A Probabilistic Approach to Gene Expression Analysis

by

Marie Lluberes

A dissertation submitted in partial fulfilment of the requirements for the degree of

DOCTOR OF PHILOSOPHY

in

COMPUTING AND INFORMATION SCIENCE AND ENGINEERING

University of Puerto Rico

Mayagüez Campus

May, 2017

Approved by:

Bienvenido Vélez, PhD Member, Graduate Committee	Date
Mauricio Cabrera Ríos, PhD Member, Graduate Committee	Date
Omar Colón, PhD Member, Graduate Committee	Date
Jaime Seguel, PhD President, Graduate Committee	Date
Wilson Rivera Gallego, PhD CISE Program Coordinator	Date
Aidsa Santiago, PhD Graduate School Representative	Date

"*Darkness took me. And I strayed out of thought and time. Stars wheeled overhead, and every day was as long as the life age of earth.*
But it was not the end. I felt light in me again. I've been sent back until my task is done."

J. R. R. Tolkien, *Gandalf the White, Lord of the Rings: The Two Towers.*

Abstract of Dissertation Presented to the Graduate School of the University of Puerto Rico in Partial Fulfilment of the Requirements for the Degree of Doctor of Philosophy

A Probabilistic Approach to Gene Expression Analysis

by

Marie Lluberes

May, 2017

Chair: Jaime Seguel

COMPUTING AND INFORMATION SCIENCE AND ENGINEERING

Technology development has considerably increased the collection and storage of biological data. Nevertheless, the challenge of transforming such data into information, prevails. Such transformation demands the involvement of several disciplines, gathered under the umbrella of Bioinformatics. One of those main challenges under Bioinformatics's extensive research area is learning the connections that govern gene activity, or gene regulatory networks (GRN). This is a very large scale problem, both because of the amount of variables involved as per the amount of possible interactions among them. Because of this, one very effective, accepted approach to inferring these networks is the use of Boolean representations of GRN. This model takes inputs from the binary domain; therefore, gene expression –which is measured as real data– needs first to be binary quantized with the use of a threshold.

But both GRN and gene expression precise mathematical models are unknown; hence, their modeling is based on conjectures, biased at times. As a consequence, different models render different results. We study the effect of the differences that some binary quantization methods have on the resulting binarized gene expression. We call this *model uncertainty*.

Furthermore, the discretization of gene expression subjects the threshold to changes as well. The number of measurements for the study of a gene may be bound, as a result of budgetary constraints, for instance. Have more data become available, this impacts the gene's expected behavior. We study the effect that these changes on

discretization have on a gene's binarization, under different methods. We call this *discretization uncertainty*.

While these uncertainties may persist due to, as aforementioned, the lack of a precise model, a unified approach may contribute to mitigate their impact. We propose a multi-algorithmic approach, with aggregation rules and voting mechanisms on several methods to countereffect model uncertainty.

Rather than relying on a particular number of measurements, we use the gene's threshold expected behavior to choose its binarization through statistical analysis, considering threshold variations, on an attempt to countereffect discretization uncertainty.

This unified approach of statistical analysis and aggregation rules is presented as a framework that allows a customized selection of the methods.

Finally, in order to measure the impact of these changes, I propose a simple evaluation method for network binarization changes. The proposed method provides specific metrics for evaluation on each network state individually for the detection of troubled binarizations. Existing network inference methods do not provide information on the binarization of each gene, making difficult to discern if the differences are due to selected binarizaton methods or to the learning mechanism of the implementation.

Resumen de la Disertación Presentada a la Escuela Graduada de la Universidad de Puerto Rico en cumplimiento parcial de los requisitos para el grado de Doctor en Filosofía

Un Enfoque Probabilístico al Análisis de Expresión Genética

Por

Marie Lluberes

May, 2017

Consejero: Jaime Seguel

CIENCIAS E INGENIERÍA DE COMPUTACIÓN E INFORMACIÓN

Los avances tecnológicos en instrumentación han aumentado considerablemente la recopilación y almacenaje de data biológica. No obstante, el desafío de transformar esta data en información, permanece. Tal transformación exige la participación de varias disciplinas, cobijadas balo la sombrilla de la Bioinformática. Uno de esos retos importantes dentro la extensa área de Bioinformática es aprender cuáles son las conexiones que rigen la actividad de los genes, o Redes Regulatorias Genéticas (RRG). Este es un problema de gran tamaño, tanto por la cantidad de variables envueltas como por la cantidad de posibles interacciones entre ellas. Debido a esto, un enfoque muy efectivo y aceptado para inferir estas redes es el uso de representaciones Booleanas de RRG. Este modelo acepta entradas del dominio binario; por consiguiente, la expresión genética –que es medida como data real– primero necesita ser quantizada en forma binaria con el uso de un umbral.

Modelos matemáticos precisos, tanto para las RRG como la expresión genética, son desconocidos; por lo tanto, su modelaje está basado en conjeturas, a veces prejuiciadas. Como consecuencia de esto, diferentes modelos ofrecen diferentes resultados. Estudiamos el efecto de las diferencias que algunos métodos de quantización binaria tienen en la expresión genética binarizada resultante. A esto le llamamos *incertidumbre del modelo.*

Mas aún, la discretización elegida para la expresión genética, también somete el umbral a cambios. La cantidad de medidas tomadas para estudiar un gen puede estar

controlada, por restricciones presupuestarias, por ejemplo. De hacerse disponible mas data, esto tiene un impacto en el comportamiento del gen. Estudiamos los efectos que estos cambios en discretización tienen en la binarización de un gen, bajo diferentes métodos. Esto lo llamamos *incertidumbre de la discretización.*

A pesar de que estas incertidumbres pueden persistir debido a, como se mencionó antes, la ausencia de un modelo preciso, un acercamiento unificado puede mitigar su impacto. Proponemos un acercamiento multi-algorítmico, con reglas de agregación y mecanismos de votación en varios métodos, para contrarrestar la incertidumbre del modelo.

En lugar de basarnos en una cantidad particular de medidas, usamos el comportamiento esperado del umbral del gen para elegir su binarización a través de análisis estadístico, teniendo en cuenta las variaciones del umbral, en un esfuerzo por contrarrestar la incertidumbre de la discretización.

Este enfoque unificado de análisis estadístico y reglas de agregación es presentado como un marco que permite adaptaciones en la selección de los métodos.

Finalmente, con el propósito de medir el impacto de estos cambios, propongo un método simple de evaluación de los cambios en una red. El método propuesto provee métricas especificas para la evaluación individual de cada estado de la red, ayudando en la identificación de binarizaciones conflictivas. Los métodos de inferencia de redes existentes no proveen información sobre la binarización de cada gen, dificultando el discernimiento de diferencias debido al método seleccionado o al mecanismo de inferencia de la implementación.

Acknowledgements

It is wholeheartedly that I want to acknowledge and express utter gratitude to my adviser, Dr. Jaime Seguel, who patiently bore with me through all my hurdles along this overextended path. Thanks for always valuing our work.

I need to acknowledge the role of my father, who instilled in me the love for learning and with much sacrifice put forth in practice his preached value for education, making of me the first college graduate in my family.

Thank you, Dr. Juan López Garriga, Rosalie and Mairim, from the RISE program, for dealing with my off-campus, non-traditional student status and accommodating me.

Thank you, Dr. Domingo Rodríguez, for taking me in and providing support when I first started, making viable my engagement in the program, and giving me the foundations of a researcher.

I'm very grateful to the staff of the CISE PhD program, Alida and Sarah, and to former program director, Dr. Néstor Rodríguez, for all their help and support during my years in the program.

I hereby acknowledge the role my daughters, Caeli and Ilea, who at some point played mom and brought me to my senses on a crucial moment of my academic life, pushing me to bring it to conclusion.

And thank you to my furry children, for all the head-bumps, so many licks and perpetual purrs; for keeping me company through endless nights and forgiving my delays on tending their needs.

This research was supported in part by grant NIH-R25GM088023 from the National Institute of General Medical Sciences.

Contents

Abstract **iv**

Resumen **vi**

Acknowledgements **viii**

List of Figures **xii**

List of Tables **xiv**

List of Abbreviations **xvi**

List of Symbols **xvii**

Preface **xviii**

Chapter 1 Introduction **1**

1.1 The Advent of Bioinformatics 2

1.2 Computational Systems Biology and its challenges 2

1.2.1 Modeling GRN and its Dynamics 4

1.2.2 Gene Expression Decision Algorithms 5

1.3 Model Uncertainty and Discretization Uncertainty 8

1.3.1 Model Uncertainty 9

1.3.2 Discretization Uncertainty 9

1.4 Proposed Work, Research Contributions And Applications 10

Chapter 2 Biological Applications and Computational Background **13**

2.1 Gene Expression and Regulation 14

2.1.1 Time Series Data Analysis 15

2.1.2 Pathway Analysis and Discovery 15
2.2 Applications of Gene Expression Quantization 16
2.2.1 Gene Regulatory Networks 17
2.2.1.1 Boolean Networks (BN) 18
2.2.1.2 Probabilistic Boolean Networks (PBN) 20
2.3 Binarization Methods 23
2.3.1 Average Jump 23
2.3.2 Multiscale Analysis 25
2.3.3 One-Step Approximation 27
2.3.4 2-means Clustering 27
2.3.5 Median Separation 29

Chapter 3 Gene Expression Threshold Computations 31

3.1 Semantics and Accuracy of GETC 31
3.2 Natural threshold, Convergence threshold and Computation 32
3.3 Assessment of Variations in Gene Expression Threshold Computations: Case Study 33
3.3.1 Numerical Variations of the Thresholds 34
3.3.2 Variations in Binarization 37
3.4 Analysis of Experiments 39

Chapter 4 A Unified Approach to the Computation and Analysis of Strings of Gene Expression States 41

4.1 Model Uncertainty and Discretization Uncertainty 42
4.2 Four Deterministic Methods 42
4.2.1 Correlation 43
4.2.2 Threshold Displacements 43
4.3 An Algorithmic Framework 44
4.3.1 Estimating Threshold Displacements 44
4.3.2 Election of a String of Expression States 46
4.3.3 Probabilistic Strings of Expression States 48
4.3.4 Post Processing 50

4.4 Experiments 51
4.4.1 Analysis of Experiments 52
4.4.2 Improving Network Resolution 54
4.4.3 Scoring Strings Probabilities 55
4.5 Value Imputation of No Decidable States 55
4.6 Evaluation of Boolean Networks Inferred from Elected Strings of States 56
4.6.1 Example A 58

Chapter 5 Summary and Future Work 62
5.1 Future Work and Related Directions 63
5.1.1 Ternary Logic Network 64
5.1.2 Threshold Convergence 64
5.1.3 GAP-Displacement Classification 64
5.1.4 Biological Validation 64
5.2 Conclusions 64

Chapter 6 Ethical Considerations on Nonhuman Animal Testing 66
6.1 Nonhuman Animal Testing in Review 67
6.1.1 IACUC Criticism 68
6.2 Alternatives to Nonhuman Animal Testing 68
6.2.1 Nonhuman Animal Testing for Cosmetic Purposes 69
6.3 A very vocal crowd 70

Appendix A Gene Expression Data Sets 72
A.1 Gene data set for Leukotriene B4 used on experiments on Chapter 4. . 72
A.2 Gene data set for yeast cell cycle used on experiments on Chapter 3 . 72

Appendix B Threshold Displacement Estimation Graphics 74
B.1 Thresholds Displacements by Range 74

References 77

List of Figures

1.1 Iterative cycle from sampling to network modelling. 3
1.2 Cardiac gene regulatory network [Paige et al., 2015] 6

2.1 Protein Synthesis in a Eukaryote. Central Dogma of Biology [*Essentials of Cell Biology*]. 14
2.2 Gene expression profiling of catenin delta 2 (CTNND2) on human fetal lung, [*Transcriptomic analysis of human lung development, GEO*]. . . 15
2.3 The Emergent Integrated Circuit of the Cell. 16
2.4 [Shmulevich et al., 2002] Left: A diagram illustrating the cell cycle regulation example. Arrowed lines represent activation and lines with bars at the end represent inhibition. Right: The logic diagram describing the activity of Rb protein in terms of 4 inputs: cdk7, cyclin H, cyclin E, and p21. 17
2.5 Boolean Network of 3 genes and its state transition diagram 18
2.6 Probabilistic Boolean Network and its predictors. 21
2.7 State Transition Diagram of a PBN. Highest probability path and attractors are collored in red. 22

3.1 Thresholds plots of the four methods for one time series on Experiment 1. Here, Jb1 is *Algorithm A*, Jb2 is *Algorithm B*, Cb1 is *Algorithm D* and Cb2, *Algorithm E*. 36
3.2 Relations between maximal distance between threshold and data range. 37

4.1 From left to right and top to bottom: threshold displacements for algorithms A, B, C and D. 44
4.2 From left to right and top to bottom: threshold displacements for algorithms A, B, C and D. 45

4.3 Algorithmic Framework Diagram 45
4.4 Probability distribution for the thresholds returned by Algorithms A, B, C and D. 49
4.5 Boolean Network of 3 genes and its states transition diagram 58
4.6 All 8 states transition diagrams for 4.5b. 59

6.1 Lung on a chip. [*Harvard Wyss Institute*] 70

B.1 Thresholds Displacements - Algorithm A. 74
B.2 Thresholds Displacements - Algorithm B. 75
B.3 Thresholds Displacements - Algorithm C. 75
B.4 Thresholds Displacements - Algorithm D. 76

List of Tables

3.1 Threshold values for random generated data. Highlighted values are matching values on both experiments. Here, Jb1 is *Algorithm A*, Jb2 is *Algorithm B*, Cb1 is *Algorithm D* and Cb2, *Algorithm E*. 35

3.2 Threshold values for cdc data set. Highlighted values are matching values on both experiments. Here, Jb1 is *Algorithm A*, Jb2 is *Algorithm B*, Cb1 is *Algorithm D* and Cb2, *Algorithm E*. 35

3.3 Euclidean distance between different methods on same experiment. Here, Jb1 is *Algorithm A*, Jb2 is *Algorithm B*, Cb1 is *Algorithm D* and Cb2, *Algorithm E*. 36

3.4 Euclidean distance between same methods on different experiments. Here, Jb1 is *Algorithm A*, Jb2 is *Algorithm B*, Cb1 is *Algorithm D* and Cb2, *Algorithm E*. 37

3.5 Comparison between binary quantization matrices of same methods using thresholds obtained from both experiments. Matching binarizations are highlighted. Cdc data set. Here, Jb1 is *Algorithm A*, Jb2 is *Algorithm B*, Cb1 is *2*-means and Cb2, *Algorithm E*. 38

3.6 Distances between all methods expressed as Hamming distance. Lowest and highest scores are highlighted. Here, Jb1 is *Algorithm A*, Jb2 is *Algorithm B*, Cb1 is *2*-means and Cb2, *Algorithm E*. 39

4.1 Correlation of the outputs of four DGEC algorithms. 43

4.2 Expected values of threshold displacements. 46

4.3 Example of an inconsistent collective decision with three decision algorithms 48

4.4 Z_e's and statistics returned by the Algorithmic Framework for the time-course expression data of gene AREG. 51

4.5 Z_e's and statistics returned by the Algorithmic Framework for the time-course expression data of gene HLX. 52
4.6 Z_e's and statistics returned for the time-course expression data of gene KIF1A by the Algorithmic Framework. 53
4.7 Z_e's and scores for the time-course expression data of gene AREG. . . 55
4.8 Z_e's and scores for the time-course expression data of gene HLX. . . . 56
4.9 Z_e's and scores for the time-course expression data of gene KIF1A. . . 57
4.10 Gene Binarization After Value Imputation 59
4.11 Network Evaluation 60
4.12 Network States Evaluation m_s 60

A.1 3 genes from Leukotriene B4 data set. 72
A.2 4 genes from yeast data set. 73

List of Abbreviations

BN	**B**oolean **N**etwork
BQ	**B**inary **Q**uantization
COD	**C**oefficient **O**f **D**etermination
DGEC	**D**eterministic **G**ene **E**xpression **C**lassifier
GAP	**G**ene **A**ctivity **P**rofile
GRN	**G**ene **R**egulatory **N**etwork
PBN	**P**robabilistic **B**oolean **N**etwork
SCSI	**S**equence of **C**ubic **S**pline **I**nterpolations

List of Symbols

$\perp$	Undecidable expression state of a gene
P_e	Probability of string Z_e
R	Aggregation rule
S	Finite set of decision algorithms
Z_e	String of expression states
α	Scalar value
δ_F	State transition
Δ_X	Displacement of threshold returned by Algorithm X
γ	Median of the vector of strong discontinuities
Γ	Finite set of logic statements (*agenda*)
Λ	Logical statement describing decision problem *doctrine*
μ	Mean
Ω_i	Average of array of probabilities of strings for gene i
$\Omega_{X,i}$	Array of probabilities of strings for gene i *with method* X
π	Vector G_i sorted in increasing order
$\rho_{X,i}$	Range of threshold displacements
σ^2	Variance
$\tau_{X,i}$	Threshold of G_i with method X
θ	Coefficient of Determination
$\Psi_X(G_i)$	Threshold of G_i with method X; G_i a sequence of cubic spline interpolations

To those while in this path I lost,

To those that have been saved,

To those I have not gotten to yet:
I forget you not...

Preface

This dissertation summarizes and further develops some of my contributions to research jointly conducted and published with my doctoral advisor.

In *Semantics and Accuracy of Gene Expression Threshold Computations: A Case Study* [Seguel and Lluberes, 2013], we expose and describe the problem –the uncertainties in binarizations of gene expression classification methods– and attempt to measure them. This discussion is reviewed in Chapter 3. In *A Unified Approach to the Computation and Analysis of Strings of Gene Expression States* [Seguel and Lluberes, 2015], we propose a framework to handle these uncertainties. This framework is reviewed and extended in Chapter 4.

In this dissertation, I describe and use five different binarizations methods. Four of them are incorporated in the proposed framework. In [Seguel and Lluberes, 2013], binarization methods are classified according to their heuristic. For the sake of balance, the 4 algorithms chosen for this semantic exercise must fall under either one classification; because of that, a fifth algorithm is added, switching with one used in the framework.

The rest of this work is organized as follows:

The first chapter introduces the problem of binary quantization of gene expressions, and surveys related work.

The second chapter is a brief biological background on data collection for gene expression and discusses Gene Regulatory Networks, one of its main applications. This chapter includes the computational background and describes and analyses the algorithms used for binary quantization of gene expression.

Chapter 5 is a summary of this dissertation, along with the conclusions and the discussion of future work.

Chapter 6 addresses ethical aspects of experimentation in animals.

Chapter 1

Introduction

"Any sufficiently advanced technology is indistinguishable from magic."

—Arthur C. Clarke

A biological system is more than the sum of its parts; a deep understanding of each of its individual components may not elucidate its intricate behavior. To model biological phenomena we need to decipher its structure and dynamics. While identifying particles and studying their response to specific stimuli is essential to reproduce the structure of a system, it is by studying their interactions that we can replicate the system's behavior. But understanding the functioning of a system as a whole, demands understanding first the dynamics of this interaction among its components [Kitano, 2002], which requires a myriad of resources, and the collaboration of several disciplines. The study of a biological system is often iterative, with each iteration potentially increasing the level of understanding and formal representation of the system. This iterative cycle uses large amounts of heterogeneous data, whose processing is unfeasible without computational resources. This is specially the case as precise formal formulation of the problem is often lacking.

1.1 The Advent of Bioinformatics

Technology developments in molecular biology instrumentation and computer machinery have considerably facilitated the collection and storage of biological information, especially information related to DNA, RNA and protein structures. As a result, a huge and constantly increasing amount of biological data is available to study molecular biology phenomena *in silica*.

Bioinformatics, which in its broadest sense is the scientific discipline of the study of biological systems from the perspective of the nature and transformation of biological information, emerged out of the need to extract knowledge from this data for scientific understanding and, often, for its applications into proactive, predictive and preventive health practices. Coined on a pre data-driven era as "the study of informatic processes in biotic systems", Bioinformatics evolved to meet the demands of analysis of the exponentially growing amounts of data throughout the development and use of computational methods [Hogeweg, 2011].

Bioinformatics uses computing, probability and statistics to model and analyze biological systems. While probability and statistics provide a basic framework within which random variation can be quantified so that systematic variation can be studied, computing provides the methods for automated information processing, model construction, systems simulation, verification and validation. Figure 1.1 depicts this learning cycle.

1.2 Computational Systems Biology and its challenges

Data-driven research, although providing valuable insight, is not enough to understand the functioning of complex systems. For that, we need to dig into their dynamics as a working system. It is then when Systems Biology emerges. Systems Biology focuses precisely on the design of biological phenomena as systems, including system structure identification and behavior analysis [Kitano, 2001].

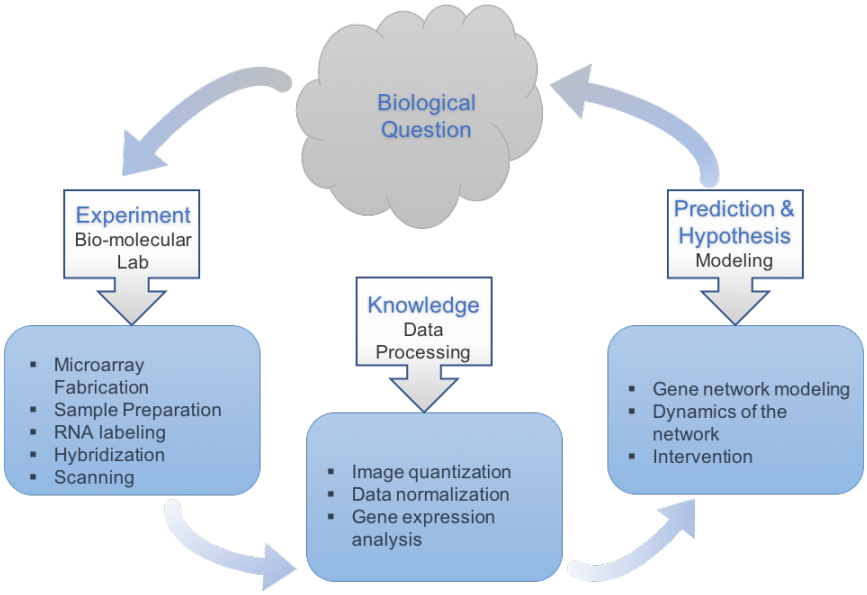


FIGURE 1.1: Iterative cycle from sampling to network modelling.

The surplus of data rose expectations of a fast-paced unveiling of the causes of disease and the efficient development of drugs to treat them. This has been impeded by the lack of understanding of the systems' dynamics. [Eils and Kriete, 2014].

The computational aspects of systems biology are particularly challenging due to the huge size of the biological data sets. Such sizes translate normally into large execution times that turn into unfeasible or unpractical algorithms that are perfectly adequate for other applications. Besides problem size, the exact algorithmic solution for biological problems is frequently impeded by the lack of explicit deterministic models for biological phenomena. A chief example illustrating this situation is the problem of modeling and simulating gene regulatory networks (GRN). This problem, which is very current in systems biology, presents many challenging aspects all across its solution steps, from the construction of the network to the study of its dynamics [Banf and Rhee, 2017]; [Gomaa, 2011].

In particular, the lack of a precise model for the interpretation of gene expression in the Boolean domain —weather or not a gene can be considered expressed at a given time instant— is a basic challenge. A number of methods have been developed attempting to offer a more precise answer to this question, yet the lack of a formal description of the phenomenon yields different results from each of them. Binarization, as this process is called, is an essential step in the construction of Boolean models of GRN.

