and simplifications that address its complexity issues. Most systems of coupled ordinary differential equations are too complicated to be solved analytically, therefore requiring numerical approximations. Moreover, differential equation models, as well as Gillespie simulations, usually depend on many kinetic parameters, which need to be properly estimated in order to accurately model biological phenomena. If enough data exists to estimate the parameters, these types of models are able to generate quantitative predictions. However, biological data is frequently particularly sparse, which limits the usefulness of highly parametrical models. Discrete dynamical systems on the other hand do not require kinetic parameters; they are solely based on the qualitative relationships among genes. Because transcriptional data is sparse, these models provide a good framework for gene regulatory networks. In general, discrete models are frequently used to reveal qualitative information in situations where quantitative information is unavailable or not necessary \[4, 115, 21\].

1.2.2 Stochastic Discrete Dynamical Systems (SDDS)

This subsection introduces the framework that is used to model GRNs throughout this document. As stated in Section 1.1, gene regulation is an inherently stochastic process. A probabilistic extension of discrete dynamical systems (see Section 1.2.1.3), called stochastic discrete dynamical systems (SDDS), captures this stochasticity, and it has been shown that this framework is an appropriate set-up to model the effect of intrinsic noise on the dynamics of GRNs \[104\].

Definition 1.2.1. A stochastic discrete dynamical system (SDDS) in the variables \(x_1, \ldots, x_n\), which in this thesis represent genes, is defined as a collection of \(n\) triplets

\[
F = (f_i, p_i^+, p_i^-)_{i=1}^n,
\]

where
The stochasticity originates from the propensity parameters $p_i^\uparrow$ and $p_i^\downarrow$, which should be interpreted as follows:

- If $x_i$ is to be activated, this activation only happens with probability $p_i^\uparrow$, i.e., if $x_i(t) = 0$ and $f_i(x_1(t), \ldots, x_n(t)) = 1$, then $x_i(t + 1) = 1$ with probability $p_i^\uparrow$.
- If $x_i$ is to be degraded, this degradation only happens with probability $p_i^\downarrow$, i.e., if $x_i(t) = 1$ and $f_i(x_1(t), \ldots, x_n(t)) = 0$, then $x_i(t + 1) = 0$ with probability $p_i^\downarrow$.

All variables are synchronously updated and one time step can be interpreted as the average time needed for transcription and translation of the fastest of the genes considered. The propensity parameters for this fastest gene will be set to one, and the propensity parameters of genes with slower transcription and translation take proportionately lower values. Thus, this framework can be interpreted as introducing a very general stochastic sequential update scheme, which also allows for a variable to not be updated at a given step, a generalization of the usual approach \[3\]. Even if the propensity parameters cannot be accurately estimated, setting them all to the same value, e.g. to 0.5, still allows the framework to capture the effect of intrinsic noise. The SDDS framework therefore allows modeling different reaction speeds, while preserving the simplicity of a Boolean network model.

**Example 1.2.2.** Consider a GRN of $n = 2$ genes, $x_1$ and $x_2$, that activate each other. Let $f_1(x_1, x_2) = x_2$ and $f_2(x_1, x_2) = x_1$ be the two update functions, and let $p_1^\uparrow = 1$, $p_1^\downarrow = 0.4$, $p_2^\uparrow = 0.8$, $p_2^\downarrow = 0.7$ be the gene-specific propensity parameters.

**Figure 1.2A** shows the state space of the deterministic discrete dynamical system (without propensity parameters) and **Figure 1.2B** shows the state space of the SDDS. The deterministic system possesses three attractors, two steady states and one limit cycle. The SDDS shares the same steady states but the limit cycle is not preserved since the system eventually transitions to either steady state when starting from any initial configuration.

The next section describes ways to measure the robustness of a GRN that is modeled using the SDDS framework.

### 1.3 Measuring Robustness

This thesis aims to shed some light on the mechanisms that confer robustness to GRNs. Robustness can refer to many different phenomena. This section introduces two ways to...
Figure 1.2: (A) Deterministic state space of the example system in Example 1.2.2. (B) Stochastic state space of the same example system.

quantify the robustness of a GRN: The Derrida value (Section 1.3.1) describes how a small network perturbation develops on average over time [37]. This measure addresses the dynamical network robustness. The Average Maximal Transition Probability (Section 1.3.2) is a measure that focuses on the stability of network attractors [72]. Since the attractors of a GRN correspond to cellular phenotypes, this measure describes the phenotypical robustness of a network. Both measures are used throughout the thesis. In Section 5.2.5 an additional measure for the evolutionary robustness of a GRN is introduced.

1.3.1 Derrida Values

The Derrida plot is a common technique to evaluate the dynamical robustness of a Boolean discrete dynamical system [37]. It describes how a perturbation of a given size propagates on average over time. If small perturbations mostly vanish over time, the system is considered to be in the ordered regime. An ordered network generally possesses many steady states and short limit cycles. If small perturbations amplify over time, the system is considered to be in the chaotic regime. A chaotic network possesses long limit cycles. Lastly, if the perturbation remains of similar size, the system is in the critical regime. Many biological systems seem to work in this critical regime; they need to be robust enough to withstand perturbations caused by environmental changes, but also flexible enough to allow adaptation [12, 108].

Formally, the concept of Derrida values can be defined as follows

**Definition 1.3.1.** Let $F$ be a SDDS. Then for an initial perturbation of size $m$, the Derrida
Table 1.1: This table shows the Hamming distance, \( d(F(x), F(y)) \), of all possible pairs of state space configurations, \( x \) and \( y \), after one update in the (A) deterministic / (B) stochastic system from Example 1.2.2. The different shades of blue represent different initial Hamming distances, \( d(x, y) \).

The value, \( D(m) \), is defined as the average size of the perturbation after one update,

\[
D(m) = \mathbb{E}\left[ d(F(x), F(y)) \mid d(x, y) = m \right],
\]

where \( d : \{0, 1\}^n \times \{0, 1\}^n \to \{0, 1, \ldots, n\} \) is the Hamming distance (the standard \( \ell^1 \) metric).

Example 1.3.2. Table 1.1 shows the Hamming distance of all possible pairs of state space configurations for the system from Example 1.2.2. The Derrida values are simply the average values of all equally shaded cells. Thus for the deterministic version, the Derrida values are

\[
\begin{align*}
D(0) &= \frac{1}{4}(0 + 0 + 0 + 0) = 0, \\
D(1) &= \frac{1}{4}(1 + 1 + 1 + 1) = 1, \\
D(2) &= \frac{1}{2}(2 + 2) = 2.
\end{align*}
\]

For the stochastic version, the Derrida values are

\[
\begin{align*}
D(0) &= \frac{1}{4}(0 + 0.42 + 0.5792 + 0) = 0.2498, \\
D(1) &= \frac{1}{4}(1.3 + 1.4 + 0.7 + 0.6) = 1, \\
D(2) &= \frac{1}{2}(1.02 + 2) = 1.51.
\end{align*}
\]

1.3.2 Average Maximal Transition Probability

In a deterministic discrete dynamical system, each initial state space configuration lies in exactly one basin of attraction. This changes when stochasticity is introduced. Now, from one initial configuration, different attractors may be reached. In the context of GRNs, the
different attractors may be interpreted as different cellular phenotypes. To quantify the pheno- 
typical robustness of a system, every initial configuration and its transition probabilities 
to the different attractors is regarded.

If each initial configuration only transitions to one attractor, the dynamics are phenotypically 
deterministic, whereas lower maximal transition probabilities to attractors lead to propor- 
tionately less phenotypically robust dynamics. The phenotypical robustness of an SDDS can 
be defined as the average maximal transition probability to its attractors \[72\].

**Definition 1.3.3.** Let \( F = (f_i, p^+_i, p^-_i)_{i=1}^n \) be an SDDS. The average maximal transition probability (AMTP) to its attractors is

\[
\mu(F) = \frac{1}{2^n} \sum_{x \in \{0,1\}^n} \left( \max_{A \in \mathcal{A}(F)} \mathbb{P}(x \xrightarrow{F} \cdots \xrightarrow{F} A) \right) \in [0, 1],
\]

(1.3.1)

where \( \mathcal{A}(F) \) is the set of all attractors of \( F \) and all \( 2^n \) state space configurations are con- 
sidered and weighted equally. \( \mathbb{P}(x \xrightarrow{F} \cdots \xrightarrow{F} A) \) describes the probability that the initial 
configuration \( x \) eventually transitions to the attractor \( A \).

**Remark 1.3.4.** When \( F \) is a deterministic system, \( \mu(F) \equiv 1 \) always holds true. In com- 
parison, for a stochastic system with \( |\mathcal{A}(F)| = a \) attractors, values as low as \( 1/a \) may be 
obtained; for stochastic systems with a single attractor, \( \mu(F) \equiv 1 \) because the single attractor 
is eventually approached from any initial configuration.

**Example 1.3.5.** The information depicted in the stochastic example state space (see Fig- 
ure 1.2B) can also be represented in a Markovian transition matrix

\[
T = \begin{pmatrix}
00 & 01 & 10 & 11 \\
00 & 1 & 0 & 0 & 0 \\
01 & 0 & 0 & 0.7 & 0.3 \\
10 & 0.08 & 0.32 & 0.12 & 0.48 \\
11 & 0 & 0 & 0 & 1
\end{pmatrix}
\]

Then,

\[
T^\infty = \lim_{n \to \infty} T^n = \begin{pmatrix}
00 & 01 & 10 & 11 \\
00 & 1 & 0 & 0 & 0 \\
01 & 0.09 & 0 & 0 & 0.91 \\
10 & 0.12 & 0 & 0 & 0.88 \\
11 & 0 & 0 & 0 & 1
\end{pmatrix}
\]

shows the final transition probabilities to the network attractors. The AMTP value of the 
system is

\[
\mu(F) = \frac{1}{4} (1 + 0.91 + 0.88 + 1) = 0.9475.
\]
1.4 Overview of Main Results

This thesis provides computational support for several hypotheses that have been proposed as an explanation of the observed robustness in gene regulation.

The concept of canalization in gene regulation has already been introduced in the 1940s as a possible explanation of the stability in embryonal development [134]. The majority of published GRNs are governed by so-called nested canalizing functions (NCFs) and these networks have mostly stable dynamics [54, 79, 102]. In Chapter 2, NCFs are introduced, generalized and analyzed with respect to their ability to confer robustness. In particular, we present a way to directly calculate the Derrida values of networks governed by NCFs. This calculation was previously based on extensive Monte Carlo simulations.

It has been hypothesized that microRNAs which operate via feedforward loops confer robustness to GRNs [97]. Chapter 3 describes a computational study, which explores the role that such network motifs play in stabilizing dynamics. The results show that certain types of feedforward loops do indeed buffer GRNs against stochastic effects. Moreover, this study constitutes the first use of the AMTP-values as a measure of phenotypical robustness.

The study in Chapter 3 is based on thousands of computer generated GRNs. Contrarily, Chapter 4 focuses on a real biological network and the role of microRNAs in this particular system. We introduce a first regulatory network of the cellular DNA mismatch repair machinery, describe model predictions that can be tested in wet-lab experiments, and analyze the system with regard to the role of microRNAs and certain other genes in conferring robustness.

Due to steady exposure to mutations, GRNs have evolved over time into their current form. In Chapter 5, a new framework for modeling the evolution of GRNs is developed and then used to identify topological features that may stabilize GRNs on an evolutionary time-scale.

Chapter 6 addresses a completely separate project in Bioinformatics. Given a set of differentially expressed genes as input, functional enrichment methods compute a set of functional categories that annotate a surprisingly large number of the genes. We introduce a novel functional enrichment method and present results that suggest its superiority over several popular methods.
Chapter 2

Nested Canalizing Functions and the Robustness of Their Networks

To be submitted for publication as: C Kadelka, JO Adeyeye, Y Li, R Laubenbacher. Nested Canalizing Functions and Their Networks.

Contributions: I conducted all the research for the results reported in Section 2.3. Most proofs in Section 2.4 & Section 2.5 were drafted by Yuan Li and I completed them.

Abstract

Nested canalizing functions (NCFs) constitute a natural extension of the concept of canalization, which is often used to explain the robustness in biological systems. Most regulatory interactions in a GRN can be described by NCFs, and networks governed by these functions exhibit surprisingly stable dynamics. Derrida values are a popular measure for the robustness of a discrete dynamical system. This chapter introduces closed formulas for the calculation of Derrida values of a network governed by NCFs, which previously required extensive simulations. Recently, the concept of NCFs has been generalized to include multistate functions, and a recursive formula for the number of NCFs has been derived. This chapter contains a detailed analysis of the class of NCFs over an arbitrary finite field. We derive a unique polynomial representation of NCFs as well as explicit formulas for the number of NCFs and the number of equivalence classes under permutation of variables. The latter is motivated by the fact that two NCFs that differ only by a permutation of the variables share many important properties. Lastly, we generalize the concept of nested canalization even further.
2.1 Introduction

Stuart Kauffman introduced canalizing functions as appropriate rules in Boolean network models of gene regulatory networks [78]. A canalizing function possesses at least one input variable, such that if this input is at certain values, then the output value is already determined, no matter the values of the remaining input variables. If there exists an importance order among all inputs of a canalizing function and if any subfunction of $x$ least important inputs is canalizing, the resulting function is a nested canalizing function (NCF). NCFs are reminiscent of the concept of canalization in gene regulation Waddington introduced already in the 1940s [134]. The number of Boolean canalizing functions and the number Boolean NCFs are known [66, 71]; the probability that a random Boolean function is canalizing decreases rapidly as the number of inputs increases (less than 0.5 for three inputs, and less than 0.01 for five inputs). Nested canalization is more restrictive and the probabilities are even lower (0.25 for three inputs, and less than $10^{-5}$ for five inputs). Interestingly, an analysis of published Boolean models of molecular regulatory networks revealed that all 139 investigated rules with at least three inputs are canalizing and that 133 are even nested canalizing [54, 79].

Networks governed by NCFs also exhibit surprisingly stable dynamics [102]. These findings clearly motivate the study of NCFs in the context of understanding the regulatory logic of gene networks.

The class of Boolean NCFs is identical to the class of unate cascade functions [66], which corresponds exactly to the class of Boolean functions with corresponding binary decision diagrams of shortest average path length [20]. Unate cascade functions are frequently used in engineering and computer science, and a detailed mathematical study of NCFs can therefore benefit these different fields as well.

Canalizing and nested canalizing functions have been generalized to the multistate case so that the range of the input values may be any finite set, and formulae for the number of such functions have been derived [94, 102, 103]. Networks governed by multistate NCFs have similar stability properties as in the Boolean case: small numbers of attractors, large attractor basins and short limit cycles [102].

This chapter is ordered as follows. In Section 2.2 we introduce the concept of (nested) canalization. In Section 2.3 we present a way to calculate the values of the Derrida plot for networks governed by NCFs. This plot is a popular tool to evaluate the robustness of molecular interaction networks and its calculation previously required extensive simulations [37]. A comprehensive analysis of Boolean NCFs was given in [93]. In the second part of this chapter, we conduct a similar analysis for the multistate case. In Section 2.4 we derive a unique polynomial representation of NCFs, as well as explicit formulas for the number of NCFs and the number of equivalence classes of NCFs for a given number of variables. Lastly, in Section 2.5 we describe an even broader generalization of the concept of NCFs. We conclude with a brief discussion of the results in Section 2.6.
2.2 Definitions and Notation

In this section we review some concepts and definitions from [102, 103] to introduce the computational concept of canalization. Let \( \mathbb{F} = \mathbb{F}_p \) be a finite field with \( p \) elements, where \( p \) is prime. Note that in earlier work \( \mathbb{F} \) only had to be a finite set. The more stringent requirement on \( \mathbb{F} \) allows the use of a wider range of mathematical tools, while all previously discovered results remain valid. In particular, network models can still be represented as polynomial dynamical systems [132].

**Definition 2.2.1.** A function \( f(x_1, x_2, \ldots, x_n) \) is essential in the variable \( x_i \) if there exist \( r, s \in \mathbb{F} \) and \( (x_1, \ldots, x_{i-1}, x_{i+1}, \ldots, x_n) \in \mathbb{F}^{n-1} \) such that
\[
f(x_1, \ldots, x_{i-1}, r, x_{i+1}, \ldots, x_n) \neq f(x_1, \ldots, x_{i-1}, s, x_{i+1}, \ldots, x_n).
\]

**Definition 2.2.2.** [93] Given \( a, b \in \mathbb{F} \) and \( i \in \{1, \ldots, n\} \), a function \( f(x_1, x_2, \ldots, x_n) \) is \( < i : a : b > \) canalizing if for all \( (x_1, \ldots, x_{i-1}, x_{i+1}, \ldots, x_n) \in \mathbb{F}^{n-1} \)
\[
f(x_1, \ldots, x_{i-1}, a, x_{i+1}, \ldots, x_n) = b.
\]
We call \( x_i \) the canalizing variable of \( f \). The set \( S \) of all \( a \in \mathbb{F} \) such that \( f \) is \( < i : a : b > \) canalizing for some \( b \) is called the canalizing input set with respect to \( x_i \), and \( b \) is called the canalized output with respect to \( x_i \) and \( a \).

We now assume that \( \mathbb{F} = \{0, 1, \ldots, p-1\} \) is ordered, in the natural order \( 0 < 1 < \cdots < p-1 \). A proper subset \( S \) of \( \mathbb{F} \) is called an interval if and only if \( S = \{0, \ldots, j\} \) or \( S^c = \mathbb{F} - S = \{0, \ldots, j\} \) for some \( 0 \leq j < p - 1 \). Hence, a proper subset \( S \) is an interval if and only if \( S^c \) is an interval.

**Definition 2.2.3.** [103] Let \( f : \mathbb{F}^n \to \mathbb{F} \) be a function in \( n \) variables and let \( S_i \) be intervals of \( \mathbb{F} \), \( i = 1, \ldots, n \). Let \( \sigma \) be a permutation of the set \( \{1, 2, \ldots, n\} \). Then \( f \) is a nested canalizing function (NCF) in the variable order \( x_{\sigma(1)}, \ldots, x_{\sigma(n)} \) with canalizing input sets \( S_1, \ldots, S_n \) and canalized output values \( b_1, \ldots, b_n, b_{n+1} \) with \( b_n \neq b_{n+1} \), if it can be represented in the form
\[
f(x_1, \ldots, x_n) = \begin{cases} 
  b_1 & x_{\sigma(1)} \in S_1, \\
  b_2 & x_{\sigma(1)} \notin S_1, x_{\sigma(2)} \in S_2, \\
  b_3 & x_{\sigma(1)} \notin S_1, x_{\sigma(2)} \notin S_2, x_{\sigma(3)} \in S_3, \\
  \vdots \\
  b_n & x_{\sigma(1)} \notin S_1, \ldots, x_{\sigma(n-1)} \notin S_{n-1}, x_{\sigma(n)} \in S_n, \\
  b_{n+1} & x_{\sigma(1)} \notin S_1, \ldots, x_{\sigma(n-1)} \notin S_{n-1}, x_{\sigma(n)} \notin S_n.
\end{cases}
\]

In short, the function \( f \) is said to be nested canalizing if \( f \) is nested canalizing in some variable order with some canalizing input sets and some canalized output values.

Let \( S = (S_1, S_2, \ldots, S_n) \) and \( \beta = (b_1, b_2, \ldots, b_{n+1}) \) with \( b_n \neq b_{n+1} \). We say that \( f \) is \( \{\sigma : S : \beta\} \) NCF if it is nested canalizing in the variable order \( x_{\sigma(1)}, \ldots, x_{\sigma(n)} \), with canalizing input sets \( S = (S_1, \ldots, S_n) \) and canalized output values \( \beta = (b_1, \ldots, b_{n+1}) \).
This definition immediately implies the following result.

**Proposition 2.2.4.** A function \( f \) is \( \{ \sigma : \mathbb{S} : \beta \} \) NCF if and only if \( f \) is \( \{ \sigma : \mathbb{S}' : \beta' \} \) NCF, where \( \mathbb{S}' = (S_1, S_2, \ldots, S_n) \) and \( \beta' = (b_1, b_2, \ldots, b_{n-2}, b_{n+1}, b_n) \).

### 2.3 Derrida Values of Networks Governed by NCFs

Gene regulatory networks are often very robust to perturbations. The so-called Derrida plot is a common technique to evaluate the robustness of a Boolean discrete dynamical system [37]. It describes how a perturbation of a given size propagates on average over time. If a small perturbation vanishes over time, the system is considered to be in the ordered regime. The network consists of many steady states and short limit cycles. If the perturbation amplifies over time, the system is in the chaotic regime. A chaotic network possesses long limit cycles. Lastly, if the perturbation remains of similar size, the system is in the critical regime. Many biological systems seem to work in this critical regime; they must be robust enough to withstand perturbations caused by environmental changes but also flexible enough to allow adaptation [12, 108].

The concept of Derrida values has been formally defined in Definition 1.3.1. Until now, the calculation of Derrida values has required extensive Monte Carlo simulations [79, 80, 118]. We derive direct formulas for the Derrida value in networks governed by NCFs.

**Definition 2.3.1.** Let \( F = (f_i)_{i=1}^n \) be a SDDS. Let \( I(i) \) be the set of essential variables of \( f_i \). We call \( k_i = |I(i)| \) the connectivity of \( f_i \). Moreover, let \( x = (x_1, \ldots, x_n), y = (y_1, \ldots, y_n) \in \mathbb{F}_2^n \) be two system configurations that differ at \( m \in \{1, \ldots, n\} \) positions, and let \( V = \{i|x_i \neq y_i\} \) be the set of positions where they differ. Lastly, let \( J(i) = I(i) \cap V \) be the set of essential variables of node \( i \) where \( x \) and \( y \) differ.

**Theorem 2.3.2.** The Derrida value of a (non-stochastic) discrete dynamical system \( F \) can be calculated as the sum of the node-specific probabilities that \( x \) and \( y \) differ after the update,

\[
D(m) = \sum_{i=1}^n \mathbb{P}\left(\left(f_i(x)\right)_i \neq \left(f_i(y)\right)_i \mid |I(i)| = c, |J(i)| = c\right) q(c, k_i),
\]

where

\[
q(c, k) = \mathbb{P}\left(\left(f_i(x)\right)_i \neq \left(f_i(y)\right)_i \left| |I(i)| = k, |J(i)| = c\right.\right)
\]

describes the probability that the \( i \)th component of \( x \) and \( y \) varies after the update, given that the number of essential variables where \( x \) and \( y \) differ and the connectivity of the update function are known.

