PATHWAYS, NETWORKS AND THERAPY: A BOOLEAN APPROACH TO SYSTEMS BIOLOGY

A Dissertation

by

RITWIK KUMAR LAYEK

Submitted to the Office of Graduate Studies of
Texas A&M University
in partial fulfillment of the requirements for the degree of

DOCTOR OF PHILOSOPHY

May 2012

Major Subject: Electrical Engineering

PATHWAYS, NETWORKS AND THERAPY: A BOOLEAN APPROACH TO SYSTEMS BIOLOGY

A Dissertation

by

RITWIK KUMAR LAYEK

Submitted to the Office of Graduate Studies of
Texas A&M University
in partial fulfillment of the requirements for the degree of

DOCTOR OF PHILOSOPHY

Approved by:

Co-Chairs of Committee,	Aniruddha Datta
	Edward R. Dougherty
Committee Members,	Shankar P. Bhattacharyya
	N. Sivakumar
Head of Department,	Costas N. Georghiades

May 2012

Major Subject: Electrical Engineering

ABSTRACT

Pathways, Networks and Therapy: A Boolean Approach to Systems Biology. (May 2012)

Ritwik Kumar Layek, B.Tech., Indian Institute of Technology, Kharagpur;
M.Tech., Indian Institute of Technology, Kharagpur

Co–Chairs of Advisory Committee: Dr.Aniruddha Datta
Dr. Edward R. Dougherty

The area of systems biology evolved in an attempt to introduce mathematical systems theory principles in biology. Although we believe that all biological processes are essentially chemical reactions, describing those using precise mathematical rules is not easy, primarily due to the complexity and enormity of biological systems. Here we introduce a formal approach for modeling biological dynamical relationships and diseases such as cancer. The immediate motivation behind this research is the urgency to find a practicable cure of cancer, *the emperor of all maladies*. Unlike other deadly endemic diseases such as plague, dengue and AIDS, cancer is characteristically heterogenic and hence requires a closer look into the genesis of the disease. The actual cause of cancer lies within our physiology. The process of cell division holds the clue to unravel the mysteries surrounding this disease. In normal scenario, all control mechanisms work in tandem and cell divides only when the division is required, for instance, to heal a wound *platelet derived growth factor* triggers cell division. The control mechanism is tightly regulated by several biochemical interactions commonly known as signal transduction pathways. However, from mathematical point of view, these pathways are marginal in nature and unable to cope with the multi-variability of a heterogenic disease like cancer.

The present research is possibly one first attempt towards unraveling the mysteries surrounding the dynamics of a proliferating cell. A novel yet simple methodology is developed to bring all the marginal knowledge of the signaling pathways together to form the simplest mathematical abstract known as the *Boolean Network*. The malfunctioning in the cell by genetic mutations is formally modeled as stuck-at faults in the underlying Network. Finally a mathematical methodology is discovered to optimally find out the possible best combination drug therapy which can drive the cell from an undesirable condition of proliferation to a desirable condition of quiescence or apoptosis. Although, the complete biological validation was beyond the scope of the current research, the process of in-vitro validation has been already initiated by our collaborators. Once validated, this research will lead to a bright future in the field on personalized cancer therapy.

To my family

ACKNOWLEDGMENTS

I would like to express my gratitude to my advisors Dr. Aniruddha Datta and Dr. Edward R. Dougherty for helping and motivating me throughout my stay at Texas A&M University. Also I would like to thank Dr. Shankar P. Bhattacharyya, Dr. J. Venkatraj, Dr. Michael Bittner and Dr. N. Sivakumar for their kind discussions regarding various research topics. Finally I am thankful to all my friends and colleagues at Texas A&M University for their supports in various forms.

TABLE OF CONTENTS

CHAPTER Page

I INTRODUCTION 1

A. Systems biology 2
1. Biology of the cell 5
2. DNA, gene, genetic code and the central dogma of molecular biology 8
3. Genetic regulation 11
4. Signal transduction pathways 11
5. Systems medicine 12
B. Dynamical systems 12
C. Dissertation outline 14

II GENETIC REGULATORY NETWORKS: MODELING AND INTERVENTION* 15

A. Systems and methods 16
1. Probabilistic Boolean networks 16
2. Infinite-horizon control: perfect modeling 19
3. Adaptive infinite-horizon control 23
B. Algorithms 27
1. Algorithm 1 27
2. Algorithm 2 30
C. Examples 31
1. Artificial example 31
2. Melanoma application 34
D. Concluding remarks 37

III FROM PATHWAYS TO NETWORKS* 39

A. Notation and digital design basics 41
1. Boolean networks 41
2. Karnaugh map representation of Boolean networks 42
B. From pathways to a family of BNs: a simple example 46
1. Iterative update of K-maps 50
C. From pathways to a family of BNs: the general procedure . . . 54
1. Definitions and preliminary observations 54

CHAPTER Page

2. Priority ordering between Boolean functions 56
3. Conflict and its resolution 58
4. Total conflict and cyclic total conflict 61

D. Simple example revisited 62

E. Network design to satisfy additional constraints 67

1. Imposition of attractor constraints 68
2. Boolean network from predictors 69

F. Modeling pathways involving the p53 gene 71

1. Boolean network modeling of the p53 pathways 72
2. Model validation using the published literature 73

G. Concluding remarks 81

IV FAULT DETECTION AND INTERVENTION IN BNS* 83

A. Modeling cancer as faults in the signaling network 83

1. Test inputs and fault detectability 87

B. Modeling drug intervention 87

C. Biological example: growth factors and cellular signal transduction 89

1. Cell cycle control, DNA mutation and cancer 89

D. Growth factor mediated pathways: combinatorial network . . . 91

1. Input-output simulation of the BN 93
2. Modeling faults and therapeutic interventions using the Boolean network 93
3. Fault analysis and classification 98
    a. Single fault simulation 98
    b. Fault classification 99
4. Simulation results for drug intervention 100
    a. Continuous real mapping of the output vector 100
    b. Interpretation of the result 103

E. p53 mediated DNA damage pathways: sequential network . . . 103

1. Fault analysis 104
2. Intervention design 104

F. Concluding remarks 106

V CONCLUSION 107

REFERENCES 111

VITA 120

LIST OF TABLES

TABLE Page

I Major Breakthroughs in Biology before 1800 AD [1]. 3

II Major Breakthroughs in Biology after 1800 AD [1]. 4

III The Genetic Code [2]. 9

IV Truth Table of the Boolean Network (Eqn. 3.1). 43

V Priority Ordering 56

VI Input-output Mapping of the Boolean Network of Fig. 31. 95

VII Steady State Attractors in the presence of Single Stuck-at Faults. 105

VIII Intervention Design for the Critical Faults in ATM-p53-Mdm2-Wip1 Boolean Network. 105

LIST OF FIGURES

FIGURE Page

1 An Eukaryotic Cell. 6

2 Central Dogma of Molecular Biology. 10

3 Dynamical Systems. 13

4 State Transition Diagrams for a Boolean Network and a Probabilistic Boolean Network. 17

5 Adaptive Control Algorithm. 28

6 Family of Probabilistic Boolean Networks. 29

7 Artificial Example:Algorithm 1. 33

8 Artificial Example: Algorithm 2. 33

9 Artificial Example: Cost Difference Comparison of the two Algorithms for different values of q. 34

10 Melanoma Application: Algorithm 1. 36

11 Melanoma Application: Algorithm 2. 37

12 Melanoma Application: Cost difference comparison of the two algorithms for different values of q. 38

13 State transition diagram of the Boolean Network (Eqn. 3.1). 43

14 Karnaugh map representation of Table IV. 44

15 Conflict in K-map of gene B. 60

16 Solvability of the conflicting pathway problem. 63

17 Steady state distribution and threshold. 68

FIGURE Page

18 State transition diagram for the Boolean network described by Equation (3.13). 71

19 ATM-p53-Wip1-Mdm2 pathways (From [3]). 74

20 State transition diagram for the Boolean Network of the p53 pathways under normal conditions. 75

21 State transition diagram for the Boolean Network of the p53 pathways in the presence of DNA damage. 76

22 Oscillation of the proteins in the presence of the DNA damage signal. . . 77

23 Timelapse fluorescence images of one cell over 29 h after 5 Gy of gamma irradiation. Nuclear p53-CFP and Mdm2-YFP are imaged in green and red, respectively. Time is indicated in hours. Adapted by permission from Macmillan Publishers Ltd: [Molecular Systems Biology] [4], copyright (2006) 78