As a result, different binarizations of gene expression may yield different network representations, adding uncertainty to an already challenged network construction.

Furthermore, the experimental study that I conducted suggests that mismatching results are obtained from the same method when the sample size of the specimen under consideration varies.

1.2.1 Modeling GRN and its Dynamics

Several approaches for the modeling and analyses of GRN have been proposed, and date as back as the 1960s [Kauffman, 1969]. Most of them have been covered in extensive reviews, both from computational [Jong, 2002]; [Bower, 2001]; [Hecker et al., 2009] and biological perspectives [Wilczynski and Furlong, 2010]; [Smolen, Baxter, and Byrne, 2000]. Figure 1.2 is an illustration of a gene regulatory network.

Continuos-time variable methods include the use of differential equations (ordinary and piece-wise differential equations), which is a prevailing formalism for modeling dynamic systems. In this approach, gene regulation is modeled as a function of messenger ribonucleic acid (mRNA) concentrations of other genes that change over time according to differential equations, at a particular rate and continuously. For solving the system is necessary to know the change-rate parameters, which are not available in most cases. Some models using this approach extract the change-rate parameter from simulations, but this is only feasible for small systems. This is the main drawback of this approach, along with the presumption that the change-rate affects continuously the concentration levels. Also, because of the number of equations and unknown parameters, the complexity of differential equation models grows exponentially. Nevertheless, these models can render very accurate representations of the system and have been successfully used as in [Csikász-Nagy et al., 2006]; [Hoon et al., 2003], among others.

Discrete-time variable methods restrict gene expression to the Boolean domain, this is, two states: expressed or not expressed, through the use of Boolean logic and deterministic rules expressing the relations between gene expression. This approach, which was pioneered by Kauffman [Kauffman, 1969], captures the dynamics of the system

in a very simple way, allowing efficient analysis of large networks. Thorough analysis of the use of Boolean networks (BN) for GRN representations have been presented [Bornholdt, 2008]; [Martin et al., 2007]. A model variation that incorporates probabilistic interactions rules are probabilistic Boolean networks (PBN) [Shmulevich et al., 2002]. In PBNs, not one but several rules may define the dynamics of each element of the system, each with a probability. BN and PBN are covered in further detail in chapter 2. This model, which will be the focus of this research, has been successfully used for modeling mammalian cell cycle [Faure et al., 2006], [Samaga et al., 2009]; yeast cell cycle [Li et al., 2004], [Davidich and Bornholdt, 2008]; Drosophila gene system [Sanchez and Thieffry, 2001]; cancer [Zhang et al., 2008]; among others like liver function [Philippi et al., 2009], [Wu, Yang, and Chan, 2009].

Bayesian networks models are worth noting, as well. These models, which can expand to discrete and continuous variable domains, capture the stochastic nature of the GRN with the representation of its elements as random variables, and describing relationships between them with probability [Markowetz and Spang, 2007]; [Han et al., 2014]. In Bayesian networks, relations are represented as acyclic graphs; autoregulatory dependencies and feed-back are modeled with dynamic Bayesian networks [Husmeier, 2003]; [Perrin et al., 2003]; [Grzegorczyk and Husmeier, 2011]. Studies comparing dynamic Bayesian networks and PBN show similar tools can be used for both models [La, 2006]; [Li et al., 2007].

Some methods use hybrid approaches to modeling GRN [Leon and Davidson, 2009]. Here, Boolean logic is added to a differential equations model of a GRN. A review of GRN modeling that includes a summary of very heterogeneous approaches is presented in Styczynski et al. [Styczynski and Stephanopoulos, 2005].

1.2.2 Gene Expression Decision Algorithms

Technologies used to sample gene expression states provide an overwhelming amount of data. One common approach for the analysis of such large amounts of data is first to group genes that exhibit similar expression patterns. Several classification algorithms are used for this and clustering is one of the most commonly used. Gene expression

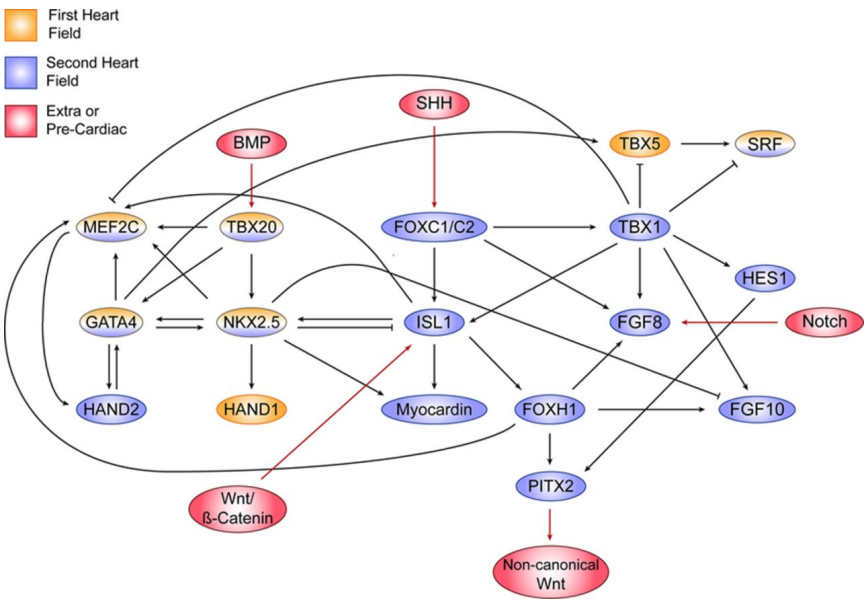


FIGURE 1.2: Cardiac gene regulatory network [Paige et al., 2015]

levels over time are recorded as time series. By comparing these expression levels, it can be decided which ones correspond to a gene that is expressed and which ones do not. This comparison needs a reference value, or threshold, to decide the state of each expression.

High-throughput technologies used to measure gene expression produce real-valued data. For the purposes of this research, gene expression decision algorithms refers to methods used to *decide* if a gene is or not expressed by mapping this data to the binary domain $\{0, 1\}$. This domain conversion has been shown to produce very little loss of information [Tuna and Niranjan, 2010]. Nevertheless, the mapping to the binary domain is affected by the heuristic used to classify the data, and by the noisy, scarce nature of the data itself. Different algorithms may render different decisions on gene expression. These differences, in turn, have an effect on the resulting Boolean network where the deterministic, rule-based construction of the network potentially propagates every change of gene state, an impact that has been widely acknowledged [Hopfensitz, Maucher, and Kestler, 2012]; [Euatham and Tongsiri, 2012].

As an essential step for modeling discrete variable GRN, the problem of weather or not a gene can be considered expressed in the binary domain is the subject of intense research. As a consequence, a variety of algorithms for this purpose has been designed.

One possible approach to the gene expression decision is to set some statistical value, as the mean or the median of the series, as the threshold [Kim, Lee, and Park, 2007]. This approach has the advantage of simplicity but the results may not be reliable in presence of data distributions with outliers.

Lloyd-Max clustering, or k-means, is other common method for threshold computation, specially with $k = 2$ [Euatham and Tongsiri, 2012]; [Berestovsky and Nakhleh, 2013]; [Müssel et al., 2016]. Computationally expensive for large data sets while conceptually simple, cluster-based approaches are subjected to random initializations of the parameter used and produce unstable results [Shmulevich and Dougherty, 2007].

Shmulevich and Zhang [Shmulevich and Zhang, 2002] proposed a method that considers the most drastic change in expression levels as a signal of change of state expression. In this method, this change of expression state is signaled by the biggest jump in data values and is used as a parameter for a time series binarization. The solution is unique but may be misleading. Without further information on the behavior of the gene, differences in expression levels may be taken as a change of state, even if this behavior happens on genes whose levels of expresson are uniformly distributed. This phenomenon is shared by most "jump-based" approaches.

The use of jump size as a signal for change on the state of a gene is also the approach taken by Hopfensitz [Hopfensitz et al., 2012]. In this method, several iterations are performed to find the step function that exhibits the data jump. The method has higher complexity but offers a unique, straightforward binarization and claims to be suited for short time series. A comparative study in the inference of Boolean networks, however, found that the networks inferred from this binarization are not always faithful [Berestovsky and Nakhleh, 2013].

Sahoo et al. proposed a method that is also based on discontinuities [Sahoo et al., 2007]. In this method, one or two steps are used to fit data series in up and down patterns for identifying both the value and the time when such discontinuity occurs. While this method does not binarize, the signaled transition value can be used as the binarization threshold. The method has the advantage of providing the instant when this change occurs; is relatively simple and fast. However, tit may render inconclusive

results for very short time series or time series where no drastic changes occur over consecutive periods, or even when changes occur slowly.

Some methods incorporate previous knowledge in threshold determination, as in Hakamada [Hakamada et al., 2004]. In Hakamada's, previous knowledge is used to find a coefficient that modifies the mean of the gene expression. This coefficient is determined with the help of the KEGG annotated database. The method yields accurate results but is very restricted to its application and demands information that may not be available.

A mixed model for binarization was proposed by Zhou et al. [Zhou, Wang, and Dougherty, 2003]. Here, cluster approach and statistical concepts are mixed to determine a threshold value.

Several other methods are proposed and used in the literature; however, many of them are a combination or based on the above discussed methods, or are methods tailored to the intended study.

As an interesting note, binarization algorithms are not exclusive of gene expression profiles. As early as 1970s, several image correction algorithms using thresholding techniques were proposed; among them, Otsu's method [Otsu, 1979] stands out.

Existing models have advantages and disadvantages, and none of them individually suffices to address the complexity of biological systems. An integrative approach seems more proper to capture biological complexity [Bower, 2001].

1.3 Model Uncertainty and Discretization Uncertainty

Gene expression is recorded as an array of measurements at time intervals j over a period of time t. Binarization of an array of expressions should be independent of the algorithm used to decide over the array. But he lack of a mathematical model that precisely describe its behavior turns the states of a gene dependant on the method used to compute it, rather than in the gene's intrinsic characteristics. Over a time period t, the same gene possesses the same behavior independently of the frequency

we choose to sample. Yet again, in absence of a model, changing the sampling rate renders different decisions.

The gene expression decision problem is stated as follows:

> *"Given the expression array $G = [G_i(j)]$, with $i = 1, ..., M$ genes and $j = 1, ...N$ observations, decide whether or not each value $G_i(j)$ corresponds to a gene in expressed state".*

In an attempt to solve this problem, different approaches have been taken for the design of gene expression decision algorithms. One approach is the use statistical analyses [Zhou, Wang, and Dougherty, 2003]; other, the of use numerical methods, called here *deterministic gene expression classifiers* (DGEC). The latter find a value, or *threshold*, where the points in the expression profile can be separated into expressed or unexpressed states. The result is a N-character binary string representing the expression states associated to a given expression profile, with unexpressed states labeled as 0 and expressed states, as 1.

1.3.1 Model Uncertainty

Let S be a set of algorithms for gene expression binarization; $X, Y \in S$. Let $B_{X,i(j)} = X(G_i(j))$ and $B_{Y,i(j)} = Y(G_i(j))$ the decisions of X, Y on $G_i(j)$, respectively.

We would expect $B_{X,i(j)} = B_{Y,i(j)}, \forall X, Y \in S$, for each G_i. But different algorithms render different decisions on G_i. We call this difference ***model uncertainty***.

1.3.2 Discretization Uncertainty

Samples are a finite set of observations of a continuous phenomenon of k number of intervals j. ***Discretization uncertainty*** is the uncertainty of missing values on the set of observations. At some point, we may want to make a refinement of the gene expression by increasing the amount k of intervals, introducing new expression values, while keeping the expression values of the original array.

Let $G_i^1, G_i^2, ..., G_i^k$ expression arrays of G_i of different interval k over time t. $\forall k, X(G_i^k)$ is expected to be the same. But decisions over arrays of different time intervals, over

the same time period, are different. This is true for every algorithm tested. To the best of my knowledge, there is no research addressing this inconsistency.

1.4 Proposed Work, Research Contributions And Applications

Our first contribution is the introduction and study of the concepts of model and discretization uncertainties on binarizations of gene expression. Through semantic and accuracy analyses of several binarization methods, we provide insights on the nature of such uncertainties.

Rather than developing a new binarization algorithm, which inevitably will suffer the same while a formal model is lacking, we propose a framework with a unified approach to deal with these uncertainties.

One of the main contributions of this research is the approach to gene expression decision with the inclusion of *not decidable* classification of some states. Here, we expose those interval measurements whose binarization cannot be determined with certainty. Instead of assigning it an arbitrary value, we identify those points as "undecidable". These serves 2 purposes: first, it flags the measurement as one that should be further reviewed; second, it allows for an assessment of the profile, where a profile with a high number of undecidable measurements may be considered as a defective sample. This novel approach helps to improve accuracy of binarization. Recently, [Müssel et al., 2016] developed an R package for binarization and "trinarization" of gene expression assessing areas that does not seem well defined for binarization. This approach, which is an extension of [Hopfensitz et al., 2012], uses 2 thresholds selected from several candidates and does not provide feedback for evaluation on the time series.

The framework provides a statistical characterization of the algorithm that can be used to decide on the states of the gene. The incorporation of the probability and displacement to determine the expression state intends to compensate for the discretization uncertainty. Statistical methods for threshold determination have been proposed [Zhou, Wang, and Dougherty, 2003]; [Nilsson et al., 2007]. Zhou et al. [Zhou, Wang,

and Dougherty, 2003] proposes a mixture model where initially the sample is clustered and needs several parameter estimation that performs better with more than 2 clusters. This approach does not consider threshold displacement. Nilsson et al. [Nilsson et al., 2007] proposes a "threshold-free" approach for gene category detection rather than binary quantization. Here, functions are assumed to be continuos and, therefore, the method is unsuitable for short data series. Euatham et al. [Euatham and Tongsiri, 2012] uses a statistical approach to characterize gene expression based on raw data "inherent" variability to create a new gene activity profile (GAP) that is, in turn, binarized. This approach, however, needs for the gene to have multiple measurements on same time instant.

Another contribution of this research is the use of a multi-algorithmic approach, which unifies heuristics of several methods. The decision process described in the above paragraph is used on a set selected algorithms rather than only one. Then, a voting mechanism, borrowed from social sciences [List, 2012], is implemented to decide the states of the gene. This was first proposed by Seguel [Seguel, 2015]; the voting system is used as a way to compensate model uncertainties. The set may have as many algorithms as the user chooses to use.

The flexibility of the proposed framework allows its adaptation to other areas where a multi-algorithmic approach to threshold computation is suitable, contributing this way, to disciplines other than the ones within the scope of this research, as may be the case with image processing.

Finally, I propose a simple method to assess the PBNs that result from this algorithm. The main advantage of this contribution is that its provides specific metrics for the assessment of each network state individually, aiding in the calibration of the network's rules and the detection of troubled binarizations. The metrics help to determine which networks are more reliable in terms of the number of decidable states. Methods for inferring Boolean networks are available. However, even when more than one binarization method may be allowed in them, in general, the output of these implementations are a few possible Boolean networks that these genes model. This kind of output does not provide information on the binarization of each gene, making

impossible to discern if the differences in the inferred networks are due to the differences between the selected binarizaton methods or to the learning mechanism of the implementation.

Drug development demands the study of the dynamics of the network and the impact of the drug under consideration in the network transitions. The proposed network evaluation method may assist in assessing such impact on the network behavior.

Because cell reproduction is regulated by genes, more accurate binarizations of gene expression may help in better understanding the role of some genes in tissue growth and development. Particularly, the role of some genes in cancer development, which has associated a high level of gene expression diversity. The proposed framework may contribute to better gene regulatory networks analysis through improved binarizations.

Development of more accurate *in silica* models as the one proposed that can, in turn, help both drug development and understanding cell regulation, may lead to reduction —and eventual elimination— of testing in non-human animals. The use and results of the animal model approach from an ethics point of view is discussed in Chapter 6.

Chapter 2

Biological Applications and Computational Background

"Computer science is to biology what calculus is to physics."

—Harold Morowitz

Cells function as an engineering system embedded in the organism. They are responsible for engaging DNA in product synthesis throughout a complex information flow, called "central dogma of molecular biology". The collective analysis of these processes provides insight on possible relation among genes, or gene networks, and on its dynamics.

Advances in technology has made possible to gather large quantities of data from cellular processes. This, and the development of complex computational methods, some aimed to extract information and others, to model phenomena, are the basis of these insights and their translation to clinical applications.

This chapter is a brief biological background, along with applications of gene expression quantizations, and a discussion of some computational methods used in the analyses and development of such applications.

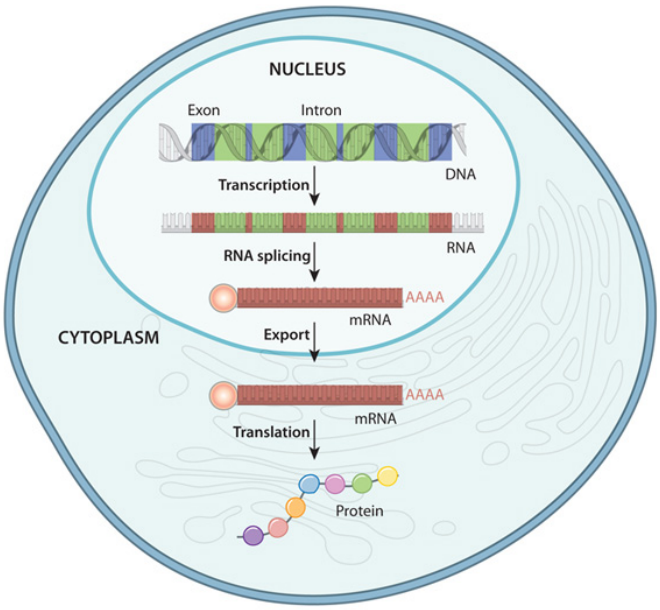


FIGURE 2.1: Protein Synthesis in a Eukaryote. Central Dogma of Biology [*Essentials of Cell Biology*].

2.1 Gene Expression and Regulation

A gene is a region of DNA that has a biological function and a specific location. Cells, as the building blocks of life, have the function of converting genes into usable products. For this, a cell must access the information on its DNA. This is done in 2-step process. In the first step, called transcription, the DNA replicates itself into mRNA. In the second step, translation, the mRNA dictates the particular connection order for the specific molecule that would be produced, or synthetized. Gene expression is the process by which the information contained on its DNA is used for product synthesis [Brooker, 2009]. Figure 2.1 illustrates this process.

A cell encodes thousands of genes on its DNA, each with a particular function. Throughout a series of mechanisms known as gene regulation, the cell controls gene expression so their specific products are synthetized only when needed by the organism. Some genes are not regulated, and thus, their expression level remains the same over time. For the rest, the expression varies in time, in reaction to different circumstances.

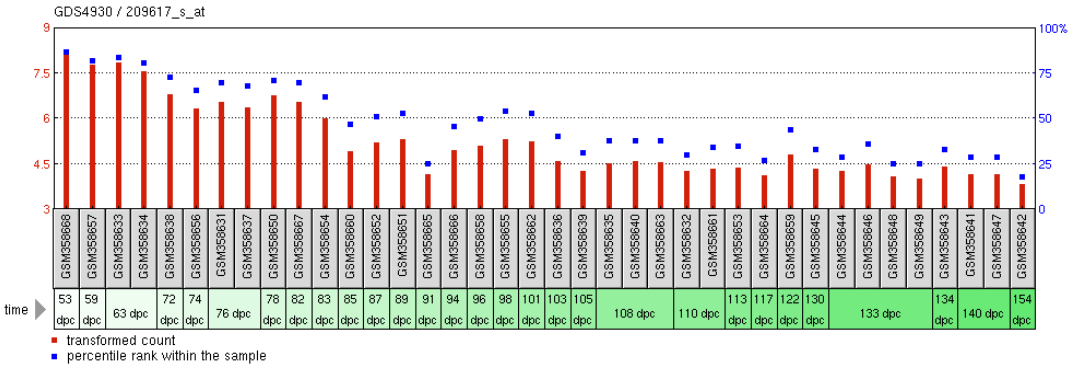


FIGURE 2.2: Gene expression profiling of catenin delta 2 (CTNND2) on human fetal lung, [*Transcriptomic analysis of human lung development, GEO*].

2.1.1 Time Series Data Analysis

The expression level of a gene can be determined by measuring the amount of mRNA present at a particular instant. Several quantitation methods are in use for different stages and elements of the gene expression process. Commonly used methods are microarray technology, real-time PCR and Next Generation Sequencing (NGS).

Several measurements of mRNA can be collected over a time period, creating a time series of gene expression level, also known as GAP. Figure 2.2 shows the expression of gene CTNND2 (catenin delta 2) in *homo sappiens* fetal lung development over a time period of 102 days.

Samples may be taken at equally spaced time intervals, uniform sampling; or not. The measurements are sequentially recorded in the form of a discrete time series over time $t; t = t_1, t_2, ..., t_n$. Time sampling interval and number of samples taken are usually subjected to budgetary constraints [Ching, Huang, and Garmire, 2014]; [Kerr and Churchill, 2001].

2.1.2 Pathway Analysis and Discovery

Cell activity is controlled by complex communication processes, both within the cell and cell-to-cell. This communication results on metabolic, signaling and regulatory pathways, that often interact with each other. Derailed information, in turn, may cause diseases. With the aid of systems biology, the network's structure resulting

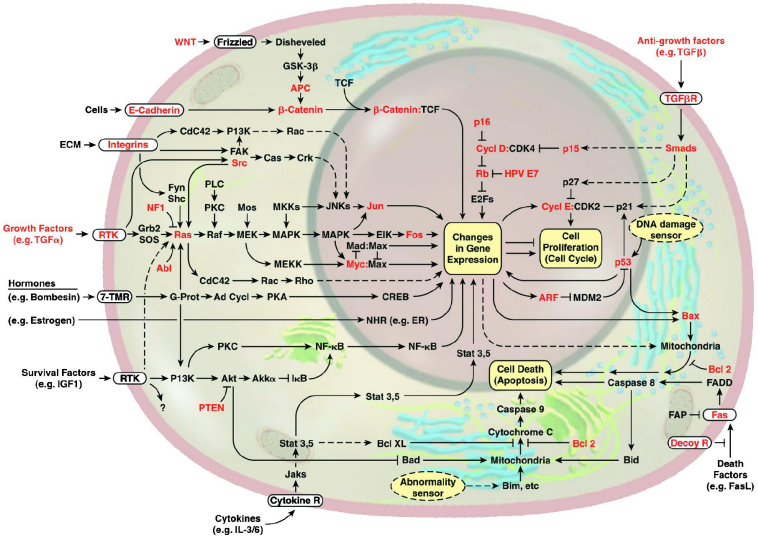


FIGURE 2.3: The Emergent Integrated Circuit of the Cell.

from these interactions can be analyzed, in hopes to decipher the rules that govern gene expression.

But learning these pathways is very challenging, and demands the use of complex algorithms and statistics. The analysis is not linear, but part of a cycle where several refinements are made and, at times, require decades of study. This is the case with the wiring diagram of the growth signaling circuitry of the mammalian cell [Hanahan and Weinberg, 2000], which is pictured in Figure 2.3.