\( |J(i)| \) follows a hypergeometric distribution,

\[
\mathbb{P}(|J(i)| = c) = \frac{\binom{m}{c} \binom{n-m}{k_i-c}}{\binom{n}{k_i}} = \frac{\binom{k_i}{c} \binom{n-k_i}{m-c}}{\binom{n}{m}}
\]
\textbf{Proof.} Let } \mathbf{x} = (x_1, \ldots, x_n), \mathbf{y} = (y_1, \ldots, y_n) \in \mathbb{F}_2^n \text{ be two system configurations that differ at } m \text{ positions. Since } \mathbf{x} \text{ and } \mathbf{y} \text{ are synchronously updated, the update of each component is independent from the update of other components. This implies that the Derrida value is simply the sum of the probabilities that } x_i \text{ and } y_i \text{ differ after the update, } i = 1, \ldots, n. \text{ Conditioning with respect to } |J(i)| \text{ leads to the second equality.}

\( J(i) \) \text{ is the intersection of two sets so that its magnitude } |J(i)| \text{ follows a hypergeometric distribution.} \qedhere

\textbf{Lemma 2.3.3.} \textit{Let } f \text{ be an NCF with } k \text{ essential variables, and let } c \text{ be the number of essential variables where two system configurations } \mathbf{x} \text{ and } \mathbf{y} \text{ differ. Then,}

\[ q(c, k) = \begin{cases} 
\frac{1}{k} & \text{if } c = 1 \\
\frac{1}{(c)} \sum_{j=1}^{k-c+1} \binom{k-j}{c-j} \left(\frac{1}{2}\right)^j & \text{if } 1 < c \leq k
\end{cases} \]

\textbf{Proof.} Boolean NCFs correspond to the class of Boolean functions with corresponding shortest binary decision diagram \cite{20}. Let } f \text{ be an NCF as in Definition 2.2.3 with } k \text{ essential variables. The value of half the states in } f \text{'s truth table is determined by only one variable, for a quarter of the states two variables matter, for an eighth three variables play a role, etc.}

(i) If } x_{\sigma(1)} \neq y_{\sigma(1)}, \text{ w.l.o.g. we can assume } x_{\sigma(1)} \in S_1, \text{ which means } f(x) = b_1 \text{ and } f(y) = b_j \text{ for some } j \in \{2, \ldots, k+1\}. \text{ Since } \mathbb{P}(b_1 = b_j) = \frac{1}{2}, f(x) = f(y) \text{ with probability } \frac{1}{2} \text{ as well. If on the other hand } x_{\sigma(1)} = y_{\sigma(1)}, \text{ either } f(x) = f(y) = b_1 \text{ or } \sigma(2) \text{ is required for the evaluation, both with probability } \frac{1}{2}. \text{ Only in the latter case, may } f(x) \text{ and } f(y) \text{ differ. If } x_{\sigma(2)} \neq y_{\sigma(2)}, \text{ w.l.o.g. we can again assume } x_{\sigma(2)} \in S_2 \text{ and obtain with the same argument as before that } f(x) = f(y) \text{ with probability } \frac{1}{2}. \text{ If } x_{\sigma(2)} = y_{\sigma(2)}, \text{ the decision moves to the third most important variable, etc. In case the least important variable, } \sigma(k), \text{ is required for the evaluation and } x_{\sigma(k)} = y_{\sigma(k)}, \text{ then } f(x) \text{ and } f(y) \text{ must differ because } \sigma(k) \text{ would not be an essential variable otherwise } (b_k \neq b_{k+1}, \text{ by Definition 2.2.3}. \text{ Therefore,}

\[ \mathbb{P}(f(x) \neq f(y)) = \begin{cases} 
\frac{1}{2} & \text{if } x_{\sigma(1)} \neq y_{\sigma(1)} \\
\frac{1}{4} & \text{if } x_{\sigma(1)} = y_{\sigma(1)}, x_{\sigma(2)} \neq y_{\sigma(2)} \\
\frac{1}{8} & \text{if } x_{\sigma(1)} = y_{\sigma(1)}, x_{\sigma(2)} = y_{\sigma(2)}, x_{\sigma(3)} \neq y_{\sigma(3)} \\
\vdots & \\
\frac{1}{2^{k-1}} & \text{if } x_{\sigma(1)} = y_{\sigma(1)}, \ldots, x_{\sigma(k-2)} = y_{\sigma(k-2)}, x_{\sigma(k-1)} \neq y_{\sigma(k-1)} \\
\frac{1}{2^{k-1}} & \text{if } x_{\sigma(1)} = y_{\sigma(1)}, \ldots, x_{\sigma(k-2)} = y_{\sigma(k-2)}, x_{\sigma(k-1)} = y_{\sigma(k-1)}, x_{\sigma(k)} \neq y_{\sigma(k)}
\end{cases} \]

(ii) There are \( \binom{k}{c} \) c-subsets in the set \{1, \ldots, k\}. \text{ Of these c-subsets, } \binom{k-1}{c-1} \text{ contain 1 as its lowest element, } \binom{k-2}{c-1} \text{ contain 2 as its lowest element, etc., and finally, } \binom{c-1}{c-1} = 1 \text{ contain } k - c + 1 \text{ as its lowest element. Thus if two configurations } \mathbf{x} \text{ and } \mathbf{y} \text{ differ at } c \text{ essential variables, there are } \binom{k-j}{c-1} \text{ possibilities that they first differ at } \sigma(j). \)

(iii) If } c > 1, \mathbf{x} \text{ and } \mathbf{y} \text{ will already first differ before } \sigma(k). \text{ The least important variable is therefore never needed to calculate the probability that } f(x) \text{ and } f(y) \text{ differ. From (i) and
(ii), we get

\[ q(c, k) = \mathbb{P}(f(x) \neq f(y)) = \frac{1}{\binom{k}{c}} \sum_{j=1}^{k-c+1} \binom{k-j}{c-1} \left(\frac{1}{2}\right)^j \]

If \( c = 1 \), \( x \) and \( y \) might differ at \( \sigma(k) \) for the first time and we get

\[ q(c, k) = \frac{1}{\binom{k}{1}} \sum_{j=1}^{k-1} \binom{k-j}{1} \left(\frac{1}{2}\right)^{j-1} \]

\[ = \frac{1}{k} \sum_{j=1}^{k} \left(\frac{1}{2}\right)^{\min(j, k-1)} \]

\[ = \frac{1}{k} \]

\[ \square \]

Lemma 2.3.3 enables us to calculate average Derrida values for a system of NCFs. The NCFs are however only characterized by the number of their essential variables, which is why this theorem only yields an average Derrida value. We can also distinguish NCFs by their Hamming weight (number of 1’s in the truth table), and find a formula for the Derrida values of a system of NCFs, specified by their Hamming weights.

**Lemma 2.3.4.** Let \( f \) be an NCF with \( k \) essential variables and Hamming weight \( w \). Let \( c \) be the number of essential variables where two system configurations \( x \) and \( y \) differ. Then,

\[ q(c, k) = \frac{1}{\binom{k}{c}} \sum_{j=1}^{k-c+1} \binom{k-j}{c-1} \left(\frac{1}{2}\right)^j p_j, \]

where \( p_0 = \frac{w}{2^k} \), \( p_{j+1} = 1 - |1 - 2p_j| \) for \( 0 \leq j \leq k - c \).

**Proof.** Let \( f \) be an NCF with \( k \) essential variables and Hamming weight \( w \).

Let \( p_j = \mathbb{P}(f(x) \neq f(y) \mid x_{\sigma(1)} = y_{\sigma(1)}, \ldots, x_{\sigma(j-1)} = y_{\sigma(j-1)}, x_{\sigma(j)} \neq y_{\sigma(j)}) \) for \( 1 \leq j \leq k - c + 1 \). If \( \frac{w}{2^k} > \frac{1}{2} \), then \( b_1 = 1 \) and if \( \frac{w}{2^k} < \frac{1}{2} \), then \( b_1 = 0 \). Therefore,

\[ p_1 = \begin{cases} 
\frac{2w}{2^k} & \text{if } \frac{w}{2^k} < \frac{1}{2} \\
2(1 - \frac{w}{2^k}) & \text{if } \frac{w}{2^k} > \frac{1}{2} 
\end{cases} \]

\[ = 1 - |1 - 2\frac{w}{2^k}| \]

More general, \( p_j < \frac{1}{2} \) implies that \( b_{j+1} = b_j \), whereas \( p_j > \frac{1}{2} \) implies that \( b_{j+1} \neq b_j \). Therefore,

\[ p_{j+1} = \begin{cases} 
2p_j & \text{if } p_j < \frac{1}{2} \\
2(1 - p_j) & \text{if } p_j > \frac{1}{2} 
\end{cases} \]

\[ = 1 - |1 - 2p_j| \]
and for conclusiveness we set \( p_0 = \frac{w}{2^k} \).

As in the previous proof,

\[
q(c, k) = \sum_{j=1}^{k-c+1} \mathbb{P}\left(\left[x_{\sigma(j)} \neq y_{\sigma(j)}\right] \land \left[x_{\sigma(q)} = y_{\sigma(q)} \forall q, 1 \leq q < j\right] \big| d(x, y) = c\right).
\]

\[
\cdot \mathbb{P}\left(f(x) \neq f(y) \big| \left[x_{\sigma(j)} \neq y_{\sigma(j)}\right] \land \left[x_{\sigma(q)} = y_{\sigma(q)} \forall q, 1 \leq q < j\right]\right)
\]

\[
= \sum_{j=1}^{k-c+1} \binom{k-j}{c-1} \binom{1}{2}^{j-1} p_j
\]

Lemma 2.3.4 enables the exact calculation of the Derrida plot for a system of Boolean NCFs of any Hamming weight. Especially systems with many regulators, this offers a huge improvement, since the time required to approximate the Derrida plot through simulations increases exponentially in the number of regulators.

Figure 2.1 depicts the Derrida values for networks of \( n = 100 \) genes, which are governed by NCFs with \( k = 5 \) regulators and different Hamming weights. The calculation of all the 800 plotted values took less than a second on a regular desktop computer. In networks of NCFs with Hamming weight 1, 3, 29 and 31, small perturbations vanish in average over time. These networks therefore operate in the stable regime. Networks of NCFs with Hamming weight 5, 7, 25 and 27 operate very close to the critical regime, while networks of NCFs with Hamming weight between 11 and 23 operate in the chaotic regime. Surprisingly, networks of NCFs with Hamming weight 11, 13, 19, and 21 are more chaotic than those governed by almost balanced NCFs of Hamming weight 15 and 17. One possible explanation for this observation may be the layer number (see Definition 4.6 in [93] or Definition 2.4.3 in this chapter). NCFs with \( k = 5 \) and Hamming weight 15 or 17 consist of two layers, while NCFs with Hamming weight 13 and 19 (11 and 21) have three (four) different layers. The number of layers seems to be positively correlated with the Derrida value for small perturbations. This observation becomes even clearer when looking at networks of \( n = 100 \) genes governed by NCFs with \( k = 8 \) regulators as in Figure 2.2. The Derrida values for networks of different layers are distinctively separated.

Gene regulation is inherently stochastic. The framework of SDDS captures this stochasticity by assigning gene-specific activation and degradation probabilities, which describe how likely a certain concentration change happens at a given update step (see Section 1.2.2).

**Theorem 2.3.5.** The Derrida value for a SDDS \( F = (f_i, p_i^+, p_i^-)_{i=1}^n \) can be calculated as

\[
D(m) = \sum_{i=1}^{n} \sum_{c=0}^{m} \mathbb{P}\{|J(i)| = c\} \left(\frac{m}{n} (q(c, k_i)r_1 + (1 - q(c, k_i))r_2) + \frac{n-m}{n} (q(c, k_i)r_3 + (1 - q(c, k_i))r_4)\right)
\]
Figure 2.1: Derrida plot for networks of $n = 100$ genes governed by NCFs with $k = 5$ regulators and varying Hamming weight

Figure 2.2: Derrida plot for networks of $n = 100$ genes governed by NCFs with $k = 8$ regulators and varying numbers of layers
Here, $J(i)$ and $q(c, k_i)$ are defined as in Theorem 2.3.3 and
\[
\begin{align*}
    r_1 &= 1 - \frac{1}{2}(p^1_i + p^\uparrow_i) + p^\uparrow_i p^\downarrow_i \\
    r_2 &= 1 - \frac{1}{2}(p^\uparrow_i + p^\downarrow_i) \\
    r_3 &= \frac{1}{2}(p^\uparrow_i + p^\downarrow_i) \\
    r_4 &= \frac{1}{2}(p^\uparrow_i (1 - p^\downarrow_i) + p^\downarrow_i (1 - p^\uparrow_i))
\end{align*}
\]

**Proof.** Let $x = (x_1, \ldots, x_n), y = (y_1, \ldots, y_n) \in \mathbb{F}_2^n$ be two system configurations that differ at $m$ positions. For each node $i \in \{1, \ldots, n\}$, define three events

- $A_i = \{x_i \neq y_i\}$,
- $B_i = \{f_i(x) \neq f_i(y)\}$ before applying $p^\uparrow_i, p^\downarrow_i$,
- $C_i = \{f_i(x) \neq f_i(y)\}$ after applying $p^\uparrow_i, p^\downarrow_i$.

Then, since $A_i$ is independent from $B_i$, we have

\[
D(m) = \sum_{i=1}^{n} \mathbb{P}(C_i)
\]
\[
= \sum_{i=1}^{n} \sum_{c=0}^{m} \mathbb{P}(|J(i)| = c) \left( \mathbb{P}(C_i | A_i, B_i) \mathbb{P}(B_i | c) \mathbb{P}(A_i) + \mathbb{P}(C_i | A_i, \neg B_i) \mathbb{P}(\neg B_i | c) \mathbb{P}(A_i) + \cdots \right)
\]
\[
= \sum_{i=1}^{n} \sum_{c=0}^{m} \mathbb{P}(|J(i)| = c) \left( \mathbb{P}(A_i) \left( \mathbb{P}(C_i | A_i, B_i) \mathbb{P}(B_i | c) + \mathbb{P}(C_i | A_i, \neg B_i) \mathbb{P}(\neg B_i | c) \right) + \cdots \right)
\]
\[
= \sum_{i=1}^{n} \sum_{c=0}^{m} \mathbb{P}(|J(i)| = c) \left( \mathbb{P}(A_i) \frac{1}{2} + \frac{1}{2} p^\uparrow_i p^\downarrow_i + (1 - p^\uparrow_i)(1 - p^\downarrow_i) \right)
\]

If $x$ and $y$ differ at $m$ out of $n$ positions, $\mathbb{P}(A_i) = \frac{m}{n}$.

$\mathbb{P}(B_i | c) = q(c, k_i)$ depends on the choice of the update function. If NCFs are chosen, Lemma 2.3.3 and Lemma 2.3.4 yield the probability.

Lastly, the probability that $x$ and $y$ differ after applying the propensity probabilities needs to be calculated. If $x_i \neq y_i$ and $f_i(x) \neq f_i(y)$ before applying $p^\uparrow_i, p^\downarrow_i$, assume w.l.o.g. that $x_i = 0, y_i = 1$. Then, either $f_i(x) = 0, f_i(y) = 1$ or $f_i(x) = 1, f_i(y) = 0$, both with probability $\frac{1}{2}$. In the first case, there is no change in values so that the propensity probabilities play no role and $f_i(x) \neq f_i(y)$ after applying $p^\uparrow_i, p^\downarrow_i$ with probability 1. In the second case, $f_i(x)$ and $f_i(y)$ only differ after applying $p^\uparrow_i, p^\downarrow_i$ if either both updates happen (probability $p^\uparrow_i p^\downarrow_i$) or neither update happens (probability $(1 - p^\uparrow_i)(1 - p^\downarrow_i)$). That means,

\[
\mathbb{P}(C_i | A_i, B_i) = \frac{1}{2} \cdot \frac{1}{2} + \frac{1}{2} (p^\uparrow_i p^\downarrow_i + (1 - p^\uparrow_i)(1 - p^\downarrow_i)) = 1 - \frac{1}{2} (p^\uparrow_i + p^\downarrow_i) + p^\uparrow_i p^\downarrow_i
\]
Similarly, we can derive
\[
\mathbb{P}(C_i|A_i, \neg B_i) = 1 - \frac{1}{2}(p_i^\uparrow + p_i^\downarrow)
\]
\[
\mathbb{P}(C_i|\neg A_i, B_i) = \frac{1}{2}(p_i^\uparrow + p_i^\downarrow)
\]
\[
\mathbb{P}(C_i|\neg A_i, \neg B_i) = \frac{1}{2}(p_i^\uparrow(1 - p_i^\uparrow) + p_i^\downarrow(1 - p_i^\downarrow))
\]

The Derrida plot is mainly used in the analysis of Boolean networks. Derrida values can however also be defined for the multistate case, where the update functions take on any value in a finite field \(F_p\) of size \(p \geq 2\). Let the Hamming distance of two system configurations still be defined as the number of positions where the configurations differ,
\[
d(x, y) = \left| \{i | x_i \neq y_i \} \right|
\]
where \(x, y \in F_p\).

Then, Theorem 2.3.5 is still valid and the function-specific values, \(q(c, k)\), can be calculated for multistate NCFs.

**Theorem 2.3.6.** Let \(f : \mathbb{F}_p^k \rightarrow \mathbb{F}_p\) be a multistate NCF with \(k\) essential variables, and let \(c\) be the number of essential variables where two system configurations \(x\) and \(y\) differ. Then,
\[
q(c, k) = r_1(p)\left(\frac{p - 1}{p}\right)^{-1} \sum_{i=1}^{c} \left[ s_i(c, k) \left( \frac{1 - r_1(p)}{2} \right)^{i-1} \right] + \frac{c}{k} \left( \frac{1}{2} \right)^{k-c} \left( \frac{1 - r_1(p)}{2} \right)^{c-1},
\]
where
\[
r_1(p) = \frac{\sum_{i=1}^{p-1} i(p - i)}{(p - 1)\left(\begin{array}{c} p \n 2 \end{array}\right)},
\]
\[
s_i(c, k) = \sum_{j=i}^{\min(i+k-c,k-1)} \left( \begin{array}{c} k-j \ n c-i \end{array} \right) \left( \begin{array}{c} j-1 \ n j-i \end{array} \right) \left( \frac{1}{2} \right)^{j-i}, 1 \leq i \leq c
\]

**Proof.** Let \(f\) be a \(\{\sigma : S : \beta\}\) NCF as in Definition 2.2.3 with \(k\) essential variables. Let \(x = (x_1, \ldots, x_k), y = (y_1, \ldots, y_k) \in \mathbb{F}_p^k\) be two system configurations that differ at \(c\) of the \(k\) positions. Let \(\tilde{\sigma}\) be the restriction of \(\sigma\) on the variables where \(x\) and \(y\) differ,
\[
\tilde{\sigma} = \{\sigma(i) | x_{\sigma(i)} \neq y_{\sigma(i)}\}.
\]
The order of the elements in \(\tilde{\sigma}\) is the same as in \(\sigma\).

(i) As an NCF, \(f\) is evaluated in an iterative process. If in the evaluation of \(f(x)\) the most important value \(x_{\sigma(1)}\) is in the canalizing set \(S_1\), the evaluation is completed. If not, the second most important value, \(x_{\sigma(2)}\), is considered, etc. In the Boolean case, whenever \(x\) and \(y\) differ at a variable \(\sigma(i)\), \(x_{\sigma(i)} \in S_i\) and \(y_{\sigma(i)} \not\in S_i\) or \(x_{\sigma(i)} \not\in S_i\) and \(y_{\sigma(i)} \in S_i\). This implies
that after evaluation of \( \sigma(1) \), the most important variable where \( x \) and \( y \) differ, \( x \) and \( y \) always follow different paths in the evaluation process, and all less important variables do not matter when determining the probability that \( f(x) \) and \( f(y) \) differ. This is no longer true in the multistate case. For instance, if \( p = 3, S_1 = \{0\}, x_{\sigma(1)} = 1, y_{\sigma(1)} = 2 \), then the second most important variable \( \sigma(2) \) needs to be considered even though \( x \) and \( y \) differ at the most important variable \( \sigma(1) \). Let

\[
  r_1(p) := \Pr\left(x_{\sigma(i)} \in S_i, y_{\sigma(i)} \notin S_i \vee x_{\sigma(i)} \notin S_i, y_{\sigma(i)} \in S_i \mid x_{\sigma(i)} \neq y_{\sigma(i)} \land S_i \in \mathcal{S}\right),
\]

where

\[
  \mathcal{S} = \{\{0\}, \{0,1\}, \ldots, \{0, \ldots, p - 2\}, \{p - 1\}, \{p - 2, p - 1\}, \ldots, \{1, 2, \ldots, p - 1\}\}
\]

is the set of all possible canalizing input intervals. Clearly, \(|\mathcal{S}| = 2(p - 1)\) and there exist two intervals with one element, two with two elements, etc. Moreover, there are \(\binom{p}{2}\) pairs of \(x_{\sigma(i)}\) and \(y_{\sigma(i)}\) so that the two values are different. For a given \(S_i\),

\[
  \Pr\left(x_{\sigma(i)} \in S_i, y_{\sigma(i)} \notin S_i \vee x_{\sigma(i)} \notin S_i, y_{\sigma(i)} \in S_i \mid x_{\sigma(i)} \neq y_{\sigma(i)} \right) = \frac{|S_i|(p - |S_i|)}{\binom{p}{2}}
\]

Thus,

\[
  r_1(p) = \sum_{S_i \in \mathcal{S}} \frac{|S_i|(p - |S_i|)}{2(p - 1)\binom{p}{2}} = \frac{\sum_{i=1}^{p-1} i(p - i)}{(p - 1)\binom{p}{2}}
\]

(ii) If the \( j \)th evaluation step is needed to determine the probability that \( f(x) \) and \( f(y) \) differ, and if \( x_{\sigma(j)} \neq y_{\sigma(j)} \), then the probability that \( x \) and \( y \) follow different paths in the evaluation process at step \( j \), i.e., the probability that no more steps are needed, is \( r_1(p) \). If \( j = k \) is the least important variable, then \( f(x) \) and \( f(y) \) differ for sure because \( b_k \neq b_{k+1} \) in Definition 2.2.3. If \( j < k \), then this probability is \( \frac{p-1}{p} \), which is the likelihood that \( b_j \) equals any other canalized output value \( b_q, j < q \leq k - 1 \) by chance.

\[
  r_2(j, k, p) := \Pr\left(f(x) \neq f(y) \mid x_{\sigma(j)} \neq y_{\sigma(j)} \land x_{\sigma(q)} \notin S_q \land y_{\sigma(q)} \notin S_q \ \forall q, 1 \leq q < j\right) = \begin{cases} r_1(p)^{p-1} & \text{if } j < k \\ r_1(p) & \text{if } j = k \end{cases}
\]

(iii) To determine the probability that \( f(x) \) and \( f(y) \) differ, the \( j \)th evaluation step is only needed if \( x_{\sigma(q)} \notin S_q \) and \( y_{\sigma(q)} \notin S_q \) for all \( 1 \leq q < j \). For any \( q \),

\[
  \Pr\left(x_{\sigma(q)} \notin S_q \land y_{\sigma(q)} \notin S_q\right) = \begin{cases} \frac{1}{2} & \text{if } x_{\sigma(q)} = y_{\sigma(q)} \\ \frac{1 - r_1(p)}{2} & \text{if } x_{\sigma(q)} \neq y_{\sigma(q)} \end{cases}
\]
If $\sigma(j)$ is the $i$th most important variable where $x$ and $y$ differ ($\sigma(j) = \bar{\sigma}(i)$), then

$$r_3(i, j, p) := \mathbb{P}(x_{\sigma(q)} \notin S_q \land y_{\sigma(q)} \notin S_q \forall q, 1 \leq q < j) = \prod_{q=1}^{j-1} \mathbb{P}(x_{\sigma(q)} \notin S_q \land y_{\sigma(q)} \notin S_q) = \left(1 - \frac{r_1(p)}{2}\right)^{i-1} \left(\frac{1}{2}\right)^{j-i}$$

(iv) The probability that the $j$th most important variable $\sigma(j)$ is the $i$th most important variable, when only considering those variables where $x$ and $y$ differ, is equal to the probability that a $c$-subset of $\{1, 2, \ldots, k\}$ contains $j$ as its $i$th lowest element. There are \(\binom{k}{c}\) $c$-subsets of $\{1, 2, \ldots, k\}$. If $j$ is the $i$th lowest element, then there are \(\binom{j-1}{i-1}\) choices for the $i-1$ lower elements and \(\binom{k-j}{c-i}\) choices for the $c-i$ higher elements. Thus for $1 \leq i \leq c, i \leq j \leq k-c+i$,

$$r_4(i, j, k) := \mathbb{P}(\sigma(j) = \bar{\sigma}(i)) = \frac{\binom{j-1}{i-1}\binom{k-j}{c-i}}{\binom{k}{c}}$$

(v) Only variables where $x$ and $y$ differ matter when deciding whether $f(x)$ equals $f(y)$. We can therefore focus the calculation of $q(c, k)$ on these $c$ variables. The probability $r_4(i, j)$ describes how likely the $i$th most important variable where $x$ and $y$ differ, $\bar{\sigma}(i)$, occurs at position $j$ when considering all variables. Logically, $i \leq j \leq k+i-c$. Thus,

$$q(c, k) = \mathbb{P}(f(x) \neq f(y)|d(x, y) = c) = \sum_{i=1}^{c} \sum_{j=i}^{k+i-c} \mathbb{P}(\sigma(j) = \bar{\sigma}(i)) \cdot \mathbb{P}(x_{\sigma(q)} \notin S_q \land y_{\sigma(q)} \notin S_q \forall q, 1 \leq q < j) \cdot \mathbb{P}(f(x) \neq f(y)\left[x_{\sigma(j)} \neq y_{\sigma(j)}\right] \wedge \left[x_{\sigma(q)} \notin S_q \land y_{\sigma(q)} \notin S_q \forall q, 1 \leq q < j\right])$$

$$= \sum_{i=1}^{c} \sum_{j=i}^{k+i-c} r_4(i, j, k) r_3(i, j, p) r_2(j, k, p)$$

$$= \sum_{i=1}^{c} \min(k+i-c,k-1) \sum_{j=i}^{r_4(i, j, k) r_3(i, j, p) r_2(j, k, p)} + \sum_{j=i}^{r_4(c, k, k) r_3(c, k, p) r_2(k, k, p)}$$

$$= \left[r_1(p)\frac{p-1}{p}\sum_{i=1}^{c} \min(k+i-c,k-1) \sum_{j=i}^{r_4(i, j, k) r_3(i, j, p) r_2(j, k, p)} + \frac{c}{k}\left(1\left(1 - \frac{r_1(p)}{2}\right)^{c-1} r_1(p)\right)$$

$$= r_1(p)\left(p\frac{1}{c}\right)^{-1} \sum_{i=1}^{c} s_i(c, k) \left(1 - \frac{r_1(p)}{2}\right)^{i-1} + \frac{c}{k}\left(1\left(1 - \frac{r_1(p)}{2}\right)^{c-1}\right),$$

where

$$s_i(c, k) = \sum_{j=1}^{\min(i+k-c,k-1)} \left(k-j\right)_{c-i} \left(j-1\right)_{j-i} \left(\frac{1}{2}\right)^{j-i}, 1 \leq i \leq c$$
2.4 Characterization of Nested Canalizing Functions

In this section we first derive a unique polynomial representation of multistate NCFs, which allows a further categorization of this class of functions. Then, we use this polynomial representation to derive explicit formulas for the number of NCFs and for the number of equivalence classes of NCFs.