24 Immunoblots of ATM-P(S1981), Chk2-P(T68), and p53 kinetics in MCF7 cells irradiated with 10Gy of gamma-irradiation. Reprinted from [5], Copyright (2008), with permission from Elsevier. 79

25 Immunoblots of Chk2-P(T68), p53, and Mdm2 kinetics in MCF7 cells treated with 400 ng/ml NCS every hour. Blots are representative of triplicate experiments. Reprinted from [5], Copyright (2008), with permission from Elsevier. 80

26 Immunoblots of p53 and Wip1 kinetics in MCF7 cells irradiated with 10 Gy of gamma-irradiation. Reprinted from [5], Copyright (2008), with permission from Elsevier. 81

27 Stuck-at faults and bridging faults in a digital circuit. 86

28 Drug intervention modeling. 88

29 The eucaryotic Cell Cycle(G0: Quiscence, G1: Gap 1, R: Restriction point, S: Synthesis, G2: Gap 2 and M: Mitosis) and the signal transduction pathways controlling the cell cycle. 90

FIGURE Page

30 A schematic diagram of the growth factor signaling pathways (the yellow color is used for the reporter proteins which will be measured in future experiments). 92

31 An input output Boolean network model of the signalling pathways of Fig. 30. 94

32 Possible fault locations and drug intervention locations: (a) proliferative stuck-at fault locations and (b) intervention locations for the available cancer drugs. 97

33 Single fault simulation: (a) output simulation in presence of all proliferative single stuck-at faults for input V = 00001 and (b) equivalent faults for input V = 00001. 99

34 Drug vector response in the presence of a single fault: (Left) output responses of the circuit for all drug vectors in presence of all single stuck-at faults and (Right) the map between the color codes and the output vectors. 102

35 Iterative update scheme of pathways and therapeutic target point knowledge in systems biology. 108

36 Personalized medicine using systems biology. 109

CHAPTER I

INTRODUCTION

Biology is a natural science concerning the study of life and living organisms. The traditional approach to biology advocates that life is also a complex manifestation of physical interactions. However, even in the 21st century, science has not been able to decipher the complete picture of this bottom-up approach towards biology. Indeed, it is fair to say that we have a long way to go to establish biology on a firm mathematical basis.

As scientific researchers, our main goal is to ferret out the inherent truth from different natural phenomena to the best of our ability, and for this we employ standard scientific techniques such as mathematical modeling, design of experiments, actual experimentation, data collection, data interpretation and validation. In the last few centuries, mathematics has grown enormously to accommodate the modeling and experimental paradigms for the elucidation of scientific theories.

However, unlike the physical sciences, biology continues to predominantly be an observational science. For instance, if we examine the work of early stalwarts like Charles Darwin or Gregor Johann Mendel, we see that their postulates and theories are mostly observational and intuitive in nature. Although several centuries have elapsed since then and technology has evolved a lot, philosophically we are still treating biology as an observational science. In addition to technical challenges, biological research through the ages has been impeded by human ethical and morality considerations. For instance, the anatomical dissection of dead bodies was prohibited during the medieval period, and even today embryonic stem cell research is restricted throughout the world although it has the potential to yield easier solutions for treating diseases requiring organ transplantation.

The journal model is *IEEE Transactions on Automatic Control.*

Before going into the actual introduction to the current dissertation, it behooves us to take a look at the timeline of the major developments in the biological sciences (Table I and II).

Although the discoveries listed in Tables I and II are by no means comprehensive, they do provide us with some flavor of the mainstream biological research. We note that there is no theorem or formula or mathematical model associated with most of these discoveries. The recent advancements in genetics, genomics and medical science have introduced a critical need for mathematically rigorous approaches in biological research. However, unfortunately, we still have a long way to go.

A. Systems biology

For a moment, let us think about the status of physics prior to Galileo Galilei(1564 AD-1642 AD) and Isaac Newton (1642 AD-1727 AD). At that time, physics was not a coherent science. Medieval physicists were busy doing research on the perpetual motion machine, the elixir of life and the sorcerer's stone, to name a few. Without proper mathematical background and systematic understanding most research during that time was in some sense an exercise in futility. A similar observation could be made about the biological research during the last century. Without proper mathematical modeling of the inherent dynamical system, research on fundamental biology and medicine mostly focussed on the good old methods of trial and error. However, even during this time, several scientists such as Erwin Schrodinger and Norbert Wiener understood that the unification of mathematics and biology could prove to be extremely beneficial. The new direction that emerged from this idea of unification is called 'Systems Biology'.

There are numerous definitions of systems biology but we want to mention the one given by The National Institute of Health (NIH).

A discipline at the intersection of biology, mathematics, engineering and the physical sci-

Table I. Major Breakthroughs in Biology before 1800 AD [1].

Year	Breakthrough
520 BC	Alcmaeon of Croton distinguished veins from arteries and discovered the optic nerve.
450 BC	Sushruta wrote the Sushruta Samhita, describing over 120 surgical instruments and 300 surgical procedures, classifying human surgery into eight categories, and introducing cosmetic and plastic surgery.
450 BC	Xenophanes examined fossils and speculated on the evolution of life.
350 BC	Aristotle attempted a comprehensive classification of animals.
300 BC	Herophilos dissected the human body.
150 AD	Claudius Galen wrote numerous treatises on human anatomy.
800 AD	Al-Jahiz describes the struggle for existence, introduces the idea of a food chain, and adheres to environmental determinism.
1628 AD	William Harvey published ’An Anatomical Exercise on the Motion of the Heart and Blood in Animals’.
1658 AD	Jan Swammerdam observed red blood cells under a microscope.
1663 AD	Robert Hooke saw cells in cork using a microscope.
1683 AD	Anton van Leeuwenhoek observed bacteria.

Table II. Major Breakthroughs in Biology after 1800 AD [1].

Year	Breakthrough
1828 AD	Friedrich Woehler synthesized urea; first synthesis of an organic compound from inorganic starting materials.
1856 AD	Louis Pasteur stated that microorganisms produce fermentation.
1858 AD	Charles R. Darwin proposed a theory of biological evolution.
1865 AD	Gregor Mendel demonstrated in pea plants that inheritance follows definite rules.
1869 AD	Friedrich Miescher discovered nucleic acids in the nuclei of cells.
1902 AD	Walter Sutton and Theodor Boveri, independently proposed that the chromosomes carry the hereditary information.
1928 AD	Alexander Fleming discovered the first antibiotic, penicillin.
1953 AD	James D. Watson and Francis Crick published a double-helix structure for DNA.
1961 AD	J. Heinrich Matthaei cracked the first codon of the genetic code.
1996 AD	Dolly the sheep was first clone of an adult mammal.
2001 AD	Publication of the first draft of the complete human genome.

ences that integrates experimental and computational approaches to study and understand biological processes in cells, tissues and organisms. Studies at the systems level are distinguished not only by their quantitative nature in data collection and mathematical modeling, but also by their focus on interactions among individual elements such as genes, proteins and metabolites. These studies often integrate data from multiple levels of the biological information hierarchy in an environmental and evolutionary context and pay particular attention to dynamic processes that vary in time and space. Successive iterations of experiment and theory development are characteristic of systems biology. When applied to human health, systems biology models are intended to predict physiological behavior in response to natural and artificial perturbations and thereby contribute to the understanding and treatment of human diseases[6].

The current dissertation will provide a preliminary but novel approach for mathematical modeling of different cellular phenomena and its possible application in systems biology. To put the subsequent chapters in proper context, in the next few sections we will discuss several biological processes which require mathematical insight and modeling. This discussion on cell biology, genetics and genomics is necessary to properly appreciate the motivation and flow of this dissertation.

1. Biology of the cell

The word ‘cell’ comes from the the Latin word ‘cellula’ meaning a small room. Cells are membrane bounded units containing different organelles performing different functions. A cell is the basic unit of life. The simplest forms of life may be solitary cells that reproduce by dividing in two, while higher organisms are ensemble of cells where a group of cells is designated for a particular functionality.

Living cells emerged on earth about 3.5 billion years ago, possibly by spontaneous reactions between molecules in an environment that was far from chemical equilibrium.