2.2 Applications of Gene Expression Quantization

A full and correct understanding of the dynamics of gene regulatory networks is an essential step in personalized medicine and genetic therapy. Indeed, a central idea in genetic therapy is to intervene some selected gene or set of genes to induce a transition in the network from an unhealthy steady state to a healthy one. Such intervention needs to be performed in a way that no path in the transition passes through states that have a potential for inducing detrimental side effects. Also, short transition paths are preferred. Achieving these goals under the given restrictions requires an accurate model of the GRN and algorithms capable of simulating effectively its dynamics. Several models have been proposed for the representation of these networks [Abate et al., 2007]; [Batt et al., 2010]; [Gebert, Radde, and Weber, 2007];

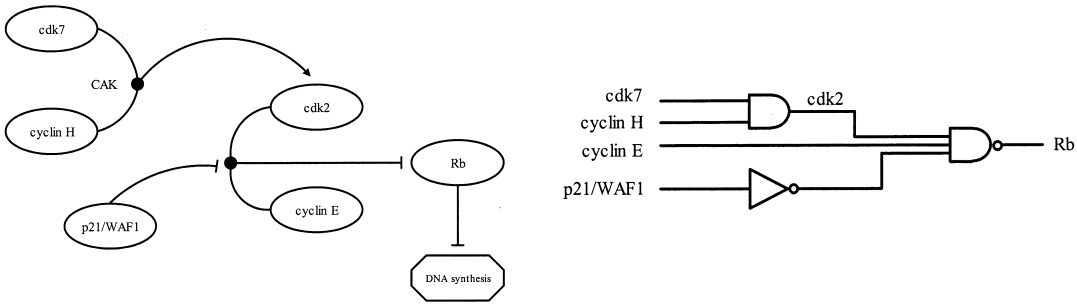


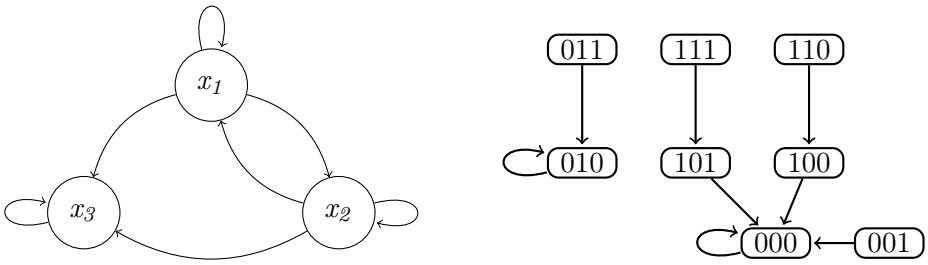
FIGURE 2.4: [Shmulevich et al., 2002] Left: A diagram illustrating the cell cycle regulation example. Arrowed lines represent activation and lines with bars at the end represent inhibition. Right: The logic diagram describing the activity of Rb protein in terms of 4 inputs: cdk7, cyclin H, cyclin E, and p21.

[Shmulevich, Dougherty, and Zhang, 2002b]; [Shmulevich, Dougherty, and Zhang, 2002a]; [Shmulevich et al., 2002]. De Jong [Jong, 2002] provides a good summary review of several of them.

2.2.1 Gene Regulatory Networks

The amounts and the temporal pattern in which gene products appear in the cell are crucial to the processes of life. The dynamics —due to both internal and external interactions— constitute the state of a system. With the aid of Computer Science and Statistics, the study of GRN dynamics has become more feasible, and several models have been developed to simulate such dynamics. Modelling is ultimately targeted to understanding the dynamic and functional characteristics of an organism. The development of an automated system capable of effectively simulating the behavior of a GRN may also provide the knowledge to alter such behavior. This alteraton of the network dynamics is referred to as intervention. The power to intervene with the network dynamics has a significant impact in diagnostics and drug design.

Although biological phenomena manifest in the continuous-time domain, in describing such phenomena we usually employ a binary language. For instance, expressed or not expressed; on or off; up or down regulated. Studies conducted restricting genes expression to only two levels (0 or 1) suggested that information retained by these when binarized remains meaningful [Shmulevich and Zhang, 2002]. This validates



(A) Boolean Network Diagram (B) State Transition Diagram

FIGURE 2.5: Boolean Network of 3 genes and its state transition diagram

the Boolean paradigm. Figure 2.4, taken from [Shmulevich et al., 2002], illustrates Boolean network modeling.

2.2.1.1 Boolean Networks (BN)

A Boolean network is a set of Boolean variables whose state is determined by other variables in the network. Formally: A Boolean network $\mathcal{N}(V, F)$ is defined by a set of nodes $V = \{x_1, ..., x_n\}$, and a list of Boolean functions $F = (f_1, ..., f_n)$. Each $x_i \in V, i = 1, ..., n$, is a binary variable representing a gene which takes value from {0, 1}. There are k_i genes assigned to gene x_i, whose value at time j determine the value at time $j + 1$ of x_i by means of a Boolean function $f_i \in F$. The BN dynamics can be described with equations (2.2) to (2.5). Figure 2.5a depicts a Boolean network of 3 genes, $V = \{x_1, x_2, x_3\}$ and F comprises the following rules:

$$f_1 : x_1(j + 1) = x_1(j) \wedge x_2(j) \tag{2.1a}$$

$$f_2 : x_2(j + 1) = \neg x_1(j) \wedge x_2(j) \tag{2.1b}$$

$$f_3 : x_3(j + 1) = x_1(j) \wedge x_2(j) \wedge x_3 \tag{2.1c}$$

A Boolean network with M genes has 2^M states. Each of these states is a string representing the pattern of expression of the individual genes, or GAPs. When the network flow gets trapped in some GAP, this GAP is known as an attractor. Attractors may be composed by more than one GAP in cycles of states. Figure 2.5b depicts the state transition flow of the Boolean network in Figure 2.5a, as per the rules given in 2.1.

Boolean network (BN) representations of gene regulatory interactions are constructed from the binary quantization of time-series of gene expression measurements. These measurements are stored in arrays of the form $G = [G_i(j)]$, $1 \leq i \leq M$, $1 \leq j \leq N$, where each i labels a gene and each j, a time instant. A binary quantization method transforms G into an $M \times N$ array $B = [B_i(j)]$ of values from the set $\{0, 1\}$. This is performed by applying to each row of G, denoted G_i, an algorithm that classifies each data-point as expressed or unexpressed, and sets its value as 1 or 0, respectively. We call these algorithms deterministic gene expression classifiers (DGEC). The vertices in a Boolean representation of a Gene Regulatory Network (GRN) are N-point binary vectors representing the states of the M genes at a particular time instant. These vertices are connected with sets of Boolean functions, each representing the influence on the state of a gene, of the states of a fixed subset of genes. In functional notation, if the state of gene i at instant $j + 1$, denoted $S_i(j + 1)$, depends on the states of the genes whose labels are in $L_i \subset \{1, ..., M\}$, and this dependence is described by a Boolean transition mapping f_i, we write

$$S_i(j + 1) = f_i([S_k(j)]),\ k \in L_i, \tag{2.2}$$

where $[S_k(j)]$ is the vector of states at instant j of the genes labelled by L_i. If the GRN is assumed to be synchronic, state transitions are defined on the basis of a fixed set $F = \{f_i : f_i \text{ transition mappings}, i = 1, ..., M\}$. In functional notation, the transition mapping based on F is

$$\delta_F : \{0, 1\}^M \to \{0, 1\}^M, \tag{2.3}$$

$$\delta_F(S_i(j)) = (f_i([S_k(j)])),\ \forall i = 1, ..., M, k \in L_i. \tag{2.4}$$

A GRN is said to be consistent with $G = [G_i(j)]$ if there is a set F such that

$$\delta_F(S_i(j)) = B_i(j + 1)\ \forall j; \tag{2.5}$$

where $B = [B_i(j)]$ is a binary quantization of G. The fundamental problem in gene regulatory inference is finding a GRN that is consistent with a given G.

The relationships between genes are determined from experimental data. A coefficient of determination (COD) is used in this endeavour to discover such associations. The COD measures the quality of a predictor in using an observed gene set to infer a target gene set, in the absence of observations. In order to further illustrate this, let x_i be a target gene, which we wish to predict by observing the set of genes $x_{i1}, x_{i2}, \ldots, x_{ik}$. Suppose that $f(x_{i1}, x_{i2}, \ldots, x_{ik})$ is an optimal predictor of x_i relative to some error measure ϵ. Let ϵ_{opt} be the optimal error achieved by f. Then, the COD for x_i relative to the set $x_{i1}, x_{i2}, \ldots, x_{ik}$ is defined as:

$$\theta = \frac{\epsilon_i - \epsilon_{opt}}{\epsilon_i} \tag{2.6}$$

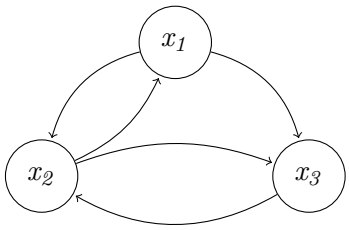
where ϵ_i is the error of the best (constant) estimate of xi in the absence of any conditional variables [Shmulevich, Dougherty, and Zhang, 2002b].

The drawback of using this formalism is that the interactions among genes are hard-wired rules. This unrealistic assumption superimposes the self-organizing nature of biological systems and, therefore, mischaracterizes their dynamics. Self-organization gives the system robustness in presence of perturbations, showing spontaneous ordered collective behavior. Boolean networks exhibit this quality through the existence of attractors and absorbing states, which act as a form of memory for the system.

Thus, there are at least two uncertainties in a solution of this problem, namely, the binary quantization of G and the existence and uniqueness of F. In Chapter 4 we propose a probability for assessing the influence of these uncertainties in a GRN solution. We do that mainly by using a method for computing probabilistic strings of gene expression states [Seguel and Lluberes, 2015].

2.2.1.2 Probabilistic Boolean Networks (PBN)

The open nature of biological systems brings about a significant uncertainty into the model. One way of coping with this difficulty is to pass the uncertainty to the predictor, by synthesizing a number of good performance predictors. Each one of them contributes its own prediction proportionally to its determinative potential, which is given by the COD. More formally, given genes $V = x_1, ..., x_n$, we assign to each



$$f_1^1 : x_1(t+1) = \neg x_2(t); \qquad c_1^1 = 1$$
$$f_1^2 : x_2(t+1) = \neg x_1(t); \qquad c_1^2 = 0.3$$
$$f_2^2 : x_2(t+1) = x_1(t) \wedge x_3(t); \qquad c_2^2 = 0.7$$
$$f_1^3 : x_3(t+1) = x_1(t); \qquad c_1^3 = 0.6$$
$$f_2^3 : x_3(t+1) = \neg x_1(t) \wedge x_2(t); \qquad c_2^3 = 0.4$$

FIGURE 2.6: Probabilistic Boolean Network and its predictors.

x_i a set $F_i = \{f_1^{(i)}, \ldots, f_{l(i)}^{(i)}\}$ of Boolean functions representing the "top" predictors for the target gene x_i. Thus, the PBN acquires the form of a graph $G(V, F)$ where $F = (F_1, \ldots, F_n)$ [Shmulevich, Dougherty, and Zhang, 2002c], and each F_i in F is as previously described. At each point in time or step of the network, a function $f_j^{(i)}$ is chosen with probability $c_j^{(i)}$ to predict gene x_i. Using a normalized COD [Shmulevich, Dougherty, and Zhang, 2002b]:

$$c_j^{(i)} = \frac{\theta_j^{(i)}}{\sum_{k=1}^{l(i)} \theta_k^{(i)}}; \tag{2.7}$$

where $\theta_j^{(i)}$ is the COD for gene x_i relative to the genes used as inputs to predictor $f_j^{(i)}$

PBNs, like Boolean networks, are rule-based. Also like Boolean networks, PBNs have a self-organizing quality. But, unlike the latter, they are not inherently deterministic but use multiple rules, or *predictors*. This makes PBNs more accurate in face of environmental and biological uncertainty. Figure 2.6 provides an example of a PBN and its predictors.

At a given instant in time, the predictors selected for each gene determine the state of the PBN. These predictors are contained on a vector of Boolean functions, where the ith element of that vector contains the predictor selected at that time instant for gene x_i. This is known as a *realization* of the PBN. If there are N possible realizations, then there are N possible vector functions, $\mathbf{f}_1, \mathbf{f}_2, \ldots, \mathbf{f}_N$, each of the form $\mathbf{f}_k = (f_{k1}^{(1)}, f_{k2}^{(2)}, \ldots, f_{kn}^{(n)})$, for $k = 1, 2, \ldots, N, 1 \leq k_i \leq l(i)$ and where $f_{ki}^{(i)} \in F_i; i = 1, ..., n$. In other words, the vector function $\mathbf{f}_k : 0, 1^n \rightarrow 0, 1^n$ acts as a transition

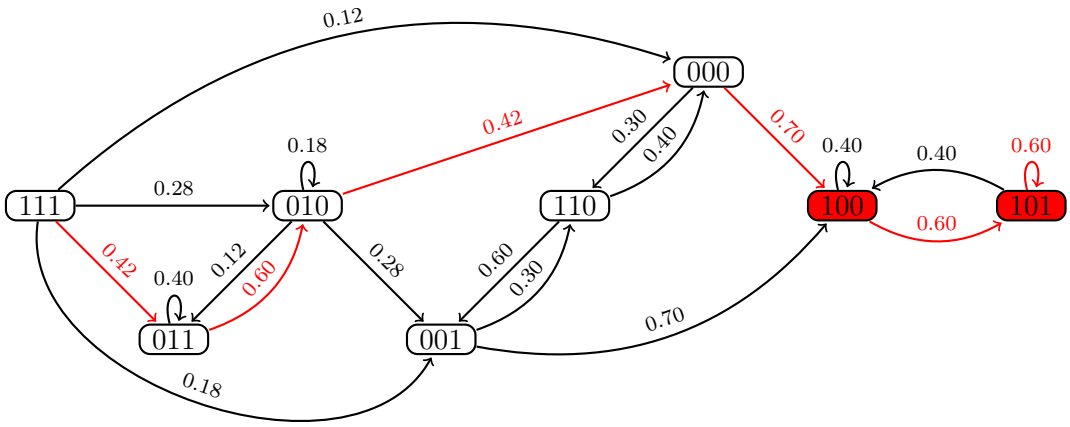


FIGURE 2.7: State Transition Diagram of a PBN. Highest probability path and attractors are collored in red.

function (mapping) representing a possible realization of the entire PBN. Thus, we have the matrix K of realizations:

$$K = \begin{bmatrix} \mathbf{f}_1 \\ \mathbf{f}_2 \\ \vdots \\ \mathbf{f}_N \end{bmatrix} \tag{2.8}$$

Assuming independence of the predictors, $N = \prod_{i=1}^{n} l(i)$. Each realization k can be selected with $P_k = \prod_{i=1}^{n} {c_{k_i}}^{i}$. The probability of transitioning from state $(x_1, \ldots, x_n)$ to $(x'_1, \ldots, x'_n)$ is given by [3]:

$$Pr\{(x_1, \ldots, x_n) \to (x'_1, \ldots, x'_n)\} = \sum_{k=1}^{N} P_k \underbrace{\left[\prod_{i=1}^{n} \left(1 - |f^i_{ki}(x_1, \ldots, x_n) - x'_i| \right) \right]}_{\in\{0,1\}} \tag{2.9}$$

Figure 2.7 shows the state transition diagram for the PBN in Figure 2.6 .

Markov theory allows the study of the dynamic behavior of PBNs in the context of Markov Chains. They explicitly represent probabilistic relationships between genes, allowing quantification of influence between genes. Because of this, PBNs are better suited than Boolean networks for modelling system uncertainties. Both BN and PBN with n genes, have 2^n states, making exhaustive analysis exponentially large.

2.3 Binarization Methods

Bellow we discuss five different binarization methods based on different heuristics. Some of the methods can be considered classical numerical methods in the sense that they provide a numerical approximation through the computation of a threshold value explicitly. That is the case of the first three methods presented. The last two algorithms classify data through clustering. For purposes of uniformity, clustering results are represented with a cluster threshold. Binarized results obtained from different binarization methods often differ from each other, creating discrepancies between binarizations.

2.3.1 Average Jump

The expression of a gene may vary along time. These variations on a time series data — or jumps— may provide information that aids in finding an expression that can be used, in turn, as a clear differentiation between expressed or not expressed states of the gene.

The use of the average of all data jumps as a threshold signal is proposed in [Shmulevich and Zhang, 2002] . This method, referred here as *Algorithm A*, compares the rate of change of each pair of consecutive data points. First, it receives a time series data vector G_i and sorts it in increasing order. It then computes the average of all jumps between points, and sets the smallest data point whose jump exceeds this average, as the threshold. The original input array is then binarized using this threshold value as criterion. This is:

$$\pi_i(j+1) - \pi_i(j),\ j = 1, ..., N-1, \tag{2.10}$$

where π_i stands for the input vector G_i sorted in increasing order, with

$$A = \frac{\pi_i(N) - \pi_i(1)}{N-1}, \tag{2.11}$$

the average rate of change. The threshold is the highest point in the first pair whose rate of change is above A.

The main advantages of *Algorithm A* are conceptual and computational simplicity, and the guarantee of a unique solution, if such solution exists. If the sorted data points correspond to a horizontal line, there are no jumps between time points and therefore *Algorithm A* returns no answer. In cases where the sorted data can be closely adjusted with a straight line, there is no difference in the size of the jumps and the threshold value produced by the method is questionable. This is also the case when jumps significantly higher than the first jump above the average occur someplace after the first jump above average.

When run as a subroutine for each gene on the set, *Algorithm A* returns the M thresholds of an $M \times N$ input array G in $O(MN)$ time, using $O(MN)$ space.

In the description below, each G_i is the k-point time series expression of the i-th gene in a gene set and B_i its binarization.

Algorithm A Binarize G_i

Input: G_i
$S_i \leftarrow$ sort $(G_{i,1}...G_{i,k})$
for $j = 1$ to $k - 1$ **do**
  $D_{i,j} \leftarrow (S_{i,j+1} - S_{i,j})$
end for
$t \leftarrow (S_{i,k} - S_{i,1})/(k - 1)$
$m = min\{j : D_{i,j} > t\}$
for $j = 1$ to k **do**
  if $G_{i,j} \geq S_{i,m+1}$ **then**
    $B_{i,j} \leftarrow 1$
  else
    $B_{i,j} \leftarrow 0$
  end if
end for
return S_m, B_i

2.3.2 Multiscale Analysis

Binarization Across Multiple Scales (BASC) algorithm [Hopfensitz et al., 2012], referred here as *Algorithm B*, approaches the input data with a sequence of step functions. The method scores the jumps before deciding which one is the threshold. An *a posteriori* analysis assesses the reliability of this choice.

The algorithm consists of several processes. In general, the method finds step functions with different number of steps, which are also the best approximations to the sorted version π_i of the input data. It starts with the one that fits exactly the input data and each subsequent step function has less steps than the previous one. At each approach, the point where the highest jump occurs is identified for further analysis. A relation between the highest jump and the approximation error to π_i incurred by the step function is then computed. A high value for this ratio indicates a strong discontinuity in the sorted data and, therefore, a potential threshold candidate. The highest ratio in each approximation is assumed to be the signal of a potential change in the state of the gene expression. The indices in π that correspond to these ratios are stored in the so-called *vector of strong discontinuities*. The threshold is defined as

$$\tau_{B,i} = \frac{\pi_i(\lfloor\gamma\rfloor) + \pi_i(\lfloor\gamma\rfloor + 1)}{2}, \tag{2.12}$$

where γ is the median of the vector of strong discontinuities.

Step functions are computed with a dynamic programming algorithm that returns a sequence of step functions of minimal Euclidian distance to π_i. With each step function approximation, a cost and break point index is calculated and stored. The cost of a step of the function is the distance to the mean of the approached data segment. The cost of the step function, in turn, is the sum of the costs of its steps. Both, the costs and break point indices are computed using the algorithms whose pseudo code are presented below. As in *Algorithm A*, the first step is sorting the points in the time series.

Algorithm B.2 reconstructs the break points from the array Ind, computed with *Algorithm B.1*. Break points are used to compute the jump size h. The error of

approximation e is the Euclidean distance of the step function to the sorted input data set. The maximum of the radios $q = h/e$, determines the strongest discontinuity. We refer the reader to [Hopfensitz et al., 2012] for further details on this method.

Algorithm B has higher time and space complexity. In particular, the dynamic programming matrix used to find the step function with minimal Euclidean distance to the original data, grows quadratically with the number of input points. *Algorithm B* returns the M thresholds of a $M \times N$ input array G in $O(MN^3)$ time, using $O(MN^2)$ space.

Just as *Algorithm A*, *Algorithm B* fails in cases where the sorted data can be closely adjusted with a straight line.

Algorithm B .1 Calculation of Optimal Steps Functions

$C_i(0) = c_{iN}, i = 1, ..., N$
for $j = 1$ to $N - 2$ **do**
  for $i = 1$ to $N - j$ **do**
    $C_i(j) \leftarrow \min_{d=i...N-j}(c_{id} + C_{d+1}(j-1))$
    $Ind_i(j) \leftarrow \text{argmin}_{d=i...N-j}(c_{id} + C_{d+1}(j-1))$
  end for
end for

Algorithm B.2 Compute Break-points of Optimal Functions

for $j = 1$ to $N - 2$ **do**
  $z = j$
  $P_i(j) = Ind_1(z)$
  if $j > 1$ **then**
    $z \leftarrow z - 1$
    for $i = 2$ to j **do**
      $P_i(j) \leftarrow Ind_{Pi-1(j)+1}(z)$
      $z \leftarrow z - 1$
    end for
  end if
end for

2.3.3 One-Step Approximation

The next method, referred to as *Algorithm C* [Seguel and Lluberes, 2015], is inspired on StepMiner computational method [Sahoo et al., 2007]. The aim of StepMiner is determining which 1-step or 2-step function best fits the data. In 2-step functions, the first and third segments have the same value. Step functions approximations are computed with linear regressions with 1 or 3 degrees of freedom. The least square errors of the approximations provide a set of F-statistics, whose P-value is used for deciding whether the error is significant. If it is not, the method makes no gene expression decision.

Unlike Step Miner, *Algorithm C* sorts the input data in increasing order, and finds only a 1-step function approximation. Data sorting provides an adequate organization of the variation of the expression profile. After sorting the data, *Algorithm C* uses the 1-step routine of StepMiner to find the subset of all one-step functions that approximate $\sigma^{(i)}$ within a significance of .05. This produces a basic subset of 1-step functions. If the subset is empty, the method returns *"undecidable"*. Otherwise, the method selects from the subset, the step function whose steps are further apart. The midpoint between the steps of the selected two-step function is the threshold returned by *Algorithm C*.

Algorithm C returns the M thresholds of a $M \times N$ input array G in $O(MN^2)$ time, using $O(MN)$ space. Here, $SSOT$ is the total sum of squares, SSE is the sum of square error, SSR is regression sum square, MSR the regression mean square and MME is the error mean square.

2.3.4 2-means Clustering

This method, refered here as *Algorithm D*, is the well-known k-means classification algorithm [MacQueen, 1967] —also called Lloyd's algorithm [Lloyd, 1982]— with $k = 2$. No data sorting is necessary. The method groups the data in two clusters around some specific points called centroids, without explicitly computing a threshold. We define the threshold of *Algorithm D* as the mid-point between the cluster's centroids.

Algorithm C One-Step Aproximation

Input: G_i, sig

$S_i \leftarrow$ sort G_i

$SSOT \leftarrow \sum(S_i - mean(S_i))^2$

for $j = 1$ to $n - 1$ **do**

$p_{1,j} \leftarrow mean(S_{i,1} \ldots S_{i,j})$

$p_{2,j} \leftarrow mean(S_{i,j+1} \ldots S_{i,n})$

$d_j \leftarrow |p_{1,j} - p_{2,j}|$

$SSE \leftarrow \sum(S_i - p)^2$

$SSR \leftarrow SSOT - SSE$

$MSR \leftarrow SSR/2$

$MME \leftarrow SSE/(n-3)$

$F_j \leftarrow MSR/MME$

end for

$Y \leftarrow pdf(F)$

$Pval \leftarrow Y/(n-1)$

$e \leftarrow \arg_j(\max d_j : Pval_j \leq sig)$

$T \leftarrow (p_{1,e} + p_{2,e})/2$

return T

The method works with almost any kind of data and is oblivious to the sizes of the data jumps or shape of the curve adjusting the sorted input data. *Algorithm D* may however, incur misclassifications, especially when the data oscillates, and in which the method may not be stable [MacQueen, 1967].