In the Boolean case, the extended monomial plays an important role in determining a novel polynomial form of NCFs [93]. In the multistate case, the product of indicator functions, which was used in [103], will take over this role.

Definition 2.4.1. Given a proper subset \( S \) of \( \mathbb{F} \), the indicator function (of \( S^c \)) is defined as
\[
Q_S(x) = \begin{cases} 
0 & x \in S, \\
1 & x \in S^c.
\end{cases}
\]

The following theorem states the main result of this section. It gives an algebraic characterization of NCFs.

Theorem 2.4.2. Given \( n \geq 2 \), the function \( f(x_1, \ldots, x_n) \) is nested canalizing if and only if it can be uniquely written as
\[
f(x_1, \ldots, x_n) = M_1 \left( M_2 \left( \cdots \left( M_{r-1} (B_{r+1} M_r + B_r) + B_{r-1} \right) \cdots \right) + B_2 \right) + B_1, \quad (2.4.1)
\]
where each \( M_i \) is a product of indicator functions of a set of disjoint variables. More precisely, \( M_i = \prod_{j=1}^{k_i} (Q_{S_{ij}}(x_j)) \), \( i = 1, \ldots, r, k_i \geq 1 \) for \( i = 1, \ldots, r, k_1 + \cdots + k_r = n, B_2, \ldots, B_{r+1} \neq 0, B_1 \in \mathbb{F}, \{ t_j | j = 1, \ldots, k_i, i = 1, \ldots, r \} = \{ 1, \ldots, n \} \), and if \( k_r = 1 \), then \( B_{r+1} + B_r \neq 0 \).

Proof. First, let \( b_i = \sum_{j=1}^{k_i} B_j \). Then it is straightforward to check that any function written as in Equation 2.4.1 is a \( \{ \sigma' : S' : \beta' \} \) NCF, where
\[
\sigma'(x_1, \ldots, x_n) = (x_{11}, \ldots, x_{1k_1}, \ldots, x_{r1}, \ldots, x_{rk_r}), \\
S' = (S_{11}, \ldots, S_{1k_1}, \ldots, S_{r1}, \ldots, S_{rk_r}), \text{ and} \\
\beta' = (b_1, b_1, b_2, \ldots, b_r, \ldots, b_r, b_{r+1}).
\]

Second, suppose \( f \) is a \( \{ \sigma : S : \beta \} \) NCF, where \( S = (S_1, S_2, \ldots, S_n) \) and \( \beta = (b_1, b_2, \ldots, b_{n+1}) \)
with $b_n \neq b_{n+1}$. Then there exist $k_i, i = 1, \ldots, r$, $k_1 + \cdots + k_r = n$, $k_i \geq 1$, such that

$$b_1 = \cdots = b_{k_1} =: C_1,$$

$$b_{k_1+1} = \cdots = b_{k_1+k_2} =: C_2,$$

$$\vdots$$

$$b_{k_1+\cdots+k_{r-1}+1} = \cdots = b_n =: C_r,$$

$$b_{n+1} =: C_{r+1},$$

and

$$C_j \neq C_{j+1}, j = 1, \ldots, r$$

Let $B_1 := C_1, B_2 := C_2 - C_1, \ldots, B_{r+1} = C_{r+1} - C_r$. Hence, $B_1 \in \mathbb{F}, B_2, \ldots, B_{r+1} \in \mathbb{F} - \{0\}$, and $f(x) = M_1(M_2(\cdots (M_{r-1}(B_{r+1}M_r + B_r) + B_{r-1}) \cdots ) + B_2) + B_1$, which shows that any NCF can be written as in Equation 2.4.1.

Finally, we need to show that each NCF has a unique polynomial representation. Without loss of generality, let $\sigma$ be the identity permutation, i.e., let $f$ be nested canalizing in the variable order $x_1, \ldots, x_n$. Besides, let $f$ be written as in Equation 2.4.1. Then all the variables of $M_1, x_1, \ldots, x_{k_1}$, are canalizing variables of $f$ with common canalized output $B_1$. We will now show that $f$ has no other canalizing variables to prove the uniqueness of $M_1$ and $B_1$. If $x_1 \in S_f^c, \ldots, x_{k_1} \in S_{k_1}^c$, then all the variables of $M_2, x_{k_1+1}, \ldots, x_{k_2}$, are canalizing variables of the subfunction $f_1 = M_2(\cdots (M_{r-1}(B_{r+1}M_r + B_r) + B_{r-1}) \cdots ) + (B_2 + B_1)$. Since $B_1 \neq B_1 + B_2$, $x_{k_1+1}, \ldots, x_{k_2}$ are not canalizing variables of $f$. In the same manner, all variables of $M_3$ are not canalizing variables of $f_1$ and thus not canalizing variables of $f$ either. Iteratively, we can prove that $x_1, \ldots, x_{k_1}$ are the only canalizing variables of $f$, which proves the uniqueness of $M_1$ and $B_1$. In the same way, the uniqueness of $M_2, \ldots, M_r$ and $B_2, \ldots, B_{r+1}$ follows. \hfill \square

Because each NCF can be uniquely written in the form of Equation 2.4.1, the number $r$ is uniquely determined by $f$, and can be used to specify the class of NCFs.

**Definition 2.4.3.** For an NCF $f$, written in the form of Equation 2.4.1, let the number $r$ be called its layer number. Essential variables of $M_1$ are called most dominant variables (canalizing variables), and are part of the first layer of $f$. Essential variables of $M_2$ are called second most dominant variables, and are part of the second layer, etc.

**Remark 2.4.4.** In the Boolean case ($p = 2$), each $M_i$ in Theorem 2.4.2 is an extended monomial, and, since $B_{r+1} + B_r = 1 + 1 = 0$, $k_r$ is greater than 1. Thus, Theorem 2.4.2 reduces to its Boolean version, already stated as Theorem 4.2 in [93]. On the other hand, if $B_{r+1} + B_r$ would be zero, then $k_r = 1$ is impossible since the function could be uniquely written in $r - 1$ layers (see comments after Lemma 2.4.6).

**Remark 2.4.5.** Equation 2.4.1 allows the use of Corollary 4.8 in [93]. The layer number of any NCF can be determined by counting the number of changes in the canalized output values. For example, if $p = 3$ and if $f$ is nested canalizing with canalized output values $\beta = (1, 1, 1, 0, 0, 0, 2, 0, 2, 2, 1)$ ($n = 11$), then the layer number of $f$ is 5.
Lemma 2.4.6. Let \( a, b \) be any nonzero elements of \( \mathbb{F} \), and let \( S \) be any interval of \( \mathbb{F} \). The number of different functions \( f = bQ_S(x) + a \), which cannot be written as \( cQ_{S'}(x) \), where \( c \neq 0 \) and \( S' \) is an interval of \( \mathbb{F} \), is \( (p - 1)^2(p - 2) \).

Proof. If a function \( f(x) = bQ_S(x) + a \) can be written as \( cQ_{S'}(x) \), then

\[
f = bQ_S(x) + a = \begin{cases} a & x \in S \\ a + b & x \in S^c \end{cases} = \begin{cases} 0 & x \in S' \\ c & x \in S'^c = cQ_{S'}(x). \end{cases}
\]

Since \( a \) and \( c \) are nonzero, \( a + b = 0 \) must hold for such a function. Since \( \mathbb{F} \) contains \( p - 1 \) nonzero numbers, there are \( p - 1 \) choices for \( b \) and \( p - 2 \) choices for \( a \), to obtain a function that cannot be written as \( cQ_{S'}(x) \). Moreover, there are \( 2(p - 1) \) different intervals \( S \), but only half of them lead to a different function since every function can be expressed in two different ways:

\[
bQ_S(x) + a = b(1 - Q_{S'}(x)) + a = -bQ_{S'}(x) + (a + b).
\]

Thus, there are \( (p - 1)^2(p - 2) \) different functions \( f = bQ_S(x) + a \) that cannot be written as \( cQ_{S'}(x) \).

This proof also shows that \( bQ_S(x) + a = cQ_{S'}(x) \) for some \( c \neq 0 \) and some interval \( S' \) of \( \mathbb{F} \), if and only if \( a + b = 0 \).

Lemma 2.4.7. Given \( a, b \neq 0 \) and intervals \( S_i, i = 1, \ldots, k \) with \( k \geq 2 \), then

1. \( f(x) = f(x_1, \ldots, x_k) = b \prod_{j=1}^{k} Q_{S_j}(x_j) + a \) cannot be written as \( c \prod_{j=1}^{k} Q_{S'_j}(x_j) \), where \( c \neq 0 \) and all \( S'_j \) are intervals, \( j = 1, \ldots, k \).

2. There are \( 2^k(p - 1)^{k+2} \) different functions of the form \( b \prod_{j=1}^{k} Q_{S_j}(x_j) + a \).

Proof. (1) Assume a function \( f(x) = b \prod_{j=1}^{k} Q_{S_j}(x_j) + a \) can be written as \( c \prod_{j=1}^{k} Q_{S'_j}(x_j) \), then

\[
b \prod_{j=1}^{k} Q_{S_j}(x_j) + a = \begin{cases} a & \exists j : x_j \in S_j \\ a + b & \forall j : x_j \in S_j^c \end{cases} = \begin{cases} a & x \in (S_1 \times \mathbb{F}^{k-1}) \cup \ldots \cup (\mathbb{F}^{k-1} \times S_k) \\ a + b & x \in S_1^c \times \ldots \times S_k^c \end{cases} = \begin{cases} 0 & x \in (S'_1 \times \mathbb{F}^{k-1}) \cup \ldots \cup (\mathbb{F}^{k-1} \times S'_k) \\ c & x \in S_1'^c \times \ldots \times S_k'^c \end{cases} = c \prod_{j=1}^{k} Q_{S'_j}(x_j).
\]

Since \( a, c \neq 0 \), \( a + b = 0 \) must hold. Hence,

\[
S_1^c \times \ldots \times S_k^c = (S'_1 \times \mathbb{F}^{k-1}) \cup \ldots \cup (\mathbb{F}^{k-1} \times S'_k).
\]
This last statement is, however, impossible. Thus, there is no function \( f(x) = b \prod_{j=1}^{k} Q_{S_j}(x_j) + a \) that can be written as \( c \prod_{j=1}^{k} Q_{S'_j}(x_j) \).

(2) The nonzero constants \( a, b \) can be arbitrarily chosen, with \( p-1 \) choices each. Contrary to the previous lemma, each choice of intervals \( S_1, \ldots, S_k \) leads to a different function because \( S_1 \times \ldots \times S_k \neq (S_1 \times \mathbb{F}^{k-1}) \cup \ldots \cup (\mathbb{F}^{k-1} \times S_k) \) and because \( a \neq a + b \). For each interval there are \( 2(p-1) \) choices, which is why altogether there are \( (p-1)(p-1)(2(p-1))^k = 2^k(p-1)^{k+2} \) different functions of the form \( b \prod_{j=1}^{k} Q_{S_j}(x_j) + a \). \( \square \)

Let \( \text{NCF}(n) \) denote the set of all nested canalizing functions in \( n \) variables.

**Theorem 2.4.8.** For \( n \geq 2 \), the number of NCFs is given by

\[
|\text{NCF}(n)| = 2^{n-1}p(p-2) \sum_{r=2}^{n} (p-1)^{n+r-1} \sum_{k_1+\ldots+k_r-1=n-1 \atop k_i \geq 1, i=1,\ldots,r} \frac{n!}{k_1!k_2!\cdots k_r!} \\
+ 2^n p \sum_{r=1}^{n-1} (p-1)^{n+r} \sum_{k_1+\ldots+k_r=n \atop k_i \geq 1, i=1,\ldots,r-1, k_r \geq 2} \frac{n!}{k_1!k_2!\cdots k_r!}
\]

**Proof.** If \( r = 1 \), then \( f = B_2 M_1 + B_1 \). Similar to Lemma 2.4.7, the number of such functions is \( (2(p-1))^n(p-1)p = 2^n(p-1)^{n+1}p \), since \( B_1 \in \mathbb{F} \) can be arbitrarily chosen, and \( k_1 = n \).

For \( r > 1 \), Equation 2.4.1 yields that for each choice of \( k_1, \ldots, k_r \), \( k_i \geq 1, i = 1, \ldots, r \), there are \( (2(p-1))^k \binom{n-k_1-\ldots-k_r-1}{k} \) ways to form \( M_j, j = 1, \ldots, r \). For those NCFs with \( k_r = 1 \), by Lemma 2.4.6 there are \( (p-1)^2(p-2) \) different functions of the form \( B_{r+1} M_r + B_r \) with \( B_r, B_{r+1} \neq 0 \). For the remaining NCFs, i.e., those with \( k_r > 1 \), Lemma 2.4.7 yields that there are \( (p-1)^2(2(p-1))^{k_r} \) ways to form \( B_{r+1} M_r + B_r \), with \( B_r, B_{r+1} \neq 0 \).

Note that there are \( p-1 \) choices for each \( B_i, 2 \leq r \leq B_{r-1}, p \) choices for \( B_1 \), and \( 2(p-1) \) choices for each canalizing input interval. Hence, the total number of NCFs with \( r > 1, k_r = 1 \), can be given by

\[
N_1 = \sum_{r=2}^{n} \sum_{k_1+\ldots+k_r-1=n-1 \atop k_i \geq 1, i=1,\ldots,r-1} (2(p-1))^{k_1+\ldots+k_r-1} \frac{n!}{(k_1)(n-k_1)!k_2!(n-k_1-k_2)!\cdots(k_r)(n-k_1-\cdots-k_{r-2})!k_{r-1}!(n-k_1-\cdots-k_{r-1})!} \\
= 2^{n-1}p(p-2) \sum_{r=2}^{n} \sum_{k_1+\ldots+k_r-1=n-1 \atop k_i \geq 1, i=1,\ldots,r-1} (p-1)^{n+r-1} \frac{n!}{k_1!k_2!\cdots k_r!}
\]

Similarly, the total number of NCFs with \( r > 1, k_r > 1 \) is
By combining all three groups of NCFs, the total number of NCFs in \( n \) variables is

\[
|\text{NCF}(n)| = 2^n(p-1)^{n+1} + N_1 + N_2
\]

\[
= 2^n - 1 p(p-2) \sum_{r=2}^{n} (p-1)^{n+r-1} \sum_{k_1 + \cdots + k_{r-1} = n-1 \atop k_i \geq 1, i = 1, \ldots, r-1} \frac{n!}{k_1! k_2! \cdots k_{r-1}!}
\]

\[
+ 2^n p \sum_{r=1}^{n} (p-1)^{n+r} \sum_{k_1 + \cdots + k_r = n \atop k_i \geq 1, i = 1, \ldots, r-1, k_r \geq 2} \frac{n!}{k_1! k_2! \cdots k_r!}
\]

(2.4.2)

Note that for \( p = 2 \), we get the same formula as in [93]. However, we are now also able to compute the number of multistate NCFs. For example, when \( p = 3 \) and \( n = 2, 3, 4 \), we get 192, 5568, 219468, respectively; when \( p = 5 \) and \( n = 2, 3, 4 \), we get 5120, 547840, 78561280, respectively. These results are consistent with those in [103].

It has been shown in [103] that the number of multistate NCFs can be calculated recursively. By expressing Equation 2.4.2 recursively, we get

**Corollary 2.4.9.** For the nonlinear recursive sequence

\[
a_2 = 4(p-1)^4, a_n = \sum_{r=2}^{n-1} \binom{n}{r-1} 2^{r-1}(p-1)^r a_{n-r+1} + 2^{n-1}(p-1)^{n+1}(2 + n(p-2)), n \geq 3
\]

it holds that

\[
|\text{NCF}(n)| = p a_n,
\]

and the explicit solution for \( a_n \) is given by

\[
a_n = 2^{n-1}(p-2) \sum_{r=2}^{n} (p-1)^{n+r-1} \sum_{k_1 + \cdots + k_{r-1} = n-1 \atop k_i \geq 1, i = 1, \ldots, r-1} \frac{n!}{k_1! k_2! \cdots k_{r-1}!}
\]

\[
+ 2^n \sum_{r=1}^{n} (p-1)^{n+r} \sum_{k_1 + \cdots + k_r = n \atop k_i \geq 1, i = 1, \ldots, r-1, k_r \geq 2} \frac{n!}{k_1! k_2! \cdots k_r!}
\]
Definition 2.4.10. Given two functions \( f(x_1, \ldots, x_n) \) and \( g(x_1, \ldots, x_n) \) over \( \mathbb{F} \). We call \( f \) and \( g \) permutation equivalent if there exists a permutation \( \sigma \) such that \( f(x_1, \ldots, x_n) = g(x_{\sigma(1)}, \ldots, x_{\sigma(n)}) \).

Equivalent functions share many properties. For example, two equivalent Boolean NCFs have the same sensitivity and the same average sensitivity \([93, 92]\). We are interested in the number of different equivalence classes of NCFs. To find this, we need the following combinatorial result.

Lemma 2.4.11. \([24]\) (Page 70) Given \( n, r \) and \( s_i, i = 1, \ldots, r \) and \( s = s_1 + \cdots + s_r \leq n \). Then the number of integer solution of the equation \( k_1 + \cdots + k_r = n \), where \( k_i \geq s_i \), is

\[
\sum_{\substack{k_1 + \cdots + k_r = n \\ k_i \geq s_i, s = s_1 + \cdots + s_r \leq n}} 1 = \binom{r + n - s - 1}{r - 1}.
\]

Theorem 2.4.12. For \( n \geq 2 \), the number of different equivalence classes of NCFs is

\[ N = 2^{n-1}(p-1)^{n+1}p^n. \]

Proof. The number of different equivalent classes of NCFs equals the number of different NCFs with a fixed canalizing variable order \( \sigma \) in Equation 2.4.1. Thus, we can follow the same enumerative schedule as we did in the proof of Theorem 2.4.8 except that we do not consider the permutation of the variables. Hence, we get

\[
N = 2^{n-1}p(p-2)\sum_{r=2}^{n}(p-1)^{n+1}r^{r-1}\sum_{\substack{k_1+\cdots+k_{r-1}=n-1 \\ k_i \geq 1, i=1,\ldots,r-1}} 1
\]

\[ + 2^np\sum_{r=1}^{n-1}(p-1)^{n-r}r^{r-1}\sum_{\substack{k_1+\cdots+k_r=n \\ k_i \geq 1, i=1,\ldots,r, k_r \geq 2}} 1
\]

\[ = 2^{n-1}p(p-1)^{n+1} \left[ (p-2)\sum_{r=2}^{n}(p-1)^{r-2}\binom{n-2}{r-2} + 2\sum_{r=1}^{n-1}(p-1)^{r-1}\binom{n-2}{r-1} \right]
\]

\[ = 2^{n-1}p(p-1)^{n+1}\left[ (p-2)p^{n-2} + 2p^{n-2} \right]
\]

\[ = 2^{n-1}(p-1)^{n+1}p^n,
\]

where we used Lemma 2.4.11 to eradicate the inner sums in the first equality and the binomial theorem to simplify the sums in the second equality.

2.5 Generalization of NCFs

We can generalize the concept of nested canalization to any finite field \( \mathbb{F}_q \), where \( q \) is a power of a prime. Moreover, the canalizing sets do not need to be restricted to intervals containing
exactly one endpoint but can be any proper subset of $F_q$. Within this setting, we obtain similar results as in the previous section. The proofs are the same as in Section 2.4; we just replace the number of elements $p$ in the finite field by $q$, the number of nonzero elements, $p - 1$ by $q - 1$, and the number of intervals, $2p - 2$, by the number of proper subsets $2^n - 2$, respectively.

**Definition 2.5.1.** Let $f : F_q^n \to F_q$ be a function in $n$ variables over $F_q$, and let $S_i$ be proper subsets of $F_q$, $i = 1, \ldots, n$. Then NCFs are defined in the same way as in Definition 2.2.3.

This definition generalizes the concept of NCFs as defined in [102]. However, NCFs defined in this way can still be uniquely represented as a polynomial as discussed above. The following theorem is a generalized version of Theorem 2.4.2 with the same arguments used to prove it.

**Theorem 2.5.2.** Given $n \geq 2$, the function $f : F_q^n \to F_q$ is nested canalizing if and only if it can be uniquely written as

$$f(x_1, \ldots, x_n) = M_1(M_2(\cdots(M_{r-1}(B_{r+1}M_r + B_r) + B_{r-1})\cdots) + B_2) + B_1,$$

where each $M_i$ is a product of indicator functions of a set of disjoint variables. More precisely, we have that $M_i = \prod_{j=1}^{k_i} (Q_{S_{ij}}(x_{ij}))$, $i = 1, \ldots, r$, $k_i \geq 1$ for $i = 1, \ldots, r$, $k_1 + \ldots + k_r = n$, $B_2, \ldots, B_{r+1} \neq 0$, $B_1 \in F$, $\{ij | j = 1, \ldots, k_i, i = 1, \ldots, r\} = \{1, \ldots, n\}$, each $S_i$, $i = 1, \ldots, n$ is any proper subset of $F_q$, and, if $k_r = 1$, then $B_{r+1} + B_r \neq 0$.