These reactions formed some simple organic molecules like amino acids, sugars, etc which, by polymerization through peptide bonds and phosphodiester bonds, then led to the formation of polypeptides and polynucleotides (RNA), that could catalyze their own replications. With time, one of these families of cooperating RNA catalysts developed the ability for direct synthesis of polypeptides. Finally, as the accumulation of additional protein catalysts allowed more efficient and complex cells to evolve, the DNA double helix replaced RNA as a more stable molecule for storing the increased amount of genetic information required by such cells[2].

A schematic diagram of a typical eukaryotic cell is shown in Fig. 1.

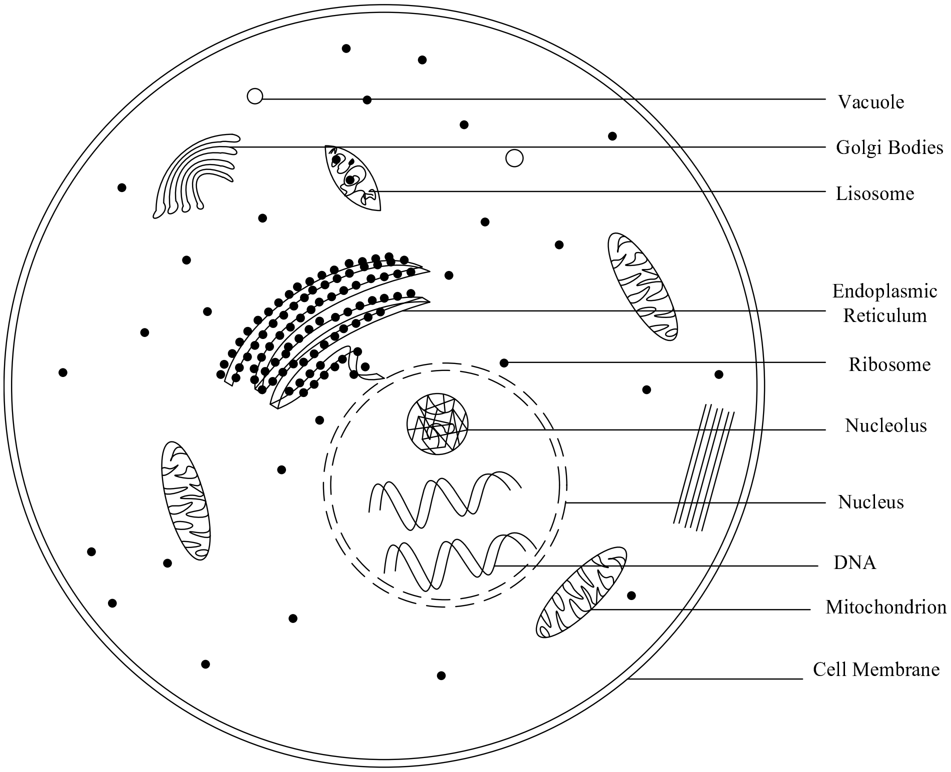


Fig. 1. An Eukaryotic Cell.

From this diagram we can see the important components of the eukaryotic cell. Brief descriptions are given below[2, 1].

- Cell membrane: The cell membrane is a selectively permeable membrane made up of a lipid bilayer and embedded proteins. It protects the intra-cellular environment and helps the cell in its motility and communication.

- Nucleus: The nucleus is a central part of the cell containing most of the cell's genetic material, organized as multiple DNA molecules in combination with a large variety of proteins to form chromosomes. Its function is to maintain the integrity of the DNA and to control the activities of the cell by regulating gene expression.

- Nucleolus: Nucleolus is a discrete densely stained structure inside the nucleus. Its main role is to transcribe ribosomal RNA (rRNA) and assemble ribosomes.

- Endoplasmic Reticulum(ER): Endoplasmic reticulum is an interconnected network of tubules and vesicles. Rough endoplasmic reticulum (with ribosomes) synthesizes and transports proteins, while smooth endoplasmic reticulum (without ribosome) synthesizes lipids, steroids and morphine, metabolizes carbohydrates, regulates drug metabolism and the attachment of receptors on cell membrane proteins.

- Ribosome: Ribosome is the protein factory in the cell. The mRNA (messenger RNA) molecule leaves the nucleus and enters the Ribosome. Ribosome reads the codons (nucleotide triplets) from the mRNA and puts the corresponding amino acids according to the genetic code.

- Mitochondrion: Mitochondrion is the power plant of the cell. This membrane enclosed organelle supplies the Adenosine triphosphate (ATP) required by the cell for meeting its energy needs.

- Lysosome: Lysosome destroys cellular debris by using its hydrolase enzymes. It helps the cell to rejuvenate by destroying old organelles. It is also known as the ‘suicide bag’ of the cell.

- Golgi apparatus: Golgi apparatus processes and packages protein molecules for delivering elsewhere. It helps in intra-cellular communication and secretion.

- Vacuoles: Vacuoles are membrane bound organelles used for carrying toxic elements out of the cell, maintaining pressure and pH inside the cell.

2. DNA, gene, genetic code and the central dogma of molecular biology

Deoxyribonucleic acid (DNA) contains most of the genetic instructions inside the cell. The DNA segments carrying these instructions are called genes. DNA consists of two long strands of nucleotides with backbones made of sugar and phosphate joined by phosphoester bonds. These two strands run in opposite directions to each other. Attached to each sugar is one of the four types of bases - Adenine(A), Guanine(G), Cytosine(C) and Thymine(T). Adenine and Guanine belong to the double ringed class of molecules called purines, whereas cytosine and thymine are single ringed pyrimidines. It is the sequence of these four bases along the backbone that encodes the genetic information. In the double helical DNA structure, Adenine always binds with Thymine and Cytosine binds with Guanine through triple and double bonds respectively. Ribonucleic acid (RNA) is the temporary carrier of genetic instructions from the DNA to the Ribosome. RNA is a single stranded polynucleotide containing Uracil(U) in liu of Thymine(T).

Amino acids serve as the building blocks of protein. There are twenty amino acids which are naturally incorporated into polypeptides.

The genetic code provides the unique map between the sequence of three consecutive bases(codon) in the mRNA and the amino acids (Table III) [7]. The mRNA molecule is decoded on ribosomes using the genetic code to synthesize the relevant protein. The steps

Table III. The Genetic Code [2].

Amino acid/control	code(s)
Alanine/Ala/A	GCU, GCC, GCA, GCG
Arginine/Arg/R	CGU, CGC, CGA, CGG, AGA, AGG
Asparagine/Asn/N	AAU, AAC
Aspartic acid/Asp/D	GAU, GAC
Cysteine/Cys/C	UGU, UGC
Glutamine/Gln/Q	CAA, CAG
Glutamic acid/Glu/E	GAA, GAG
Glycine/Gly/G	GGU, GGC, GGA, GGG
Histidine/His/H	CAU, CAC
Isoleucine/Ile/I	AUU, AUC, AUA
Leucine/Leu/L	UUA, UUG, CUU, CUC, CUA, CUG
Lysine/Lys/K	AAA, AAG
Methionine/Met/M	AUG
Phenylalanine/Phe/F	UUU, UUC
Proline/Pro/P	CCU, CCC, CCA, CCG
Serine/Ser/S	UCU, UCC, UCA, UCG, AGU, AGC
Threonine/Thr/T	ACU, ACC, ACA, ACG
Tryptophan/Trp/W	UGG
Tyrosine/Tyr/Y	UAU, UAC
Valine/Val/V	GUU, GUC, GUA, GUG
Start	AUG
Stop	UAA, UGA, UAG

for going from DNA to protein are as follows.

DNA sequences are copied into RNA molecules in the process termed *Transcription*; a gene that is transcribed is said to be actively expressed, while a gene that is not transcribed is considered as repressed. Normally transcription of a gene yields an RNA molecule of length similar to the gene itself. Once synthesized, the base sequences of the RNA molecule are *translated* by the ribosomes into a sequence of amino acids. The resulting molecule folds up into a unique three-dimensional configuration and becomes a functional protein [8]. The complete unidirectional information flow for protein biosynthesis from the gene is referred to as the *central dogma of molecular biology* (Fig. 2).