An interesting variation of the classical 2-mean classification of gene expression states is the iterative clustering method [Berestovsky and Nakhleh, 2013], a method designed to smooth the effects of small data oscillations in the classification. However, the statistical behavior of iterative clustering turned out to be similar to that of the 2-means algorithm. Thus, for our purposes and for the sake of simplicity, we selected the latter.

Algorithm D returns the M thresholds of an $M \times N$ expression array in $O(kiMN)$ time, where i is the number of iterations needed until centroids converge, using $O((N+k)M)$ space.

Algorithm D 2-means Clustering

Input: $G_i; k = 2$

$c_{nj} \leftarrow$ random pick from $\{G_i\}; j = 1, ..., k$ // Initialize centroids

while $c_{nj} \neq c_{oj}$ **do**

  $c_{oj} \leftarrow c_{nj}$

  $l_i \leftarrow \text{argmin } \|G_i - c_{oj}\|^2$

  $S_j \leftarrow \{G_i : l_i = j\}$ // Construct clusters

  $c_{nj} \leftarrow \frac{1}{|S_j|} \sum_{G_i \in S_j} G_i$ // Compute new clusters' centroids

end while

return c_{nj}

2.3.5 Median Separation

This method, proposed in Seguel and Lluberes [Seguel and Lluberes, 2013], is a variant of 2-means that replaces the mean with the median. We refer to this method as *Algorithm E*.

Algorithm E first sorts the m input data points, and for each value $j, 1 \leq j \leq m$, computes the median of the first j points and that of the last $m - j - 1$ data points.

It then finds the pair of medians that are farther apart and returns their average as the threshold. In both *Algorithm E* and 2-means, two points —an upper and a lower value— determine the binary quantization. Therefore, if there is a threshold, this is most probably given by their average.

Algorithm E returns the M thresholds of an $M \times N$ input array G in $O(MN)$ time, using $O(MN)$ space.

Algorithm E Median Separation

Input: G_i

$S \leftarrow$ sort $(G_{i,1}...G_{i,m})$

for $j = 1$ to $m - 1$ **do**

  $lm_j \leftarrow median(S_1...S_j)$

  $um_j \leftarrow median(S_{j+1}...S_m)$

  $A_j \leftarrow |um_j - lm_j|$

end for

$Ind \leftarrow$ argmax $_{d=1...m}(A)$

$lmMax \leftarrow lm_{Ind}$

$umMax \leftarrow um_{Ind}$

$T \leftarrow (lmMax + umMax)/2$

return T

The next chapter analyses the variation of the results of these methods in light of discretization error and model uncertainty.

Chapter 3

Gene Expression Threshold Computations

"Nothing in life is to be feared, it is only to be understood. Now is the time to understand more, so that we may fear less." — Marie Skłodowska-Curie

This chapter examines the roots of model and discretization uncertainties, and proposes methods for estimating them.

3.1 Semantics and Accuracy of GETC

In standard scientific computing practices, algorithms for phenomena study and representation are usually designed on the basis of explicit, well-defined mathematical models. These models are abstractions of the phenomenon under consideration; they are used for verification and validation purposes, and for numerical simulations. However, because accurate inner working of cellular mechanisms remains unknown, most computer methods designed for their study are based on implicit models of the phenomenon and its hypothetical manifestation in the data. As a consequence, the phenomenon simulated and the interpretation of its results, is concealed by the absence of algorithmic-independency. This also has an impact on how the phenomenon is perceived and interpreted for further analyses or modeling endeavors [Kim et al., 2013].

This lack of abstract mathematical models makes computer algorithms the expression of a conjecture, which cannot be formally stated and is not independently verifiable. This matter is particularly noticeable in the case that each method would render a different threshold value for the same input data.

3.2 Natural threshold, Convergence threshold and Computation

Gene expressions are the result of a cascade of processes that are stochastic in nature. However, by the Law of Large Numbers, a smooth non-negative real valued function can approximate the average expression behavior of a significantly large number of cells, in an interval of time [Klebanov and Yakovlev, 2008]. Provided that the variations in this function are large enough, the lowest and highest values can be associated with expressed and non-expressed gene states, respectively. Hypothetically, someplace within the function range, there should be the point in which Nature separates the two states. We call this hypothetical point *Natural Threshold* (NT). We plan to answer the semantics question by investigating to what extent the result of the methods reveal a NT.

In all the methods considered, the threshold varies with the number of input data points. In order to bound these variations, we compute the *Convergence Threshold* (CT). This threshold is obtained by iterative refinements of the method's thresholds, up until the values fall within a predetermined error tolerance *tol*. The data for the iterative refinements is taken from a cubic spline interpolation of the input time series [Bar-Joseph et al., 2003]; [Chiu et al., 2015].

For time-course data, it is reasonable to assume that each G_i is an N-point sample of a continuous, real-valued gene expression function g_i, which is defined on $[0, 1]$. The samples are taken at instants t_j, with $t_{j-1} < t_j$. Thus, for each $i = 1, ..., M$,

$$G_i(j) = g_i(t_j), \forall j = 1, ..., N. \tag{3.1}$$

We assume that g_i can be approached with a cubic spline interpolator f constructed on the basis of G_i. Thus, given an N-point G_i, we construct a $(2N-1)$-point vector, denoted G_i^1, by assigning

$$G_i^1(2j) = \begin{cases} f\left(\frac{t_{j-1}+t_j}{2}\right), & f\left(\frac{t_{j-1}+t_j}{2}\right) \geq 0 \\ 0, & \text{otherwise.} \end{cases} \tag{3.2a}$$

$$G_i^1(2j-1) = G_i(j) \tag{3.2b}$$

A sequence $G_i^0, \ldots, G_i^L$ is called *sequence of cubic spline interpolations* (SCSI) of G_i if

1. $G_i^0 = G_i$, and
2. $\forall n \geq 1, G_i^n$ is constructed from G_i^{n-1} through (3.2).

Given G_i and $X \in \{A, B, C, D\}$, we define

$$\Psi_X(G_i) = \{X(G_i^n) : G_i^0, ..., G_i^L \text{ a SCSI}\}. \tag{3.3}$$

The CT is then defined as the $\Psi_X(G_i^n)$ such that

$$|\Psi_X(G_i^n) - \Psi_X(G_i^{n-1})| \leq tol. \tag{3.4}$$

The CT is also used to assess the accuracy of the methods. We do this by comparing the method's CT with the threshold of the original time series G_i. We also compare the binary expression matrices derived from these thresholds.

3.3 Assessment of Variations in Gene Expression Threshold Computations: Case Study

For the purposes of this study, we classify the four methods considered according to their approach to the search for a threshold. This renders what we call *jump-based* and *cluster-based* methods. These methods were discussed in section 2.3.

Jump-based methods use data variations —or jumps—, between data points, as a reference for the determination of a threshold value. *Algorithm* A and *Algorithm* B, for this study, are also referred as *Jb1* and *Jb2*, respectively.

Cluster-based methods partition the input data set into a predetermined number k of disjoint data subsets, referred as clusters. The partition is made on the basis of the nearest center of each cluster. Here, we compare *Algorithm* D and *Algorithm* E, also referred as *Cb1* and *Cb2*, respectively.

In order to assess the variations in the threshold value returned by each method, we conducted two experiments. The first uses the threshold computed by each method and the second, the CT of each method. Both experiments were performed with two different data sets. The first data set [Cho et al., 1998], which corresponds to the mitotic cell cycle of yeast, is taken from [*Yeast Cell Cycle Analysis Project*]. We took the 17-point time series of four genes, namely cdc24, cdc19, cdc15, and cdc27. The second data set is a 6 x 8 randomly generated matrix of real values between 0 and 1, mimicking an eight point time series of six genes.

Supplementary information for these experiments is included in Appendix A. All experiments were run with Matlab 7.0.12.635 (R2011a) for Mac.

3.3.1 Numerical Variations of the Thresholds

Figure 3.1 shows the thresholds obtained in the first experiment. Algorithms B and D (*Jb2* and *Cb1*) exhibit the closest numerical values, while the thresholds returned by Algorithms A and E (*Jb1* and *Cb2*) are significantly farther apart. All threshold values are shown in Tables 3.1 and 3.2.

In order to quantify these observations, we compute the ratio $d_{max}/range$, where d_{max} is the largest distance between thresholds for a given time series, and *range* is the difference between the largest and smallest values in the time series. The results of these computations are depicted in Figure 3.2.

For a more algorithmic-centered classification, we define the *distance between methods* as the Euclidian distance between the vectors formed by the thresholds of each time

TABLE 3.1: Threshold values for random generated data. Highlighted values are matching values on both experiments. Here, Jb1 is *Algorithm A*, Jb2 is *Algorithm B*, Cb1 is *Algorithm D* and Cb2, *Algorithm E*.

Experiment 1				Experiment 2			
Jb1	Jb2	Cb1	Cb2	Jb1	Jb2	Cb1	Cb2
0.28072	0.54384	0.50574	0.45214	0.28072	0.48069	0.4671	0.18751
0.2805	0.4284	0.45131	0.41986	0.2805	0.37147	0.40742	0.23781
0.34758	0.45847	0.49741	0.44159	0.34758	0.77806	0.70254	0.37476
0.40338	0.50213	0.51927	0.47716	0.32267	0.50211	0.51097	0.47145
0.51041	0.71765	0.66833	0.64514	0.34708	0.67214	0.64782	0.32805
0.13834	0.60205	0.30395	0.61079	0.13834	0.50511	0.52622	0.15539

TABLE 3.2: Threshold values for cdc data set. Highlighted values are matching values on both experiments. Here, Jb1 is *Algorithm A*, Jb2 is *Algorithm B*, Cb1 is *Algorithm D* and Cb2, *Algorithm E*.

Experiment 1				Experiment 2			
Jb1	Jb2	Cb1	Cb2	Jb1	Jb2	Cb1	Cb2
0.1981	0.27476	0.27286	0.30952	0.18095	0.25524	0.25531	0.19169
5.6067	4.1914	3.7892	3.8212	0.85382	3.2098	3.3162	0.89969
0.057143	0.080952	0.082453	0.087976	0.057143	0.080952	0.082453	0.087976
0.11143	0.1181	0.12625	0.125	0.11143	0.11976	0.12571	0.1196

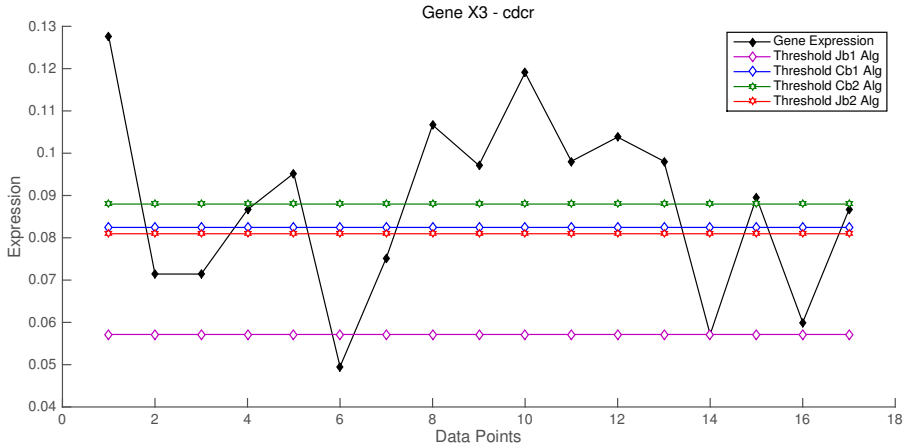


FIGURE 3.1: Thresholds plots of the four methods for one time series on Experiment 1. Here, Jb1 is *Algorithm A*, Jb2 is *Algorithm B*, Cb1 is *Algorithm D* and Cb2, *Algorithm E*.

series of genes. Table 1 shows the distances between each pair of methods. The shortest and largest distances are highlighted, as well.

In both experiments and with the cdc data, the distance from *Algorithm* A to *Algorithm* D is the largest. In turn, *Algorithm* D and *Algorithm* E are separated by the shortest distance, followed closely by algorithms *A* and *E*. We also compared the variations of each algorithm with respect to the input data sets. The results are reported in Table 3.3. Table 3.4 shows the Euclidean distance between the two experiments, for each of the four methods. Among the methods, *Algorithm A* turned out to be the less stable as it has both the shortest and largest distances between data sets.

TABLE 3.3: Euclidean distance between different methods on same experiment. Here, Jb1 is *Algorithm A*, Jb2 is *Algorithm B*, Cb1 is *Algorithm D* and Cb2, *Algorithm E*.

		Jb1—Jb2	Jb2—Cb1	Cb1—Cb2	Jb1—Cb1	Jb2—Cb2	Jb1—Cb2
Exp 1	cdc	1.41759	0.40229	0.04899	1.81927	0.37196	1.78929
	random	0.60920	0.30835	0.32162	0.40995	0.12134	0.55177
Exp 2	cdc	2.35729	0.10658	2.41736	2.46367	2.31099	0.05689
	random	0.71131	0.09108	0.67506	0.67320	0.71282	0.18449

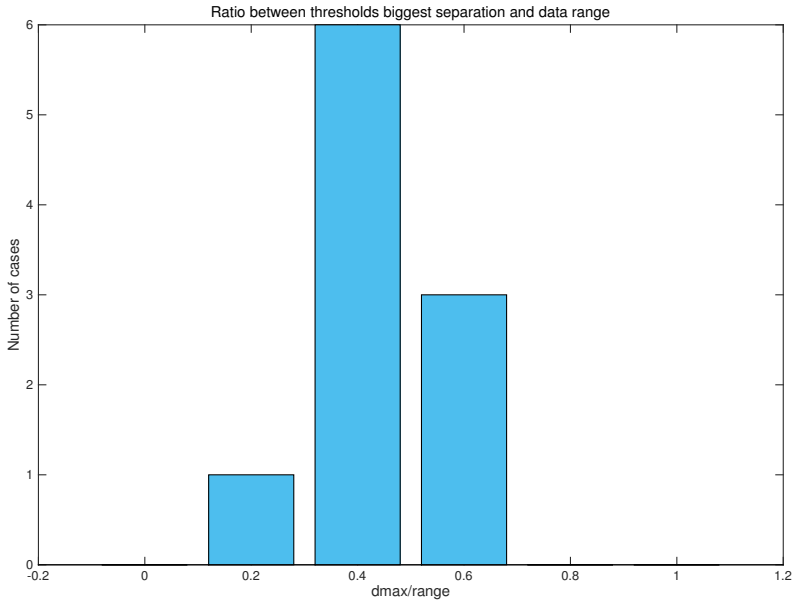


FIGURE 3.2: Relations between maximal distance between threshold and data range.

3.3.2 Variations in Binarization

We use the Hamming distances divided by the number of data points to measure the differences between the binarizations derived with each threshold on the same time series. As an example, Table 3.5 shows the binarizations results for the cdc data set. The highlighted rows are pairs of different methods whose binarizatons coincide. The methods with the largest number of coincidences are *Algorithm* B and *Algorithm* D.

Table 3.6 shows the Hamming distances between the binarizations computed with the convergence threshold of each method. The closest methods are again, *Algorithm* B and *Algorithm* D. In turn, the distances between algorithms A and E, and algorithms A and B are the largest.

TABLE 3.4: Euclidean distance between same methods on different experiments. Here, Jb1 is *Algorithm A*, Jb2 is *Algorithm B*, Cb1 is *Algorithm D* and Cb2, *Algorithm E*.

		Jb1—Jb1	Jb2—Jb2	Cb1—Cb1	Cb2—Cb2
Exp 1 vs Exp 2	cdc	4.75291	0.98180	0.47333	2.92389
	random	0.18218	0.34761	0.30885	0.64467

TABLE 3.5: Comparison between binary quantization matrices of same methods using thresholds obtained from both experiments. Matching binarizations are highlighted. Cdc data set. Here, Jb1 is *Algorithm A*, Jb2 is *Algorithm B*, Cb1 is *2*-means and Cb2, *Algorithm E*.

	Original	Convergence
Jb1	11011111111111111	11111111111111111
	00000000000000000	11111111111111111
	11111011111111111	11111011111111111
	01111111111011110	01111111111011110
Jb2	00001011010010100	10001011110010100
	00000000010000000	00000000010000000
	10011001111110101	10011001111110101
	01111111111010110	01111111111010110
Cb1	00001011010010100	10001011110010100
	00000000010000000	00000000010000000
	10011001111110101	10011001111110101
	01111111111000110	01111111110010110
Cb2	00001010010000000	11011111111111111
	00000000010000000	11111111111111111
	10001001111110100	10001001111110100
	01111111110010110	01111111111010110

TABLE 3.6: Distances between all methods expressed as Hamming distance. Lowest and highest scores are highlighted. Here, Jb1 is *Algorithm A*, Jb2 is *Algorithm B*, Cb1 is *2*-means and Cb2, *Algorithm E*.

		Original			Convergence		
		Jb2	Cb1	Cb2	Jb2	Cb1	Cb2
cdc data	Jb1	0.23529	0.25	0.32353	0.45588	0.47059	0.13235
	Jb2	—	0.014706	0.088235	—	0.014706	0.38235
	Cb1	—	—	0.073529	—	—	0.39706
Random data	Jb1	0.3125	0.22917	0.3125	0.4375	0.41667	0.10417
	Jb2	—	0.083333	0	—	0.020833	0.33333
	Cb1	—	—	0.083333	—	—	0.3125

3.4 Analysis of Experiments

The results show that the thresholds computed by the four methods are significantly different. This is a clear rejection of the hypothesis that the methods compute the algorithmic-independent value, referred in this chapter as Natural Threshold. As expected, convergence threshold differs from thresholds. This sensitivity to the sample size is also a negative answer to the question of the accuracy of the methods. Also, the convergence thresholds produced binary expression matrices that are significantly different to the ones obtained by the thresholds of each method. An important implication that can be drawn from these observations is that the models of gene regulatory networks, whose construction uses binarization as a first step, are biased by the choice of the binarization method. The success of some PBN representations of GRNs suggests that this bias is being corrected, in part, with the incorporation of prior gene interconnection knowledge, and expected results.

The difficulties in determining a numerical threshold may arise from the intrinsic nature of gene expressions. Both assumptions, jumps in the data or statistical separation in two groups may be too strict, in some sense, as data may have some natural perturbation or noise. It may be the case that on average, expressed and not expressed gene states are separated in nature by an interval, not a point. In the interval model,

expressed states will correspond to values above the interval's upper limit while non-expressed states, to values below its lower limit. And these expression values that fall within the interval shall be declared noisy data-points. Threshold intervals in gene expression time series may be investigated by adding filters that eliminate expression values that are too close to the threshold points returned by the previous methods.

Threshold computation is not a large-scale problem, at least not with the amount of data compilation currently available. However, this may change as models evolve and parameters, such as time, are incorporated. In such cases, parallel and distributed computing versions of the algorithms will be a necessary algorithmic development. Most probably, because of the strong interdependence of data expression, these methods will be mostly implemented in shared memory systems.

Chapter 4

A Unified Approach to the Computation and Analysis of Strings of Gene Expression States

"Imagination is more important than knowledge. Knowledge is limited. Imagination encircles the world." —Albert Einstein

In the previous chapter, it was shown that the problem of deciding whether or not a gene is in expressed state is crucial in the Boolean representation of gene regulatory networks. The absence of independent mathematical models for gene expression affects the answers to this problem; the model used for algorithm design and the availability of measurements brings uncertainties to the solution provided.

In this chapter, we propose an algorithmic framework to handle these uncertainties. In the proposed framework, first we compute the probability distributions of heterogeneous methods and use aggregation rules on them, in a unifying approach. The result is a set of probabilistic strings with metrics for comparative analysis. One string is elected from the set through a voting mechanism on the state of the gene for each measurement. The method is applied on biological data sets and the results are compared with those of four previously published algorithms.

4.1 Model Uncertainty and Discretization Uncertainty

The absence of a mathematical formulation of the gene expression phenomenon makes bounding the uncertainties described in 1.3 impossible. As a result, the classification of the expression values that are close to a computed threshold is not reliable. In fact, some DGECs filter out data points that are close to the numeric threshold, rendering the expression state associated to these points undecidable. The lack of independent mathematical descriptions is, on the other hand, filled in part with the rules that underly a DGEC design. In our opinion, this is one of the main values of the DGEC approach to the gene expression decision problem. This research adds two new elements to this approach. First, an expansion of the DGEC rules through the aggregation of existing rules and a collective threshold decision by vote. The vote algorithm is, in fact, a generalization of existing DGECs, as it includes them as particular cases. The method returns a string over the alphabet $\{0, 1, \perp\}$, where $\perp$ indicates that the state of the gene is undecidable. Second, the assignment to each value in the expression profile of three probabilities, namely, the probability of corresponding to an unexpressed gene, the probability of corresponding to an expressed gene, and the probability that the state of the gene is undecidable. These, in turn, define a set of string of states, each with its own probability. The mean of these probabilities and other basic statistics help providing a sense of the potential statistical significance of the elected string.

4.2 Four Deterministic Methods

We selected four DGEC, A, B, C and D, for this work (see section 2.3 for a detailed description of each method). We use the functional notation

$$\tau_{X,i} = X(G_i),\ X \in \{A, B, C, D\}, \tag{4.1}$$

to represent the input-output relation. The thresholds $\tau_{X,i}$ are independent of the time elapsed between measurements, and the input-output relations are linear with

TABLE 4.1: Correlation of the outputs of four DGEC algorithms.

	A	B	C	D
A	1	0.1007	0.1907	0.2345
B	0.1007	1	0.1810	0.7030
C	0.1907	0.1810	1	0.3237
D	0.2345	0.7030	0.3237	1

respect to scalar multiplication. This is,

$$\alpha\tau_{X,i} = X(\alpha G_i), X \in \{A, B, C, D\}, \tag{4.2}$$

where α is a scalar.

4.2.1 Correlation

The Pearson correlation of the thresholds returned by the algorithms on $1,000$ randomly generated 10-point expression vectors is shown in Table 4.1. As the table shows, the outputs show no significant correlations. This is a direct consequence of what we have called model uncertainty.

4.2.2 Threshold Displacements

The *threshold displacement* of an expression profile G_i under algorithm X is defined as

$$d_{X,i} = |\max \Psi_X(G_i) - \min \Psi_X(G_i)|. \tag{4.3}$$

Figures 4.1 and 4.2 illustrate threshold displacements of two gene expression profiles. The thresholds for these profiles and their spline interpolations are basically stabilized after 4 iterations. Thus, the displacements in the figures correspond to sequences $G_i^0, ..., G_i^4$, for each algorithm. Data for the expression profiles was taken from the Gene Expression Omnibus (GEO) repository [*Gene Expression Omnibus*].