As in the previous section, we can count the number of NCFs and get

**Theorem 2.5.3.** For $n \geq 2$, the number of NCFs over $F_q$, where NCFs are defined as in Definition 2.5.1, is given by

$$|\text{NCF}(n)| = 2^{n-1}q(q - 2)(2^{q-1} - 1)^n \sum_{r=2}^{n}(q - 1)^{r-1} \sum_{\substack{k_1 + \ldots + k_{r-1} = n-1 \atop k_i \geq 1, i = 1, \ldots, r-1}} \frac{n!}{k_1!k_2! \cdots k_{r-1}!}$$

$$+ 2^n q(2^{q-1} - 1)^n \sum_{r=1}^{n-1}(q - 1)^r \sum_{\substack{k_1 + \ldots + k_r = n \atop k_i \geq 1, i = 1, \ldots, r-1, k_r \geq 2}} \frac{n!}{k_1!k_2! \cdots k_r!}$$

**Theorem 2.5.4.** The number of different equivalence classes of NCFs over $F_q$, where NCFs are defined as in Definition 2.5.1, is

$$N = 2^{n-1}(q - 1)q^n(2^{q-1} - 1)^n.$$

2.6 Discussion

In this chapter, we introduced explicit formulas for the calculation of Derrida values of networks governed by NCFs. Moreover, we generalized the concept of nested canalization...
to finite fields without any restriction on the canalizing input sets. By using the product of indicator functions, a generalization of extended monomials, we successfully generalized the characterization of Boolean NCFs [93] to the multistate case. Besides, we discussed the permutational equivalence of NCFs, and obtained a number of the different equivalence classes.

The main contributions of this work are threefold. First, we introduced formulas for the calculation of the Derrida value of (stochastic) discrete dynamical systems governed by NCFs, which are the first of their kind, and abolish the need for simulations in the calculation of the Derrida plot. Second, we established a closed form formula for the number of multistate NCFs as defined in [102], improving on the recursive formula given there, and introduced a new closed form formula for the number of equivalence classes of NCFs under permutation equivalence. Third, we established a very general definition of multistate NCFs, for which many of the analytical results remain true.
Chapter 3

Stabilizing Gene Regulatory Networks Through Feed Forward Loops


Contributions: David Murrugarra, Reinhard Laubenbacher and I initially discussed the broad idea of this project. I designed and implemented the study, and finally wrote the paper. Reinhard Laubenbacher guided me in all of these steps through helpful discussions and editing.

Abstract

The global dynamics of gene regulatory networks are known to show robustness to perturbations in the form of intrinsic and extrinsic noise, as well as mutations of individual genes. One molecular mechanism underlying this robustness has been identified as the action of so-called microRNAs that operate via feedforward loops. We present results of a computational study, using the modeling framework of stochastic Boolean networks, which explores the role that such network motifs play in stabilizing global dynamics. We introduce a new measure for the robustness of stochastic networks. The results show that certain types of feedforward loops do indeed buffer the network against stochastic effects.
3.1 Introduction

The term *canalization* was coined by the geneticist C. Waddington [134] to describe the theory that embryonal development is buffered against genetic and environmental perturbations. It is only recently that a molecular basis for this phenomenon has been suggested. Recent research has highlighted how the intrinsic stochasticity of gene expression can drive changes in phenotypes [9]. Short segments of single-stranded RNA, so-called microRNAs, represent an entirely novel agent of gene regulation discovered relatively recently [6, 14], and have been proposed to function as canalizing agents that buffer the effects of stochasticity in gene expression [57, 121]. According to this theory, when microRNA expression is perturbed, stochasticity in gene expression can result in transitions to distinct cellular phenotypes. As microRNAs bind to gene targets they downregulate translation of target mRNA into protein. Embedded in several different types of so-called feedforward loops (FFLs), microRNAs help smooth out noise and generate canalizing effects in gene regulation by overriding the effect of certain genes on others [97]. Complex networks (viewed as graphs) ranging from the transcriptional networks in yeast and *E. coli* to engineered systems are enriched for certain graph-theoretic motifs that include feedforward loops [99].

An understanding of canalization in evolutionary biology is important as a cornerstone of natural selection and the emergence of new phenotypes [133], as well as for the understanding of diseases such as cancer. Transitions to new phenotypes have been implicated as one of the driving forces of tumorigenesis [22, 59, 14, 74, 89]; and, interestingly, significantly altered expression of microRNAs is a feature of several cancers. Much experimental work remains to be done in elucidating this concept, and recent years have seen an explosive growth of publications in this area. There have also been a number of computational studies focused on canalization. Several papers have studied computational models that capture the evolution of canalization in networks and their ability to support significant mutation without change in the phenotype [15, 61], while others have studied models of stochastic gene expression as the internal source of noise in regulatory networks [67]. A detailed stochastic model of the ability of microRNAs to buffer gene expression noise in a feedforward loop has been proposed [109], providing evidence that this type of network motif can in fact play a canalizing role. There is evidence that microRNAs do not regulate their target genes directly; rather they act as post-transcriptional regulators by reducing the amount of mRNA and by repressing mRNA translation [131], e.g., by binding to the 3’-UTR of a mRNA, which prevents this mRNA from being translated.

In this chapter, we carry out a computational study of the ability of the feedforward loop motif to buffer a gene regulatory network against intrinsic noise. This is done using stochastic Boolean network models as a computational instantiation of gene regulatory networks. We introduce a measure on networks that captures their “distance-to-deterministic” characteristics in terms of the stability of their attractors. For a given network, we successively introduce feedforward loops and track the resulting change in dynamics. The results show clearly that the feedforward loop motif buffers the network phenotype, in terms of stability of attractors, against perturbations from intrinsic noise.
<table>
<thead>
<tr>
<th>Network Size</th>
<th>5</th>
<th>15</th>
<th>30</th>
<th>50</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cycle Length</td>
<td>1</td>
<td>2</td>
<td>4</td>
<td>8</td>
</tr>
<tr>
<td>1</td>
<td>2.835</td>
<td>3.6577</td>
<td>4.3492</td>
<td>4.8709</td>
</tr>
<tr>
<td>2</td>
<td>0.1529</td>
<td>0.1522</td>
<td>0.1540</td>
<td>0.1486</td>
</tr>
<tr>
<td>4</td>
<td>0.0244</td>
<td>0.0378</td>
<td>0.0415</td>
<td>0.0415</td>
</tr>
<tr>
<td>8</td>
<td>0.0002</td>
<td>0</td>
<td>&lt;0.0001</td>
<td>&lt;0.0001</td>
</tr>
<tr>
<td>16</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Sample Size</td>
<td>120000</td>
<td>42000</td>
<td>40000</td>
<td>62500</td>
</tr>
</tbody>
</table>

Table 3.1: This table shows the average number of steady states and limit cycles that remain attractors in the SDDS framework for different network sizes. We found fewer than thirty such 8-cycles among more than 250,000 networks of different sizes, and no 16-cycles at all. This shows that including only limit cycles of length 16 and less is not restricting the study.

### 3.2 Modeling Framework

In this study, the framework of stochastic discrete dynamical systems (SDDS) is used to model gene regulatory networks (see Section 1.2.2 for details of the framework).

Most limit cycles that are attractors in a deterministic system are no longer attractors in an SDDS because a cycle can be exited with a certain probability. Nevertheless, one particular kind of limit cycle, which consists of $2^k$ elements and in which all but $k$ bits are fixed, remains an attractor even in an SDDS. One such example is a 2-cycle, in which just one bit switches states, e.g., 000 $\leftrightarrow$ 001. Table 3.1 shows that such limit cycles occur very rarely by chance, and for computational reasons, we decided to include only steady states and limit cycles of length $2^5$ or less in this study. This restriction does not influence the study since longer limit cycles that remain attractors in the SDDS practically do not occur.

The basic procedure underlying the computational study is, for a given SDDS, referred to as the “basic” network, to construct several “extended” networks by successively adding nodes, which we shall refer to as microRNAs, together with one or more feedforward loops involving the new microRNAs in a specific way. We then measure the change in phenotypic robustness using the AMTP measure (see Section 1.3.2 for details). Let $F = (f_i, p_i^+, p_i^-)_{i=1}^{n_1}$ be the basic network, and let $F^* = (f_i^*, q_i^+, q_i^-)_{i=1}^{n_2}$ be the extended network, $n_1 \leq n_2$. Furthermore, let $\mu(F)$ be the AMTP value of $F$. We hypothesize that the feedforward loop enriched network is more robust. To compare the phenotypic robustness of two systems, we consider their difference $m$ in AMTP values,

$$m(F, F^*) = \begin{cases} 0 & \text{if } \mu(F^*) - \mu(F) = 1, \\ \frac{\mu(F) - \mu(F^*)}{1 - \min(\mu(F), \mu(F^*))} & \text{otherwise.} \end{cases} \quad (3.2.1)$$

The denominator scales this difference into the range $[-1, 1]$. Here, $m = 0$ means that both networks have equal phenotypic robustness. If $m$ is positive, the extended network $F^*$ is
more robust than the basic network $F$, and a negative value of $m$ suggests the opposite. The magnitude describes the difference in degree of phenotypic robustness. A magnitude of 1 means that one of the networks is as robust as a deterministic system, whereas a magnitude of 0.5 suggests that one system is 50% more phenotypically robust than the other.

### 3.3 Methods

For each set of input nodes we generated a certain number $N$ of basic Boolean SDDS $F$, introduced microRNAs, in a way that will be specified later in this section, to obtain the extended network $F^*$, and then compared their degree of determinism via $m(F, F^*)$. We considered 4 network sizes $n$: 5, 15, 30, 50 nodes. The corresponding number $N$ is 20000, 7000, 5500, 5500, respectively. The networks generated are random, subject to the following restrictions.

Large-scale studies of *B. subtilis*, *E. coli* and *S. cerevisiae* strongly suggest that the in-degrees of nodes in a transcriptional regulatory network are Poisson distributed with a mean of about two [5]. Thus, we chose a Poisson distribution with parameter $\gamma = 2$ for the basic network. The only other restriction is that each gene is regulated by at least one other gene, which raises the average in-degree to approximately 2.2. The regulators of each gene are randomly chosen from the set of all $n$ genes in the network, allowing self-regulation.

All propensity parameters for transcription factors and microRNAs are also randomly chosen from $[0, 2, 1]$. For computational reasons, the full interval $[0, 1]$ is not used since a propensity parameter close to 0 might strongly decelerate convergence to attractors, slowing the performance of the simulation. However, 0.2 as the lower limit for the propensity parameters still allows one process to happen up to five times more frequently than another. Each gene is regulated by a certain number of genes, depending on its in-degree, and random Boolean functions that actually depend on all input variables are used as update functions.

After creating the basic network, we generate an extended network in a way reminiscent of post-transcriptional regulation by microRNAs. (Since we do not include a corresponding protein node for each gene node, the analogy is limited). Starting with the basic SDDS, microRNAs are iteratively added to this system by randomly choosing one gene as a transcription factor (TF) that induces transcription of the introduced microRNA. This microRNA, in turn, reduces the mRNA level of its own transcription factor; one example for such coregulation is the interplay between miR-133b and Pitx3 in midbrain dopamine neurons [81]. A lower TF mRNA level leads to a lower TF protein level, which then affects the regulation of all TF target genes. In the Boolean framework, a gene-specific threshold is used to distinguish between low (0) and high concentration (1). For some target genes, even after the TF mRNA reduction, there might still be enough transcription factor so that the target concentration is on the same side of the threshold as if no reduction had taken place; for other target genes, the target concentration might change significantly because of the TF mRNA reduction. Since this reduction is caused by the microRNA presence, the microRNA becomes a new regulator of the target genes in the latter case. One input variable in this
Table 3.2: Example of how an additional microRNA is embedded as an input variable in a target gene’s update function. The light grey rectangle contains the original random update function, in which $a_1, \ldots, a_4 \in \{0, 1\}$ can be any Boolean values, with the sole restriction that the update function must depend on both inputs. Unless both microRNA and transcription factor are present, the microRNA has no influence since there is no TF mRNA that can be degraded or there is no microRNA that can catalyze this degradation. Only if both are present (gray rows), can the microRNA reduce the TF mRNA level to an extent that the target concentration changes because of the microRNA.

<table>
<thead>
<tr>
<th>microRNA(t)</th>
<th>TF(t)</th>
<th>other input(t)</th>
<th>target(t+1)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>$a_1$</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>$a_2$</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>$a_3$</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>$a_4$</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>$a_1$</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>$a_2$</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>$a_1$</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>$a_2$</td>
</tr>
</tbody>
</table>

study, called microRNA strength, describes the probability that transcription factor-target gene pairs fall into this latter case, i.e., that the TF mRNA level is significantly reduced, so that the Booleanized target concentration is the same as if no transcription factor had been present at all. If, for instance, the microRNA strength is 0.5, any microRNA regulates on average half of its transcription factor’s target genes. However, we require each microRNA to regulate at least one target gene. This restriction ensures that each microRNA is part of at least one feedforward loop, consisting of transcription factor, microRNA, and target gene. Table 4.1 depicts an example of how the update function of target genes regulated by a microRNA is expanded, taking into consideration the mode of action of microRNAs. Only if microRNA is present and TF mRNA is transcribed, the mRNA reduction takes place. In this case we see the same output as if no TF mRNA were present.

Because all networks with exactly one attractor already have totally deterministic dynamics, in the sense defined earlier, i.e., $\mu \equiv 1$ for those networks, only basic networks with multiple attractors are considered in this study. Those are the interesting networks, in which actual stabilization of the dynamics might be observed. Particularly interesting dynamics occur if at least two attractors possess a relatively large basin of attraction. The network selection process therefore favors networks with multiple large basins of attraction by picking a network only if at least two attractors are found more than once starting from twenty random initial configurations.

If the extended network $F^*$ does not possess multiple attractors, $m(F, F^*) = 1$ by definition. One could argue that the loss of attractors in the extended network is one feature of stabilization through feedforward loops. On the other hand, however, this could be seen as an experimental bias. To consider both views, we use $m$ to define two output measures, $m_1$
and $m_2$, one regarding any extended network and the other considering only those network pairs in which the extended network also possesses multiple attractors with at least two large basins of attraction. For a given set of input variables, we generate $N$ basic and extended networks, and measure

\[ m_1 = \frac{1}{N} \sum_{i=1}^{N} \{ m(F, F^*) \mid \text{basic network } F \text{ and extended network } F^* \text{ have multiple attractors} \}, \]

(3.3.1)

\[ m_2 = \frac{1}{N} \sum_{i=1}^{N} \{ m(F, F^*) \mid \text{only the basic network } F \text{ is required to have multiple attractors} \}. \]

(3.3.2)

For any set of input variables, we expect $m_2 \geq m_1$ since all network pairs with less than two attractors in the extended network, which are omitted in $m_1$, have $\mu \equiv 1$ and thus a mean value closer to 1. However, we do not want to prefer one or the other measure and thus we report results for both, which have been obtained independently, i.e., a network pair that was used for $m_1$ is not used for $m_2$.

A full analysis of the state space of a SDDS is only possible for small networks, so we used random sampling of initial configurations and an estimate of transition probabilities to attractors to approximate $m(F, F^*)$. We created a set of 100 random initial configurations, which were used in both networks to find the transition probabilities to attractors by updating each configuration 50 times, until an attractor was reached. In a small preliminary study, we found that these two values yield a good trade-off between accuracy and efficiency.

### 3.4 Results

#### 3.4.1 Average Maximal Transition Probability

Overall, we created over 300,000 pairs of basic and feedforward loop enriched networks. The results for networks with sizes ranging from 5 to 50 genes can be seen in Figure 3.1. None of these networks were required to be strongly connected, and in all of them the introduced microRNAs had full strength, meaning that each microRNA regulates all of its transcription factor’s target genes in a feedforward loop structure. The main result is that both measures, $m_1$ and $m_2$, are indeed positive for all network sizes and numbers of introduced microRNAs, indicating that microRNA-mediated FFLs can actually stabilize networks. It can also be seen that the impact of a single microRNA/FFL decreases when the network becomes larger. This means that larger networks require more microRNAs/FFLs for the same degree of stabilization.

Table 3.3 shows the results for networks of size 50. We see that more microRNAs and thus more FFLs stabilize the dynamics. Whereas one microRNA of full strength with $m_1 \approx$
Figure 3.1: $m_1$ and $m_2$ are plotted against the number of introduced microRNAs. Networks are not necessarily strongly connected, and the microRNA has full strength. The size of the considered networks varies from 5 (solid line) to 50 (dotted line). The impact of a single FFL on the dynamics is larger in smaller networks, which suggests that larger networks require more FFLs for the same amount of stabilization.
### Table 3.3: Comparison of the degree of stochasticity via $m_1$ and $m_2$ for not necessarily strongly connected networks of 50 genes, in which various numbers of microRNAs with full strength (A) and with strength 0.5 (B) are introduced. Overall, the more microRNA-mediated FFLs are introduced, the less stochastic the network dynamics become.

4% only has a small impact, five such microRNAs already lead to $m_1 \approx 12\%$, and the introduction of thirty microRNAs of full strength stabilizes the stochastic system quite a lot ($m_1 \approx 0.37\%$). As expected, $m_2$ yields higher values and thirty microRNAs already reduce the stochasticity in the dynamics by more than 50% ($m_2 \approx 0.57\%$). In the case that each microRNA regulates on average only half its transcription factor’s target genes (but at least one), all values are considerably lower; the general behavior does not change, however.

These results raise the question whether a given network can be fully stabilized by introducing a sufficient number of microRNAs. Indeed, under certain conditions this is possible by ensuring the existence of a unique steady state. If an $n$-gene network contains no self-regulating genes, then $n$ microRNAs with full strength, each regulated by another gene, suffice to have fully deterministic network dynamics. Since the microRNA has full strength, it will downregulate any present mRNA, which ensures that only the value $a_1 \in \{0, 1\}$ (compare Table 3.4) can be taken by the target gene at a steady state. Each gene is regulated by at least one other gene. Hence, each gene and its regulated microRNA can only take the value $a_1$ in its truth table and the existence of a unique steady state is guaranteed. Thus, $\mu \equiv 1$ for such networks, which is equivalent to fully deterministic dynamics in the sense defined in Section 3.2.

### 3.4.2 Derrida Values

In this study, we introduce a new measure for the robustness of stochastic networks by quantifying the degree of determinism of network dynamics. Another measure that can be used in the Boolean context was suggested by Derrida in 1986 [37]. Pairs of initial configurations of fixed Hamming distance are sampled from the entire state space, and
<table>
<thead>
<tr>
<th>microRNA1(t)</th>
<th>microRNA2(t)</th>
<th>TF1(t)</th>
<th>TF2(t)</th>
<th>target(t+1)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>$a_1$</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>$a_2$</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>$a_3$</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>$a_4$</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>$a_1$</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>$a_1$</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>$a_3$</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>$a_3$</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>$a_1$</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>$a_2$</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>$a_1$</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>$a_1$</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>$a_1$</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>$a_1$</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>$a_1$</td>
</tr>
</tbody>
</table>

Table 3.4: If each of a target gene’s transcription factor regulates a microRNA that degrades the transcription factor mRNA, then only the fixed value $a_1 \in \{0, 1\}$ can be taken on at steady state because each transcription factor and its regulated microRNA have to take on the same value, 0 or 1, at a steady state (gray rows). Here, the light gray rectangle contains the original update function, and $a_1, \ldots, a_4$ are any Boolean values such that the update function depends on both inputs.

Their mean normalized Hamming distance, after being updated using update functions and propensity parameters, is defined as the Derrida value for a given initial Hamming distance. Lower Derrida values reflect more stable dynamics. To take time dependencies into account, we also considered the mean Hamming distance after two and three time steps as has been done earlier [79]. Table 3.5 displays the percent change in Derrida values starting with a basic 50-gene network and introducing 30 microRNAs. In all cases, the change is negative, i.e., the Derrida values decreased, indicating that the extended network exhibits more stable dynamics than the basic network, which we observed for different network sizes as well. Thus, another commonly used robustness measure also agrees with our hypothesis, which suggests that our findings are independent of the choice of robustness measure.

### 3.5 Discussion

We have examined the effect of feedforward loop motifs in stochastic Boolean network models of transcriptional networks, in analogy to the regulatory effects of microRNAs. Our goal was to test the hypothesis that these regulatory motifs have the effect of buffering the network against stochastic effects in the sense that they stabilize the basins of attraction. To this end, we conducted a computational experiment on a large number of randomly generated networks. The networks were modified by introducing additional nodes and feedforward motifs.
Table 3.5: Derrida values for initial small disturbances of a Hamming distance up to 5 were simulated for a basic 50-gene network and an extended network with an additional 30 microRNAs of full strength. Multiple time steps were taken into account to consider time dependencies. The table shows that the percent change of the Derrida values from the basic to the extended network is always negative, indicating that our findings do not depend on the choice of the robustness measure.

<table>
<thead>
<tr>
<th>Time Steps</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>-3.6709</td>
<td>-5.0620</td>
<td>-6.3319</td>
<td>-7.3764</td>
<td>-8.3597</td>
</tr>
</tbody>
</table>

motifs in a way that suggests regulation by microRNAs. To capture the effect on network stability we introduced a new measure of stochasticity of a network suitable for this purpose. Using this measure, as well as the classical measure of Derrida values, we showed that indeed the introduction of microRNAs has the hypothesized buffering effect.

The number of microRNAs that are introduced strongly influences the magnitude of the stabilizing effect, so that one might wonder how many feedforward loops can be expected to be found in actual gene regulatory networks. In a data set from *E. Coli*, among 424 nodes with 519 edges, 40 FFLs have been found [117]. In *S. cerevisiae*, among 685 genes with 1,052 interactions, there are at least 70 FFLs [101]. However, restricting the data to subnetworks, we find other occurrence frequencies of FFLs. A subnetwork of *E. Coli* of 67 nodes with 102 edges containing 42 FFLs was identified (some new FFLs had been found by then), and in *D. melanogaster*, a subnetwork of 54 nodes and 167 edges contained as many as 157 FFLs [64]. These numbers indicate that the question of how many FFLs are reasonable in a gene network of a certain size seems to depend strongly on the average in-degree of the nodes; whereas even large networks with average in-degree of less than 2 have few FFLs, this number can rise quickly when the network becomes more highly connected, as indicated by the considered network of *D. melanogaster*, with an average in-degree of approximately 3.

Additionally, we looked at the correlations between the number of attractors in both networks and the number of common attractors, where we defined a configuration in both networks to equal if the states of all genes, i.e., the first $n$ bits, coincide. Figure 3.2 shows the observed correlations, and we notice expected decreasing correlations between all three variables when more microRNAs are introduced. Surprisingly, the number of attractors of the extended network is much more strongly correlated with the number of common attractors than the respective number for the basic network, the cause of which remains to be explored.

This study can be extended in several ways, which we are planning to pursue. To make the study design more realistic, it is useful to introduce additional nodes for proteins, in order to be able to implement more mechanistic details of microRNA regulation. Also, here we do not restrict the regulatory rules to those that correspond to activation and inhibition only, which does not allow the classification of feedforward loops into coherent and incoherent,
Figure 3.2: The correlation between the number of attractors in the basic and the extended network and the number of common attractors is compared pairwise. These values have been obtained from not necessarily strongly connected networks of 50 genes and the introduced microRNAs had full strength.

an important distinction. Also, a more careful study remains to be done on the effect of microRNAs relative to their position in the network and the local network topology into which they are embedded. Finally, another limitation of this work is that only intrinsic noise is being considered as a perturbation. It is important, however, to also take extrinsic noise into account, which requires an extension of the SDDS framework.