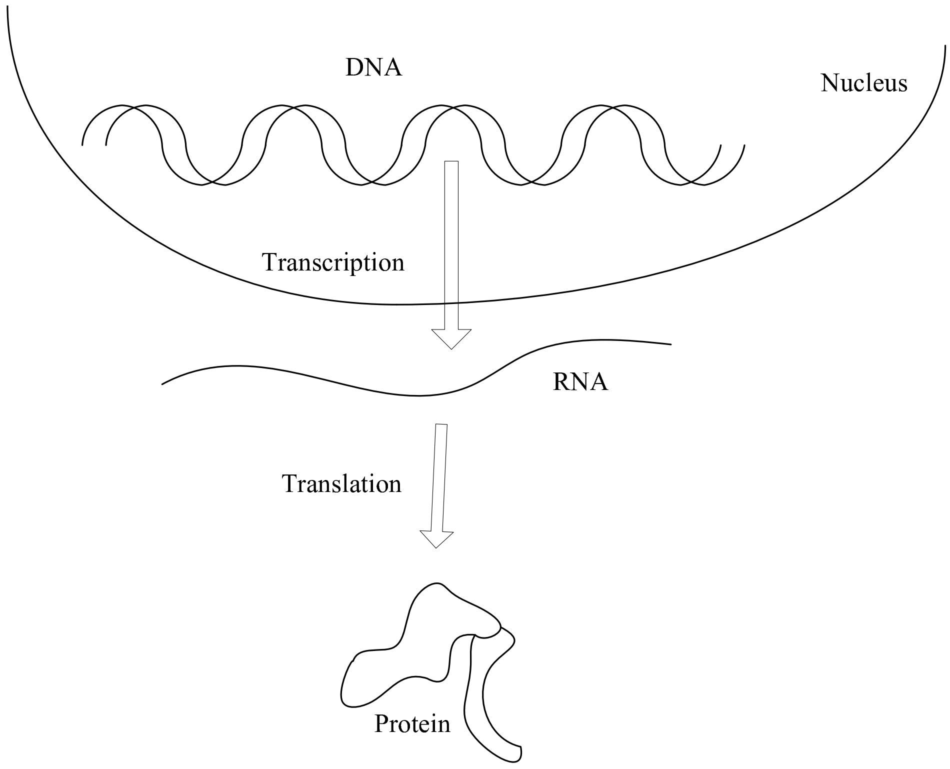


Fig. 2. Central Dogma of Molecular Biology.

3. Genetic regulation

Some proteins in the cell are called housekeeping proteins. Their corresponding genes are constitutively active to maintain the protein concentration. However, there are certain other proteins which are normally not present inside the cell all the time. Only when the protein is required, the corresponding gene is turned ‘ON’. The mechanism by which a particular gene is turned ‘ON’ or ‘OFF’ is called genetic regulation. The proteins which can bind to the DNA to start the transcription process are called *transcription factors*. These transcription factors are also regulated by other transcriptional or enzymatic activities. The complex gene-protein-RNA interactions are instrumental in maintaining cellular homeostasis.

From a systems viewpoint, the behavior of a living cell is analogous to that of a multi-input-multi-output (MIMO) feedback system. Understanding this system is the most important challenge in systems biology.

4. Signal transduction pathways

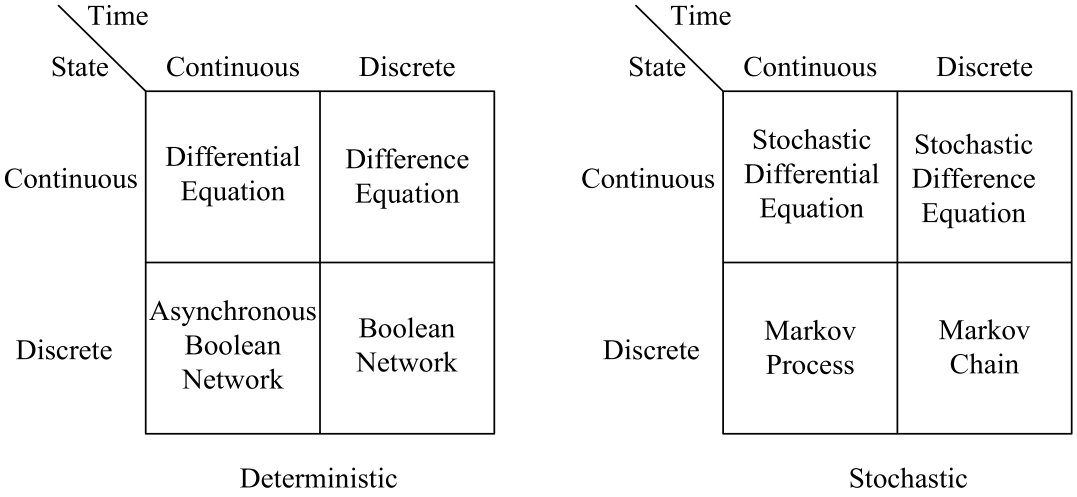
In multi-cellular organisms, life is sustained by a systematic coordination between different cells and all extra cellular signals. Each cell has its own functionality and its future is determined by various intrinsic and extrinsic biological signals. For instance, a cell’s proliferation, differentiation or induction of apoptosis are determined by a number of different signals. From the time of a cell’s birth (by division of its parent cell), the cell’s state is tightly controlled by different biological regulations. Cell signaling is a form of communication between different cells. These signals can be chemical or electrical impulses. Communication via electrical impulses is typically associated with nerve cells (neurons) which are attached to each other and the action potential transmits from neuron to neuron. For general somatic cells, proteins are usually the signaling molecules used for communication. The interactions between the different signaling molecules are multivariate in nature and hence difficult to study. As a result, historically biologists have focussed on studying the marginal interaction between the signaling molecules, leading to what is called *pathway* information. Although pathway knowledge cannot provide the complete multivariate picture of the overall cellular signal transduction, it is clear that one has to have a mechanism for incorporating this prior information into any signal transduction model that one develops. An approach to do precisely that will be discussed elaborately in CHAPTER III and is also reported in [9].

5. Systems medicine

The area of systems medicine focusses on the problem of treating a complex disease such as cancer. Any disease is nothing but the lack of order in the system. Systems diseases such as cancer are possibly caused by mutations in the DNA. Malfunctions in the interactions between the genes and proteins cause disruption in the normal cellular dynamics. Systems medicine seeks to restore the earlier dynamics of the cell or terminate the cell if such restoration is not possible. This problem is different from that in systems biology because here controlling the dynamics of the system is more important than knowing the system accurately. In systems theory, there are different approaches for controlling a system even if the system is not fully known. CHAPTERs II and IV discuss therapeutic intervention and systems medicine and these results have been also reported in [10] and [11].

B. Dynamical systems

Before starting the mathematical modeling of biological systems, it is appropriate to introduce dynamical systems. A *dynamical system* can be thought of as a rule for determining the time evolution of a system state (vector). Although any real world dynamical system is continuous both in time and space, for modeling simplicity we often discretize these variables. In addition, the mathematical rule determining the state transition can be either deterministic or stochastic. Based on these considerations, we can get different kinds of dynamical systems as shown in Fig. 3.



State \ Time	Continuous	Discrete
Continuous	Differential Equation	Difference Equation
Discrete	Asynchronous Boolean Network	Boolean Network

Deterministic

State \ Time	Continuous	Discrete
Continuous	Stochastic Differential Equation	Stochastic Difference Equation
Discrete	Markov Process	Markov Chain

Stochastic

Fig. 3. Dynamical Systems.

Although the actual protein concentrations in the cell are continuous variables, there are at least three reasons why a discrete type of modeling would be preferred. First, although the continuous model may dictate the exact dynamics, using the current technology it is impossible to reliably measure the concentration of each protein inside the cell in real time. Second, many of the genes/proteins inside the cell exhibit ON/OFF switch-like behavior which is more readily accommodated using quantization within the digital domain [12], [13]. Third, the discrete-time systems are easier to analyze, model and control in real time in comparison to continuous-time systems [14]. Hence, in this dissertation we talk mostly about the two discrete-time discrete-state models namely, Boolean Network (BN) and Markov Chain/Probabilistic Boolean Network (PBN). BN and PBN are formally introduced in CHAPTERs II and III.