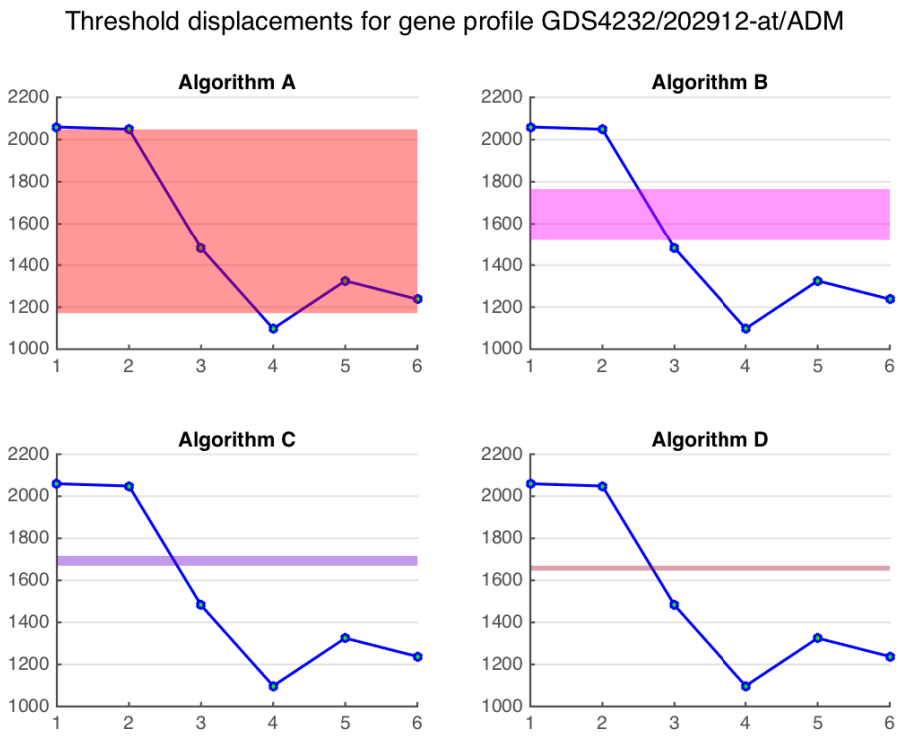


FIGURE 4.1: From left to right and top to bottom: threshold displacements for algorithms A, B, C and D.

4.3 An Algorithmic Framework

This section describes an algorithmic framework for electing a string of expression states denoted by Z_e and computing the set Z with 3^N probabilistic strings over the alphabet $\{0, 1, \perp\}$ which Z_e belongs in. It is worth remarking that most of the strings in Z do not correspond to the output of a DGEC. The framework provides some statistics of Z_e with respect to the set of strings in Z. Figure 4.3 depicts the main processes in this framework.

4.3.1 Estimating Threshold Displacements

We use the expected value $\mathbb{E}[\Delta_X]$ of the random variable

$$\Delta_X =: \text{“}\textit{Displacement of threshold returned by Algorithm } X\text{”} \tag{4.4}$$

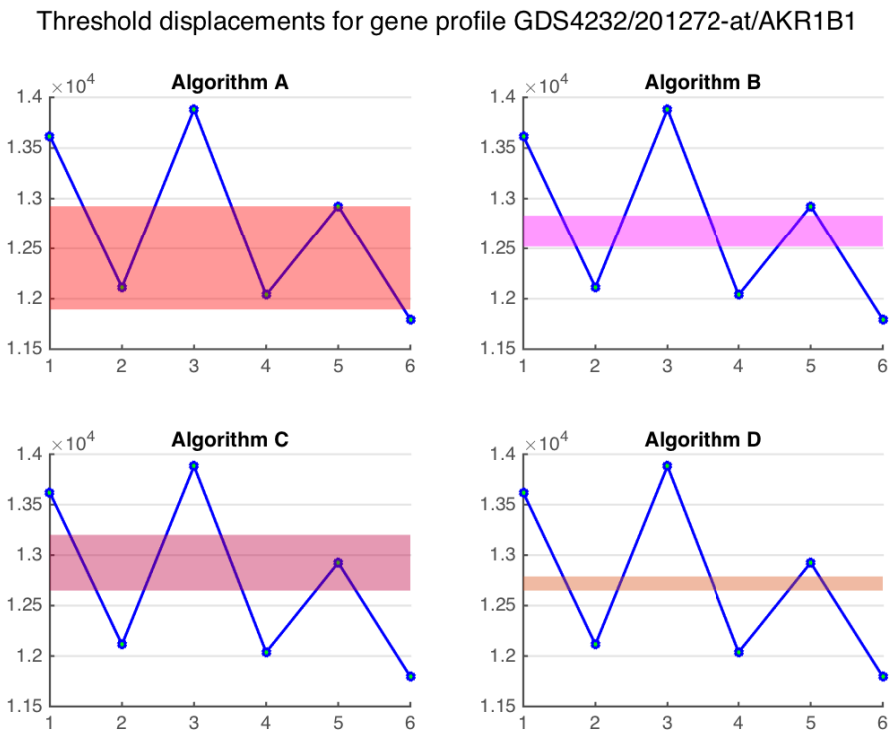


FIGURE 4.2: From left to right and top to bottom: threshold displacements for algorithms A, B, C and D.

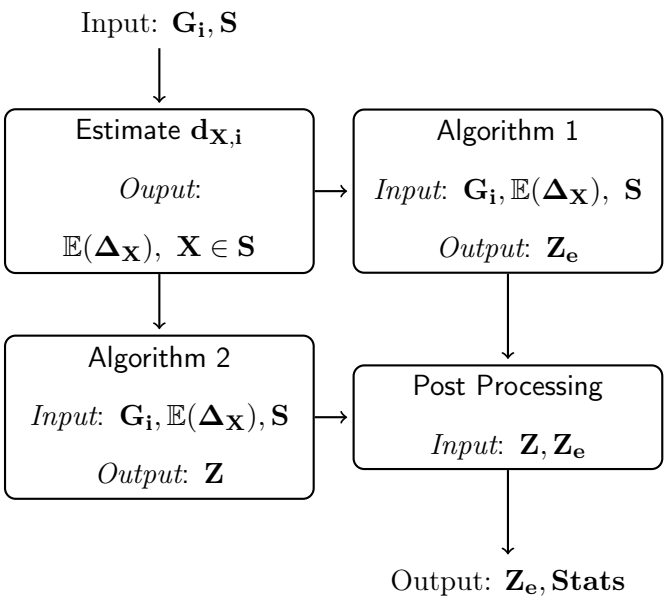


FIGURE 4.3: Algorithmic Framework Diagram

to estimate the threshold displacement $d_{X,i}$ described in equation (4.3). Since threshold displacements are approximately bounded by the range

$$\rho_{X,i} = \max(G_i) - \min(G_i), \tag{4.5}$$

for each $k = 1, \ldots, 10$, we estimated $\mathbb{E}[\Delta_X]$ with 100 randomly generated profiles of range $\frac{k}{10}$. The results are shown in Table 4.2.

4.3.2 Election of a String of Expression States

The election algorithm is based on concepts originally developed in the context of jurisprudence [List, 2012]. Underlying the design of the method is a quadruple (S, Γ, R, Λ), where S is finite set of decision algorithms, Γ a finite set of logic statements called *agenda*, R an aggregation rule, and Λ, the *doctrine*, is a logical statement describing the decision problem in terms of the statements in the agenda [Seguel, 2015]. In this particular instance, $S \subset \{A, B, C, D\}$ and $\Gamma = \{U, N\}$ where, U and N are the Boolean variables defined by the predicates:

$$U =: (G_i(j) + d_{X,i} < \tau_{X,i}) \wedge |\tau_{X,i} - G_i(j)| > d_{X,i}, \tag{4.6}$$

$$N =: |\tau_{X,i} - G_i(j)| \leq d_{X,i}. \tag{4.7}$$

Table 4.2: Expected values of threshold displacements.

Range	$\mathbb{E}[\Delta_A]$	$\mathbb{E}[\Delta_B]$	$\mathbb{E}[\Delta_C]$	$\mathbb{E}[\Delta_D]$
0.1	0.0297	0.0234	0.0165	0.0122
0.2	0.0490	0.0366	0.0292	0.0188
0.3	0.0699	0.0580	0.0508	0.0252
0.4	0.0845	0.0785	0.0480	0.0307
0.5	0.1107	0.0938	0.0660	0.0397
0.6	0.1356	0.0967	0.0823	0.0432
0.7	0.1435	0.1107	0.0796	0.0502
0.8	0.1795	0.1425	0.0975	0.0570
0.9	0.1949	0.1557	0.1389	0.0685
1.0	0.2244	0.1732	0.1487	0.0691

Clearly, $U = 1$ means that *Algorithm X* decides that $G_i(j)$ corresponds to an unexpressed gene. In turn, $N = 1$ means that *Algorithm X* decides that the state of $G_i(j)$ is not decidable. The doctrine Λ defines the new Boolean variable

$$E \iff \neg(U \vee N). \tag{4.8}$$

Thus, $E = 1$ means that the decisions made by *Algorithm X* are equivalent to decide that $G_i(j)$ corresponds to a gene in expressed state.

Algorithm 1 Election of a String of Gene Expression States

```
Input: G_i(j), j = 1, ..., N, S, and d_{X,i}, ∀X ∈ S
Initialize: Z_e = ϵ, where ϵ is the null string
for X ∈ S do
  Compute τ_{X,i}
end for
for j = 1 to N do
  for X ∈ S do
    Use (4.6) and (4.7) to evaluate N and U
    E ← ¬(U ∨ N)
    Store N, U, and E
  end for
  Compute majority of N, U and E
  if E ≠ ¬(U ∨ N) or N = 1 or the vote is a tie then
    Z_e ← cat(Z_e, ⊥), where cat is concatenation
  else if E = 1 then
    Z_e ← cat(Z_e, 1)
  else
    Z_e ← cat(Z_e, 0)
  end if
end for
return Z_e
```

In *Algorithm 1*, each method $X \in S$ computes $\tau_{X,i}$ and use it to decide the truth

value of U and N. Then, equation (4.8) is used to compute the value of E, in each case. Next, a collective decision is made by assigning to U, N and E the values of the majority of each of these decisions. Then, the consistency of the values of the collective decisions on U, N and E are verified by entering them in equation (4.8). If the values do not correspond to a valid evaluation of the formula, the collective decision is declared *inconsistent*. An inconsistent collective decision renders the corresponding state undecidable. Table 4.3 is an example of an inconsistent collective decision. The first three rows of the table are the decisions of some hypothetical algorithms I, II, and III on N and U, and the value of E computed by entering the former to equation (4.8). The forth row is the collective decision made by taking the majority of the values given to N, U and E in the previous process. This collective decision is inconsistent with the doctrine since $N = 0$ and $U = 0$ give $\neg(N \vee U) = 1 \neq E = 0$.

It is worth remarking that if $S = \{X\}$ is a singleton, then Z_e corresponds to a binarization performed with threshold $\tau_{X,i}$.

4.3.3 Probabilistic Strings of Expression States

For each algorithm $X \in \{A, B, C, D\}$ we computed the probability distribution of the random variable

$$T_X =: \text{“}\textit{threshold returned by } X\text{.”} \tag{4.9}$$

These distributions, which were constructed with $1,000$ randomly generated 10-point expression profiles, are shown in Figure 4.4.

Table 4.3: Example of an inconsistent collective decision with three decision algorithms

Algorithms	N	U	E
I	1	0	0
II	0	0	1
III	0	1	0
Majority	0	0	0

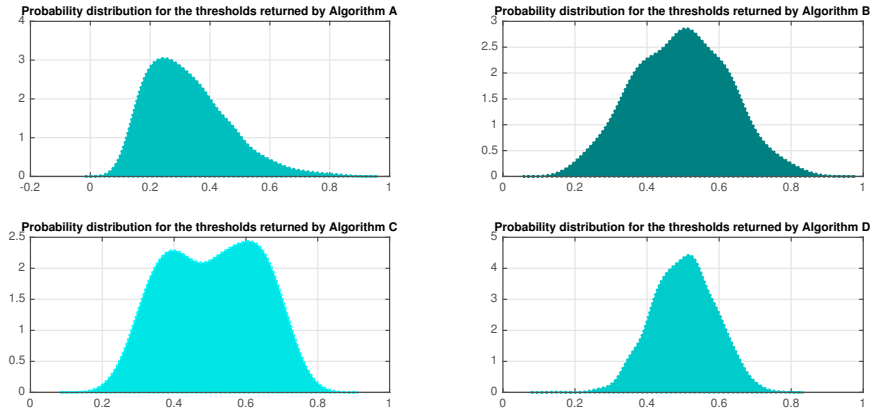


FIGURE 4.4: Probability distribution for the thresholds returned by Algorithms A, B, C and D.

Given an expression profile G_i, an algorithm X and an estimation of the threshold displacement $d_{X,i}$, we use the probability distribution of T_X to compute

$$P_{X,j}(0) = P(T_X > G_i(j) + d_{X,i}), \tag{4.10}$$

$$P_{X,j}(1) = P(T_X < G_i(j) - d_{X,i}), \text{and} \tag{4.11}$$

$$P_{X,j}(\perp) = 1 - (P_{X,j}(0) + P_{X,j}(1)), \tag{4.12}$$

for each $j = 1, \ldots, N$. These probabilities are stored in a $3 \times N$ array

$$\Omega_{X,i} = [P_{X,j}(Y)]\,,\, Y \in \{0, 1, \perp\}. \tag{4.13}$$

We use the average

$$\Omega_i = [P_j(Y)] = \frac{1}{|S|} \sum_{X \in S} \Omega_{X,i}, \tag{4.14}$$

over a set S of selected DGECs, to assign a single probability $P_j(Y)$ to each of the three possible states $Y = 0, 1, \perp$ of $G_i(j)$. Then, the set $\{Z_k : k = 1, \ldots, 3^N\}$ of strings of length N over $\{0, 1, \perp\}$ is produced in lexicographic order and stored in a $3^N \times (N + 1)$ array

$$Z = [Z_k(j), P_k], k = 1, \ldots, 3^N \text{ and } j = 1, \ldots, N; \tag{4.15}$$

where P_k is the probability of string Z_k computed as

$$P_k = \prod_{j=1}^{N} P_j(Z_k(j)). \tag{4.16}$$

Algorithm 2 is a pseudo code of this process.

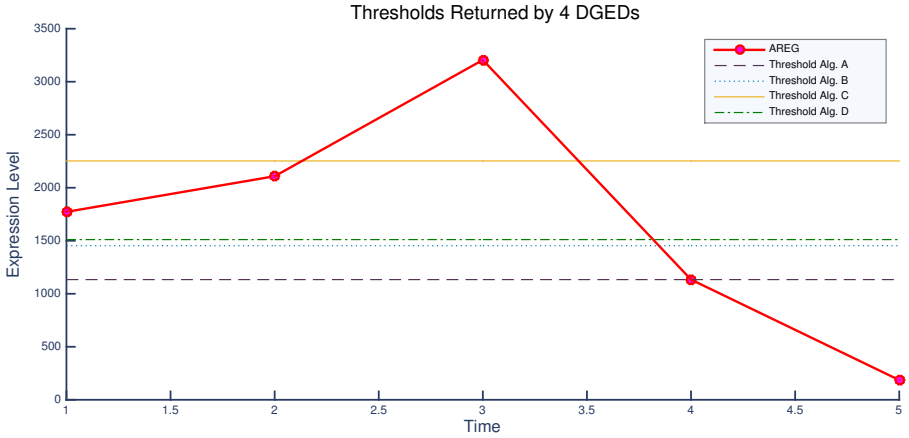
Algorithm 2 Probabilistic Strings of Expression States

Input: $G_i(j), j = 1, \ldots, N$, S, and $d_{X,i}, \forall X \in S$
for $X \in S$ **do**
  for $j = 1$ to N **do**
    Use (4.10), (4.11) and (4.12) to compute $\Omega_{X,i}$
  end for
end for
$\Omega_i = \frac{1}{|S|} \sum_{X \in S} \Omega_{X,i}$
for $k = 1$ to 3^N **do**
  Use the lexicographic order to produce Z_k
  $P_k \leftarrow 1$
  for $j = 1$ to N **do**
    Use Ω_i to update $P_k \leftarrow P_k P_j(Z_k(j))$
  end for
  Write Z_k, P_k in the k-th row in Z
end for
return Z

4.3.4 Post Processing

We divide the interval $[0, 1]$ into h segments of length $\frac{1}{h}$ and produce a histogram of the frequencies of occurrence of strings in Z with probabilities in each of these segments. The mean μ, variance σ^2 and the distance of the probability P_e of Z_e to μ are computed as a way of assessing the significance of Z_e.

TABLE 4.4: Z_e's and statistics returned by the Algorithmic Framework for the time-course expression data of gene AREG.

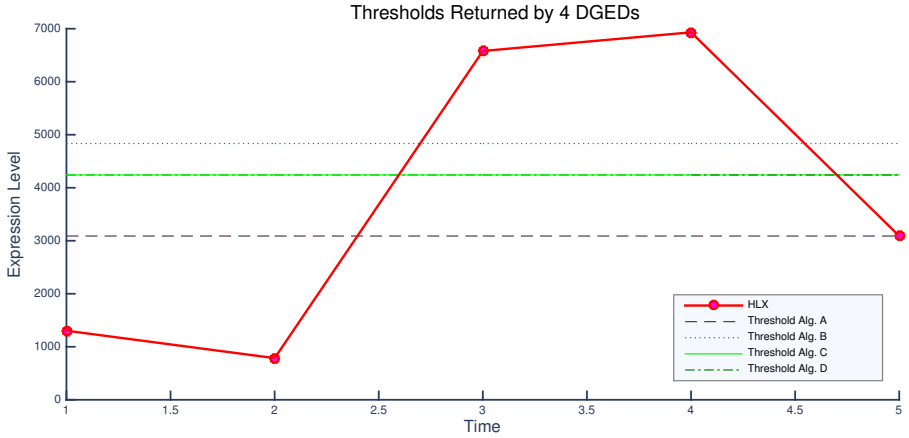


S	Z_e	P_e	μ	$\lvert P_e - \mu \rvert$	σ^2
$\{A\}$	$\perp$ 1 1 $\perp$ 0	0.01711	0.0041	0.0130	0.00009
$\{B\}$	$\perp\perp$ 1 $\perp$ 0	0.00248	0.0041	0.0016	0.00028
$\{C\}$	0 $\perp$ 1 0 0	0.04957	0.0041	0.0455	0.00021
$\{D\}$	$\perp$ 1 1 0 0	0.00004	0.0041	0.0041	0.00135
$\{A, B\}$	$\perp\perp$ 1 $\perp$ 0	0.01431	0.0041	0.0102	0.00012
$\{B, C, D\}$	$\perp\perp$ 1 0 0	0.00802	0.0041	0.0039	0.00039
$\{A, B, C, D\}$	$\perp\perp$ 1 $\perp$ 0	0.00398	0.0041	0.0001	0.00021

4.4 Experiments

We applied the algorithmic framework described in Section 4.3 to the vehicle control used in the study of Leukotriene B4 (LTB4) effect on monocytes [Sun et al., 2007]. This is a five-point time-course data set GDS3088, available in the GEO repository [*Gene Expression Omnibus*]. The results returned by different choices of sets of algorithms S on three gene expression profiles are displayed together with their statistics in Tables 4.4-4.6. The set $S = \{A, B\}$ comprises the least correlated DGEC whereas $S = \{B, C, D\}$ the highest correlated, as shown in Table 4.1.

TABLE 4.5: Z_e's and statistics returned by the Algorithmic Framework for the time-course expression data of gene HLX.

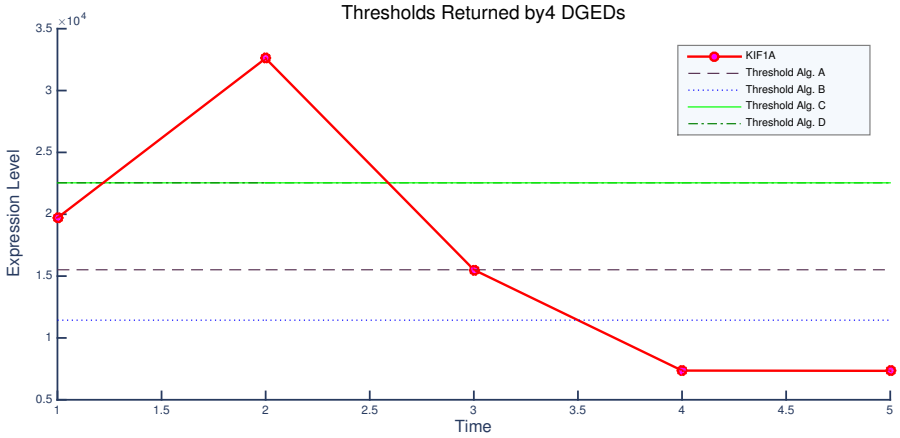


S	Z_e	P_e	μ	$\lvert P_e - \mu \rvert$	σ^2
$\{A\}$	0 0 1 1 $\perp$	0.1072	0.0041	0.1031	0.0004
$\{B\}$	0 0 1 1 0	0.1797	0.0041	0.1755	0.0007
$\{C\}$	0 0 1 1 0	0.1945	0.0041	0.1903	0.0006
$\{D\}$	0 0 1 1 0	0.5452	0.0041	0.5410	0.0017
$\{A, B\}$	0 0 1 1 $\perp$	0.2104	0.0041	0.2063	0.00042
$\{B, C, D\}$	0 0 1 1 0	0.2817	0.0041	0.2776	0.0008
$\{A, B, C, D\}$	0 0 1 1 0	0.1717	0.0041	0.1676	0.0005

4.4.1 Analysis of Experiments

The algorithmic framework described in Section 4.3 is designed to assist deciding the states of a gene in the presence of model and discretization uncertainties. The method aggregates a variety of models and their statical behaviors in a flexible manner, providing, thus, several points of view to support gene state decision-making, together with a measure of the likelihood of a particular choice, and other general statistics. The framework can be expanded by adding other DGECs or other kind of gene expression analysis methods, provided that their results are expressible through a gene expression decision threshold. The experiments conducted with biological data support this claim. For instance, in Table 4.4, a quick column-by-column revision of strings Z_e's

TABLE 4.6: Z_e's and statistics returned for the time-course expression data of gene KIF1A by the Algorithmic Framework.



S	Z_e	P_e	μ	$\lvert P_e - \mu \rvert$	σ^2
$\{A\}$	$\perp$ 1 $\perp$ 0 0	0.05957	0.0041	0.0555	0.00009
$\{B\}$	1 1 $\perp\perp\perp$	0.00001	0.0041	0.0041	0.00041
$\{C\}$	$\perp$ 1 0 0 0	0.06141	0.0041	0.0573	0.00024
$\{D\}$	0 1 0 0 0	0.22668	0.0041	0.2226	0.00144
$\{A, B\}$	$\perp$ 1 $\perp\perp\perp$	0.00094	0.0041	0.0032	0.00016
$\{B, C, D\}$	$\perp$ 1 0 0 0	0.04201	0.0041	0.0379	0.00049
$\{A, B, C, D\}$	$\perp$ 1 $\perp$ 0 0	0.01517	0.0041	0.0111	0.00028

supports declaring AREG expressed in the third time instant of the time-series, and not expressed in the fifth. There should be also little doubt in declaring the state of the gene undecidable in the first time instant. As for the second time instant, although two options declare it expressed, the state is undecidable for the rest of the options, including that with the highest P_e. Thus, the state of the gene in the second time instant may be safely declared undecidable. The results returned for the forth time instant are almost evenly split between unexpressed and not decidable, with minority leaning towards the former but including the string with highest P_e. This state may require some further analysis and eventually, the use of remaining selections of S for a more dependable decision. Table 4.5 confirms, in turn, that the string of expression states corresponding to the time-course sample of HLX is 00110, since only

Algorithm A has a different symbol for the fifth time instant. Finally, a decision on the string of states of the time-course expression data of KIF1A from Table 4.6 is subject to a similar analysis to that made with Table 4.4. For the sake of brevity, the statistical parameters μ, $|P_e - \mu|$ and σ^2 were used but not mentioned in the previous analyses. In general, a Z_e whose probability is to the right of μ is better than one whose probability is to the left of μ, since the former is closer to the string with the highest probability in Z. This is specially significant if the variance is small and the distance $|P_e - \mu|$ is comparatively large.