### 3.6 Conclusions

This study provides computational evidence that microRNA-mediated feedforward loops have the effect of buffering the network against phenotypic variation due to stochastic effects. Introducing a feedforward loop motif has a local effect on network dynamics that propagates to a generally much smaller global effect on attractor stability. Thus, as the number of feedforward loop motifs increases, the overall stabilizing effect increases as well. In our study, the number of microRNAs introduced is of a relative order of magnitude that might be expected in an actual transcriptional network. Thus, our computational experimental setup can be used in conjunction with an appropriate experimental system to investigate the effects of individual microRNA actions.
Chapter 4

The Regulatory Network of DNA Mismatch Repair

Most of the work described in this chapter was conducted as part of an REU summer program at Virginia Tech’s Bioinformatics Institute. Ross Donatelli, Marschall Furman and Madeleine Weinstein participated in this program, I served as their mentor and Reinhard Laubenbacher lead the Modeling and Simulation in Systems Biology REU program, which was supported by the National Science Foundation under Award Number 1062878. Moreover, Reinhard Laubenbacher and I were supported by the National Science Foundation under Grant Nr. CMMI-0908201.

Parts of this chapter will soon be submitted for publication as: C Kadelka, S Arat, R Donatelli, M Furman, M Weinstein, C Heinen, R Laubenbacher. The Regulatory Network of DNA Mismatch Repair.

Contributions: Reinhard Laubenbacher and I discussed the general idea of this study. The three REU students, Ross Donatelli, Marschall Furman, Madeleine Weinstein, and I conducted the extensive literature study, assembled the regulatory network, and conducted most of the robustness experiments together. At the end of their REU program, the students wrote a final paper, which served as a rough draft of this section. I generated network predictions that can possibly be validated in the laboratory by Seda Arat and Chris Heinen. Throughout the whole project, Chris Heinen served as biology expert.

Abstract

Although failure of DNA Mismatch Repair (MMR) is associated with microsatellite instability and colorectal cancer, little is known about MMR except for its biochemical pathway. By assembling known regulatory interactions, we built a novel gene regulatory network of
MMR, based on a time- and state-discrete modeling framework, which accounts for the cell’s inherent stochasticity. This model describes MMR’s response to hypoxia and DNA damage-inducing factors like UV radiation. We support the hypothesis that microRNAs can stabilize network dynamics, thus enhancing genomic stability, by showing that over-expressing microRNAs increases robustness while knocking them out seems to have the opposite effect. In addition to providing a gene regulatory network of MMR, the model yields various experimentally verifiable predictions and enables further analysis of the potential stabilizing effect of microRNAs on gene regulatory networks.

4.1 Introduction

Robustness is the biological phenomenon of maintaining persistent phenotypes in the face of mutations and cellular noise. One type of biological regulator thought to play a role in robustness is microRNAs. MicroRNAs are noncoding segments of RNA that are approximately 22 nucleotides long. They act as post-transcriptional inhibitors by binding to their target mRNAs, catalyzing their degradation, and preventing their translation in protein. The hypothesis that microRNAs confer robustness to gene networks has been supported in computational studies like the one in Chapter 3. In this chapter, we test the effect of microRNAs on the particular regulatory network of DNA Mismatch Repair.

DNA Mismatch Repair (MMR) is a process mammalian cells employ to fix base-base mismatches, which are one of the errors that can occur during DNA replication. A deficiency in MMR can cause a 50- to 1000-fold increase in mutation rates, and a working MMR machinery is therefore considered an important guardian of genome stability. [65] Loss-of-function mutations in MMR genes cause hereditary nonpolyposis colorectal cancers [58, 110].

Despite MMR’s biological significance, most research has focused on its direct biochemical pathway. This chapter introduces a model describing the regulatory logic behind the biochemical pathway, which can be used to explain possible feedback effects in MMR effectiveness. We use the SDDS framework to model the regulatory logic behind MMR. Using two different robustness measures, the AMTP-values and the Derrida values, we investigate the effect of microRNAs and other regulatory proteins on the activity of MMR. MicroRNAs seem to confer robustness to the regulatory network.

4.2 Methods

4.2.1 Modeling Framework

To capture a cell’s inherent stochasticity, we use the SDDS framework to model the gene regulatory network of DNA Mismatch Repair (see Section 1.2.2 for details of the framework).
To keep the regulatory logic as simple as possible we use Boolean logic. A directed graph, called the wiring diagram, describes the regulatory network. Nodes represent genes, proteins, microRNAs, external factors, or biological phenomena. Edges represent activations or inhibitions of one node by another. The update function for each node is based on biological information about how its regulators combine to affect the node. Moreover, to account for the inherent stochasticity of gene regulation, activation and degradation propensities are assigned to each node.

4.2.2 Quantifying Robustness

An examination of the mechanisms that confer robustness to a cellular program requires rigorously defined measures. In this study we use the AMTP-values (see Section 1.3.2 for details) and the Derrida values (see Section 1.3.1 for details) to measure robustness. The latter is a well-known measure that has been used in the field of Boolean networks since almost thirty years [37], whereas the first measure constitutes a new way of quantifying robustness, focusing on the phenotypic stability of network attractors [72]. To take time dependencies, e.g. through feedback or feedforward loops, into account, we also considered the mean Hamming distance after multiple time steps as has been done earlier [72, 79]. Generally, lower Derrida values reflect more robust dynamics.

4.2.3 Building the Regulatory Network of MMR

The biochemical pathway that describes how a cell responds to DNA damage is well-understood [58, 65, 88]. Two proteins, MSH2 and MSH6, form a complex, MutSα, which can detect a short nucleotide mismatch and attach to it. Another protein complex, called MutLα that consists of the proteins MLH1 and PMS2, then binds to MutSα and starts the excision of part of the DNA strand containing the mismatch. Various other proteins such as Exo1, PCNA and MSH3 are also involved in this response. MSH3 for instance forms another complex with MSH2, called MutSβ, which works similar as MutSα but detects longer nucleotide mismatches.

Little is, however, known about the regulatory network that governs this pathway and, to our knowledge, no regulatory network of MMR exists. Information about specific proteins involved in MMR is readily available in numerous publications. We therefore conducted a large literature search with the goal of assembling a gene regulatory network that governs the key MMR proteins, MSH2, MSH3, MSH6, MLH1 and PMS2. Focusing not only on their direct regulators but also on regulators of the regulators, etc., we found over 70 regulators with over 250 interactions as the largest version of such a network. To build the regulatory network reported in this chapter, we started with the key MMR proteins and added only those regulators and interactions which seemed absolutely essential.

We simplified the regulatory network in a couple of ways. First, we only considered the four
Table 4.1: Truth table of the default update function in the DNA Mismatch Repair network.

<table>
<thead>
<tr>
<th>target(t)</th>
<th>TF(t)</th>
<th># active miRNAs(t)</th>
<th>target(t+1)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0 (0.00)</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0 (0.00)</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>&gt;1</td>
<td>0 (0.00)</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1 (0.75)</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0 (0.38)</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>&gt;1</td>
<td>0 (≤0.19)</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0 (0.25)</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0 (0.25)</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>&gt;1</td>
<td>0 (0.25)</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1 (1.00)</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1 (0.63)</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>&gt;1</td>
<td>0 (≤0.43)</td>
</tr>
</tbody>
</table>

main MMR proteins, MSH2, MSH6, MLH1 and PMS2. In particular, we did not include MSH3 because we could not find any experimental evidence for regulators of MSH3. The network therefore primarily captures the regulation of single nucleotide mismatch repair as MutSα detects primarily those mismatches, whereas the MSH3-involving MutSβ is needed for the detection of longer mismatches. Second, we abbreviated some indirect pathways by removing the intermediate proteins; consequently, an activation or inhibition of a target by another protein in the network does not necessarily mean that this protein is an actual regulator of the target.

### 4.2.4 Update Functions

Each node in the network is assigned an update function. If biological evidence did not suggest otherwise, we used the following function as default update rule (see Table 4.1 for the truth table),

\[
\text{Target}(t + 1) = \begin{cases} 
0, & \text{if } \left(\frac{1}{4}\text{Target}(t) + \frac{3}{4}\left(\frac{1}{2}\#\text{active miRNAs}(t) \times \text{TF}(t)\right)\right) \leq \frac{1}{2} \mid \text{inhibitors}(t) \\
1, & \text{if } \left(\frac{1}{4}\text{Target}(t) + \frac{3}{4}\left(\frac{1}{2}\#\text{active miRNAs}(t) \times \text{TF}(t)\right)\right) > \frac{1}{2} \& \neg\text{inhibitors}(t)
\end{cases}
\]

\[
= \text{TF}(t) == 1 \& (\text{miRNAs}(t) == 0 \mid (\text{miRNAs}(t) == 1 \& \text{Target}(t) == 1))
\]

This formula incorporates assumptions about protein degradation, as well as the different regulatory mechanisms of microRNAs, other inhibitors, and transcription factors.
We assume that proteins degrade between consecutive time steps. This assumption matters in determining how a node’s current concentration level affects its concentration level at the next time step. The constants in the formula are chosen in a way that regardless of a node’s concentration at a current time step, it cannot be expressed at the next time step unless one of its transcription factors is currently present to activate it. On the other hand, if a node is actively inhibited by microRNAs, its current concentration level does affect its level at the next time step.

We assume that microRNAs have an additive[^1] and weak inhibitory[^11] effect on their target proteins. The additive effect leads to the following rules: If only one of a node’s regulatory microRNAs is expressed, then in the presence of transcription factor the concentration level of the node at the next time step depends on its current level. If two or more of a node’s regulatory microRNAs are expressed, they degrade possibly present mRNA to an extent so that the concentration of the node at the next time step is low.

Other inhibitors regulate their targets in general more strongly than single microRNAs. If such inhibitors are present, they completely prevent presence of the target at the next time step. This situation only applies to two nodes in the system: cMyc can be entirely inhibited by the presence of hypoxia or p53[^2][^82], and mir-17/92 can be completely inhibited by p53[^135].

The final assumption built into the default update function is that transcription factors have no additive effect unless they are known to act as a complex. Thus, most nodes can be fully activated by the presence of any one of their transcription factors. The sole exception is mir-21; experiments have shown that p53, NFkB, and STAT3 must all be present together in order to start transcription of mir-21[^28]. We therefore assume that mir-21 is not present unless all three of its regulators are present.

While the default update rule can be used for 12 of the 14 regulated nodes in the system, there are two exceptional nodes which require a customized update function: DNA damage and p53. DNA damage can be caused by DNA damage-inducing factors like UV radiation and eliminated by a functioning mismatch repair machinery. Unlike the proteins, however, DNA damage remains present without continued activation. It can only return to a low level if the MMR machinery is working. P53 can only actively work in the presence of DNA damage since it needs to be phosphorylated by DNA damage[^26]. Thus, DNA Damage and one of p53’s three transcription factors must be present, for p53 to become active.

### 4.2.5 Knockout and Over-expression Experiments

In order to investigate the impact of each node on the network, we created different versions of the original network. In each version, one individual node was *knocked-out* or *over-expressed*; rather than using the node’s update function, we set it to a constant value of zero for knock-outs or one for over-expressions.
Using AMTP-values and Derrida values, we then compared the robustness of all these versions to the original network as control. Since no particular values for propensity parameters are known, we used $p^i_1, p^i_2 = 0.5$ for all propensity parameters whenever we investigated stochastic networks.

Since the entire state space consists of $2^{16}$ states, the AMTP-values are calculated by sampling. For a number of $n_{IC}$ randomly generated state space configurations, we performed $n_{samples}$ simulations per configuration to approximate its maximal transition probability to an attractor. Thus, instead of Equation 1.3.1, we used the approximation

$$
\mu(F) = \frac{1}{n_{IC}} \sum_{i=1}^{n_{IC}} \left( \max_{A \in k(F)} \left( \frac{1}{n_{samples}} \sum_{j=1}^{n_{samples}} (x_i \xrightarrow{F} \cdots \xrightarrow{F} A) \right) \right) \in [0, 1].
$$

(4.2.1)

For all knockout and over-expression experiments, we sampled $n_{IC} = 5000$ initial configurations and $n_{samples} = 100$ trajectories to attractors per configuration. This means that we sampled about 7.5% of the entire state space.

To obtain Derrida values we sampled 10000 pairs of initial configurations and observed their average development over the course of 10 time steps. We only allowed the pairs of initial configurations to differ in regular nodes, not in any of the microRNAs or the external factors.

### 4.2.6 Rewiring Experiments

To further understand why microRNAs might increase the robustness of the MMR network, we performed a robustness analysis on slightly altered, ”rewired” networks. We randomly rewired the outgoing edges of microRNAs, thereby randomly selecting new miRNA targets. Possible targets were the nine nodes in the network that were neither microRNAs, ideological nodes (DNA Damage and MMR), nor external factors. Each microRNA still targeted the same number of nodes as in the original control network.

For 1000 rewired networks, we calculated AMTP-values and Derrida values, as well as the number of attractors and maximum basin sizes to compare with the original network. Because of the large number of generated networks, we used only $n_{IC} = 200$ initial configurations with $n_{samples} = 75$ trajectories to attractors per configuration for the sampling of the AMTP-values, and 1000 pairs of initial configurations for the Derrida values.
4.3 Results and Discussion

4.3.1 Regulatory Network

We conducted a large literature study to assemble a regulatory network model of DNA Mismatch Repair. Figure 4.1 shows its wiring diagram. We focused on the response of MMR to the external factors, Hypoxia and DNA-damaging factors like UV radiation. In short, UV leads to increased MMR activity through E2F and cMyc [56], whereas hypoxia reduces MMR activity through NFKB and STAT3 [35, 75]. Several microRNAs influence MMR, and we included the three most important ones in the model [88, 27]. Mir-21 and mir-155 directly inhibit MMR proteins, while mir-17-92 inhibits the MMR process via inhibition of E2F, which is one of the key transcription factors of the MMR proteins [119, 111, 123].

The wiring diagram in Figure 4.1 shows the (direct and indirect) regulators of each node, as well as the type of regulatory interaction, activation or inhibition. Based on this information, each node is assigned an update function, as described in Section 4.2.4. The set of all update functions, together with gene-specific propensity parameters, forms the SDDS $F = (f_i, p^i_↑, p^i_↓)_{i=1}^{16}$ that describes MMR regulation.

Figure 4.2 depicts the attractors of the system for all four different combinations of external conditions. Overall, there are 19 attractors, all of which are steady states. Many attractors are very similar and are grouped in one row. Attractors 5 and 6, as well as 16 and 17 only differ in the concentration level of MLH1, and attractors 18 and 19 only differ in the concentration level of STAT3. Finally, attractors 7-13 share the same concentration levels for all nodes but the MMR proteins, MLH1, MSH2 and MSH3; at least one of these three proteins is at a low concentration. The fact that we obtain multiple attractors in these cases can be considered an artifact of the model. Although presence of DNA Damage is all that distinguishes attractors 1 from attractor 2, we do not group them because these two attractors correspond to distinct phenotypes (see Section 4.3.2).

Among the attractors, certain patterns can be discerned that are already supported by the literature. For example, in cases of hypoxia, mismatch repair is off due to excess levels of NFKB and STAT3. Moreover, since E2F is a necessary transcription factor for certain MMR proteins, only attractors in which E2F is expressed at a high level have an active MMR machinery. Overall, the visual representation of all attractors clarifies which nodes correspond with MMR functioning and failure, indicating new relationships that cannot immediately be seen in such a complex network.

4.3.2 Network Validation

As part of a model validation, we regarded the individual effect of each node on the phenotype of the system, which can then be compared to the experimentally observed effect. A good
Figure 4.1: This visualization shows the wiring diagram of the regulatory network of DNA Mismatch Repair. Green solid arrows signify activation, red dashed arrows stand for inhibition. Gray nodes denote external factors, and ellipsoid shapes represent microRNAs.
Figure 4.2: This table depicts the steady states of the MMR model. One row may contain multiple phenotypically equal steady states. Bold lines separate different external conditions. Dark blue represents high level of expression, white represents low level, light blue means that both levels of expression lead to separate steady states, which correspond however to the same phenotype.

Based on the level of MMR activity and DNA damage, we defined three different phenotypes. Each network attractor corresponds to one of these phenotypes. The system is in an abnormal state if DNA damage is high and the level of MMR activity is low. It is in a normal state if the opposite is true, DNA damage is low and MMR activity is high. If both DNA damage and MMR activity are high or both are low, the system is in a mediocre state. We assign a score of 0 to the abnormal, 1 to the mediocre, and 2 to the normal phenotype. The higher the score, the more normal is the state of the system. For $x = (x_{MMR}, x_{p53}, \ldots, x_{UV})$, define the score as

$$
    s(x) = \begin{cases} 
        2 & \text{if } x_{MMR} = 1 \text{ and } x_{DNA \text{ Damage}} = 0 \\
        1 & \text{if } x_{MMR} = x_{DNA \text{ Damage}} \\
        0 & \text{if } x_{MMR} = 0 \text{ and } x_{DNA \text{ Damage}} = 1.
    \end{cases}
$$

We measured the effect of an individual node on the network phenotype in two different ways. First, we looked at all initial configurations where the node under consideration is at a fixed value and calculated the mean score of the attractors reached by these configurations, e.g.,

$$
    m_1(\text{NfkB}, 0) = \frac{1}{2^{n-1}} \sum_{x \in \{0,1\}^n, x_{\text{NfkB}} = 0} s(y|x \xrightarrow{F} \cdots \xrightarrow{F} y) \in [0, 2].
$$

For the second measure, we knocked out or over-expressed the node under consideration by changing the node’s update function to a constant value. We then calculated the mean score
of the attractors reached by all initial configurations in this new network, e.g.,

\[ m_2(\text{NfkB}, 0) = \frac{1}{2^{n-1}} \sum_{x \in \{0, 1\}^n, x_{\text{NfkB}}=0} s\left( y | x \xrightarrow{F^*} \cdots \xrightarrow{F^*} y \right) \in [0, 2] \]

where \( F^* = (f_{\text{MMR}}, \ldots, f_{\text{NfkB}} = 0, \ldots, f_{\text{UV}}) \).

For each non-MMR protein and miRNA in the network, we compared the score of the original network and the score when the node is knocked-out and over-expressed. Figure 4.3 is based on \( m_1 \), while Figure 4.4 is based on \( m_2 \).

As could be expected, completely changing a node’s update function (as in \( m_2 \)) has a larger effect than just setting a node at an initially high or low level (as in \( m_1 \)). Since the two measures are defined differently, we do not focus on the magnitude of the effect but only on the relative height of red and green bars. If a high level of a protein leads to a more normal phenotype, this protein is predicted to have a positive effect on the system. Otherwise, the predicted effect is negative. Thus, under normal conditions (that is, in a normoxic environment and low levels of DNA damage inducing factors like UV), p53, NfkB, and the three miRNAs have a negative effect, while E2F and cMyc have a positive effect. For STAT3, the two measures do not agree.

Some of these predictions are currently being tested experimentally (no results yet). The MMR machinery is working properly if all four MMR genes are expressed. We therefore compare the expression level of MSH2, MLH1, MSH6 and PMS2 in a normal cell (control) and in a cell in which a certain protein is knocked out. The experimental procedure for E2F knockout is as follows.

Human cervical adenocarcinoma cells (HeLa) are cultured in Earle’s modified Eagle’s medium (MEM) (Invitrogen, Carlsbad, CA) supplemented with 10% fetal bovine serum (FBS) and antibiotics at 37°C in a humidified incubator containing 5% \( \text{CO}_2 \). The control siRNA (Silencer Negative Control#1 Cat. No. 4611) and predesigned siRNAs (ID114508 for E2F1) have been purchased from Ambion (Austin, TX). Transfections are performed using the Lipofectamine 2000 reagent (Invitrogen), according to the manufacturer’s instructions. Cells are plated into 60mm dishes (Corning, New York, NY) 24 hours prior to transfections. At the time of transfections, the cell confluency is about 40%. Briefly, 10µl of siRNA and 1 ml of transfection reagent are incubated in 1 ml of Opti-MEM I Reduced Serum Medium (Invitrogen) for 20 min to facilitate complex formation at room temperature. The resulting mixture is added to the cells. After 4 hours, 2 ml of DMEM is added to the cells (day 0). siRNA-treated cells are harvested on day 2, 3, 4 and 5 for time-course data. After centrifugation, we process the supernatant (cytosolic fraction) using Western blotting and examine the effect of E2F1 knockdown on the DNA mismatch proteins: MSH2, MLH1, MSH6 and PMS2.
Figure 4.3: The effect of an individual node on the network phenotype is examined by splitting up the state space. The relative height of the red and green bars determines the effect of the node on the system. If the red (green) bar is higher, the node has a negative (positive) effect on MMR.
Figure 4.4: The effect of an individual node on the network phenotype is examined by knocking out and over-expressing the node. The relative height of the red and green bars determines the effect of the node on the system. If the red (green) bar is higher, the node has a negative (positive) effect on MMR.
4.3.3 Robustness Analysis

We hypothesized that the presence of miRNAs stabilizes the regulatory network. Figure 4.5 exhibits that over-expressing microRNAs, i.e., constantly holding their values at 1, increased the AMTP-value (compared to the value of the original control network). A positive correlation between the number of over-expressed microRNAs and the AMTP-values seems to exist; the simultaneous over-expression of all three miRNAs leads to the highest average AMTP-value, while the over-expression of two miRNAs leads to higher average AMTP-values than other experiments. However, over-expressing solely miR-155 leads to AMTP-values as high as over-expressing the two other miRNAs, which can be interpreted as miR-155 confers the most robustness among the miRNAs. The AMTP-values of the knockout experiments on the other hand do not fully follow the expected pattern. We expected that any miRNA knockout would lower the AMTP-value, which is true for miR-21 and miR-17-92. Knocking out miR-155, together with possibly other miRNAs, leads, however, to higher AMTP-values.

The results obtained with the second robustness measure, the Derrida values, almost perfectly support with our hypothesis. Figure 4.6 shows how pairs of configurations develop over the span of three times steps. Over-expression of miRNAs leads to considerably lower Derrida values as the control network, which, in turn, has lower values than any knockout experiment. Moreover, the more miRNAs are simultaneously fixed at a value, the larger is
Figure 4.6: Single miRNAs, pairs of miRNAs and all three miRNAs are over-expressed or knocked out and the Derrida plot after three time steps of the corresponding network is shown in this figure. The x-axis shows the initial perturbation size, while the y-axis exhibits the average size of such a perturbation three time steps later. A blue dashed line $y = x$ is plotted to separate the chaotic regime above the line from the ordered regime below the line (see Section 1.3.1 for details).

the difference in Derrida values (compared to the original network). In other words, over-expression of microRNAs leads to more stable systems, whereas knocking out microRNAs leads to more chaotic systems. Figure 4.7 emphasizes on the development over time of the possibly smallest perturbation. While after one time step, the original network has the highest Derrida value among all 15 considered networks, this quickly changes and the Derrida values settle down after five time steps since attractors are being reached. In the long-run, a small perturbation has the largest impact in a network, where all three miRNAs are knocked out, and much smaller influence on networks where miRNAs are over-expressed. According to Derrida values, miR-17-92 seems to be the most influential miRNA as a network, in which only miR-17-92 is knocked out, has considerably higher Derrida values than a network, in which both other miRNAs are knocked out.
Development of Perturbations of Initial Size 1

Figure 4.7: Single miRNAs, pairs of miRNAs and all three miRNAs are over-expressed or knocked out and the Derrida values after a variable number of time steps of the corresponding network is shown in this figure. The x-axis shows the number of time steps that both configurations being updated, while the y-axis exhibits the average size of a single perturbation x time steps later.
Table 4.2: This table displays the AMTP-values for various networks, in which always one regulatory protein or miRNA is knocked out or over-expressed.