C. Dissertation outline

The dissertation is organized as follows:

- CHAPTER II: Boolean networks and probabilistic Boolean networks are formally introduced as two simple models of genetic regulatory networks. Adaptive intervention in generic probabilistic Boolean networks(BN is a trivial subset of PBN) is described as a method for arriving at an intervention strategy that is practically implementable.

- CHAPTER III: Cell signaling pathways are considered to be the knowledge base for building Boolean networks. The synthesis algorithm is explained in detail and illustrative examples are included. Some experimental validation results from the existing literature are also presented.

- CHAPTER IV: Intervention strategies are designed for both combinatorial and sequential Boolean networks based on some realistic modeling of therapeutic intervention. An example of the growth factor mediated pathways is presented, and this example is relevant to cell cycle control and cancer. An intervention strategy is also designed for a sequential Boolean network (or a feedback network). An example from DNA damage stress response pathways is presented.

- CHAPTER V: Finally a futuristic research direction for systems biology is outlined where the starting point for experimental design is the existing knowledge from past biological research.

CHAPTER II

GENETIC REGULATORY NETWORKS: MODELING AND INTERVENTION*

There are two major objectives for modeling of genetic regulatory networks: (i) to better understand inter-gene (and protein) interactions and relationships on a holistic level, thereby facilitating the diagnosis of disease; and (ii) to design and analyze therapeutic intervention strategies for shifting the state of a diseased network from an undesirable location to a desirable one. Many different approaches have been proposed in the literature for modeling the behaviour of genetic regulatory networks. Of these, the model which has received the most attention in the context of therapeutic intervention is the probabilistic Boolean network (PBN). To date, a number of approaches have been proposed for carrying out interventions in PBNs based on stochastic optimal control theory for Markov chains [15, 16, 17]. These assume perfect knowledge of the underlying PBN, an assumption which, when not satisfied in practice, can lead to degraded or unacceptable performance. To remedy the situation, one could design a fixed intervention strategy that is “robust”, or somewhat insensitive, to modeling errors, in particular, to the effect of uncertainties in the transition probability matrix of a PBN, Another approach is to “tune” the intervention strategy to the actual network via on-line adaptation. The aim of this chapter is to demonstrate the feasibility of such an adaptive approach in the framework of PBNs. At the very outset, it is important to point out that such a scheme is feasible only if the uncertainty belongs to a specific class and prior knowledge about this class can be incorporated into the design.

*Part of this chapter is reprinted with permission from “Adaptive intervention in probabilistic boolean networks” by R. Layek, A. Datta, R. Pal, and E. R. Dougherty, 2009, Bioinformatics, vol. 25, no. 16, pp. 2042-2048, Copyright [2009], Oxford University Press. (http://bioinformatics.oxfordjournals.org/content/25/16/2042.short)

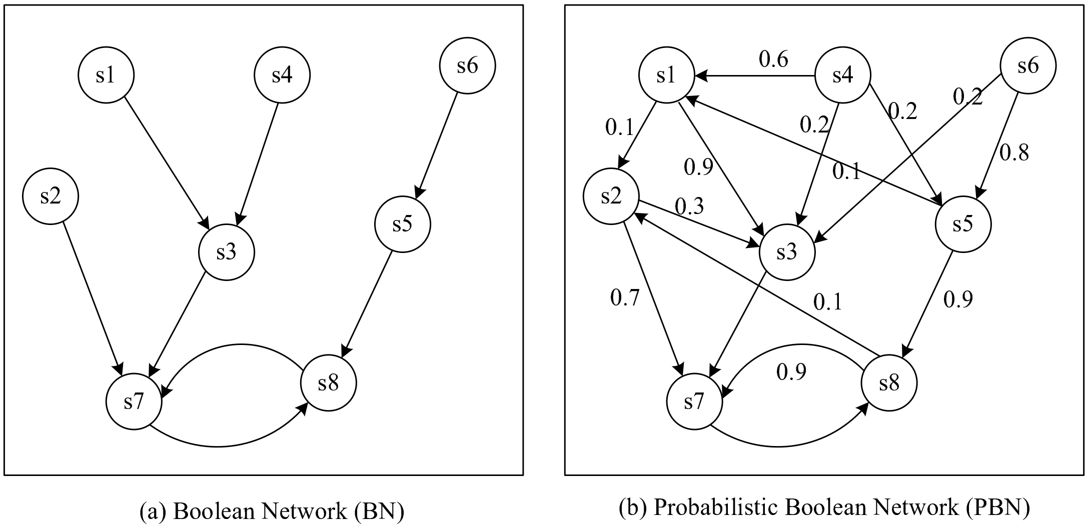
A. Systems and methods

1. Probabilistic Boolean networks

A *Boolean Network* (*BN*), $\Upsilon = (V, F)$, on n genes is defined by a set of nodes/genes $V = \{x_1, ..., x_n\}$, $x_i \in \{0, 1\}$, $i = 1, ..., n$, and a list $F = (f_1, ..., f_n)$, of Boolean functions, $f_i : \{0, 1\}^n \rightarrow \{0, 1\}$, $i = 1, ..., n$ [18]. Each node x_i represents the state/expression of the i^{th} gene, where $x_i = 0$ means that gene i is OFF and $x_i = 1$ means that gene i is ON. The function f_i is called the *predictor function* for gene i. Updating the states of all genes in B is done synchronously at every time step according to their predictor functions. At time t, the network state is given by $x(t) = (x_1(t), x_2(t), ..., x_n(t))$, called a *gene activity profile* (*GAP*). The state (GAP) transition diagram of a typical BN is shown in Fig. 4(a). A *Probabilistic Boolean Network* (*PBN*) consists of a set of nodes/genes $V = \{x_1, ..., x_n\}$, $x_i \in \{0, 1, ..., d\}$, $i = 1, ..., n$, and a set of vector valued network functions, $\mathbf{f}_1, \mathbf{f}_2, ..., \mathbf{f}_k$, governing the state transitions of the genes. For $j = 1, 2, ..., k$, $\mathbf{f}_j = (f_{j1}, f_{j2}, ..., f_{jn})$, where $f_{ji} : \{0, 1, ..., d\}^n \rightarrow \{0, 1, ..., d\}$, $i = 1, ..., n$ [19, 20] In most applications, the discretization is either binary or ternary. Here we use binary quantization, $d = 1$, which presents no theoretical limitation on the development. At each time point a random decision is made as to whether to switch the network function for the next transition, with the probability q of a switch being a system parameter. If the decision is to switch, then a new function is chosen from among $\mathbf{f}_1, \mathbf{f}_2, ..., \mathbf{f}_k$, with c_j being the probability of choosing $\mathbf{f}_j$ (network selection is not conditioned by the current network, which can itself be selected). Each network function $\mathbf{f}_j$ determines a BN, the individual BNs being called the *contexts* of the PBN. The PBN behaves as a fixed BN until a decision is made to switch contexts according to the probabilities $c_1, c_2, ..., c_k$ from among $\mathbf{f}_1, \mathbf{f}_2, ..., \mathbf{f}_k$. If $q = 1$, the PBN is said to be *instantaneously random*; if $q < 1$ [21], the PBN is said to be *context-sensitive.* We consider PBNs with perturbation, meaning that at each time point there is a probability p of any gene flipping its value uniformly randomly. Since there are n genes, the probability of a random perturbation at any time point is $1-(1-p)^n$. A context-sensitive PBN determines a Markov chain whose states are (context, GAP) pairs. The transition probability from $(s,\mathbf{y})$ to $(r,\mathbf{x})$ is given by

$$\begin{aligned} P_{s,\mathbf{y}}(r,\mathbf{x}) &= \mathbf{1}_{[r=s]}((1-q)+qc_s)\{\mathbf{1}_{[\mathbf{f}_s(\mathbf{y})=\mathbf{x}]}(1-p)^n \\ &+ \mathbf{1}_{[\mathbf{x}\neq\mathbf{y}]}p^{\eta(\mathbf{x},\mathbf{y})}(1-p)^{n-\eta(\mathbf{x},\mathbf{y})}\} \\ &+ \mathbf{1}_{[r\neq s]}qc_r\{\mathbf{1}_{[\mathbf{f}_r(\mathbf{y})=\mathbf{x}]}(1-p)^n \\ &+ \mathbf{1}_{[\mathbf{x}\neq\mathbf{y}]}p^{\eta(\mathbf{x},\mathbf{y})}(1-p)^{n-\eta(\mathbf{x},\mathbf{y})}\}, \end{aligned} \tag{2.1}$$