4.4.2 Improving Network Resolution

A string with a large number of non decidable states ($\perp$) is considered a string with low resolution. This may be indicative of noisy input data and thus, the binarizaton obtained is non dependable. In order to select a string of states with a dependable binarization, we need to process all strings to determine their amount of no decidable states. We will, then, filter the strings based on their resolution. Let

$$\upsilon_{Z_c} = \sum_{j=1}^{N} \mathbb{1}_{\perp}\Big(Z_c(j)\Big) \tag{4.17}$$

where

$$\mathbb{1}_{\perp}\Big(Z_c(j)\Big) = \begin{cases} 1, \text{ if } Z_c(j) = \perp \\ 0, \text{ otherwise.} \end{cases} \tag{4.18}$$

the amount of undecidable states of string Z_c; $0 \leq \upsilon \leq N$. Then, we define the resolution of string Z_c as

$$res = 1 - \frac{\upsilon_{Z_c}}{N} \tag{4.19}$$

Here, a resolution of 1 means the strings have no undecidable states. A resolution value of 0, on the other hand, means all states in the string have been assigned $\perp$.

4.4.3 Scoring Strings Probabilities

The probability value of string of states presents us with an expectancy of the occurrence of the binarization among other possible values within its distribution. However, to assist us in choosing a meaningful value, we need to rank the string probabilities. z scores provide a measure on how above or below the population mean is the string probability. A high, positive z score is an indicative of a probability close to the highest value and, thus, is a desirable quality. We compute the z score of a string Z as

$$z_Z = \frac{P_Z - \mu}{\sigma} \tag{4.20}$$

Table 4.7 to 4.9 show these statistics.

4.5 Value Imputation of No Decidable States

The last step of post-processing is the value imputation of $\perp$ states. Although $\perp$ states play a valuable role in analyzing, assessing and validating the binary quantization of a gene, both BN and PBN are defined over the set {0,1}. The mapping

$$I_{Z_c} : \{0, 1, \perp\}^N \rightarrow \{0, 1\}^N \tag{4.21}$$

TABLE 4.7: Z_e's and scores for the time-course expression data of gene AREG.

S	Z_e	P_e	res	z	σ
$\{A\}$	$\perp$ 1 1 $\perp$ 0	0.01711	0.6	1.3647	0.0095
$\{B\}$	$\perp\perp$ 1 $\perp$ 0	0.00248	0.4	−0.0962	0.0170
$\{C\}$	0 $\perp$ 1 0 0	0.04957	0.8	3.1293	0.0145
$\{D\}$	$\perp$ 1 1 0 0	0.00004	0.8	−0.1112	0.0367
$\{A, B\}$	$\perp\perp$ 1 $\perp$ 0	0.01431	0.4	0.9286	0.0110
$\{B, C, D\}$	$\perp\perp$ 1 0 0	0.00802	0.6	0.1968	0.0198
$\{A, B, C, D\}$	$\perp\perp$ 1 $\perp$ 0	0.00398	0.4	−0.0096	0.0144

produces the set $\mathcal{I}$ of all possible imputed values of a string Z_c; $|\mathcal{I}| = 2^{\upsilon_{Z_c}}$.

4.6 Evaluation of Boolean Networks Inferred from Elected Strings of States

The state of a GRN at a particular instant is the state of each of its genes at that instant. The set of network states is the set of all states of its genes at every instant, and the sequence of these states is a states transition mapping.

Due to model and discretization uncertainties, we may obtain more than one string of states for any network component. This, in turn, will produce more than one set of network states, each set corresponding to different states transition mappings. Thus, different rules may be inferred from each set, which will construct different networks.

The method described below is a simple way to evaluate the inferred networks. Some implementations to infer and model Boolean networks are available. Even when more than one binarization method may be allowed in them, in general, these implementations take as input a matrix of gene expressions, binarize them and learn the network from such binarization. The output are a few possible Boolean networks that these genes model. This kind of output does not provide information on the binarization of each gene, making impossible to discern if the differences in the inferred networks

TABLE 4.8: Z_e's and scores for the time-course expression data of gene HLX.

S	Z_e	P_e	res	z	σ
$\{A\}$	0 0 1 1 $\perp$	0.1072	0.8	4.9924	0.0206
$\{B\}$	0 0 1 1 0	0.1797	1	6.5893	0.0266
$\{C\}$	0 0 1 1 0	0.1945	1	7.9736	0.0239
$\{D\}$	0 0 1 1 0	0.5452	1	12.9801	0.0417
$\{A, B\}$	0 0 1 1 $\perp$	0.2104	0.8	10.0507	0.0205
$\{B, C, D\}$	0 0 1 1 0	0.2817	1	9.7850	0.0284
$\{A, B, C, D\}$	0 0 1 1 0	0.1717	1	7.2336	0.0232

TABLE 4.9: Z_e's and scores for the time-course expression data of gene KIF1A.

S	Z_e	P_e	res	z	σ
$\{A\}$	$\perp$ 1 $\perp$ 0 0	0.05957	0.6	5.6051	0.0099
$\{B\}$	1 1 $\perp\perp\perp$	0.00001	0.4	-0.2014	0.0204
$\{C\}$	$\perp$ 1 0 0 0	0.06141	0.8	3.7135	0.0154
$\{D\}$	0 1 0 0 0	0.22668	1	5.8578	0.0380
$\{A, B\}$	$\perp$ 1 $\perp\perp\perp$	0.00094	0.2	-0.2488	0.0127
$\{B, C, D\}$	$\perp$ 1 0 0 0	0.04201	0.8	1.7051	0.0222
$\{A, B, C, D\}$	$\perp$ 1 $\perp$ 0 0	0.01517	0.6	0.6615	0.0167

are due to the differences between the selected binarizaton methods or to the learning mechanism of the implementation. The advantage of the proposed method is that its assessment provides, through specific metrics, information on each network state individually, in this way aiding in the calibration of the network's rules and the detection of troubled binarizations.

Let $\mathcal{S}_{\mathcal{N}}$ be the set of all possible transition mappings of binary states for network $\mathcal{N}$ and $\mathcal{S}_{\mathcal{N},o}$ the known set of string states for network $\mathcal{N}$. For any $\mathcal{S}_{\mathcal{N},c} \subset \mathcal{S}_{\mathcal{N}}, c = 1, ..., 2^{N \times M}$, we define the matching of the network resulting from the states transitions on the set as

$$m = 1 - \frac{\mathcal{H}_{c,o}}{N \times M} \tag{4.22}$$

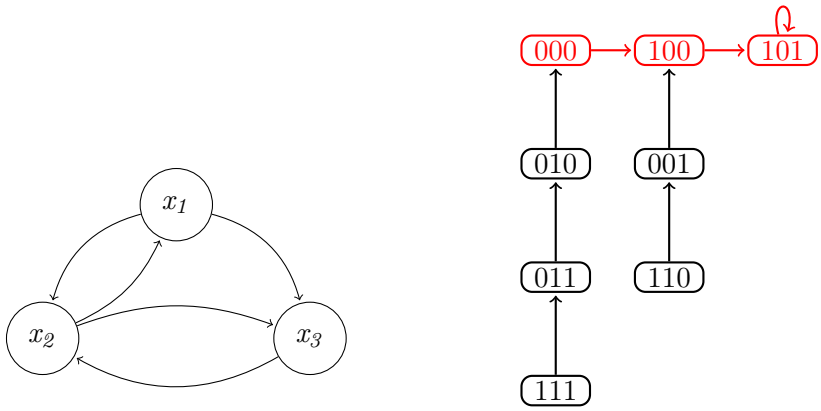
where $\mathcal{H}_{c,o}$ is the Hamming distance between $\mathcal{S}_{\mathcal{N},o}$ and $\mathcal{S}_{\mathcal{N},c}$; $0 \leq m \leq 1$. A m value of 1 will indicate a perfect matching and $m = 0$ would mean no matching in any state. The amount of transition mappings after gene value imputation depends on the amount of genes whose values where undecidable, thus,

$$|\mathcal{S}_{\mathcal{N}}| = 2^{\upsilon_{\mathcal{N}}} \tag{4.23}$$

where

$$\upsilon_{\mathcal{N}} = \sum_{i=1}^{M} \upsilon_{Z_i}. \tag{4.24}$$

We also add two more metrics to this assessment. The first one, m_g, will indicate



(A) Boolean Network Diagram (B) Known State Transition Diagram

FIGURE 4.5: Boolean Network of 3 genes and its states transition diagram

the percentage of genes that match all states transitions. The second, m_s, serves to evaluate each individual string of states in a scale from 0 to M, where strings of states that score 0 indicates that none of the genes states matches with its corresponding string in $\mathcal{S}_{\mathcal{N},o}$, while $m_s = M$ indicates all gene states match with its corresponding string.

4.6.1 Example A

Let Boolean network $\mathcal{N}(V, F)$ be defined by set of nodes $V = \{x_1, x_2, x_3\}$ and list of Boolean functions $F = (f_1, f_2, f_3)$ where F comprises the following rules:

$$f_1 : x_1(j+1) = \neg x_2(j) \tag{4.25a}$$

$$f_2 : x_2(j+1) = x_2(j) \wedge x_3(j) \tag{4.25b}$$

$$f_3 : x_3(j+1) = x_1(j) \tag{4.25c}$$

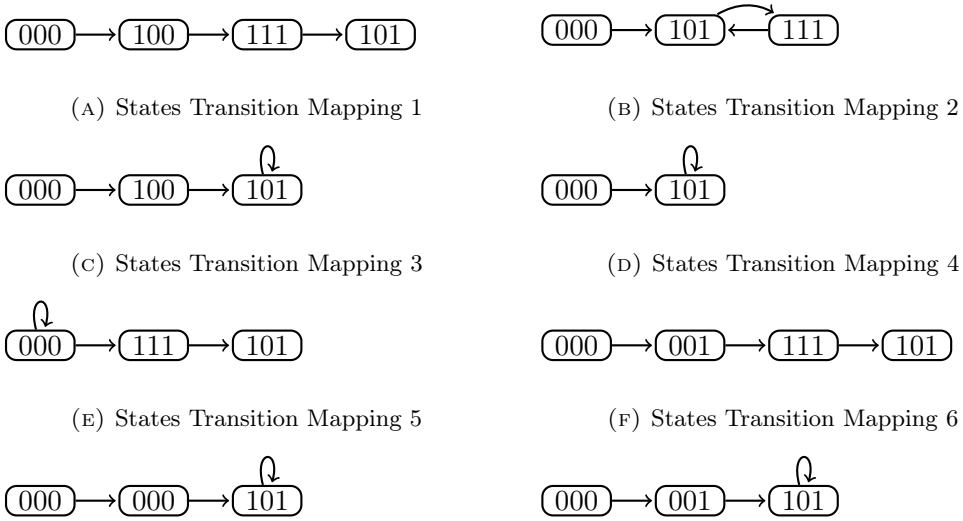
The network defined by equations 4.25 and its state transitions are represented in Figure 4.5.

Assume that Table 4.10 is the binarization of genes x_1, x_2, x_3. Here, $*$ indicates that at that particular time instant, the expression of the gene was undecidable and thus, subjected to value imputation. Gene profiles consist of a 4-point time series.

The set $\mathcal{S}_{\mathcal{N}}$ of possible transition mappings are depicted on Figures 4.6a throgh 4.6h.

TABLE 4.10: Gene Binarization After Value Imputation

	t_1	t_2	t_3	t_4
x_1	0	1*	1	1
	0	0*	1	1
x_2	0	0	1*	0
	0	0	0*	0
x_3	0	0*	1	1
	0	1*	1	1



(A) States Transition Mapping 1

(B) States Transition Mapping 2

(C) States Transition Mapping 3

(D) States Transition Mapping 4

(E) States Transition Mapping 5

(F) States Transition Mapping 6

(G) States Transition Mapping 7

(H) States Transition Mapping 8

FIGURE 4.6: All 8 states transition diagrams for 4.5b.

Using the evaluation metrics we propose, each network assessment is shown in Table 4.11.

The results show that only network of states transition mapping 3 is a perfect matching to the known network. 3 of the networks (1, 4 and 7) have perfect matching in 2 out of 3 genes. Network 6 shows the worst matching with all 3 genes failing in some state. All networks have perfect matching in the start and end states. The inconsistencies on network of states transition mapping 8, although on 2 genes, happens in same network state, which may provide information about the sampling process at that particular time instant. States transition flows 2 and 5, on the other hand,

TABLE 4.11: Network Evaluation

	$\mathcal{N}_1$	$\mathcal{N}_2$	$\mathcal{N}_3$	$\mathcal{N}_4$	$\mathcal{N}_5$	$\mathcal{N}_6$	$\mathcal{N}_7$	$\mathcal{N}_8$
m	0.92	0.83	1	0.92	0.83	0.75	0.92	0.83
$m_g(\%)$	67	33	100	67	33	0	67	33

TABLE 4.12: Network States Evaluation m_s

	$\mathcal{N}_1$	$\mathcal{N}_2$	$\mathcal{N}_3$	$\mathcal{N}_4$	$\mathcal{N}_5$	$\mathcal{N}_6$	$\mathcal{N}_7$	$\mathcal{N}_8$
State 1	3	3	3	3	3	3	3	3
State 2	3	2	3	2	2	1	2	1
State 3	2	2	3	3	2	2	3	3
State 4	3	3	3	3	3	3	3	3

while having similar metrics than states transition flow 8, present inconsistencies on 2 different states (with 2 different genes each time). Therefore, a sampling problem is less likely in these networks. This is assessed through m_s, where time instants with inconsistent binarizations can be identified. It is important to note that not all network states considered in a Boolean model necessarily occur in reality, but the model is a representation of how the network is expected to transition have those states happen. This is the case with gene states (010), (011) and (110) in 4.6.1.

Each network state assessment is shown in Table 4.12.

The impact of the uncertainties on gene binarization on a GRN may be influenced by several aspects, as the robustness of the network and the particular genes with not decidable states. In the example, almost all instances of intermediate states are inconsistent with the known states. This may lead to wrong conclusions if the dynamics of these transitions is what we are interested on observing, as in the case of the effect of a drug in the network behavior. It is interesting that, although all 3 genes in the network have one not decidable state, all possible transition mappings end on the known final state. This may be an indicative of the robustness of this specific network; but this also may be due to the fact that none of the genes in the network have a not decidable state in that final time instant.

The algorithmic framework introduced in this chapter may be modified by identifying the subset S of algorithms whose thresholds represent more accurately the observed state of a gene, creating thus a correspondence between subsets S and genes. By using a more specific subset of algorithms, the performance of the framework may be improved.

Chapter 5

Summary and Future Work

"Humans are allergic to change. They love to say, 'We've always done it this way.' I try to fight that. That's why I have a clock on my wall that runs counter-clockwise."

—Grace Hopper

Technology advances in data collection have made available large quantities of information on gene expression activity. The observed behavior of the genes, along with this information, aids in the construction of gene regulatory networks. However, the question of what is the state of a gene at a particular instant needs first to be answered.

Several methods have been proposed that compute a threshold, the binarization criterium. Nevertheless, there are discrepancies in the results provided by them, in spite of the chosen heuristic. This is due to the lack of a mathematical formulation that accurately models the phenomenon; to this we call *model uncertainty.* This research shows that there are also discrepancies in the results obtained from time series of different lengths, causing the results to displace when more data points are added to the series. This is what we call *discretization uncertainty.*

The robustness of the network, the specific genes and instants with non decidable states, among other aspects, affect the impact of the uncertainties on gene binarization on a GRN. However, even if a network reaches the observed final state, the changes on the dynamics of the transition states may be misleading, as in the study of the effect of a drug on network behavior, for instance.

The proposed framework handles these uncertainties with a unified approach, incorporates statistics to the analysis of the gene expressions and uses a set of methods where a voting mechanism takes place. One of the main advantages of this framework is the capability of identifying "noisy strips" in the data series, which can go one way or the other in the binarization process, labeling such states as *not decidable*. This novel approach aids in improving the accuracy of the binarization, which in turn, would provide more reliable networks. The performance of this framework may be improved by creating a correspondence between subsets S of algorithms and genes, choosing those methods whose thresholds represent more accurately the observed state of the gene.

5.1 Future Work and Related Directions

While the algorithmic framework introduced in this chapter helps in coping with model and discretization uncertainties, some calibration could be made with biological reality. This adjustment can be done in several ways in the framework; for instance, identifying the subset S of algorithms whose thresholds represent more accurately the observed state of a gene, creating thus a correspondence between subsets S and genes. One possible improvement to the proposed framework may be to use previous knowledge to impose weights to the probabilities to modify the computation of the mean in *Algorithm 2*. This would render a probabilistic network. A refinement in the estimation of threshold displacements will also improve accuracy.

This research suggests a line of research that may be worth pursuing. Its ultimate aim should be a mathematical framework for validating implicit models from their different algorithmic approaches. This validation might eventually lead to, or replace an explicit abstract mathematical representation of the reality that the implicit model attempts to represent. The development of such a framework will support current tendencies of using multi-algorithmic approaches to data based computational modeling.

This research has been a "seed" as several paths have emerged during the development of this dissertation; some of them are worthy of their own devoted research.

5.1.1 Ternary Logic Network

The construction of a network using ternary logic over the set $\{0, 1, \perp\}$ may provide interesting results. This would be consonant with the binarization process followed by the proposed framework.

5.1.2 Threshold Convergence

For the discretization uncertainty study, a tolerance was imposed to the threshold computation. Some methods, however, reached this convergence error faster than others. The accepted tolerance largely varies among methods; asymptotic and oscillating behaviors are observed. This arises questions like weather or not the threshold computed by a method converges and what is the tolerance accepted by the method. The answer to these questions is currently subject of research.

5.1.3 GAP-Displacement Classification

The algorithm used to compute the threshold does not bound the displacement. Another question that arises is what role, if any, plays the shape of the GAP in the displacement of the threshold. Seizing on clustering classification efforts of gene expression [Eisen et al., 1998]; [Alon et al., 1999]; [Jiang, Tang, and Zhang, 2004], a GAP classification for the estimation of thresholds displacements could be established.

5.1.4 Biological Validation

The methods could gain biological meaning by constructing the PDFs using biological data, rather than random profiles. It would also be interesting to study the possible charaterization of such PDFs.

5.2 Conclusions

This research introduces the concepts of uncertainties on binarizations of gene expressions due to the lack of a mathematical model that describes them. The analyses

conducted provide insights on the nature of such uncertainties and we developed mechanisms to measure them. The proposed framework for gene expression decission unifies heuristics, incorporating commonly used methods with statistics, based on the expected behavior of the GAP. The inclusion of thresholds displacement on the analysis of binarization states provides information to improve accuracy. This is a novel approach to a rather traditionally studied problem. The results show the framework provides reliable binarizatons when compared with other methods, and has the flexibility to be used with any set of existing methods, making its adaptation to any known threshold computation method feasible.

Chapter 6

Ethical Considerations on Nonhuman Animal Testing

"Any intelligent fool can make things bigger, more complex and more violent. It takes a touch of genius —and a lot of courage— to move in the opposite direction."

—E. F. Schumacher

Every year, millions animals, including nonhuman primates, dogs, cats, pigs and farm animals, mice, rats, guinea-pigs and other rodents, birds, fish, and other species, are used worldwide in laboratory research. The subjects of these tests, while quite diverse, may be summarized as assessing the potential benefits or damage of products in human health and the environment. The nature of these products range from drugs, cosmetics, household and agricultural products.

Weather or not nonhuman animals should be employed in research is a subject of great polarization, even within the scientific community, due to its ethical implications. But the outcomes of animal testing are controversial as well, and its benefits for human has become questioned.

This chapter is a brief review of nonhuman animal testing from an ethical perspective.

6.1 Nonhuman Animal Testing in Review

In 1959, Russell and Burch published "The Principles of Human Experimentation" [Russell et al., 2005], which introduced what later was coined as the 3Rs: Replacement (of animals with alternative methods), Reduction (of the amount of animal required for testing) and Refinement (of the tests in aims to eliminate or reduce pain and distress) of nonhuman animal subjects. More recently, a forth R has been considered: Rehabilitation.

The Animal Welfare Act (AWA) [*Animal Welfare Act*] was signed into federal law in 1966, with the mission of regulating the treatment nonhuman animal research, as well as other activities like breading and dealing, exhibition and transport. Animals under the scope of AWA are nonhuman primates, dogs, cats, rabbits, guinea pigs, hamsters. Mice and rats, farm animals used for agricultural research, birds and cold-blooded animals, are exempt from AWA protection. The law has undergone several revisions. The amendments of 1985 introduced the establishment of Institutional Animal Care and Use Committee (IACUC) [*IACUC*], following requirements of the Public Health Services (PHS) agency. This requires that each grantee institution using animals have a nonhuman animal care committee of five members, including at least one veterinarian [*IACUC Regulations*]. In 1993 with the NIH Revitalization Act [*NIH Act*], NIH was directed to implement the 3Rs, it is, to support research and to develop and validate methods to replace, reduce and refine nonhuman animal use in biomedical research.

The exact amount of nonhuman animals used in test is uncertain due to failure from concerned agencies to release information [Taylor et al., 2008]. Data from the USDA, which is the agency enforcing AWA, shows on average over a million animals used yearly, for the las 8 years [*APHIS Reports*]. This number, however, amounts only for species under AWA protection. Lab-designed animals, those which are bred for the purpose of experimentation and are out of AWA protection, are reported to account for over 90% the total animals used [Trull and Rich, 1999]. These animals are regulated under the PHS.

6.1.1 IACUC Criticism

The effectiveness of IACUCs as an ethical assessment body is contended. Reports from their overseeing agency, showing inadequate care for the specimens and poor test monitoring, have been consistently presented ever since their inception [*USDA AWA Report 2014*]; [*USDA AWA Report 2005*]. Fails to pursue enforcement actions against violations is also noted on those reports as well. Some argue that this is due, in part, to the conflict of interest that arises because most of their members are nonhuman animal testers themselves [Hansen, Goodman, and Chandna, 2012]; [Carbone, 2004]; [Hansen, 2013].

But the purposed advance of human health through nonhuman animal testing effectiveness is contended as well [Lawrence et al., 2001]; [Cohen and Lawson, 1995]. Some argue that the recurrent translational failure to the clinic is due to methodological flaws on these studies, like lacking randomization and blindness, leading, thus to biased results [Plous and Herzog, 2001]; [Worp et al., 2010]. The lack of a systematic reviews of animal models and studies is also argued to contribute to the failure [Pound et al., 2004].

Although this research didn't involve animal testing, its relationship with biological sciences make pertinent to state that this research advocates for collaborative and responsible practices that seek alternatives to nonhuman animal testing.

6.2 Alternatives to Nonhuman Animal Testing

Since the 1950s, it has been proposed the use of alternative methods for nonhuman animal testing. Its impact, however, was little to none, showing that the resistance to changing the status quo is one hard to overcome [Balls, 1994].

In 1981, the Johns Hopkins Center for Alternatives to Animal Testing (CAAT) was founded, as part of the Johns Hopkins University Bloomberg School of Public Health. With a database of hundreds of publications [*Johns Hopkins University Jonhs Hopkins Center for Alternatives to Animal Testing*], either fruition of CAAT research of

disseminated through its workshops, CAAT has gained sponsorship of several gubernamental agencies as members of the private industry.