<table>
<thead>
<tr>
<th>Name of Node</th>
<th>AMTP-value of Knockout</th>
<th>AMTP-value of Over-expression</th>
</tr>
</thead>
<tbody>
<tr>
<td>Control</td>
<td>0.7955</td>
<td>0.7955</td>
</tr>
<tr>
<td>E2F</td>
<td>0.8635</td>
<td>0.8102</td>
</tr>
<tr>
<td>c-Myc</td>
<td>0.8249</td>
<td>0.7893</td>
</tr>
<tr>
<td>NFKB</td>
<td>0.8327</td>
<td>0.7963</td>
</tr>
<tr>
<td>p53</td>
<td>0.7724</td>
<td>0.8362</td>
</tr>
<tr>
<td>STAT3</td>
<td>0.7573</td>
<td>0.8138</td>
</tr>
<tr>
<td>miR-21</td>
<td>0.7688</td>
<td>0.8690</td>
</tr>
<tr>
<td>miR-155</td>
<td>0.8324</td>
<td>0.8775</td>
</tr>
<tr>
<td>miR-17-92</td>
<td>0.7918</td>
<td>0.8236</td>
</tr>
</tbody>
</table>

To conclude that the three miRNAs really play a special role in the network, we conducted knockout and over-expression experiments for all other regulatory proteins, E2F, c-Myc, p53, STAT3 and NFKB. Table 4.2 shows the AMTP-values for all single knockouts and over-expressions of regulatory proteins and miRNAs. For E2F, c-Myc and NFKB, the knockout AMTP-value is higher than the over-expression AMTP-value, which does not at all agree with what can be observed for the miRNAs. That is, an absence of those three regulatory proteins leads to a more robust system than a constitutive presence. On the other hand, fixing p53 and STAT3 leads to similar effects on the robustness as fixing miRNAs.

Comparing the Derrida plots (Figure 4.8) shows that only p53 plays a similar role in conferring robustness to the network as the miRNAs. The Derrida value for the p53-knockout network is much higher than the value for the p53-over-expression network, whereas networks, in which any of the four other regulatory proteins is fixed, do not show this miRNA-like behavior. Figure 4.9 exhibits that a p53 over-expression confers even more robustness to the network than the miRNAs, which agrees with the fact that miRNAs are known to be weak regulators.

To further validate the network and analyze the mechanisms by which miRNAs confer robustness, we conducted a rewiring experiment, described in Section 4.2.6. The results strongly support the idea that the specific way the microRNAs are built into the network maximizes their ability to confer robustness. Among 1000 different rewiring experiments, the original network had by far the lowest Derrida value for any size of perturbation after 10 time steps (Figure 4.10). That is, the proposed network is in the long run not as affected by perturbations as any of the randomly altered networks, and can thus maintain phenotypic stability in the face of intrinsic noise. Moreover, the original network is clearly in the ordered regime, whereas some rewired networks operate in the critical, and some even in the chaotic regime. Note the interesting gap in the figure between trajectories of networks operating in the chaotic and the non-chaotic regime.
Figure 4.8: This figure shows the Derrida values after a variable number of time steps for various networks, in which one regulatory protein is knocked out (dashed line) or over-expressed (solid line). The x-axis shows the number of time steps that both configurations being updated, while the y-axis exhibits the average size of a single perturbation x time steps later.

Figure 4.9: This figure shows the Derrida values after a variable number of time steps for various networks, in which p53 or a miRNA is knocked out (dashed line) or over-expressed (solid line). The x-axis shows the number of time steps that both configurations being updated, while the y-axis exhibits the average size of a single perturbation x time steps later.
Figure 4.10: The Derrida plot after 10 time steps for 1000 randomly rewired networks (blue lines), and the original network (black line) for comparison. The red line $y = x$ displays the border between ordered and chaotic regime for small Hamming distances.

Furthermore, the original network has the 53rd lowest AMTP-value compared with 1000 values for the rewired networks. This result is not as strong as the Derrida values but it becomes stronger when looking at the maximum basin size of the original network. This measure for the original network is about in the middle of the total ranking but it is known to be highly correlated to the Average Maximal Transition Probability. Thus, it is indeed surprising that only about 5% of all rewired networks have lower AMTP-values while about 50% have a larger maximal basin of attraction than the original network. One possible reason that the original network had high AMTP-values could be that it had a relatively low number of attractors. Thus, one way, in which microRNAs confer robustness, is possibly through the reduction of steady states, which correspond to biological phenotypes.
4.4 Conclusion

We introduced a first regulatory network for the DNA Mismatch Repair machinery, and provided computational evidence that microRNAs confer robustness to this network using the well-established measure of Derrida values as well as the AMTP-values. We also showed that the specific topology of the network locally maximizes robustness by comparing it to randomly generated rewired networks. The model yields experimentally verifiable predictions about MMR’s response to hypoxia and UV, and enables further analysis of the stabilizing effect of microRNAs on DNA mismatch repair and gene regulatory networks in general.
Chapter 5

Evolutionary Robustness of Gene Regulatory Networks

Contributions: This chapter describes my latest research, which was supported by biweekly discussions with Seda Arat, Dr. Chris Heinen and Dr. Reinhard Laubenbacher.

5.1 Introduction

DNA mutations are very rare events. Nevertheless, they have influential implications by irreversibly altering the way a cell behaves. On the one hand, mutations can lead to changes in the structure of encoded proteins, which is the cause for various diseases. Sickle cell anemia, for instance, is caused by a single nucleotide change of the \( \beta \)-globin gene [62]. On the other hand, mutations can also lead to changes in the transcriptional regulation of a gene. Transcription factors regulate certain genes by attaching to matching binding sites upstream of the protein-coding region of a gene. Mutations can mainly be acquired in two ways. DNA can get damaged (1) by environmental factors like UV radiation, chemicals and viruses or (2) through randomly occurring errors in the DNA replication process. Cellular repair mechanisms like DNA mismatch repair (MMR) detect and fix the majority of these errors. Undetected errors lead however to mutations. This chapter focuses on the second kind of mutation acquisition and presents a modeling framework, which can be used to study the long-term effect of certain stimuli on a cell, whose DNA is subject to mutation.

The effect of mutations on gene regulatory networks has already been investigated [129]. To our knowledge, the framework presented in this chapter is however novel in the way genes and mutations are modeled. It seems closer to biology as the regulatory region of genes is represented by an actual DNA strand, which undergoes point mutations.

Robustness is the biological phenomenon of maintaining persistent phenotypes in the face of
mutations and cellular noise. In previous chapters we focused on the intrinsic cellular noise and showed that microRNA-mediated feed forward loops allow gene regulatory networks to better maintain their phenotypes. Genes are constantly subject to mutations and although a mutation rate of $10^9 - 10^{-10}$ per nucleotide and per replication seems negligibly small, these random nucleotide mismatches can have large impacts on the structure and stability of gene regulatory networks on an evolutionary time scale. This motivates us to investigate the evolutionary robustness of a gene network. In both prokaryotes and eukaryotes, sequence-specific transcription factors are considered the most important and diverse mechanism of cellular gene regulation [112]. We therefore study a network of transcription factors, in which regulations occur if the sequence of a gene’s binding site is recognized by a transcription factor DNA binding domain. The nucleotides in the binding sites undergo mutations and we describe topological features that allow the cell to maintain a stable phenotype on an evolutionary time-scale despite the mutations.

5.2 Methods

5.2.1 Modeling Framework

The fate of any cell depends on its genes and their expressions. Genes interact with each other to orchestrate a cellular program, thereby forming GRNs, which can be modeled using the SDDS framework (see Section 1.2.2 for details). It has been shown that this framework is an appropriate set-up to model the effect of intrinsic noise on the network dynamics. As the focus of this chapter is evolutionary robustness, we model each gene network as dynamically deterministic. That is, we set the propensity parameters for all gene to one.

We use Boolean threshold functions as update rules. This class of functions is well-suited to model transcriptional regulation because almost all transcription factors either activate or inhibit transcription, which can be directly represented in threshold functions. As in [91, 36], we define the dynamics of the network by

$$x_i(t + 1) = f_i(x_{i1}(t), \ldots, x_{ik}(t)) = \begin{cases} 1 & \text{if } \sum_{j=1}^{k_i} a_{ij} x_{ij}(t) > 0 \\ b & \text{if } \sum_{j=1}^{k_i} a_{ij} x_{ij}(t) = 0 \\ 0 & \text{if } \sum_{j=1}^{k_i} a_{ij} x_{ij}(t) < 0, \end{cases}$$

(5.2.1)

where $\{x_{i1}, \ldots, x_{ik}\}$ are the $k_i$ regulators of $x_i$. In general, if the strength of expressed activators equals the strength of expressed inhibitors, the concentration level of a gene remains at the previous level, that is, we set $b = x_i(t)$. Only if all regulators are activators (i.e., if $a_{ij} > 0$ for all $j = 1, \ldots, k$), we make an exception to this rule and set $b = 0$, and if all regulators are inhibitors (i.e., if $a_{ij} < 0$ for all $j = 1, \ldots, k$), we set $b = 1$.

Moreover, we restrict $a_i \in \{-1, 1\}$ for all $i = 1, \ldots, k$, which means that all transcription factors have equal strength. Thus, for the case of $k = 2$ regulators, only the three functions shown in Table 5.1 are possible update rules.
5.2.2 Modeling Mutations

To model the evolution of a gene regulatory network, we make the following assumptions.

(1) Each gene consists of a coding region and an upstream regulatory region. The coding region determines the sequence and structure of the encoded protein, which may act as a transcription factor for other genes. The sequence of the regulatory region decides which transcription factors can bind to it and thereby activate or inhibit transcription of the downstream gene.

(2) The coding region of each gene contains the blueprint for a single protein (transcription factor, TF). This TF contains a single DNA binding domain (DBD), which enables the transcription factor to bind to certain nucleotide sequences.

(3) The regulatory region of each gene contains $b$ TF binding sites (BS) of length $L$. We model each binding site as a random sequence of the four nucleotides A,C,G,T.

(4) TFs can often bind to DNA even if the match between their DBD and a BS is not perfect. When modeling specific genes, known sequence logos could be used to decide whether a TF can bind to a certain BS. In a general study like this, we assume that the DBD can only bind to the BS if at least $L_b$ of the $L$ nucleotides match complementarily ($L_b \leq L$).

(5) Gene X regulates Gene Y if and only if X’s TF DBD can bind to at least one of Y’s binding sites.

Assumption (5) implies that knowledge of the binding sites and DBDs of all genes of interest suffices to obtain the network topology. The following assumptions describe how we model single nucleotide mutations and how they can lead to changes in the topology. Let $N = nbL$ be the total number of nucleotides under consideration in all of the genes’ BSs.

(6) Every DNA replication, any nucleotide in the regulatory region mutates with probability $p$. We call $p$ the mutation rate. If cellular repair mechanisms like DNA mismatch

<table>
<thead>
<tr>
<th>$A_1(t)$</th>
<th>$A_2(t)$</th>
<th>$x(t+1)$</th>
<th>$I_1(t)$</th>
<th>$I_2(t)$</th>
<th>$x(t+1)$</th>
<th>$A_1(t)$</th>
<th>$I_1(t)$</th>
<th>$x(t+1)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>x(t)</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>x(t)</td>
</tr>
</tbody>
</table>

Table 5.1: There are only three possible update functions for the case $k = 2$ regulators. The left table depicts the truth table of the update function if both regulators are activators, the middle table is for the case of two inhibitors, and the right table is for the case of one activator and one inhibitor.
Claus T. Kadelka Chapter 5 64

repair are working, $p \approx 10^{-10}$, otherwise $p \approx 10^{-8}$ [105]. These low mutation rates imply that it is highly unlikely that two nucleotides mutate in the same replication step,

$$\mathbb{P}(1 \text{ mutation}) = Np(1-p)^{N-1} \approx Np \gg N(N-1)p^2 \approx N(N-1)p(1-p)^{N-2} = \mathbb{P}(2 \text{ mutations}).$$

For computational reasons, we therefore neglect this possibility and assume that during every replication, none or only one of the considered nucleotides change.

(7) After a nucleotide change happens, TF X may no longer be able to bind to one of Gene Y’s binding sites. Gene X is then dropped from the list of the regulators of Gene Y. On the other hand, new regulations may occur if TF X can now bind to at least one of Y’s binding sites. Gene X is then added to the list of regulators of Gene Y.

(8) If a nucleotide mutates, each change is equally likely. For instance, if C mutates, it becomes $A, G, T$ with probability $1/3$.

(9) The average number of nucleotide matches between a DBD and a BS of length $L$ is $L/4$. As long as the required number of matches for a regulation $L_b$ is close to $L$, it is therefore highly unlikely that a regulation occurs [19]. To keep the number of regulations in the gene network at a constant average level, we assume that all nucleotides in binding sites that currently match a DBD are $s$ times less likely to mutate. We call $s$ the selective advantage constant,

$$s = \frac{x}{1 - x} \cdot \frac{B_1(L_b, L)}{B_2(L_b, L)}$$

(5.2.2)

where $B_1(L_b, L) = \sum_{i=0}^{L_b-1} \prod_{j=i}^{L-1} \frac{3(j + 1)}{L - j}$, $B_2(L_b, L) = 1 + \sum_{i=L_b}^{L-1} \prod_{j=i}^{L-1} \frac{3(j + 1)}{L - j}$, and $x$ is the desired number matches between DBDs and BSs over the total number of possible matches. See Section 5.2.4 for details of the calculation.

Biologically, this assumption makes sense because a loss of a transcriptional regulation is mostly deleterious and often causes the cell to die. From a population point of view, this assumption means that only 1 out of $s$ cells survives a loss of a transcriptional regulation.

(10) DNA replications happen on a much slower time-scale than gene-gene interactions. We therefore assume that the gene regulatory network always reaches an attractor before the next DNA replication and possible change in network topology occurs.

### 5.2.3 Refinements of the Model

In this study, we present and analyze a model based on the simplistic assumptions described in the previous subsection. Various refinements could be made, e.g.
(3a) Each gene can have a different number $b$ of BSs.

(3b) The various binding sites can also have different lengths.

(3c) The relative proportion of C and G, the so-called GC-content, varies strongly among different organisms and even among different regions of the same organism \[44, 46\]. Instead of assuming equal occurrence of all four nucleotides, this information can be used to model the initial composition of the binding sites more accurately.

(8a) Phylogenetic research has indicated that some nucleotide changes are more likely to occur than others \[39\]. CpG sites (sites where C is directly followed by a G) are particularly prone to DNA damage for various reasons (methylated state of many cytosines in CpG dinucleotides, high susceptibility of guanine to transversions (changes among purines (A and G) or pyrimidines (C and T)) because of oxygen radicals, etc.). Comparing the number of nucleotide differences per position in the genome of humans and chimpanzees leads to the identification of two main trends.

1. *Transversions* are more frequent than *transitions* (changes from purines to pyrimidines, and vice versa).
2. CpG sites are much more likely to mutate than other sites.

Based on these findings, each possible nucleotide change in any of the binding sites is assigned a certain positive probability, and the nucleotide change is chosen based on this probability.

### 5.2.4 Selective Advantage Constant

The selective advantage constant $s$ from Assumption (9) is calculated in a way that the overall number of matches between TF DBDs and DNA BSs remains at a constant average level over time.

Consider one DBD and one BS, and let $M(t) \ (0 \leq M(t) \leq L)$ be the number of nucleotides, at which the DBD matches the BS at time $t$. Let us further assume that at every time step exactly one of the $L$ nucleotides in the BS mutates with equal probability. Then,

$$
\mathbb{P}(M(t+1) = i-1 | M(t) = i) = \frac{i}{L}, \quad \forall i = 1, \ldots, L
$$

$$
\mathbb{P}(M(t+1) = i | M(t) = i) = \frac{2L-i}{3L}, \quad \forall i = 0, \ldots, L
$$

$$
\mathbb{P}(M(t+1) = i+1 | M(t) = i) = \frac{1L-i}{3L}, \quad \forall i = 0, \ldots, L-1
$$

$$
\mathbb{P}(M(t+1) = j | M(t) = i) = 0, \quad \forall i, j = 0, \ldots, L \text{ with } |i - j| > 1
$$

Let

$$
T(i, j) = \mathbb{P}(M(t+1) = j | M(t) = i) \text{ for } 0 \leq i, j \leq L.
$$
Then $T$ is the transition matrix of an irreducible and aperiodic Markov chain so that any starting distribution eventually converges to a unique stationary distribution $\pi = (\pi_0, \ldots, \pi_L)$, for which $\pi T = \pi$. This stationary distribution can be calculated using

$$\pi_0 + \ldots + \pi_L = 1 \quad (5.2.3)$$

and the detailed balance equations

$$T_{i,i+1}\pi_i = T_{i+1,i}\pi_{i+1}, \quad i = 0, \ldots, L - 1. \quad (5.2.4)$$

In this study, we assumed (Assumption 4) that a TF can regulate a gene if its DBD matches at least $L_b$ of the $L$ nucleotides of one of the gene’s BSs. Therefore, the probability $x$ that a random DBD matches a random BS in the long-run is

$$x = \pi_{L_b} + \ldots + \pi_L.$$ 

To keep the number of regulations at a constant average level, we want $x$ to equal the ratio of the desired number of matches between DBDs and BSs over the total number of possible matches, and we assume that a nucleotide in a BS that currently matches a DBD is $s$ times less likely to mutate. That means,

$$\mathbb{P}\left(M(t+1) = i - 1 \mid M(t) = i\right) = \frac{1}{s} \frac{i}{L}, \quad \forall i = L_b, \ldots, L$$

$$\mathbb{P}\left(M(t+1) = i \mid M(t) = i\right) = \frac{s - 1}{s} + \frac{1}{s} \frac{L - i}{L}, \quad \forall i = L_b, \ldots, L$$

$$\mathbb{P}\left(M(t+1) = i + 1 \mid M(t) = i\right) = \frac{1}{s} \frac{L - i}{L}, \quad \forall i = L_b, \ldots, L - 1$$

Using Equations 5.2.3 and 5.2.4 to calculate the stationary distribution of this altered Markov chain yields

$$1 = \pi_0 + \pi_1 + \ldots + \pi_L$$

$$1 = \left(\sum_{i=0}^{L-1} \prod_{j=0}^{L-1} \frac{T_{j+1,i}}{T_{j,j+1}}\right)\pi_L + \pi_L$$

$$1 = \left(\sum_{i=0}^{L_b} \prod_{j=0}^{L-1} \frac{T_{j+1,i}}{T_{j,j+1}} + \sum_{i=L_b}^{L-1} \prod_{j=0}^{L-1} \frac{T_{j+1,i}}{T_{j,j+1}} + 1\right)\pi_L$$

$$1 = \left(\frac{1}{s} \sum_{i=0}^{L_b} \prod_{j=0}^{L-1} \frac{3(j + 1)}{L - j} + \sum_{i=L_b}^{L-1} \prod_{j=0}^{L-1} \frac{3(j + 1)}{L - j} + 1\right)\pi_L$$

$$1 = \left(\frac{1}{s} B_1(L_b, L) + B_2(L_b, L)\right)\pi_L$$

$$s = \frac{B_1(L_b, L)\pi_L}{1 - B_2(L_b, L)\pi_L}, \quad (5.2.5)$$
Figure 5.1: For different choices of $L_b$ and $L$, the selective advantage constant $s$ is plotted against the desired long-run proportion of random matches $x$.

where

$$B_1(L_b, L) = \sum_{i=0}^{L_b-1} \prod_{j=i}^{L-1} \frac{3(j + 1)}{L - j}, \quad B_2(L_b, L) = 1 + \sum_{i=L_b}^{L-1} \prod_{j=i}^{L-1} \frac{3(j + 1)}{L - j}.\$$

Plugging

$$x = \pi_{L_b} + \ldots + \pi_L = B_2(L_b, L)\pi_L$$

into Equation 5.2.5 yields that

$$s = \frac{x}{1 - x} \cdot \frac{B_1(L_b, L)}{B_2(L_b, L)}. \quad (5.2.6)$$

5.2.5 Quantifying Evolutionary Robustness

A cell constantly has to maintain a stable phenotype in the face of various perturbations. An examination of the mechanisms that confer robustness to a cellular program requires rigorously defined measures. In the previous chapters, we have used the Average Maximal Transition Probability and the Derrida values as measures for the robustness. Here, we are primarily interested in the evolutionary robustness of gene networks, which is why we need to define a new measure.
The general idea is to define evolutionary robustness of a network as the average number of nucleotide changes required before a network phenotype is lost. In an evolutionarily stable network, many mutations can accumulate before the phenotype changes, whereas a single nucleotide change may lead to a loss of phenotype in an unstable network. Let \( x_0 \in \{0, 1\}^n \) be a randomly chosen initial network configuration, and let \( F = F_0 = (f_i^1, p_i^1, p_i^2)_{i=1}^n \) be the original SDDS. Due to the finite number of configurations in the state space, \( x_0 \) will eventually transition to an attractor, \( A(F, x_0) \), which can be a steady state or a limit cycle. As before, this attractor corresponds to a cellular phenotype \[76, 78\]. Furthermore, let \( F_t \) be the evolved SDDS after \( t \) nucleotide changes occurred, and let \( A(F_t, x_0) \) denote the attractor to which \( x_0 \) transitions in the evolved SDDS. Then, the evolutionary robustness \( \mu_R \) of the gene regulatory network \( F \) is

\[
\mu_R(F) = \frac{1}{2^n} \sum_{x_0 \in \{0, 1\}^n} \mathbb{E}\left[ \min_{t > 0} |A(F_t, x_0) \cap A(F, x_0) = \emptyset| \right]
\]

That is, we define the evolutionary robustness of a network as the average number of nucleotide changes required until an initial configuration transitions to a different attractor, where we call two attractors different if they do not share any configuration. A very similar definition of mutational robustness has already been successfully used to study the evolvability of robustness \[29, 30\].

### 5.3 Results

The goal of this study was to understand the effect of various network properties on the evolutionary robustness of a GRN. We therefore generated more than 2000 networks of sizes 10, 30, 50 and 100. A network is strongly connected if there exists a path from every node to every other node. Strongly connected networks are therefore the only networks in which all genes regulate each other, at least indirectly. Since we wanted to investigate the influence of network size on evolutionary robustness, we ensured that every initial network was strongly-connected. Large-scale studies of \( B. subtilis, E. coli \) and \( S. cerevisiae \) strongly suggest that the average number of regulators is about 2 \[5\]. We therefore assumed that each gene was initially regulated by 2 other genes. The length \( L \) of TF binding sites in eukaryotes varies between 5-30 nucleotides. One study reports a peak value of seven and a mean value of ten \[124\], while another study reports a peak value of ten and a prevalence of even numbers \[69\]. For simplicity, we assumed that each DNA binding site has length \( L = 10 \). Moreover, we fixed the number of binding sites to be \( b = 10 \). A visual analysis of eukaryotic sequence logos indicated that on average 80 – 90% of the nucleotides need to match for successful TF binding to DNA. Thus, if \( L = 10, L_6 = 8 \) or \( L_6 = 9 \) seem to be suited values for the number of complementary nucleotide matches required between a TF DBD and a BS for binding to happen.

For every initial network \( F \), we used 1000 randomly chosen pairs of configurations with
Figure 5.2: This plot shows the average Pearson correlation between various network properties for $n = 10, L_b = 8$. A red value represents positive correlation, blue stands for negative correlation and green means no correlation.
Hamming distance 1 to estimate the Derrida value $D(1)$ of the network for the smallest possible perturbation (see Section 1.3.1 for details). We sampled the attractor landscape by letting 1000 randomly chosen configurations transition to their attractors. This provided a lower bound for the number of attractors and an approximation for the size of the largest basin of attraction. Sampling the attractor landscape also allowed one to estimate another commonly used dynamical robustness measure, the entropy of the network. Let $A(F)$ be the set of all attractors of $F$. Then, the entropy is defined as

$$\text{entropy}(F) = - \sum_{A \in A(F)} p(A) \ln[p(A)] \in (0, \infty),$$

where $p(A)$ is the normalized basin size of attractor $A$, i.e., $p(A)$ is the probability that a randomly chosen configuration eventually reaches attractor $A$. Each gene can either act as an activator or an inhibitor. We recorded the number of activators in the network. Lastly, we calculated the number of different types of FFLs.