The state (GAP) transition diagram of a typical PBN (or Markov Chain) is shown in Fig. 4(b). where r,s denote the rth and sth BNp (Boolean Network with perturbation), which



(a) Boolean Network (BN)

(b) Probabilistic Boolean Network (PBN)

Fig. 4. State Transition Diagrams for a Boolean Network and a Probabilistic Boolean Network.

are the BNps at time $t+1$ and t, where $\eta(\mathbf{x},\mathbf{y})$ is the Hamming distance between $\mathbf{x}$ and $\mathbf{y}$, and $\mathbf{1}_{[\mathbf{f}(\mathbf{y})=\mathbf{x}]}$ is the indicator function that takes value 1 if $\mathbf{f}(\mathbf{y})=\mathbf{x}$ according to the rule structure and is equal to 0 otherwise. The random perturbation makes the Markov chain irreducible and ergodic. Thus, it possesses a steady-state distribution. Since there are k contexts and 2^n GAPs in each network, the Markov chain possesses $k2^n$ states and we can relabel the states with $z(t) \in \{0, 1, 2, \cdots, 2^n k - 1\}$ being the state that is occupied by the network at time t. For an instanteously random PBN, the Markov chain reduces so that its states are the GAPs of the PBN. The transition probability expression (2.2) can be used to track the time evolution of the (context, GAP) state. In practice it may be impossible to detect context, only the GAP. We obtain the transition probabilities between the GAPs by taking the expectation of the (context, GAP) transition probabilities over the networks, the transition probability from GAP $\mathbf{y}$ to GAP $\mathbf{x}$ being given by

$$P_{\mathbf{y}}(\mathbf{x}) = (1-p)^n \sum_{i=1}^{k} \mathbf{1}_{[\mathbf{f}_i(\mathbf{y})=\mathbf{x}]} c_i + \mathbf{1}_{[\mathbf{x} \neq \mathbf{y}]} p^{\eta(\mathbf{x},\mathbf{y})} (1-p)^{n-\eta(\mathbf{x},\mathbf{y})} \tag{2.2}$$

Using the above equations we can compute the $2^n \times 2^n$ transition probability matrix corresponding to the averaged context-sensitive PBN. As shown in [22], the transition probability matrix for an averaged context-sensitive PBN is the same as that of an instantaneously random PBN that makes use of the same constituent Boolean networks. It is possible that some of the transition probabilities computed using (2.2) may evaluate out to zero. The corresponding transitions are referred to as *forbidden transitions* and the adaptive algorithms to be presented in this chapter require that the set F of such forbidden transitions be known.

Remark 1. *The transition probability expressions derived in this subsection allow for the possibility of different selection probabilities for the different constituent Boolean networks of a PBN. However, in the absence of any prior knowledge, we will henceforth assume a uniform distribution of the selection probabilities, i.e.* $c_i = \frac{1}{k},\ i = 1, 2, \cdots, k$.

2. Infinite-horizon control: perfect modeling

In this section, we summarize some results on the infinite-horizon control of PBNs, assuming perfect modeling. A PBN with control can be modeled as a stationary discrete-time dynamic system

$$z_{t+1} = f(z_t, u_t, w_t),\ t = 0, 1, ..., \tag{2.3}$$

where for all t, the state z_t is an element of a state space S , the control input u_t is an element of a control space C, the disturbance w_t is an element of a space D and $f : S \times C \times D \mapsto S$. [1] In the particular case of a PBN with n genes composed of m Boolean networks with perturbation probability p and network transition probability q, $S = \{0, 1, 2, ..., m(2^n - 1)\}$ and the control input u_t is constrained to take values in the space $C = [0, 1, ..., 2^k - 1]$, where k is the number of binary control inputs. The disturbance w_t is manifested in terms of change of network based on the network transition probability q or change of state due to perturbation probability p. w_t is independent of prior disturbances $w_0, w_1....w_{t-1}$. The objective is to derive a sequence of control inputs, a *control strategy*, such that some cost function is minimized over the entire class of allowable control strategies. We define a cost per stage, $\tilde{g}(i, u, j)$, depending on the origin state i, the destination state j, and the applied control input u. [2] The actual design of a “good” cost function is application dependent and is likely to require considerable expert knowledge. In finite-horizon control one can sum the costs over the number of time points constituting the horizon and take the expectation; however, this cannot safely be done with infinite horizon because the summation of the

[1] In the rest of this chapter, we will be denoting the time dependence of z, u and w by the subscript t. In all other situations, the context will make it clear whether a subscript denotes time dependence or reference to the particular component of a vector.

[2] Note that while finite horizon control problems in the literature allow for costs-per-stage functions that vary from one stage to another, infinite horizon control problems in the literature have typically been derived assuming that the same cost per stage function is used for all stages. For PBNs (both context sensitive and otherwise), this is not of any consequence since all of our earlier finite horizon results also used the same cost per stage function for all stages.

one-stage costs might diverge to infinity (for all controls), thereby leading to an ill-posed optimization problem. One way to avoid the problem of a possibly infinite total cost is by considering the *average cost per stage* which is defined by

$$J_\pi(z_0) = \lim_{M\to\infty} \frac{1}{M} E\{\sum_{t=0}^{M-1} \tilde{g}(z_t, \mu_t(z_t), z_{t+1})\} \tag{2.4}$$

where the expectation is with respect to both origin and destination states. In this formulation, a control policy $\pi = \{\mu_0, \mu_1, \cdots\}$ is chosen to minimize the above cost and the problem is referred to as the *average cost per stage problem*. Minimization of the total cost is feasible if $J_\pi(z_0)$ is finite for at least some admissible policies π and some admissible states z_0. If there is no zero-cost absorbing state (which is the case in context-sensitive PBNs with perturbation), then the total cost will frequently go to ∞. Hence the *average cost per stage* formulation is essential when we are interested in the condition of the patient in the long run and equal importance is given to the patient's condition in all stages. In general, the cost $\tilde{g}(i, u, j)$ of moving from state i to state j under control u may depend on the starting state i; however, in the case of PBNs, we have no obvious basis for assigning different costs based on different initial states. Accordingly, we assume that the penalty $\tilde{g}(i, u, j)$ is independent of the starting state i and its value is based on the control effort and the terminal state j. The penalty is high if the end state is a bad state regardless of the starting state, and vice-versa. Hence $\tilde{g}(i, u, j) = \tilde{g}(u, j)$. Moreover, since in Eq. 2.4 the cost is obtained by taking the expectation with respect to the origin and destination states, it is possible to replace $\tilde{g}(z_t, u_t, z_{t+1})$ by an equivalent cost per stage that does not depend on the destination state by taking the expectation with respect to the destination state and leaving only the expectation with respect to the original state. More specifically, we use as cost per stage the expected cost $g(i, u)$ given by [23]:

$$g(i, u) = \sum_{j=0}^{2^n-1} p_{ij}(u)\tilde{g}(i, u, j) = \sum_{j=0}^{2^n-1} p_{ij}(u)\tilde{g}(u, j) \tag{2.5}$$

where $p_{ij}(u)$ is the transition probability under control u.

To solve the average-cost-per-stage optimal control problem, let Π denote the set of all *admissible* policies π, i.e., the set of all function sequences $\pi = \ \mu_0, \mu_1, ...$ with $\mu_t(x) : S \to C, t = 0, 1, ...$. The optimal cost function J^*,which is independent of the initial state [23], is defined by

$$J^* = \min_{\pi \in \Pi} J_\pi(z), z \in S \text{ is arbitrary}. \tag{2.6}$$

A *stationary policy* is an admissible policy of the form $\pi = \ \mu, \mu, ...$ Its corresponding cost function is denoted by J_μ. A stationary policy $\pi = \ \mu, \mu....$ is optimal if $J_\mu(z) = J^*(z)$ for all states z.