In 2000, the Interagency Coordinating Committee on the Validation of Alternative Methods (ICCVAM) Authorization Act was presented, "to establish, wherever feasible, guidelines, recommendations, and regulations that promote the regulatory acceptance of new or revised scientifically valid toxicological tests that protect human and animal health and the environment while reducing, refining, or replacing animal tests and ensuring human safety and product effectiveness." [*NIH Alternative Methods*]. In 2002, ICCVAM evaluated and accepted some alternative methods already validated by the EU [*NIH Evaluation of EU Methods*].

Developing alternative methods, however, may be a problem of great complexity. Some test can be replaced with a single, simple alternative, while some others demand an assembly of methods in an integral approach. After development, these alternatives need to be scientifically validated before being accepted. All of the above demands tremendous resources, which may explain the delay on producing them. An updated database on validated and accepted alternative methods can be found in [*Validated And Accepted Alternative Methods*]. Alternative methods include cell culture, medica imaging, computer simulation, among others.

Of particular interest is the development of "organ-in-a-chip", a technology regarded as to have great potential to advance the study of tissue development, organ physiology and disease, as well as aid in drug discovery and development, toxicity testing and biomarker identification [Bhatia and Ingber, 2014]; [Baker, 2011]. Figure 6.1 is a chip of a lung developed at Harvard Wyss Institute [*Harvard Wyss Institute*].

6.2.1 Nonhuman Animal Testing for Cosmetic Purposes

Testing cosmetic products in nonhuman animals is banned in several counties, including the European Union (EU), India and an Norway and, as of 2014, Brazil.

The ban on the EU started in 2004, with prohibition on testing finished products, extended its coverage to ingredients in 2009 and, as of 2013, includes all cosmetic

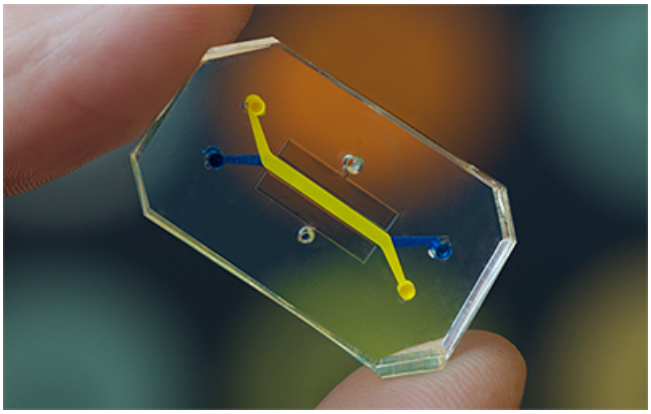


FIGURE 6.1: Lung on a chip. [*Harvard Wyss Institute*]

products regardless its effects or alternative tests availability. Since 1990, alternative methods in Europe have been proposed; this is a statement to the long road to acceptance to alternative methods [Liebsch et al., 2011].

In the United States, cosmetics safety is oversight by the Food and Drugs Administration (FDA). The FDA states that: "The FD&C Act does not specifically require the use of animals in testing cosmetics for safety, nor does the Act subject cosmetics to FDA premarket approval. However, the agency has consistently advised cosmetic manufacturers to employ whatever testing is appropriate and effective for substantiating the safety of their products. It remains the responsibility of the manufacturer to substantiate the safety of both ingredients and finished cosmetic products prior to marketing." [*Food And Drug Administration*]. Following European lead, in 2015, the "Humane Cosmetics Act" (H.R. 2858) was introduced proposing a ban on cosmetics testing [*Congress Cosmetic Testing Ban*].

With a proposed model for *in silico* testing, this research strives to contribute to the development of alternative models to nonhuman animal testing.

6.3 A very vocal crowd

Humane groups and animal activist have played a vital role in advancing alternatives to nonhuman animal testing. Not only in the form of whistleblowers and protesters, exerting pressure on gubernamental agencies to seek alternatives and intervene in

unethical, abusive scenarios, but as educational agents as well. Some major humane groups sponsor many of the research conducted on this area.

IACUCs are attributed to have been created in response to public outrage stemming from revelations of animal abuse in research laboratories [Holden, 1986]. In a similar fashion, several military medical training activities had been said to be reduce and eventually substituted by alternative methods after public pressure following exposure [*Military to curtail use of live animals in medical training*].

Contended opinions exist within the scientific community as well. On 2006, it Nature conducted an anonymous survey about scientist views on nonhuman animal testing [*Nature News*]. While divided in opinion, with the majority lending towards regarding the tests as necessary, many noted that "We have not addressed legitimate issues that animal rights groups have raised, ...a mouse is not a human and the question to be tested will not be fully answered... We need to admit this but point out that it is more complex than that."

The core of the search for alternatives to nonhuman animal testing is elimination of violence, even in its more subtle ways, and compassion towards all sentient beings, human included. This research regards the role of the public in general, and scientist in particular, in holding agencies and practitioners accountable, as essential, but rejects the use of violence and destruction of property as means to attain those goals.

Appendix A

Gene Expression Data Sets

A.1 Gene data set for Leukotriene B4 used on experiments on Chapter 4.

Full data set can be found in [Sun et al., 2007].

Table A.1: 3 genes from Leukotriene B4 data set.

Gene	t:0	t:1	t:2	t:3	t:4
AREG	1772.47	2108.5	3205.1	1133.63	185.885
KIF1A	19709	32599.8	15510.8	7360.89	7339.8
HLX	1302.86	785.806	6579.7	6931.75	3088.99

A.2 Gene data set for yeast cell cycle used on experiments on Chapter 3

Full data set can be found in [*Yeast Cell Cycle Analysis Project*].

TABLE A.2: 4 genes from yeast data set.

Gene	t:0	t:10	t:20	t:30	t:40	t:50	t:60	t:70	t:80	t:90	t:100	t:110	t:120	t:130	t:140	t:150	t:160
CDC24	273	214	190	208	353	210	337	324	271	391	250	244	306	265	304	253	231
CDC19	2020	1284	2191	1651	2696	1069	2084	1779	1261	5887	2676	2411	1523	2277	2915	2426	2863
CDC15	134	75	75	91	100	52	79	112	102	125	103	109	103	60	94	63	91
CDC27	112	155	142	150	137	135	150	159	153	147	131	111	153	117	139	160	113

Appendix B

Threshold Displacement Estimation Graphics

B.1 Thresholds Displacements by Range

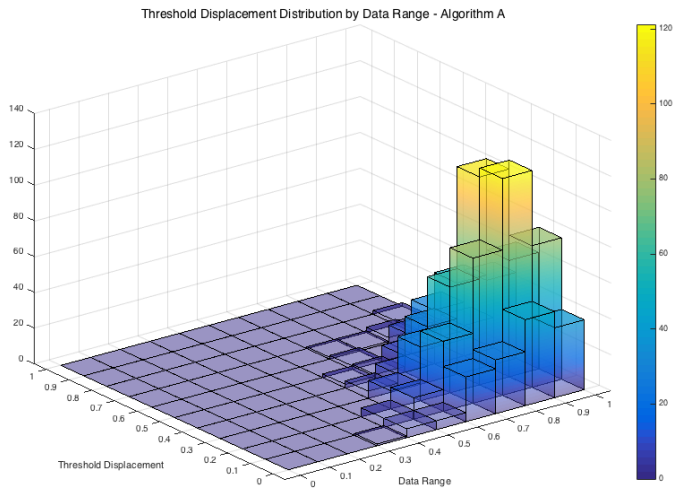


FIGURE B.1: Thresholds Displacements - Algorithm A.

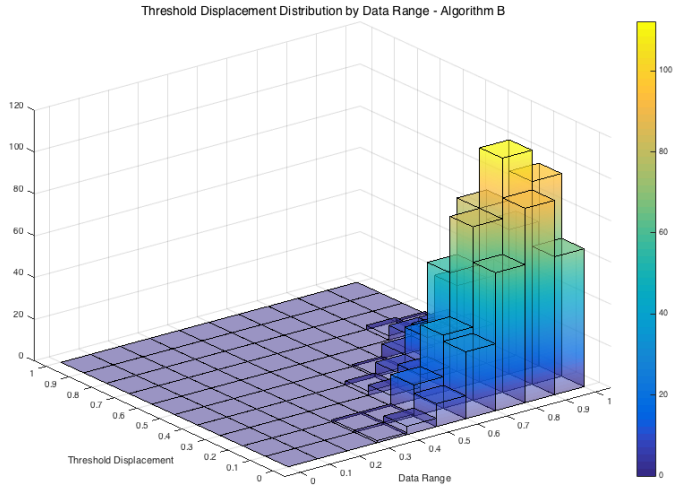


FIGURE B.2: Thresholds Displacements - Algorithm B.

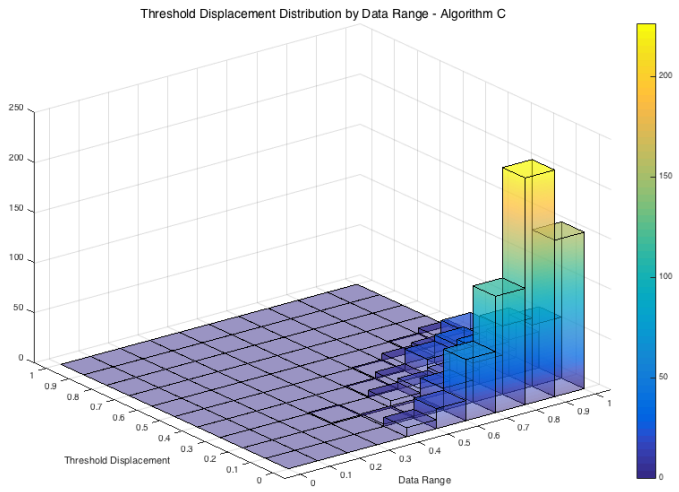


FIGURE B.3: Thresholds Displacements - Algorithm C.

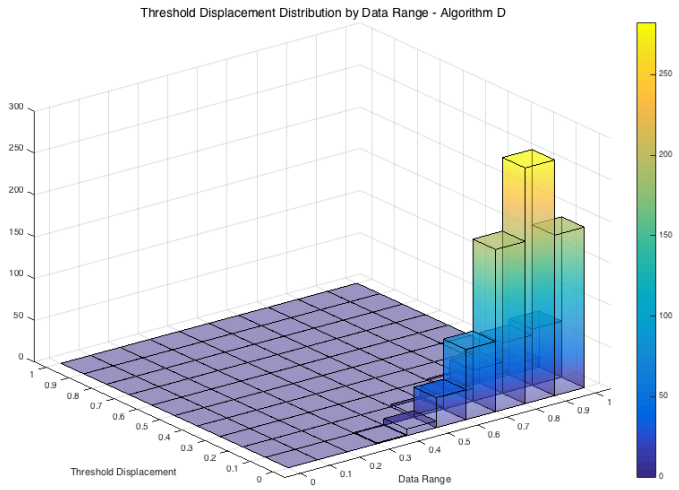


FIGURE B.4: Thresholds Displacements - Algorithm D.

References

Abate, A. et al. (2007). "Quantitative and Probabilistic Modeling in Pathway Logic". In: *2007 IEEE 7th International Symposium on BioInformatics and BioEngineering*, pp. 922–929. DOI: 10.1109/BIBE.2007.4375669 (cit. on p. 16).

Alon, U. et al. (1999). "Broad patterns of gene expression revealed by clustering analysis of tumor and normal colon tissues probed by oligonucleotide arrays". In: *Proceedings of the National Academy of Sciences* 96.12, pp. 6745–6750. DOI: 10.1073/pnas.96.12.6745. eprint: http://www.pnas.org/content/96/12/6745.full.pdf. URL: http://www.pnas.org/content/96/12/6745.abstract (cit. on p. 64).

Animal Welfare Act. URL: https://www.aphis.usda.gov/animal_welfare/downloads/awa/awa.pdf (cit. on p. 67).

APHIS Reports. URL: https://www.aphis.usda.gov/aphis/ourfocus/animalwelfare/sa_awa/AWA-Inspection-and-Annual-Reports (cit. on p. 67).

Baker, M. (2011). "Tissue models: A living system on a chip". In: *Nature* 471.7340, pp. 661–665. URL: http://dx.doi.org/10.1038/471661a (cit. on p. 69).

Balls, M. (1994). "Replacement of animal procedures: alternatives in research, education and testing". In: *Laboratory Animals* 28.3. PMID: 7967458, pp. 193–211. DOI: 10.1258/002367794780681714. eprint: http://dx.doi.org/10.1258/002367794780681714. URL: http://dx.doi.org/10.1258/002367794780681714 (cit. on p. 68).

Banf, M. and S. Y. Rhee (2017). "Computational inference of gene regulatory networks: Approaches, limitations and opportunities". In: *Biochimica et Biophysica Acta (BBA) - Gene Regulatory Mechanisms* 1860.1. Plant Gene Regulatory Mechanisms and Networks, pp. 41 –52. ISSN: 1874-9399. DOI: http://dx.doi.org/10.

1016/j.bbagrm.2016.09.003. URL: http://www.sciencedirect.com/science/article/pii/S1874939916301882 (cit. on p. 3).

Bar-Joseph, Z. et al. (2003). "Continuous Representations of Time-Series Gene Expression Data". In: *Journal of Computational Biology* 10.3-4, pp. 341–356. DOI: 10.1089/10665270360688057. URL: http://dx.doi.org/10.1089/10665270360688057 (cit. on p. 32).

Batt, G. et al. (2010). "Efficient parameter search for qualitative models of regulatory networks using symbolic model checking". In: *Bioinformatics* 26.18, pp. i603–i610. DOI: 10.1093/bioinformatics/btq387. URL: http://www.ncbi.nlm.nih.gov/pmc/articles/PMC2935427/ (cit. on p. 16).

Berestovsky, N. and L. Nakhleh (2013). "An Evaluation of Methods for Inferring Boolean Networks from Time-Series Data". In: *PLoS ONE* 8.6, e66031. DOI: 10.1371/journal.pone.0066031. URL: http://dx.doi.org/10.1371%2Fjournal.pone.0066031 (cit. on pp. 7, 29).

Bhatia, S. N. and D. E. Ingber (2014). "Microfluidic organs-on-chips". In: *Nat Biotech* 32.8, pp. 760–772. URL: http://dx.doi.org/10.1038/nbt.2989 (cit. on p. 69).

Bornholdt, S. (2008). "Boolean network models of cellular regulation: prospects and limitations". In: *Journal of The Royal Society Interface* 5.Suppl 1, S85–S94. ISSN: 1742-5689. DOI: 10.1098/rsif.2008.0132.focus. URL: http://rsif.royalsocietypublishing.org/content/5/Suppl_1/S85 (cit. on p. 5).

Bower, J. M. (2001). *Computational Modeling of Genetic and Biochemical Networks*. Ed. by J. M. Bower and H. Bolouri. MIT Press (cit. on pp. 4, 8).

Brooker, R. J. (2009). *Genetics: Analysis and Principles*. Ed. by P. E. Reidy. 3rd. 1221 Avenue of the Americas, New York, NY 10020: McGraw-Hill (cit. on p. 14).

Carbone, L. (2004). *What Animals Want: Expertise and Advocacy in Laboratory Animal Welfare Policy*. Oxford University Press (cit. on p. 68).

Ching, T., S. Huang, and L. X. Garmire (2014). "Power analysis and sample size estimation for RNA-Seq differential expression". In: *RNA* 20.11, pp. 1684–1696. DOI: 10.1261/rna.046011.114. eprint: http://rnajournal.cshlp.org/content/20/11/1684.full.pdf+html. URL: http://rnajournal.cshlp.org/content/20/11/1684.abstract (cit. on p. 15).

Chiu, T.-Y. et al. (2015). "Interpolation based consensus clustering for gene expression time series". In: *BMC Bioinformatics* 16, p. 117. DOI: 10.1186/s12859-015-0541-0. URL: http://www.ncbi.nlm.nih.gov/pmc/articles/PMC4407314/ (cit. on p. 32).

Cho, R. J. et al. (1998). "A Genome-Wide Transcriptional Analysis of the Mitotic Cell Cycle". In: *Molecular Cell* 2.1, pp. 65–73. DOI: 10.1016/S1097-2765(00)80114-8. URL: http://dx.doi.org/10.1016/S1097-2765(00)80114-8 (cit. on p. 34).

Cohen, S. M. and T. A. Lawson (1995). "Rodent bladder tumors do not always predict for humans." eng. In: *Cancer Lett* 93.1, pp. 9–16. ISSN: 0304-3835 (Print); 0304-3835 (Linking). DOI: 10.1016/0304-3835(95)03785-U (cit. on p. 68).

Congress Cosmetic Testing Ban. URL: https://www.congress.gov/bill/114th-congress/house-bill/2858 (cit. on p. 70).

Csikász-Nagy, A. et al. (2006). "Analysis of a Generic Model of Eukaryotic Cell-Cycle Regulation". In: *Biophysical Journal* 90.12, pp. 4361–4379. DOI: 10.1529/biophysj.106.081240. URL: http://dx.doi.org/10.1529/biophysj.106.081240 (cit. on p. 4).

Davidich, M. I. and S. Bornholdt (2008). "Boolean network model predicts cell cycle sequence of fission yeast." eng. In: *PLoS One* 3.2, e1672. ISSN: 1932-6203 (Electronic); 1932-6203 (Linking). DOI: 10.1371/journal.pone.0001672 (cit. on p. 5).

Derveaux, S., J. Vandesompele, and J. Hellemans (2010). "How to do successful gene expression analysis using real-time {PCR}". In: *Methods* 50.4. The ongoing Evolution of qPCR, pp. 227 –230. ISSN: 1046-2023. DOI: http://dx.doi.org/10.1016/j.ymeth.2009.11.001. URL: http://www.sciencedirect.com/science/article/pii/S1046202309002461.

Eils, R. and A. Kriete (2014). "Chapter 1 - Introducing Computational Systems Biology". In: *Computational Systems Biology (Second Edition)*. Ed. by A. Kriete and R. Eils. Second Edition. Oxford: Academic Press, pp. 1 –8. ISBN: 978-0-12-405926-9. DOI: http://dx.doi.org/10.1016/B978-0-12-405926-9.00001-0. URL: http://www.sciencedirect.com/science/article/pii/B9780124059269000010 (cit. on p. 3).

Eisen, M. B. et al. (1998). "Cluster analysis and display of genome-wide expression patterns". In: *Proceedings of the National Academy of Sciences of the United States of America* 95.25, pp. 14863–14868. URL: http://www.ncbi.nlm.nih.gov/pmc/articles/PMC24541/ (cit. on p. 64).

Essentials of Cell Biology. URL: http://www.nature.com/scitable/ebooks/essentials-of-cell-biology-14749010/122996928 (cit. on p. 14).

Euatham, K. and N. Tongsiri (2012). "Effects of expression profile data variation on boolean gene regulatory network predictions". In: *International Research Journal of Biochemistry and Bioinformatics* 2.10, pp. 208–215 (cit. on pp. 6, 7, 11).

Faure, A. et al. (2006). "Dynamical analysis of a generic Boolean model for the control of the mammalian cell cycle." eng. In: *Bioinformatics* 22.14, e124–31. ISSN: 1367-4811 (Electronic); 1367-4803 (Linking). DOI: 10.1093/bioinformatics/btl210 (cit. on p. 5).

Food And Drug Administration. URL: https://www.fda.gov/cosmetics/scienceresearch/producttesting/ucm072268.htm (cit. on p. 70).

Gebert, J, N Radde, and G Weber (2007). "Modeling gene regulatory networks with piecewise linear differential equations". In: *European Journal of Operational Research* 181.3, pp. 1148–1165. ISSN: 03772217. DOI: 10.1016/j.ejor.2005.11.044. URL: http://linkinghub.elsevier.com/retrieve/pii/S0377221706001512 (cit. on p. 16).

Gene Expression Omnibus. URL: https://www.ncbi.nlm.nih.gov/geo/query/acc.cgi?acc=GSE30536 (cit. on pp. 43, 51).

Gomaa, W. E. (2011). "Modeling gene regulatory networks: A survey". In: *2011 9th IEEE/ACS International Conference on Computer Systems and Applications (AICCSA)*, pp. 204–208. DOI: 10.1109/AICCSA.2011.6126584 (cit. on p. 3).

Greenbaum, D. et al. (2003). "Comparing protein abundance and mRNA expression levels on a genomic scale". In: *Genome Biology* 4.9, p. 117. ISSN: 1474-760X. DOI: 10.1186/gb-2003-4-9-117. URL: http://dx.doi.org/10.1186/gb-2003-4-9-117.

Grzegorczyk, M. and D. Husmeier (2011). "Improvements in the reconstruction of time-varying gene regulatory networks: dynamic programming and regularization by

information sharing among genes". In: *Bioinformatics* 27.5, p. 693. DOI: 10.1093/bioinformatics/btq711. eprint: /oup/backfile/Content_public/Journal/bioinformatics/27/5/10.1093/bioinformatics/btq711/2/btq711.pdf. URL: +http://dx.doi.org/10.1093/bioinformatics/btq711 (cit. on p. 5).

Hakamada, K. et al. (2004). "A preprocessing method for inferring genetic interaction from gene expression data using Boolean algorithm". In: *Journal of Bioscience and Bioengineering* 98.6, pp. 457 –463. ISSN: 1389-1723. DOI: http://dx.doi.org/10.1016/S1389-1723(05)00312-9. URL: http://www.sciencedirect.com/science/article/pii/S1389172305003129 (cit. on p. 8).

Han, S. et al. (2014). "A Full Bayesian Approach for Boolean Genetic Network Inference". In: *PLoS ONE* 9, e115806. ISSN: 1932-6203. DOI: 10.1371/journal.pone.0115806. URL: http://dx.plos.org/10.1371/journal.pone.0115806 (cit. on p. 5).

Hanahan, D. and R. A. Weinberg (2000). "The Hallmarks of Cancer". In: *Cell* 100.1, pp. 57–70. DOI: 10.1016/S0092-8674(00)81683-9. URL: http://dx.doi.org/10.1016/S0092-8674(00)81683-9 (cit. on p. 16).

Hansen, L. A., J. R. Goodman, and A. Chandna (2012). "Analysis of Animal Research Ethics Committee Membership at American Institutions". In: *Animals* 2.1, pp. 68–75. ISSN: 2076-2615. DOI: 10.3390/ani2010068. URL: http://www.mdpi.com/2076-2615/2/1/68 (cit. on p. 68).

Hansen, L. A. (2013). "Institution animal care and use committees need greater ethical diversity". In: *Journal of Medical Ethics* 39.3, pp. 188–190. ISSN: 0306-6800. DOI: 10.1136/medethics-2012-100982. eprint: http://jme.bmj.com/content/39/3/188.full.pdf. URL: http://jme.bmj.com/content/39/3/188 (cit. on p. 68).

Harvard Wyss Institute. URL: https://wyss.harvard.edu/ (cit. on pp. 69, 70).

Hecker, M. et al. (2009). "Gene regulatory network inference: Data integration in dynamic models—A review". In: *Biosystems* 96.1, pp. 86 –103. ISSN: 0303-2647. DOI: http://dx.doi.org/10.1016/j.biosystems.2008.12.004. URL: http://www.sciencedirect.com/science/article/pii/S0303264708002608 (cit. on p. 4).