For every initial network $F = F_0$, we randomly picked 100 initial configurations, and for each of these configurations, we let the network evolve according to the assumptions from Section 5.2.2. Once the original phenotype was lost, we stopped the evolution process. For each of the 100 final networks, we recorded the number of nucleotide changes that happened before the phenotype was lost, as well as the number of edges that were added or deleted during the evolution process. The former served as an estimate for the evolutionary robustness of the initial network defined in Section 5.2.2. Like for the initial network, we also estimated the (1) number of attractors, (2) largest basin size of an attractor, (3) entropy, (4) Derrida value for a perturbation of size 1, and (5) number of different types of FFLs. In addition, we calculated the number of strongly connected components as the evolved network may have lost its strong connectedness. Lastly, we recorded the average number of regulations per gene.

As a first step towards finding causes of evolutionary robustness, we looked at the pairwise Pearson correlation of the different recorded measurements. Figure 5.2 shows all pairwise correlations for a network of $n = 10$ genes with $L_b = 8$. Although most measurements are only approximations, this figure clearly exhibits general relationships. For instance, it is not surprising that the number of nucleotide changes and the number of edge changes (until a phenotype is lost) are highly correlated as both measurements depend directly on the duration of the evolution process. Also, the strong positive correlations between initial and final Derrida value and initial and final number of FFLs are expected since most of the time only a few regulations change before the phenotype is lost (see Figure 5.3I) so that the Derrida value and the number of FFLs does not change much. More surprisingly, the number of attractors in the original network is positively correlated with the Derrida value of the final network, and the number of FFLs in the original network is positively correlated with the number of components in the final network. Unaware of any obvious explanation, we looked at the development of these correlations when the network size increases (see Figure 5.4 and Figure A.1). Several correlations vanish in larger networks, however the correlation between the evolutionary robustness (i.e., the number of nucleotide changes until the phenotype is lost) and the average connectivity becomes stronger.
Figure 5.3: Empirical distribution of several measurements for $L_b = 8$ and different network sizes.
5.4 Discussion

This study assumes that one nucleotide mutates every "slow" time step. In reality, the duration between mutation events is, however, not constant. It depends on the number of nucleotides and the functionality of repair mechanism. Let $N$ be the overall number of nucleotides that can undergo mutation. In the simplest version of the model, $N = nbL$. Then, the probability that a mutation happens in a given replication step is $q = 1 - (1 - p)^N$, and the number of replications until the next mutation occurs is a geometrically distributed random variable with success probability $q$. This enables us to link the abstract notion of "slow" time steps to real time. For every "slow" time step, we can simulate the number of replications that the cell has undergone since the beginning. Being able to link the abstract time clock of the framework to real time allows a proper interpretation of the model results. Investigating after how many replication steps a particular phenotype is on average gained or lost, and comparing this for different external conditions is only one of many interesting questions, for which the model can generate predictions.

The developed framework is particularly useful when modeling GRNs that include a node, which describes the activity of the repair machinery. The node 'MMR' in the GRN of DNA mismatch repair, developed in Chapter 4, is such a node. It takes on the value one if all essential MMR proteins are present (the MMR machinery is then active), otherwise it is zero. The replicative error rate, $p$, in such a model is no longer constant but it changes roughly between $10^{-9} - 10^{-10}$ (if the MMR machinery is active) and $10^{-7}$ (if it is inactive) [65]. The
number of replications until the next mutation event (and thereby the passed real time from the beginning) now actively depends on the current phenotype of the model.

In this study, we used correlation coefficients to describe relationships between different properties. It is a common misconception to use correlation coefficients as a predictor of a causal connection. If two properties, X and Y, are correlated, it could be that X causes Y, or vice versa that Y is responsible for X. It could however also be that both A and B are caused by another factor Z, or that they have completely independent causes that just happen to run in parallel. The described results are thus only a first step towards finding properties that cause a gene regulatory network to be evolutionary robust.

The ideas described in this chapter shall be seen as a starting point for further studies. In Section 5.2.3 we described possible minor generalizations of the assumptions from Section 5.2.2. One could extend this framework even more, e.g. by allowing for insertion or deletion of one or more nucleotides or by modeling random gene duplication and divergence of the duplicated gene. Particularly the inclusion of gene duplication could prove helpful as this mechanism is considered one of the main drivers of genome growth and evolution of new phenotypes [96, 126, 136].
Chapter 6

Concise Functional Enrichment of Ranked Gene Lists (CRFE)

To be submitted for publication as: C Kadelka, M Brandon, TM Murali. Concise Functional Enrichment of Ranked Gene Lists.

Contributions: This project started as a class project in Computational Systems Biology, taught by T. M. Murali. Through discussions, Madison Brandon and I developed the key ideas of this chapter together. I wrote almost all the code, created the web application for online use of our method, and was the primary author of Subsections 6.2.2 6.2.3 & 6.3.1 and Section 6.4. Subsection 6.3.2 is mainly the work of Madison Brandon. I included it for conclusiveness.

Abstract

The integration of large scale biology and technology has led to an explosion of genome-wide expression data. Given a set of differentially-expressed genes as input, functional enrichment methods compute a set of functional categories that annotate a surprisingly large number of the genes. Many functional enrichment methods work in a binary mode, i.e., they consider a gene to be either expressed or not. These approaches may disregard useful information provided by the ranking. The few enrichment methods that do operate on ranked lists output highly redundant or non-specific functional categories.

We present a new functional enrichment algorithm, called Concise Ranked Functional Enrichment (CRFE), that effectively utilizes the ranking in gene expression data sets and computes a small set of non-redundant functional categories that are significantly over-represented in highly ranked genes. We apply CRFE and three existing functional enrichment methods to two treatment-control microarray data sets and compare their performance based on utiliza-
tion of ranking information, conciseness and specificity. CRFE performs well in all criteria, whereas every other method performs poorly in at least one. Lastly, we report a high-level interpretation of the functional categories returned by CRFE for each data set.

The CRFE source code is freely available from www.math.vt.edu/people/claus89/CRFE/

6.1 Introduction

Over the past two decades technological advances, such as the advent of the microarray chip and next-generation sequencing, have heralded a new era of biomedical research. Scientists can now routinely perform high throughput profiling studies at relatively low cost. The integration of large scale biology and technology has led to an explosion of genome-wide expression data. These “-omics” data sets often take the form of long lists of genes, which can be ranked by expression, confidence, or some other attribute of the data.

Functional enrichment methods take as input a list of thousands of genes and a list of annotations describing which genes are involved in which functional categories. The output is a list of functional categories that annotate a surprisingly large number of genes in the input list [2, 16, 40, 50, 95, 98, 113]. An essential enabling tool of these algorithms is an annotation database such as the Gene Ontology (GO) [8]. The GO is a controlled vocabulary of cellular components, molecular functions and biological processes, which can be represented as a directed acyclic graph where parent nodes describe more general terms and child nodes describe more specific terms. Every term in the GO is annotated to a particular set of genes, and parent nodes inherit all gene annotations from their children.

The output of microarray or RNA-seq experiments is a list of genes, often ranked by differential expression. Many functional enrichment methods work in a binary mode [2, 16, 50, 95, 98]. MGSA [16], for instance, considers all genes above (respectively below) a certain differential expression cutoff as active (respectively inactive). This approach disregards information provided by the ranking, which could be used to obtain a set of functional categories that annotate significantly many of the highly ranked genes.

There are several enrichment methods that do operate on ranked lists [17, 40, 113]. However, all these methods share a common limitation. They consider the functional categories individually. This approach cannot account for potential overlap among categories, which arises from the hierarchal structure of annotation databases like the GO. Thus, the set of returned categories may be highly redundant, i.e., it may include many categories which annotate very similar sets of genes. Selecting a meaningful and non-redundant subset of categories can be very difficult.

By calculating the likelihood of enrichment for sets of functional categories rather than one category at a time, MGSA addresses the problem of redundancy, thereby returning a small set of meaningful functional categories. However, MGSA does not use the valuable information
available in the ranking of the genes. Consequently, an enriched functional category may annotate both highly-ranked and poorly-ranked genes. We present a new method, Concise Ranked Functional Enrichment (CRFE), which is a generalization of MGSA that operates with fully ranked lists and computes a concise (non-redundant) set of specific functional categories. These categories preferentially annotate highly ranked genes.

To address the challenge posed by ranked lists, we develop a generative model that explains why one may observe a ranking of genes. Specifically, we design an appropriate representation of noise sources that lead to erroneous observations in the data. In order for CRFE to find functional categories that are preferentially over-represented by highly ranked genes, we introduce false positive and false negative rates that scale with the ranking.

A functional enrichment method for fully ranked lists should satisfy three criteria. First, it should compute categories that preferentially annotate highly ranked genes. Second, it should return a concise set of categories that do not share many gene annotations. Third, since categories with few annotations are more clearly defined and hence more informative, it should prefer specific categories over those with many gene annotations.

We apply CRFE to two treatment-control microarray data sets and compare its performance to MGSA (as one of the best methods operating in binary mode), FuncAssociate2.0 [17] and GOrilla [40] (two methods that use the ranking). We introduce an intuitive measure for the quality of a set of functional categories, and show that CRFE computes sets of functional categories of overall higher quality than MGSA, FuncAssociate2.0, or GOrilla. Furthermore, categories obtained by CRFE share few gene annotations, while FuncAssociate2.0 and GOrilla return many categories with overlapping annotations. Finally, we show that CRFE
returns specific categories, and therefore satisfies all three criteria for a good functional enrichment method. In addition to benchmarking experiments, we demonstrate the usefulness of our method by providing a biological interpretation of the functional categories computed for each data set.

6.2 Methods

6.2.1 Generative Model

We developed a generative model that describes how a genome-wide experiment can yield a list of genes, ranked by some measured value. For ease of exposition, we describe the model in the setting of a case-control DNA microarray experiment, which ranks the genes in order of statistical significance of differential expression. We model the response of genes in the experiment as the result of the perturbation of several functional categories. In a departure from previous approaches [16, 95], we allow for the possibility that the experiment may perturb the categories to different extents, leading to a ranking among the genes.

We assume that the experiment seeks to detect gene states based on differential expression, and that a gene can be in one of multiple states, e.g., “very perturbed,” “perturbed,” and “unchanged.” In the extreme, each gene can have its own state. The true state of every gene is hidden. We model the results of the experiment by assuming that, in the absence of any noise, every gene that is not in the “unchanged” state is annotated to one of the perturbed functional categories. However, since experimental data can contain errors, we further assume that unknown false positive and false negative rates determine the measured ranking of the genes. For example, genes annotated to the most (respectively, less) perturbed categories are likely to appear toward the top of (respectively, further down in) the list of differentially-expressed genes.

We now describe the generative model formally (see Figure 6.1 for an illustration). We use biological processes in the Gene Ontology to represent functional categories. Let \( T \) be the set of all these processes. We use two sets of parameters, \( \alpha \)'s and \( \beta \)'s, corresponding to false positive and false negative rates. An additional parameter \( q \) describes the probability that the cell activates any given process. Our model will yield two ranked lists \( \tilde{P} \) and \( \tilde{U} \), whose concatenation is the final ranking of genes. The model has the following elements.

1. (Active processes) The cell activates each process in \( T \) with probability \( q \). The activated processes form a ranked list \( C = (c_1, c_2, \ldots, c_m) \) of \( m \) processes in \( T \), where \( c_i \) is more activated than \( c_{i+1} \), \( 1 \leq i < m \). Ideally, the list \( \tilde{P} \) will contain all the genes annotated to at least one of the processes in \( C \) and \( \tilde{U} \) will contain the remaining genes. The remaining steps in the model partition \( \tilde{P} \) into \( m \) sub-lists \( \tilde{P}_i \), \( 1 \leq i \leq m \).

2. (False negatives) For each \( i, 1 \leq i \leq m \), we assume that every gene annotated to
process \( c_i \) is in the list \( \tilde{P}_i \) with probability \( 1 - \beta_i \) and in the list \( \tilde{U} \) with probability \( \beta_i \). Here, \( \beta_i \leq \beta_{i+1} \), for every \( 1 \leq i < m \) so that a gene annotated to \( c_i \) is less likely to be a member of \( \tilde{U} \) than a gene annotated to \( c_{i+1} \). If a gene has multiple annotations, we consider only the annotation to the most activated process in \( C \).

3. (False positives) Every gene not annotated to any of the \( m \) processes is in \( \tilde{P} \) with probability \( \alpha \), the false positive rate, and to be in \( \tilde{U} \) with probability \( 1 - \alpha \). If such a gene is in \( \tilde{P} \), the probability that it is in the list \( \tilde{P}_i \) is proportional to \( \alpha_i \), where \( \alpha_i \leq \alpha_{i+1}, 1 \leq i < m \).

4. (Ranking) We assume that every gene in \( \tilde{P}_i \) has a lower (numerical) rank than every gene in \( \tilde{P}_{i+1} \), for every \( 1 \leq i < m \). To generate a ranking of all the genes, we assume for every \( 1 \leq i \leq m \) that the genes in \( \tilde{P}_i \) can appear in any order. We also assume that the genes in \( \tilde{U} \) can appear in any order.

For instance, as per our model, a gene annotated to \( c_1 \), the most perturbed process, will appear in the set \( \tilde{U} \) of unperturbed genes with a (small) probability of \( \beta_1 \). Moreover, a gene that is truly unperturbed will appear among the perturbed genes annotated to \( c_1 \) in the ranked list with a (small) probability proportional to \( \alpha_1 \).

### 6.2.2 Algorithm

CRFE takes as input a ranked list of genes, a ranking cutoff, and a belief parameter, which we formally define later. We first use the cutoff to divide the list into a ranked list \( P \) of perturbed genes and a list \( U \) of unperturbed genes. Since the true number of activated processes and their number of annotations are unknown, we partition \( P \) into \( k \) equally-sized subsets \( P_i \), where every gene in \( P_i \) has a lower (numerical) rank than every gene in \( P_{i+1} \). Through this partitioning, our algorithm exploits the ranking of genes in \( P \) by assigning specific error rates to each subset \( P_i \). In this chapter we use \( k = |P| \). We describe the algorithm, however, for any value of \( k \) to make the connection to MGSA \((k = 1)\) explicit.

Given a set of processes \( C \), let \( E(C) \) be the set of genes annotated to at least one of the processes in \( C \). We call elements in \( E(C) \) explained genes, and say that \( C \) explains the set of genes \( E(C) \). We can now define four different types of sets of genes:

(i) \( EP_i = E(C) \cap P_i, 1 \leq i \leq k \), the perturbed genes in subset \( P_i \) that are annotated to (one of the processes in) \( C \),

(ii) \( NP_i = P_i - E(C), 1 \leq i \leq k \), the perturbed genes in subset \( P_i \) that are not annotated to (any process in) \( C \),

(iii) \( EU = E(C) \cap U \), the set of unperturbed genes that are annotated to \( C \), and

(iv) \( NU = U - E(C) \), the set of unperturbed genes that are not annotated to \( C \).
Note that each of these sets is a function of the set \( C \). We omit \( C \) from the notation from now on to facilitate readability.

Using a “belief” parameter \( b \), we assign a specific false positive and false negative rate to each subset \( P_i \), \( 1 \leq i \leq k \) such that

\[
\text{(i)} \quad \alpha_k = b\alpha_1, \beta_k = b\beta_1
\]

\[
\text{(ii)} \quad \alpha_i, \beta_i \text{ increase linearly with } i, \text{ i.e.,}
\]

\[
\alpha_i = \frac{(k-i)\alpha_1 + (i-1)\alpha_k}{k-1} \quad \text{and} \quad \beta_i = \frac{(k-i)\beta_1 + (i-1)\beta_k}{k-1}
\]

These conditions lead to two separate linear systems of \( k-1 \) equations with \( k \) unknowns each, namely \((\alpha_1, \ldots, \alpha_k)\) and \((\beta_1, \ldots, \beta_k)\). The solutions to these systems are

\[
\alpha_i = \frac{(k-i)\alpha_1 + (i-1)\alpha_k}{\sum_{j=1}^{k}(k-j+(j-1)b)|P_j|}
\]

\[
\beta_i = \frac{(k-i)\beta_1 + (i-1)\beta_k}{\sum_{j=1}^{k}(k-j+(j-1)b)|P_j|}
\]

These solutions are unique up to a scaling factor \( \alpha \) (respectively, \( \beta \)), which corresponds to the average false positive (respectively, false negative) rate. When learning this scaling factor, the possible range needs to be restricted. A perturbed gene should never be penalized more than an unperturbed gene. Therefore, \( \beta_k \leq 1 - \alpha \) and \( \alpha_k \leq 1 - \beta \). These conditions imply that \( \alpha_k \leq 1 - \alpha \) and \( \beta_k \leq 1 - \beta \), and the maximal possible value for \( \alpha \) and \( \beta \), which we call \( r_{\text{max}} \), can be derived

\[
\alpha_k \leq 1 - \alpha \\
\frac{(k-1)b|P|}{\sigma} \leq 1 - \alpha \\
\alpha \left(1 + \frac{(k-1)b|P|}{\sigma}\right) \leq 1 \\
\alpha \leq \left(1 + \frac{(k-1)b|P|}{\sigma}\right)^{-1},
\]

where \( \sigma = \sum_{j=1}^{k}(k-j+(j-1)b)|P_j| \). Thus,

\[
r_{\text{max}} = \left(1 + \frac{(k-1)b|P|}{\sigma}\right)^{-1}.
\]

If each perturbed gene forms its own category (i.e., \( k = |P|, |P_j| = 1 \)), then \( r_{\text{max}} \) reduces to

\[
r_{\text{max}} = \frac{1+b}{1+3b} \in \left(\frac{1}{3}, \frac{1}{2}\right) \text{ for } b \geq 1.
\]
The belief parameter $b > 1$ acts as a scaling factor. A large value of $b$, for instance $b = 10$, puts emphasis on highly perturbed genes, whereas $b = 1$ assigns the same rates to all perturbed genes as in MGSA. The assignment of a greater belief parameter intuitively translates to an experimentalist having a greater confidence that the genes measured to be highly differentially expressed are the most relevant to the experiment.

Based on the generative model, the probability of a particular set $C$ of processes is given by

$$
P(C|\alpha, \beta, b, q) = \prod_{i=1}^{k} (1 - \beta_i)^{|EP_i|} \prod_{i=1}^{k} \alpha_i^{|NP_i|} \beta^{|EU|} (1 - \alpha)^{|NU|} q^{|C|} (1 - q)^{|T - C|} \quad (6.2.3)$$

Here, $q$ is a parameter that penalizes increases in the size of $C$.

Finding a set $C^*$ of biological processes that “best” explains the data now reduces to maximizing this probability function with respect to $C$ as well as $(\alpha, \beta, q)$,

$$C^* = \arg\max_{C \times (\alpha, \beta, q) \subseteq T \times \mathcal{P}} P(C|\alpha, \beta, b, q),$$

where $\mathcal{P}$ is the space that contains all allowed parameter configurations. The size of the space $T$ is already $2^{|T|}$, too large to search exhaustively. We thus employ a Markov Chain Monte Carlo (MCMC) approximation to estimate $C^*$ and the parameters $(\alpha, \beta, q)$. For convenience, we maximize the log-likelihood function $L(C|\alpha, \beta, b, q) = \log P(C|\alpha, \beta, b, q)$.

We initialize $C_0$ as the empty set. During each iteration $j \geq 0$, we create a proposal set $C_{j+1}$ by

1. adding an unselected process (from $T - C_j$) to the current set $C_j$,
2. deleting a process from the current set $C_j$, or
3. swapping a process in $C_j$ with an unselected process.

Note that there are $|T| + |C_j||T - C_j|$ possible proposals. We choose each proposal with equal probability. We accept the proposed change with probability

$$P(C_{j+1} = C_{j+1}') = \begin{cases} 1 & \text{if } L(C_{j+1}') \geq L(C_j) \\ e^{L(C_j') - L(C_j)} & \text{if } L(C_{j+1}') < L(C_j) \end{cases} \quad (6.2.4)$$

In other words, we always accept the proposal if the likelihood of the proposed set is higher. We may accept proposals with lower likelihood with a positive probability, which is inversely proportional to the difference in likelihoods. This possibility allows the search to escape local maxima. The designed Markov chain is finite (there are only $2^{|T|}$ states), irreducible (each state can be reached with positive probability from any other state within $|T|$ steps) and aperiodic (the chain has cycles of length one and two), which secures convergence to a stationary distribution.
In our implementation, after an initial burn-in period of $l_1 = 10^5$ iterations, we record the current set $C_j$ at every step, and stop the algorithm after additional $l_2 = 10^6$ iterations. We define the posterior probability of a term $t \in T$ as the fraction of recorded steps in which the current set $C_j$ contained the term,

$$P_{\text{posterior}}(t) = \frac{1}{l_2} \left| \{ j = 1, \ldots, l_2 : t \in C_j \} \right|$$  \hspace{1cm} (6.2.5)

All terms with a posterior probability above or at a user-defined cutoff $\tau \in (0, 1]$ form the returned explanatory set, ordered by their posterior probability.

During the MCMC approximation we can learn the model parameters, $\alpha$, $\beta$ and $q$ as well. Instead of always changing the current set $C$ of processes, with probability 0.2 we change the average false positive rate $\alpha$ or false negative rate $\beta$ by selecting from a pre-defined discretization of the interval $(0, r_{\text{max}}]$. In addition, whenever we add or delete a process from the current set $C$, we optimize $q$ using the estimate $\min(|C|/|T|, 20/|T|)$, as in MGSA. This upper limit on $q$ penalizes choices of $C$ with more than 20 processes. The posterior probability of a parameter ($\alpha$, $\beta$, or $q$) taking on a particular discrete value is defined as the fraction of recorded steps in which the parameter takes on this value.

### 6.2.3 Quality measures

Functional enrichment methods seek to find a set of biological processes that annotate as many perturbed genes as possible and as few unperturbed genes as possible. These two goals conflict with each other: an improvement in one of them usually leads to a deterioration in the other. To quantify this tradeoff, we define a measure called quality. Given a set $C$ of biological processes and a subset $X$ of the perturbed genes $P$, we define quality $Q(C, X)$ of $C$ with respect to $X$ as the ratio of (a) the proportion of genes in $X$ that are explained by $C$ and (b) the proportion of unperturbed genes that are explained by $C$, i.e.,

$$Q(C, X) = \frac{|E(C) \cap X|/|X|}{|EU|/|U|} = \frac{|E(C) \cap X||U|}{|X||EU|}$$  \hspace{1cm} (6.2.6)

By defining the quality for any subset of the perturbed genes, we can ask how it varies for different prefixes of the perturbed gene list.

### 6.2.4 Data sets

We applied our method to two treatment-control microarray data sets. The first data comparison the gene expression in *Rattus norvegicus* (rat) hepatocytes between two organotypic *in vitro* models of the liver [83], which we call “2-cell LM” and “3-cell LM”. Both models contained hepatocytes and liver sinusoidal endothelial cells (LSECs), separated from each other by a polyelectrolyte membrane. The “3-cell LM” contained Kupffer cells in addition
to the other two cell types. We used RMA to normalize the data across three replicates [63]. For each gene, we computed a signal-to-noise ratio that measured the difference between the gene’s expression in the 3-cell LM replicates and the 2-cell LM replicates. We converted the identifiers of the rat genes to human gene names.

The second dataset comes from a meta-analysis study of microarray experiments related to adenocarcinoma of lung reported in the National Center for Biotechnology Information (NCBI) Gene Expression Omnibus (GEO) database [38]. For each gene the study computed a $p$-value that reflected the statistical significance of the gene’s expression in the disease samples compared to healthy controls.