To minimize the cost function of Eq. 2.4, first defiine the mapping

$$J_t(i) = \min_{u \in C} \left[g(i, u) + \sum_{j=0}^{2^n - 1} p_{ij}(u) J_{t+1}(j) \right] \tag{2.7}$$

which, although we will not go into detail, provides the dynamic programming solution for the finite-horizon problem [23]. Secondly, for any cost function $J : S \to \Re$, define the mapping $TJ : S \to \Re$ by

$$(TJ)(i) = \min_{u \in C} [g(i, u) + \sum_{j=0}^{2^n - 1} p_{ij}(u) J(j)], \ i \in S. \tag{2.8}$$

We note in passing that TJ is the optimal cost function for the one-stage (finite horizon) problem that has stage cost g and terminal cost J. For the average-cost-per-stage problem, the value iteration $J_{t+1}(i) = TJ_t(i)$ cannot be used directly because it may diverge to infinity. Thus, calculating the average cost by taking $\lim_{M \to \infty}(J_M/M)$ is not feasible. Instead, we consider a *differential cost* h_t obtained by subtracting a fixed component of J_t, say $J_t(n_1)$, from each element of J_t, i.e.,

$$h_t(i) = J_t(i) - J_t(n_1), \forall i \in S. \tag{2.9}$$

Letting $e = [1, 1, 1, \cdots, 1]^T$, the above relationship can be rewritten in vector form as

$$h_t = J_t - J_t(n_1)e.$$

Some algebraic manipulations [17] yield

$$h_{t+1} = Th_t - (Th_t)(n_1)e$$

as the *value iteration algorithm* for the differential cost. Using some additional arguments, we can arrive at the following *policy iteration* algorithm for the average cost case [23, 17]:

- 1. (Initialization): An initial policy μ_0 is selected.

- 2. (Policy Evaluation): Given a stationary policy μ_k, we obtain the corresponding average and differential costs λ_k and $h_k(i)$ satisfying

  $$\lambda_k + h_k(i) = g\left(i, \mu_k(i)\right) + \sum_{j=0}^{2^n-1} p_{ij}(\mu_k(i))h_k(j), i \in S \tag{2.10}$$

  This linear system of equations can be solved utilizing the fact that $h_k(n_1) = 0$, where $n_1 \in S$ is any particular reference state.

- 3.(Policy improvement): An improved stationary policy μ_{k+1} satisfying

  $$\begin{aligned} & g\left(i, \mu_{k+1}(i)\right) + \sum_{j=0}^{2^n-1} p_{ij}(\mu_{k+1}(i))h_k(j) \\ = & \min_{u \in C} \left[g\left(i, u\right) + \sum_{j=0}^{2^n-1} p_{ij}(u)h_k(j) \right]. \end{aligned} \tag{2.11}$$

  is obtained. The iterations are stopped if $\mu_{k+1} = \mu_k$, else we return to Step 2 and repeat the process.

3. Adaptive infinite-horizon control

We now consider an adaptive intervention strategy that can be used in the presence of model uncertainty. We assume that the underlying network is modeled by a member of a known finite family of PBNs and we have no *a priori* knowledge about which member of that family models the actual network. In such a situation, a natural approach is to estimate the model number on-line and then use policy iteration to determine the corresponding controller. This is the principle of adaptive control and considerable theoretical research has been aimed at showing that such *certainty equivalence* schemes can provide the required performance [24, 25]. Our focus will be to demonstrate via simulations the feasibility of adaptive intervention in the context of gene regulatory networks. We will use a variation of an adaptive control algorithm developed in [26] for unknown Markov chains, to which we refer for technical proofs of convergence. While the scheme in [26] attempts to estimate all entries of the transition probability matrix, our adaptive algorithm will estimate only the model number since our underlying assumption is that the transition probabilities of the PBN are completely determined, once we know the model number.

There are a number of ways in which one can possess a list of PBNs and thereby be presented with the problem of adaptively determining a model number. Several inference procedures produce PBNs by way of first producing Boolean networks satisfying some desired relation to the data. In [27], Boolean networks are constructed whose attractor structures coincide with data points assumed to be in attractors in the true biological network, along with the networks satisfying certain constraints, such as the number of predictors. Then one or more PBNs are constructed from these Boolean networks by comparing the steady-state distributions of potentially inferred PBNs with the full set of experimental data. In [28], Boolean networks are inferred by first using a Bayesian approach to generate regulatory graphs (topologies) most compatible with the data and then inferring the predictors via a nonlinear perceptron model, using a reversible jump Markov chain Monte Carlo

(MCMC) method. Then one or more PBNs are constructed from the Boolean networks by using Bayesian scores. In [29], a single PBN is constructed such that each constituent Boolean network is consistent with the data, the estimate of the expected distribution of the data generated by the PBN using its steady-steady state distribution agrees with the distribution of the data, and the latter condition cannot be accomplished with less than the number of constituent networks in the inferred PBN. While this leads to a single PBN, in order that the inferred PBN not overfit the data, and in the process be composed of an inordinately large number of Boolean networks, the data are first filtered. Thus, different filtering techniques can lead to different PBNs.

In each of the preceding cases, rather than settle on a single PBN model when applying control, one can take the view that there is a list of potential PBNs and that new data are to be used to adaptively determine the control policy. Moreover, in the cases of [27] and [28],one might not even form a PBN and simply treat the problem in the framework of a collection of Boolean networks, in which the adaptation is aimed at selecting a control policy for the governing Boolean network, a view compatible with our proposed algorithms. This latter view, that one has a collection of Boolean networks, absent a PBN structure, was taken in [30], where a finite-horizon control policy was determined that performed optimally relative to the family of networks. Here we would proceed adaptively.

In addition to inference, there is another way in which a list of PBNs can naturally occur. In [31] and [32], a PBN is derived from a mammalian cell cycle network proposed in [33] by assuming a mutation that leads to a cancerous phenotype. Specifically, in the mutation, the gene p27 can never be activated, the result being that the cell can cycle in the absence of any growth factor. A different mutation will lead to a different PBN. Thus, based on a given network, in this case the one proposed in [33], if one is unsure of the mutation that has led to a cancerous phenotype, then new data utilized in an adaptive fashion can be used to design an intervention strategy.

Suppose the family of controlled PBNs is parametrized by the parameter $\alpha \in A$ where, for any $\alpha \in A$, $\sum_{j \in S} p(i,j,u,\alpha) = 1$ for any $(i,u) \in S \times C$. [3] The only constraint on A is that every element of A results in a set of bonafide transition probabilities. The cardinality, $|A|$, of A determines the total number of possible PBNs. For each $\alpha \in A$, we can compute the uncontrolled transition probability matrix by using (2.2). In addition, for a given control gene, the rows of the *controlled* transition probability matrix can be determined as a linear transformation of the rows of the uncontrolled transition probability matrix. As shown in [34], this is a consequence of restricting the class of allowable interventions to the flipping of a chosen control gene. We use the adaptive control algorithm originally derived in [26] by maximizing a modified likelihood criterion. For each $\alpha \in A$, let $J^*(\alpha)$ be the optimal long term average cost obtained for model α using the method of the last sub-section and let $\phi(.,\alpha) : S \to C$ be the corresponding control law attaining it. Let $f : \mathbb{R} \to \mathbb{R}$, $o : \mathbb{Z} \to \mathbb{R}$, and constant m be defined as follows: f is a strictly monotonically increasing continuous function such that $f(\inf_{\alpha \in A} J^*(a)) > 0$; o is any function such that $\lim_{t \to \infty} o(t)t^{-\theta}$ is a positive finite number for some $\theta \in (0,1)$; and m is any integer such that $m > |S| + 1$.For our implementation purposes we take f as the logarithmic function and $o(t)$ as the function $o(t) = 2\sqrt(t)$, for which $\theta = 0.5$. The value of m can be satisfactorily chosen depending on the cardinality of the state space. The adaptive controller consists of two separate operations, estimation and control:

- ***Estimator:*** At each time step $0, m, 2m, 3m, ..km, (k+1)m, ...$, estimate α by

$$\hat{\alpha_t} := \text{argmax}_{a \in A} \bar{D}_t(\alpha), \tag{2.12}$$

where

$$\bar{D}_t(\alpha) := K \prod_{(i,j,u) \in F^c} p(i,j,u,\alpha)^{n_t(i,j,u)}, \tag{2.13}$$

[3] In this section, $p(i,j,u,\alpha)$ denotes $p_{ij}(u)$ when the model α has been selected.

$$K = \left[\frac{1}{f\left\{J^*(\alpha)\right\}} \right]^{o(t)}, \tag{2.14}$$

and F^c is the complement of the set of forbidden transitions F, which is assumed to be known *a priori*. These transitions correspond to zero values for $p(i, j, u, \alpha)$. In (2.13), $n_t(i, j, u)$ is defined as

$$n_t(i, j, u) = 1 + \sum_{s=0}^{t-1} \mathbf{1}(z_s = i, z_{s+1} = j, u_s = u) \tag{2.15}$$

and can be interpreted as measuring the number of times a transition occurs from i to j under control u. Here $\mathbf{1}(.)$ denotes the indicator function. At time km, knowing the parameter estimate $\hat{\alpha_{km}}$, we can find the optimal cost function $J^*(\hat{\alpha_{km}})$ and the optimal control law $\phi(z_t, \hat{\alpha_{km}})$ which will be used for the next m time steps. The parameter estimate is kept constant at $\hat{\alpha_{km}}$ between time steps km and $(k+1)m - 1$.