Hogeweg, P. (2011). "The Roots of Bioinformatics in Theoretical Biology". In: *PLoS Computational Biology* 7.3. Ed. by D. B. Searls, e1002021. DOI: 10.1371/journal.pcbi.1002021. URL: http://www.ncbi.nlm.nih.gov/pmc/articles/PMC3068925/ (cit. on p. 2).

Holden, C (1986). "A pivotal year for lab animal welfare". In: *Science* 232.4747, pp. 147–150. ISSN: 0036-8075. DOI: 10.1126/science.3952503. eprint: http://science.sciencemag.org/content/232/4747/147.full.pdf. URL: http://science.sciencemag.org/content/232/4747/147 (cit. on p. 71).

Hoon, M. J. L. de et al. (2003). "Inferring gene regulatory networks from time-ordered gene expression data of Bacillus subtilis using differential equations." eng. In: *Pac Symp Biocomput*, pp. 17–28. ISSN: 2335-6936 (Print) (cit. on p. 4).

Hopfensitz, M. et al. (2012). "Multiscale Binarization of Gene Expression Data for Reconstructing Boolean Networks". In: *IEEE/ACM Transactions on Computational Biology and Bioinformatics* 9.2, pp. 487–498. ISSN: 1545-5963. DOI: http://doi.ieeecomputersociety.org/10.1109/TCBB.2011.62 (cit. on pp. 7, 10, 25, 26).

Hopfensitz, M., M. Maucher, and H. A. Kestler (2012). "Fuzzy Boolean Network Reconstruction". In: *Challenges at the Interface of Data Analysis, Computer Science, and Optimization: Proceedings of the 34th Annual Conference of the Gesellschaft für Klassifikation e. V., Karlsruhe, July 21 - 23, 2010*. Ed. by W. A. Gaul et al. Berlin, Heidelberg: Springer Berlin Heidelberg, pp. 263–270. ISBN: 978-3-642-24466-7. DOI: 10.1007/978-3-642-24466-7_27. URL: http://dx.doi.org/10.1007/978-3-642-24466-7_27 (cit. on p. 6).

Husmeier, D. (2003). "Sensitivity and specificity of inferring genetic regulatory interactions from microarray experiments with dynamic Bayesian networks". In: *Bioinformatics* 19.17, p. 2271. DOI: 10.1093/bioinformatics/btg313. eprint: /oup/backfile/Content_public/Journal/bioinformatics/19/17/10.1093/bioinformatics/btg313/2/btg313.pdf. URL: +http://dx.doi.org/10.1093/bioinformatics/btg313 (cit. on p. 5).

IACUC. URL: https://www.nal.usda.gov/awic/public-law-99-198-food-security-act-1985-subtitle-f-animal-welfare (cit. on p. 67).

IACUC Regulations. URL: https://grants.nih.gov/grants/olaw/guidebook.pdf (cit. on p. 67).

Jiang, D., C. Tang, and A. Zhang (2004). "Cluster analysis for gene expression data: a survey". In: *IEEE Transactions on Knowledge and Data Engineering* 16.11, pp. 1370–1386. ISSN: 1041-4347. DOI: 10.1109/TKDE.2004.68 (cit. on p. 64).

Johns Hopkins University Jonhs Hopkins Center for Alternatives to Animal Testing. URL: http://caat.jhsph.edu/publications/ (cit. on p. 68).

Jong, H. de (2002). "Modeling and Simulation of Genetic Regulatory Systems: A Literature Review". In: *Journal of Computational Biology* 9.1, pp. 67–103. DOI: 10.1089/10665270252833208. URL: http://dx.doi.org/10.1089/10665270252833208 (cit. on pp. 4, 17).

Kauffman, S. (1969). "Metabolic stability and epigenesis in randomly constructed genetic nets". In: *Journal of Theoretical Biology* 22.3, pp. 437 –467. ISSN: 0022-5193. DOI: http://dx.doi.org/10.1016/0022-5193(69)90015-0. URL: http://www.sciencedirect.com/science/article/pii/0022519369900150 (cit. on p. 4).

Kerr, M. K. and G. A. Churchill (2001). "Experimental design for gene expression microarrays". In: *Biostatistics* 2.2, p. 183. DOI: 10.1093/biostatistics/2.2.183. URL: +http://dx.doi.org/10.1093/biostatistics/2.2.183 (cit. on p. 15).

Kim, H., J. K. Lee, and T. Park (2007). "Boolean networks using the chi-square test for inferring large-scale gene regulatory networks". In: *BMC Bioinformatics* 8.1, p. 37. DOI: 10.1186/1471-2105-8-37. URL: http://dx.doi.org/10.1186/1471-2105-8-37 (cit. on p. 7).

Kim, Y. et al. (2013). "Inference of dynamic networks using time-course data". In: *Briefings in Bioinformatics* 15.2, p. 212. DOI: 10.1093/bib/bbt028. eprint: /oup/backfile/Content_public/Journal/bib/15/2/10.1093/bib/bbt028/2/bbt028.pdf. URL: +http://dx.doi.org/10.1093/bib/bbt028 (cit. on p. 31).

Kitano, H. (2001). *Foundations of Systems Biology*. Ed. by H. Kitano. 1st. MIT Press (cit. on p. 2).

Kitano, H. (2002). "Systems biology: a brief overview." In: *Science (New York, N.Y.)* 295.5560, pp. 1662–1664. ISSN: 1095-9203. DOI: 10.1126/science.1069492. URL: http://dx.doi.org/10.1126/science.1069492 (cit. on p. 1).

Klebanov, L. B. and A. Y. Yakovlev (2008). "A nitty-gritty aspect of correlation and network inference from gene expression data". In: *Biology Direct* 3.1, p. 35. ISSN: 1745-6150. DOI: 10.1186/1745-6150-3-35. URL: http://dx.doi.org/10.1186/1745-6150-3-35 (cit. on p. 32).

La, H. (2006). "Relationships between probabilistic Boolean networks and dynamic Bayesian networks as models of gene regulatory networks". In: *Signal Processing* 86, pp. 814–834. DOI: 10.1016/j.sigpro.2005.06.008 (cit. on p. 5).

Lawrence, J. W. et al. (2001). "Differential gene regulation in human versus rodent hepatocytes by peroxisome proliferator-activated receptor (PPAR) alpha. PPAR alpha fails to induce peroxisome proliferation-associated genes in human cells independently of the level of receptor expresson." eng. In: *J Biol Chem* 276.34, pp. 31521–31527. ISSN: 0021-9258 (Print); 0021-9258 (Linking). DOI: 10.1074/jbc.M103306200 (cit. on p. 68).

Leon, S. B.-T. de and E. H. Davidson (2009). "Modeling the dynamics of transcriptional gene regulatory networks for animal development". In: *Developmental biology* 325.2, pp. 317–328. DOI: 10.1016/j.ydbio.2008.10.043. URL: http://www.ncbi.nlm.nih.gov/pmc/articles/PMC4100934/ (cit. on p. 5).

Li, F. et al. (2004). "The yeast cell-cycle network is robustly designed". In: *Proceedings of the National Academy of Sciences of the United States of America* 101.14, pp. 4781–4786. DOI: 10.1073/pnas.0305937101. URL: http://www.ncbi.nlm.nih.gov/pmc/articles/PMC387325/ (cit. on p. 5).

Li, P. et al. (2007). "Comparison of probabilistic Boolean network and dynamic networks". In: *BMC Bioinformatics* 8. DOI: 10.1186/1471-2105-8-S7-S13 (cit. on p. 5).

Liebsch, M. et al. (2011). "Alternatives to animal testing: current status and future perspectives". In: *Archives of Toxicology* 85.8, pp. 841–858. ISSN: 1432-0738. DOI: 10.1007/s00204-011-0718-x. URL: http://dx.doi.org/10.1007/s00204-011-0718-x (cit. on p. 70).

List, C. (2012). "The theory of judgment aggregation: an introductory review". In: *Synthese* 187.1, pp. 179–207. ISSN: 1573-0964. DOI: 10.1007/s11229-011-0025-3. URL: http://dx.doi.org/10.1007/s11229-011-0025-3 (cit. on pp. 11, 46).

Lloyd, S. (1982). "Least squares quantization in PCM". In: *IEEE Transactions on Information Theory* 28.2, pp. 129–137. ISSN: 0018-9448. DOI: 10.1109/TIT.1982.1056489 (cit. on p. 27).

Lluberes, M., J. Seguel, and J. Ramirez Vick (2011). "Markov Model Checking of Probabilistic Boolean Network Representations of Genes". In: *Proceedings of the 2011 Int. Conf. on Bioinformatics and Computational Biology*. Vol. 1, pp. 63–67.

MacQueen, J. (1967). "Some methods for classification and analysis of multivariate observations". In: *Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability, Volume 1: Statistics*. Berkeley, Calif.: University of California Press, pp. 281–297. URL: http://projecteuclid.org/euclid.bsmsp/1200512992 (cit. on pp. 27, 29).

Markowetz, F. and R. Spang (2007). "Inferring cellular networks – a review". In: *BMC Bioinformatics* 8.6, S5. ISSN: 1471-2105. DOI: 10.1186/1471-2105-8-S6-S5. URL: http://dx.doi.org/10.1186/1471-2105-8-S6-S5 (cit. on p. 5).

Martin, S. et al. (2007). "Boolean dynamics of genetic regulatory networks inferred from microarray time series data". In: *Bioinformatics* 23.7, p. 866. DOI: 10.1093/bioinformatics/btm021. eprint: /oup/backfile/Content_public/Journal/bioinformatics/23/7/10.1093/bioinformatics/btm021/2/btm021.pdf. URL: +http://dx.doi.org/10.1093/bioinformatics/btm021 (cit. on p. 5).

Matsumura, H. et al. (2005). "SuperSAGE". In: *Cellular Microbiology* 7.1, pp. 11–18. ISSN: 1462-5822. DOI: 10.1111/j.1462-5822.2004.00478.x. URL: http://dx.doi.org/10.1111/j.1462-5822.2004.00478.x.

Military to curtail use of live animals in medical training. URL: https://www.bostonglobe.com/news/nation/2014/11/11/pentagon-takes-major-steps-phase-out-use-live-animals-medical-training/2XOfgaevD80qsHs1A1SbNJ/story.html (cit. on p. 71).

Müssel, C. et al. (2016). "BiTrinA—multiscale binarization and trinarization with quality analysis". In: *Bioinformatics* 32.3, p. 465. DOI: 10.1093/bioinformatics/

btv591. URL: +http://dx.doi.org/10.1093/bioinformatics/btv591 (cit. on pp. 7, 10).

Nature News. URL: http://www.nature.com/news/2006/061211/full/news061211-9.html (cit. on p. 71).

Needham, C. J. et al. (2009). "From gene expression to gene regulatory networks in Arabidopsis thaliana". In: *BMC Systems Biology* 3.1, p. 85. ISSN: 1752-0509. DOI: 10.1186/1752-0509-3-85. URL: http://dx.doi.org/10.1186/1752-0509-3-85.

NIH Act. URL: https://grants.nih.gov/grants/olaw/pl103-43.pdf (cit. on p. 67).

NIH Alternative Methods. URL: https://ntp.niehs.nih.gov/iccvam/docs/about_docs/pl106545.pdf (cit. on p. 69).

NIH Evaluation of EU Methods. URL: https://ntp.niehs.nih.gov/pubhealth/evalatm/test-method-evaluations/dermal-corr-irrit/in-vitro-dermal-corr/tmer/index.html (cit. on p. 69).

Nilsson, B. et al. (2007). "Threshold-free high-power methods for the ontological analysis of genome-wide gene-expression studies". In: *Genome Biology* 8.5, R74–R74. DOI: 10.1186/gb-2007-8-5-r74. URL: http://www.ncbi.nlm.nih.gov/pmc/articles/PMC1929143/ (cit. on pp. 10, 11).

Otsu, N. (1979). "A Threshold Selection Method from Gray-Level Histograms". In: *IEEE Transactions on Systems, Man, and Cybernetics* 9.1, pp. 62–66. ISSN: 0018-9472. DOI: 10.1109/TSMC.1979.4310076 (cit. on p. 8).

Paige, S. L. et al. (2015). "Molecular Regulation of Cardiomyocyte Differentiation". In: *Circulation Research* 116.2, pp. 341–353. ISSN: 0009-7330. DOI: 10.1161/CIRCRESAHA.116.302752. eprint: http://circres.ahajournals.org/content/116/2/341.full.pdf. URL: http://circres.ahajournals.org/content/116/2/341 (cit. on p. 6).

Perrin, B.-E. et al. (2003). "Gene networks inference using dynamic Bayesian networks". In: *Bioinformatics* 19.suppl$_2$, p. ii138. DOI: 10.1093/bioinformatics/btg1071. eprint: /oup/backfile/Content_public/Journal/bioinformatics/19/suppl_2/10.1093/bioinformatics/btg1071/2/btg1071.pdf. URL: +http://dx.doi.org/10.1093/bioinformatics/btg1071 (cit. on p. 5).

Philippi, N. et al. (2009). "Modeling system states in liver cells: survival, apoptosis and their modifications in response to viral infection." eng. In: *BMC Syst Biol* 3, p. 97. ISSN: 1752-0509 (Electronic); 1752-0509 (Linking). DOI: 10.1186/1752-0509-3-97 (cit. on p. 5).

Plous, S. and H. Herzog (2001). "Reliability of Protocol Reviews for Animal Research". In: *Science* 293.5530, pp. 608–609. ISSN: 0036-8075. DOI: 10.1126/science.1061621. eprint: http://science.sciencemag.org/content/293/5530/608. URL: http://science.sciencemag.org/content/293/5530/608 (cit. on p. 68).

Pound, P. et al. (2004). "Where is the evidence that animal research benefits humans?" In: *BMJ : British Medical Journal* 328.7438, pp. 514–517. URL: http://www.ncbi.nlm.nih.gov/pmc/articles/PMC351856/ (cit. on p. 68).

Russell, W. M. S. et al. (2005). *The principles of humane experimental technique*. URL: http://altweb.jhsph.edu/pubs/books/humane_exp/foreword (cit. on p. 67).

Sahoo, D. et al. (2007). "Extracting binary signals from microarray time-course data". In: *Nucleic Acids Research*. DOI: 10.1093/nar/gkm284. eprint: http://nar.oxfordjournals.org/content/early/2007/05/21/nar.gkm284.full.pdf+html. URL: http://nar.oxfordjournals.org/content/early/2007/05/21/nar.gkm284.short (cit. on pp. 7, 27).

Samaga, R. et al. (2009). "The Logic of EGFR/ErbB Signaling: Theoretical Properties and Analysis of High-Throughput Data". In: *PLoS Computational Biology* 5.8. Ed. by A. R. Asthagiri, e1000438. DOI: 10.1371/journal.pcbi.1000438. URL: http://www.ncbi.nlm.nih.gov/pmc/articles/PMC2710522/ (cit. on p. 5).

Sanchez, L and D Thieffry (2001). "A logical analysis of the Drosophila gap-gene system." eng. In: *J Theor Biol* 211.2, pp. 115–141. ISSN: 0022-5193 (Print); 0022-5193 (Linking). DOI: 10.1006/jtbi.2001.2335 (cit. on p. 5).

Seguel, J. and M. Lluberes (2015). "A unified approach to the computation and analysis of strings of gene expression states". In: *2015 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*, pp. 1687–1693. DOI: 10.1109/BIBM.2015.7359929 (cit. on pp. xix, 20, 27).

Seguel, J. (2015). "Multi-Algorithmic Approaches to Gene Expression Binarization". In: *Proceedings of the International Conference on Bioinformatics Models, Methods*

and Algorithms (BIOSTEC 2015), pp. 109–115. ISBN: 978-989-758-070-3. DOI: 10.5220/0005203701090115 (cit. on pp. 11, 46).

Seguel, J. and M. Lluberes (2013). "Semantics and Accuracy of Gene Expression Threshold Computations A Case Study". In: *IARIA*, pp. 1–6. ISSN: 2308-4499 (cit. on pp. xix, 29).

Shmulevich, I. and E. Dougherty (2010). *Probabilistic Boolean Networks*. Society for Industrial and Applied Mathematics. DOI: 10.1137/1.9780898717631. eprint: http://epubs.siam.org/doi/pdf/10.1137/1.9780898717631. URL: http://epubs.siam.org/doi/abs/10.1137/1.9780898717631.

Shmulevich, I, E. R. Dougherty, and W Zhang (2002a). "Control of stationary behavior in probabilistic Boolean networks by means of structural intervention". In: *Journal of Biological Systems* 10.4, pp. 431–446. URL: http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.18.7179{\&}rep=rep1{\&}type=pdf (cit. on p. 17).

Shmulevich, I. and E. R. Dougherty (2007). *Genomic Signal Processing (Princeton Series in Applied Mathematics)*. Princeton, NJ, USA: Princeton University Press. ISBN: 0691117624, 9780691117621 (cit. on p. 7).

Shmulevich, I., E. R. Dougherty, and W. E. I. Zhang (2002b). "From Boolean to Probabilistic Boolean Networks as Models of Genetic Regulatory Networks". In: *IEEE* 90.10. DOI: 10.1109/JPROC.2002.804686 (cit. on pp. 17, 20, 21).

Shmulevich, I., E. R. Dougherty, and W. Zhang (2002c). "Gene perturbation and intervention in probabilistic Boolean networks." In: *Bioinformatics (Oxford, England)* 18.10, pp. 1319–31. ISSN: 1367-4803. URL: http://www.ncbi.nlm.nih.gov/pubmed/12376376 (cit. on p. 21).

Shmulevich, I. and W. Zhang (2002). "Binary analysis and optimization-based normalization of gene expression data". In: *Bioinformatics* 18.4, p. 555. DOI: 10.1093/bioinformatics/18.4.555. eprint: /oup/backfile/Content_public/Journal/bioinformatics/18/4/10.1093/bioinformatics/18.4.555/2/180555.pdf. URL: +http://dx.doi.org/10.1093/bioinformatics/18.4.555 (cit. on pp. 7, 17, 23).

Shmulevich, I. et al. (2002). "Probabilistic Boolean Networks: a rule-based uncertainty model for gene regulatory networks." In: *Bioinformatics (Oxford, England)* 18.2,

pp. 261–74. ISSN: 1367-4803. URL: http://www.ncbi.nlm.nih.gov/pubmed/11847074 (cit. on pp. 5, 17, 18).

Smolen, P., D. A. Baxter, and J. H. Byrne (2000). "Modeling transcriptional control in gene networks—methods, recent results, and future directions". In: *Bulletin of Mathematical Biology* 62.2, pp. 247–292. ISSN: 1522-9602. DOI: 10.1006/bulm.1999.0155. URL: http://dx.doi.org/10.1006/bulm.1999.0155 (cit. on p. 4).

Styczynski, M. P. and G. Stephanopoulos (2005). "Overview of computational methods for the inference of gene regulatory networks". In: *Computers And Chemical Engineering* 29.3. Computational Challenges in Biology, pp. 519 –534. ISSN: 0098-1354. DOI: http://dx.doi.org/10.1016/j.compchemeng.2004.08.029. URL: http://www.sciencedirect.com/science/article/pii/S009813540400242X (cit. on p. 5).

Sun, B. et al. (2007). *Study of Leukotriene B4 (LTB4) effect on primary human monocytes transcription profile.* Tech. rep. PRJNA103009. URL: http://www.ncbi.nlm.nih.gov/bioproject/PRJNA103009 (cit. on pp. 51, 72).

Tarca, A. L., R. Romero, and S. Draghici (2006). "Analysis of microarray experiments of gene expression profiling". In: *American Journal of Obstetrics & Gynecology* 195.2, pp. 373–388. DOI: 10.1016/j.ajog.2006.07.001. URL: http://dx.doi.org/10.1016/j.ajog.2006.07.001.

Taylor, K. et al. (2008). "Estimates for worldwide laboratory animal use in 2005." eng. In: *Altern Lab Anim* 36.3, pp. 327–342. ISSN: 0261-1929 (Print); 0261-1929 (Linking) (cit. on p. 67).

Transcriptomic analysis of human lung development, GEO. URL: https://www.ncbi.nlm.nih.gov/geo/query/acc.cgi?acc=GSE14334 (cit. on p. 15).

Trull, F. L. and B. A. Rich (1999). "More Regulation of Rodents". In: *Science* 284.5419, pp. 1463–1463. ISSN: 0036-8075. DOI: 10.1126/science.284.5419.1463. eprint: http://science.sciencemag.org/content/284/5419/1463. URL: http://science.sciencemag.org/content/284/5419/1463 (cit. on p. 67).

Tuna, S. and M. Niranjan (2010). "Inference from Low Precision Transcriptome Data Representation". In: *Journal of Signal Processing Systems* 58.3, pp. 267–279. ISSN:

1939-8115. DOI: 10.1007/s11265-009-0363-2. URL: http://dx.doi.org/10.1007/s11265-009-0363-2 (cit. on p. 6).

USDA AWA Report 2005. URL: https://www.usda.gov/oig/webdocs/33002-03-SF.pdf (cit. on p. 68).

USDA AWA Report 2014. URL: https://www.usda.gov/oig/webdocs/33601-0001-41.pdf (cit. on p. 68).

Validated And Accepted Alternative Methods. URL: http://alttox.org/mapp/table-of-validated-and-accepted-alternative-methods/ (cit. on p. 69).

Wilczynski, B. and E. E. Furlong (2010). "Challenges for modeling global gene regulatory networks during development: Insights from Drosophila". In: *Developmental Biology* 340.2. Special Section: Gene Regulatory Networks for Development, pp. 161 –169. ISSN: 0012-1606. DOI: http://dx.doi.org/10.1016/j.ydbio.2009.10.032. URL: http://www.sciencedirect.com/science/article/pii/S0012160609012913 (cit. on p. 4).

Worp, H. B. van der et al. (2010). "Can animal models of disease reliably inform human studies?" In: *PLoS medicine* 7.3, e1000245+. ISSN: 1549-1676. DOI: 10.1371/journal.pmed.1000245. URL: http://dx.doi.org/10.1371/journal.pmed.1000245 (cit. on p. 68).

Wu, M., X. Yang, and C. Chan (2009). "A dynamic analysis of IRS-PKR signaling in liver cells: a discrete modeling approach." eng. In: *PLoS One* 4.12, e8040. ISSN: 1932-6203 (Electronic); 1932-6203 (Linking). DOI: 10.1371/journal.pone.0008040 (cit. on p. 5).

Yamamoto, M. et al. (2001). "Use of serial analysis of gene expression (SAGE) technology". In: *Journal of Immunological Methods* 250.1–2. Gene Expression Technologies, pp. 45 –66. ISSN: 0022-1759. DOI: http://dx.doi.org/10.1016/S0022-1759(01)00305-2. URL: http://www.sciencedirect.com/science/article/pii/S0022175901003052.

Yeast Cell Cycle Analysis Project. URL: http://genome-www.stanford.edu/cellcycle/ (cit. on pp. 34, 72).

Zhang, R. et al. (2008). "Network model of survival signaling in large granular lymphocyte leukemia". In: *Proceedings of the National Academy of Sciences* 105.42,

pp. 16308–16313. DOI: 10.1073/pnas.0806447105. eprint: http://www.pnas.org/content/105/42/16308.full.pdf. URL: http://www.pnas.org/content/105/42/16308.abstract (cit. on p. 5).

Zhou, X., X. Wang, and E. R. Dougherty (2003). "Binarization of Microarray Data on the Basis of a Mixture Model1". In: *Molecular Cancer Therapeutics* 2.7, pp. 679–684. ISSN: 1535-7163. eprint: http://mct.aacrjournals.org/content/2/7/679.full.pdf. URL: http://mct.aacrjournals.org/content/2/7/679 (cit. on pp. 8–10).