### 6.3 Results

#### 6.3.1 Comparison of CRFE to other Functional Enrichment Algorithms

We compared our approach CRFE to three popular functional enrichment methods: MGSA [16], FuncAssociate2.0 [17] and GOrilla [40]. Recall that MGSA is a model-based approach that considers the enrichment of sets of biological processes but does not use the ranking of the genes. On the other hand, FuncAssociate2.0 and GOrilla do use the ranking of genes but they evaluate the enrichment of processes on an individual basis. We based the comparison on the three criteria mentioned in the introduction: ranking, conciseness, and specificity.

As CRFE and MGSA are randomized algorithms, we averaged their results over 20 independent runs. For both data sets, we considered the top 30% of all measured genes as perturbed, and we set $k = |P|$, i.e., we put each perturbed gene in its own category. We used three values for the belief parameter $b = 2, 5, \text{and } 10$; a larger value puts more emphasis on highly perturbed genes. Lastly, we only considered biological processes that annotate between 20–200 measured genes. For GOrilla, we returned all processes with a $p$-value threshold of $10^{-3}$ (as the default choice in the GOrilla web application). For Funcassociate 2.0 we considered all returned terms (internal threshold values).

**Ranking.**

Since MGSA does not take the ranks of perturbed genes into account, we expected its quality to be independent of the choice of perturbed genes. To test this hypothesis and to compare all the algorithms, we varied both the number $r$ of processes and the percentage $p$ of top perturbed genes we considered. As we varied $r$, we plotted the quality against the proportion of top $p\%$ of the perturbed genes that were explained by the $r$ processes. For example, if $p$ was 50, we considered only the top half of the perturbed genes. Figures [6.2] & [6.3] show
Figure 6.2: Plot of the quality against the proportion of the highest $p\%$ perturbed genes that are explained as we increase the rank $r$. We show curves for three different values of $p$. We denote the points where the posterior probability of the process at rank $r$ is 0.75 (circle), 0.5 (diamond), 0.1 (square), and 0.01 (triangle).
Figure 6.3: Plot of the quality against the proportion of the highest $p\%$ perturbed genes that are explained as we increase the rank $r$. We show curves for three different values of $p$. The first row is for MGSA, the second for FuncAssociate2.0, and the third for GOrilla. For MGSA, we denote the points where the posterior probability of the process at rank $r$ is 0.75 (circle), 0.5 (diamond), 0.1 (square), and 0.01 (triangle).
the results for MGSA, CRFE, FuncAssociate2.0, and GOrilla, respectively. The quality of CRFE’s processes increased as we decreased $p$. We observed the same dependence on $p$ for FuncAssociate2.0 and GOrilla. However, for MGSA the quality did not seem to depend on the choice of $p$. These observations confirmed that methods that used the ranking of the genes (CRFE, FuncAssociate2.0 and GOrilla) paid greater attention to explaining highly ranked genes.

Furthermore, for every method, we observed larger variations in quality with change in $p$ for the liver models than we saw for adenocarcinoma of lung. This trend may be attributed to the fact that, for the liver models, the top 25% perturbed genes were on average annotated to 20.55 processes, while this number was 18.76 (respectively, 17.25) for the top 50% (respectively, all) perturbed genes. For adenocarcinoma of lung, we did not observe such a change in the average number of annotations (21.34, 20.07, and 20.25 for the top 25%, 50%, and all perturbed genes, respectively).

In addition, we compared the algorithms with each other for $p = 50\%$. As shown in Figure 6.4, CRFE consistently explained a larger proportion of the top 50% perturbed genes with a higher quality than MGSA. Both FuncAssociate and GOrilla achieved high quality scores for a small proportion of the top 50% perturbed genes but the quality of their explanatory sets quickly dropped as more of the perturbed genes were explained. The initial high quality values are not surprising since the $p$-value of the hypergeometric test used by both methods is likely to correlate highly with quality. These trends held when we instead considered the top 25% or all the perturbed genes (see Figure S3).

Figure 6.4 also shows that a higher belief parameter led to a higher quality score for smaller proportions of perturbed genes. Furthermore, a higher belief parameter yielded fewer pro-
cesses with comparatively lower posterior probabilities because the conditions imposed on acceptance of a process in the MCMC implementation became more restrictive. Therefore, the choice of the belief parameter should depend on what proportion of perturbed genes a user wants to explain with a high quality.

Conciseness.

A common problem of functional enrichment algorithms is a lack of conciseness in the explanatory set, i.e., multiple returned processes share many gene annotations. In these cases, many returned processes are very similar, making the interpretation of the results difficult.

To quantify the conciseness of a set of processes (returned by an algorithm), we define the redundancy of a process $t$ with respect to a set $C$ as

$$R(t, C) = \max_{c \in C, c \neq t} \left| \frac{|E(t) \cap E(c)|}{|E(t) \cup E(c)|} \right|,$$

where $E(t)$ is the set of genes annotated to $t$. The quantity $R(t, C)$ measures the largest overlap of $t$’s annotations with the annotations of any other process in $C$. Intuitively, a set of processes is concise if very few processes in that set are redundant.

Figure 6.5 displays the redundancy of the top 20 processes returned by the four methods. For both data sets, CRFE and MGSA return a much more concise set of processes than FuncAssociate2.0 and GOrilla. While the choice of top 20 processes is arbitrary, the trends did not change for other numbers (see Figures A.2 & A.3). The simultaneous optimization of sets of processes as well as the penalization of the size of the explanatory set allows CRFE and MGSA to avoid returning similar processes. FuncAssociate2.0 and GOrilla, on the other hand, consider a single process at a time.
Specificity.

A good functional enrichment method preferably returns specific processes, i.e., processes with few gene annotations. These processes represent biological phenomena which are more clearly defined and hence may be more informative for the user. Recall that we restricted the analysis to processes with 20–200 gene annotations, which have 61 annotations on average. For each method, we measured the average number of annotations to the top \( r \) ranked processes. As shown in Figure 6.6, CRFE and Funcassocate2.0 yielded more specific processes than expected, whereas MGSA and GOrilla returned processes that are as specific or less specific than expected. This result shows that CRFE does return a set of specific processes.

6.3.2 Interpretation of the Enriched Set of Processes

Liver models.

We present a high-level interpretation of the results of CRFE for both datasets. For the liver data, we only considered positively differentially expressed genes as perturbed in our analysis; therefore, each enriched GO term reported in Table 6.1 represents activation in the 3-cell LM as compared to the 2-cell LM. Not surprisingly, the set of enriched terms largely describes additional liver functionality mediated by Kupffer cells, the cell type included in the 3-cell, but not in the 2-cell model.

Kupffer cells are specialized, liver resident macrophages. A major function of these cells is to recognize and scavenge waste, which they then pass along to hepatocytes.
to degrade and clear as bile. Correspondingly, the term “Bile acid metabolic process”, an essential hepatocyte function, was enriched. Other important liver metabolic processes were also up-regulated in the 3-cell model, including those involving alditol phosphates and prostagladins, both of which are important for glucose metabolism in hepatocytes.

In addition to gains in metabolic functioning, several enriched GO terms suggest the added Kupffer cells are secreting pro-inflammatory and pro-growth factors. For instance, CRFE returns “Cytokine Production”, and in fact it has been shown that in response to stimulation by Kupffer cell-produced cytokines (e.g. TNF-α), hepatocytes secrete their own array of cytokines in order to amplify the inflammatory response [114]. Platelet-derived growth factor (PDGF) is an important growth factor regulating cell proliferation, differentiation, and growth. Enrichment of “Platelet-derived growth factor signaling pathway” may imply that Kupffer cells are secreting PDGF and eliciting a pro-growth response in neighboring hepatocytes. More generally, the top enriched term “Receptor-mediated endocytosis” refers to internalization of receptor-ligand complexes and indicates an overall increase in signaling through membrane receptors binding to external ligands such as PDGF.

<table>
<thead>
<tr>
<th>GO Biological Process Term</th>
<th>Number of Annotations</th>
<th>Posterior Probability</th>
</tr>
</thead>
<tbody>
<tr>
<td>receptor-mediated endocytosis</td>
<td>108</td>
<td>0.999</td>
</tr>
<tr>
<td>phagocytosis</td>
<td>62</td>
<td>0.991</td>
</tr>
<tr>
<td>amino sugar metabolic process</td>
<td>28</td>
<td>0.976</td>
</tr>
<tr>
<td>protein activation cascade</td>
<td>43</td>
<td>0.947</td>
</tr>
<tr>
<td>actin polymerization or depolymerization</td>
<td>25</td>
<td>0.946</td>
</tr>
<tr>
<td>alditol phosphate metabolic process</td>
<td>20</td>
<td>0.910</td>
</tr>
<tr>
<td>platelet-derived growth factor receptor signaling pathway</td>
<td>23</td>
<td>0.906</td>
</tr>
<tr>
<td>cytokine production</td>
<td>64</td>
<td>0.887</td>
</tr>
<tr>
<td>prostaglandin metabolic process and prostanoid metabolic process</td>
<td>21</td>
<td>0.870</td>
</tr>
<tr>
<td>dendrite morphogenesis</td>
<td>31</td>
<td>0.849</td>
</tr>
<tr>
<td>neural precursor cell proliferation</td>
<td>45</td>
<td>0.836</td>
</tr>
<tr>
<td>regulation of erythrocyte differentiation</td>
<td>21</td>
<td>0.820</td>
</tr>
<tr>
<td>actin cytoskeleton reorganization</td>
<td>32</td>
<td>0.806</td>
</tr>
<tr>
<td>sprouting angiogenesis</td>
<td>29</td>
<td>0.796</td>
</tr>
<tr>
<td>bile acid metabolic process</td>
<td>32</td>
<td>0.778</td>
</tr>
</tbody>
</table>

Table 6.1: The 15 processes with the highest posterior probability computed by CRFE with a belief parameter of 5.
The enriched term “Sprouting angiogenesis” implies that Kupffer cells may also be secreting pro-angiogenic factors which would in turn signal hepatocytes and LSECs to begin the sprouting angiogenesis process, a crucial step for liver regeneration post injury [130]. Likewise the term “Regulation of erythrocyte differentiation” suggests hepatocytes in the 3-cell model are modulating (likely increasing) the rate of red blood cell maturation more so than in the 2-cell model.

Curiously, one highly ranked term “Phagocytosis” is a well known function of Kupffer cells but is not typically associated with hepatocytes [86]. There is, however, evidence that hepatocytes have phagocytic capabilities [73, 120]. Particularly in response to liver damage, hepatocytes and LSECs have been shown to compensate for an impairment of Kupffer cell phagocytosis [85, 122]. The enriched GO terms “Dendrite morphogenesis” and “Neural precursor cell proliferation” are puzzling, but it is conceivable that some growth factors which stimulate liver regeneration also activate neuroregeneration and hence these genes are co-annotated to brain-related developmental process terms.

Overall we observe processes in the 3-cell liver model (relative to the 2-cell model) that are indicative of post-injury, liver regeneration, i.e., secretion of growth factors and cytokines and up-regulation of important metabolic processes. Actin cytoskeleton enriched terms may imply an increase in cell-cell and cell-matrix interactions. Terms related to signaling seem to imply an increase in cell cross-talk and collaboration, a key aspect of healthy tissue functioning. Ultimately, we found support in the literature for a majority of the most enriched terms. Our results clearly demonstrated how functional enrichment can lead to useful insights, in this case providing a diagnosis of functionality gained by adding another cell type to a specific organotypic, *in vitro* liver model.

**Adenocarcinoma of Lung.**

In this data set, since genes were ranked by significance of differential expression, there was no distinction between up-regulated and down-regulated genes [38]. Furthermore, gene expression measurements were likely drawn from a multitude of different cell types. These two points made interpretation of enrichment results for these data somewhat less straightforward than for the liver models data. Nonetheless, functional enrichment provided interesting tissue-level insights.

Several of the ten hallmarks of cancer appeared among the top enriched GO terms reported in Table 6.1 [53]. Terms such as “Endothelial cell migration”, “Lamellipodium assembly”, “Cell-substrate junction assembly”, and “Platelet degranulation” were related to two hallmarks: *Inducing angiogenesis* and *Activating invasion and metastasis*. Endothelial cells are recruited to the tumor site where they integrate and initiate angiogenesis, thus ensuring nutrient and oxygen supplies [43]. Cell cytoskeleton projections at the leading edge, called lamellipodia, enable migrating endothelial cells and invading cancer cells to move through the extracellular matrix. An increased population of migratory endothelial cells may lead to expression differences in cell junction proteins since migration would favor weaker cell-
substrate contacts. In fact, proteins that provide cell-cell and cell-matrix contacts are known to be dis-regulated in cancer tissue [87]. Weaker cell-cell contacts between endothelial cells may also contribute to metastasis during later cancer stages [34]. Finally, “platelet degranulation” refers to platelet exocytosis of secretory granules containing pro-angiogenesis and pro-inflammatory factors. Platelets regulate endothelial cell migration, angiogenesis, and blood clotting and may transmit pro-survival signals to cancer cells [41]. A necessary precursor to degranulation is the transport of secretory proteins across the endoplasmic reticulum, which is described by the GO term “Cotranslational protein targeting to membrane”. The release of pro-inflammatory factors from platelets is also directly related to another hallmark of cancer, Tumor-promoting inflammation [31].

Other hallmarks such as Evading growth suppressors, Sustaining proliferative signaling, and Enabling replicative immortality were also represented in the top enriched terms. “Mitotic cell cycle checkpoint” is a process we expect to be enriched in cancer cells. Deactivating cell cycle checkpoints is one way cancer cells evade the built-in regulation that typically suppresses limitless replication [53]. Similarly, enrichment of “Activation of MAPKK activity” means genes involved in the mitogen- (or growth factor-) activated signaling pathway were differentially expressed in the lung cancer tissue. In fact, genes in this pathway are often mutated in cancers, resulting in continuous proliferative signaling. Mutations in the MEK gene, or the MAPKK, have been specifically implicated in lung cancers [1]. Further, several stem cell- and development-related terms were enriched, which is indicative of the complexity of the tumor microenvironment. A significant population of stromal cells are recruited to tumor sites and begin releasing growth factors and synthesizing the structural framework of the tumor microenvironment [15]. Additionally, cancer cells and stem cells share many genetic properties, including the ability to undergo unlimited division [52]. Finally, as described by the hallmark Reprogramming Energy Metabolism, cancer cells make adjustments to their cellular metabolic processes in order to sustain limitless replication [53]. The enriched terms, “Regulation of glycoprotein biosynthetic process”, “Water-soluble vitamin metabolic process”, “Anaphase-promoting protein catabolic process”, and “Glycosphingolipid metabolic process” likely fall under this category of metabolic processes which are dis-regulated in lung cancer tissue.

6.4 Discussion

The main strength of this new method is that by accounting for the differential gene expression of perturbed genes, we use more of the available information in the given ranked gene list. However, two seemingly minor details considerably improve the performance and could also be used to refine other functional enrichment methods.

First, when learning the overall false positive and false negative rate, choosing a maximal value $r_{\text{max}}$ forces the algorithm to never penalize a perturbed gene more than any unperturbed gene. All perturbed genes that are not in the lowest category, $P_k$, are even penalized strictly less than any unperturbed gene. The value of $r_{\text{max}} = 0.5$ used by MGSA implies that
Figure 6.7: Plot of the optimal choice for the false negative rate $\beta$ against the ratio of explained perturbed genes to explained unperturbed genes. Through different line styles, this relation is shown for cases where 10% (solid), 30% (dashed) and 50% (dotted) of all genes are considered perturbed.

if $\beta^* = 0.5$ ($\alpha^* = 0.5$), every explained (unexplained) gene, no matter whether perturbed or unperturbed, is multiplied by the same value, which does not allow for meaningful functional enrichment. Figure 6.7 shows that if the proportion of genes that are considered perturbed is small, $\beta^* = 0.5$ unless the returned set of processes has really high quality. A closer examination of the MCMC implementation can explain this behavior.

Our MCMC implementation is designed to maximize the likelihood function with respect to $C, \alpha, \beta$ and $q$. Given a set $C$ of biological processes, we can also analytically maximize with respect to the error rates, as we explain below. We do not employ this way of optimization in the algorithm since it is much slower than sampling a discrete value in a predefined range.

Let $w_1 \leq \ldots \leq w_k$ be weights that satisfy the same equations as $\alpha_i$ (Equations 6.2.1 & 6.2.2) except that they are centered around 1 instead of $\alpha$. Then, by setting $\alpha_i = \alpha w_i$, we obtain an optimal false positive rate $\alpha^*$ of

$$\alpha^* = \frac{\sum_{i=1}^{k} |NP_i|}{|NU| + \sum_{i=1}^{k} |NP_i|}$$
By setting $\beta_i = \beta w_i$, we obtain an optimal false negative rate $\beta^*$ as the solution of

$$0 = \frac{\delta L}{\delta \beta} = \sum_{i=1}^{k} \frac{-w_i}{1 - \beta w_i} |E P_i| + \frac{|E U|}{\beta}$$

$$0 = |E U| - \sum_{i=1}^{k} \frac{\beta^* w_i}{1 - \beta^* w_i} |E P_i|$$

Thus, finding the optimal $\beta^*$ requires finding the root of a polynomial of degree $k$, which can be done numerically. Figure 6.7 shows the optimal false negative rate for different proportions of perturbed genes and different qualities of the explanatory set. The optimal false negative rate decreases as the proportion of genes that are considered perturbed (which is a predefined model parameter) or the quality of the explanatory set increases. In other words, $\beta^*$ decreases as $(\sum_{i=1}^{k} |E P_i|)/|E U|$ increases.

All this shows that the choice of how many genes are considered perturbed heavily influences, which false rates are learnt by CRFE and MGSA. If we consider only a small proportion of genes as perturbed, i.e., $|P| \ll |U|$, then $\sum_{i=1}^{k} |N P_i| \ll |N U|$, which means that CRFE and MGSA will learn a small false positive rate, and since $\sum_{i=1}^{k} |E P_i| \ll |E U|$, the methods will learn a large false negative rate - often even larger than the maximally allowed value, $r_{max}$. Therefore, we can compare the learned false rates of different data sets only if we use the same proportion of perturbed genes.

### 6.5 Conclusion

We presented a new functional enrichment method called Concise Ranked Functional Enrichment (CRFE) that uses both the ranking of genes in a list to compute functional categories that annotate many of the highly ranked genes. As a generalization of MGSA, CRFE calculates the likelihood of enrichment for sets of processes rather than for individual processes, and it penalizes the size of the returned set of processes.

We examined four functional enrichment methods with respect to three criteria. CRFE performed well in all criteria: it returned a concise set of specific processes that explained highly perturbed genes with high quality. Every other method performed poorly in at least one criterion. GOrilla and FuncAssociate2.0 explained only a small proportion of perturbed genes with high quality. FuncAssociate2.0 returned specific processes but they were redundant. GOrilla performed poorly in terms of both conciseness and specificity. Overall, MGSA had the lowest quality score. MGSA returned less specific processes than CRFE although it performed as well as CRFE in terms of conciseness.

The high-level interpretations of the results serve as an argument that the CRFE enriched GO terms make sense within the context of the data set being analyzed. We attempted to give a flavor of the type of meaningful information that can be extracted from functional enrichment analysis using our algorithm. A deeper investigation by an expert of the specific
genes enriching these top processes would likely lead to interesting, useful insights and may even prompt the discovery of new biology.
Bibliography


## Index

<table>
<thead>
<tr>
<th>Term</th>
<th>Pages</th>
</tr>
</thead>
<tbody>
<tr>
<td>Attractor</td>
<td>5, 32, 36, 40, 41, 44, 47, 48, 50, 51, 53, 69, 64, 68, 70</td>
</tr>
<tr>
<td>Average Maximal Transition Probability</td>
<td>9, 33, 44</td>
</tr>
<tr>
<td>Basin of Attraction</td>
<td>5, 12, 36, 70</td>
</tr>
<tr>
<td>Central Dogma of Molecular Biology</td>
<td>1</td>
</tr>
<tr>
<td>Coding Region</td>
<td>1</td>
</tr>
<tr>
<td>Derrida Plot</td>
<td>18</td>
</tr>
<tr>
<td>Derrida Value</td>
<td>7, 14, 18, 44</td>
</tr>
<tr>
<td>Differential Equation Model</td>
<td>4</td>
</tr>
<tr>
<td>Discrete Dynamical System</td>
<td>4</td>
</tr>
<tr>
<td>Extrinsic Noise</td>
<td>3</td>
</tr>
<tr>
<td>Gene Expression</td>
<td>1, 32</td>
</tr>
<tr>
<td>Gene Regulation</td>
<td>1</td>
</tr>
<tr>
<td>Gene Regulatory Network</td>
<td>2</td>
</tr>
<tr>
<td>Gillespie Algorithm</td>
<td>4</td>
</tr>
<tr>
<td>Hamming Weight</td>
<td>16, 17</td>
</tr>
<tr>
<td>Intrinsic Noise</td>
<td>3, 5</td>
</tr>
<tr>
<td>Knock-Out</td>
<td>43, 46, 51, 57</td>
</tr>
<tr>
<td>Limit Cycle</td>
<td>5, 6, 7, 12, 14, 33, 68</td>
</tr>
<tr>
<td>Nested Canalizing Function</td>
<td>11, 30</td>
</tr>
<tr>
<td>Definition</td>
<td>13, 29</td>
</tr>
<tr>
<td>Layer Number</td>
<td>18, 25, 25</td>
</tr>
<tr>
<td>Number of</td>
<td>26, 30</td>
</tr>
<tr>
<td>Unique Polynomial Representation</td>
<td>24, 29</td>
</tr>
<tr>
<td>Over-expression</td>
<td>43, 46, 51, 57</td>
</tr>
<tr>
<td>Phenotype</td>
<td>2, 3, 5, 9, 32, 43, 48, 50, 53, 59, 68</td>
</tr>
<tr>
<td>Promoter Region</td>
<td>1</td>
</tr>
<tr>
<td>Robustness</td>
<td>3, 7, 8</td>
</tr>
<tr>
<td>dynamic</td>
<td>7, 35, 36</td>
</tr>
<tr>
<td>evolutionary</td>
<td>68, 68</td>
</tr>
<tr>
<td>phenotypic</td>
<td>9, 44</td>
</tr>
<tr>
<td>SDDS</td>
<td>18</td>
</tr>
<tr>
<td>State Space</td>
<td>4, 6, 8, 9, 36, 38, 47, 52</td>
</tr>
<tr>
<td>Steady State</td>
<td>5, 7, 14, 33, 38, 39, 50, 68</td>
</tr>
<tr>
<td>Stochasticity</td>
<td>3, 5, 18, 32, 38, 40, 43, 44</td>
</tr>
<tr>
<td>strongly connected</td>
<td>68, 70</td>
</tr>
<tr>
<td>components</td>
<td>70</td>
</tr>
<tr>
<td>Transcription Factor</td>
<td>1, 34, 36, 38, 39, 45, 46</td>
</tr>
<tr>
<td>48, 61, 63, 105</td>
<td></td>
</tr>
</tbody>
</table>
Nomenclature

BS    (DNA) binding site
CRFE  Concise Ranked Functional Enrichment
DBD   (Transcription factor) DNA binding domain
DNA   Deoxyribonucleic acid
FFL   Feedforward loop
GO    Gene Ontology
GRN   Gene regulatory network
MCMC  Markov chain Monte Carlo method
MMR   Mismatch Repair
mRNA  Messenger RNA
NCF   Nested canalizing function
RMA   Robust multichip average
RNA   Ribonucleic acid
SDDS  Stochastic Discrete Dynamical System
TF    Transcription factor
UV    Ultraviolet light
Appendix A

Additional Figures
Figure A.1: Development of the correlation of various measurements when the network size increases
Figure A.2: Histogram of the redundancy of the top 10 processes for each method. Each column shows the result for a different method, each row for a different data set.

Figure A.3: Histogram of the redundancy of the top 40 processes for each method. Each column shows the result for a different method, each row for a different data set.