- *Controller:* At each time t, the control applied is

$$u_t := \phi(z_t, \hat{\alpha}_t). \tag{2.16}$$

The optimal cost function and optimal control law are determined by applying policy iteration to the estimated model.

Remark 2. *The adaptive algorithm presented here is based on the transition probability expression (2.2). Since this expression accurately models an instantaneously random PBN, it is only to be expected that performance degradation will occur as the value of q is reduced from 1 to 0. This will be borne out by our simulations in the next section.*

Remark 3. *From a practical point of view, the expectation is that the constituent Boolean networks of a PBN switch very infrequently. In other words, the value of q can be reasonably assumed to be very small. In such a scenario, one could consider each constituent Boolean network to be a possible model to be identified by the estimation algorithm. Although this increases the cardinality of the set of possible models, it is expected to result* *in improved performance especially since a small value of q means that the constituent networks will change very infrequently so that the estimation algorithm will have enough time to identify the current Boolean network. This will also be borne out by the simulation results in the chapter.*

B. Algorithms

The schematic diagram of the adaptive control algorithm is shown in Fig. 5. The controller and estimator modules are shown separately with the model set A for the estimator module explicitly indicated. Two different choices for the model set A will lead to the two different algorithms presented in this chapter. The family of PBNs is shown schematically in Fig. 6.Each member of the family consists of a number of constituent BNps. The underlying PBN is assumed to come from the family. Any switching from one underlying PBN to another is assumed to be deterministic and very infrequent so that, for all practical purposes, the estimator does not need to track a model changing with time.

1. Algorithm 1

In Algorithm 1, we assume that the family of PBNs constitutes the model set A. Note that this formulation encompasses context-sensitive PBNs, instantaneously random PBNs, and BNs with perturbation (BNps) as they are all special cases of PBNs. For each model (PBN), we can compute the transition probability matrix for the extended state space using Eqn. 2.2, but it is very difficult to determine the context number from the output state data of the actual PBN. So, constructing the transition counter matrix for the extended state space is practically impossible. For example, suppose each PBN consists of 4 contexts (4 BNps) and the actual underlying PBN is the 2nd PBN in the model set. In addition, suppose at time t there is a transition from state 5 of BNp2 (i.e, context 2) to state 8 of BNp3 (i.e, context 3). In that case, we will observe the $5 \rightarrow 8$ transition; however, in the transition counter

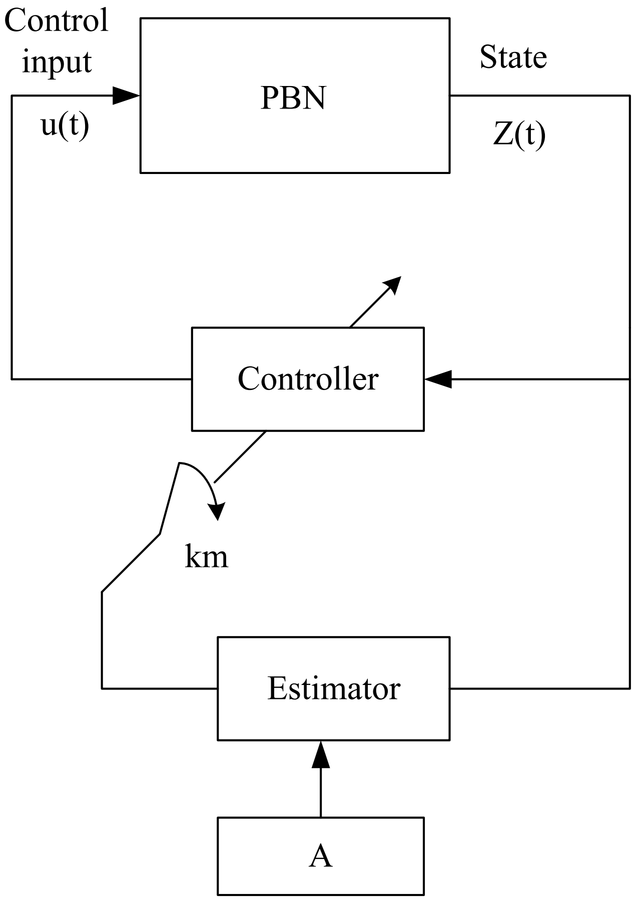


Fig. 5. Adaptive Control Algorithm.

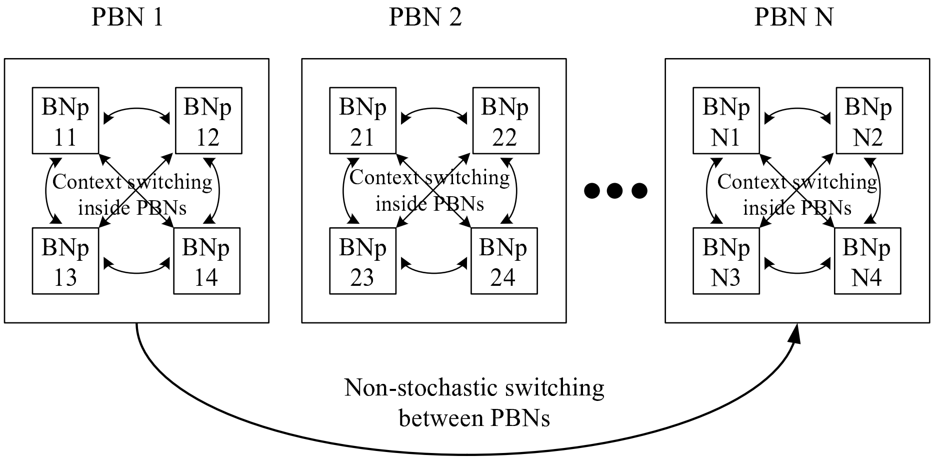

PBN 1
PBN 2
PBN N
BNp 11
BNp 12
BNp 13
BNp 14
BNp 21
BNp 22
BNp 23
BNp 24
BNp N1
BNp N2
BNp N3
BNp N4
Context switching inside PBNs
Context switching inside PBNs
Context switching inside PBNs
Non-stochastic switching between PBNs

Fig. 6. Family of Probabilistic Boolean Networks.

matrix there would be 16 elements for that particular $5 \rightarrow 8$ transition (corresponding to the different combinations of 4 source contexts and 4 destination contexts) and there is no way of figuring out which precise context switching occurred. Faced with this hurdle, we compress the transition probability matrix in such a way so that we don't need to find the extended transition counter matrix. This can be done by using equation 2.2, where the individual transition probability matrices for the different contexts have been averaged out. This averaging out causes no loss of context information when the PBN is instantaneously random since in that case there is no context information to start with; however, even when the PBN is not instantaneously random, and context information is lost, we can still use the averaged transition probability matrix to estimate the model (PBN) number of the underlying PBN. Such an algorithm using the averaged transition probability matrix will henceforth be referred to as Algorithm 1. Clearly, one would expect such an algorithm to perform well for $q = 1$ (i.e, instantaneously random PBN) with performance degradation occurring as the value of q is reduced (i.e, we are moving further and further away from an instantaneously random PBN.)

2. Algorithm 2

The main problem with Algorithm 1 is that for small values of the switching probability q (which are typically the more realistic ones), it doesn't perform well. The attractor basin structures of the different constituent BNps of a particular PBN vary significantly and so, averaging of the transition probability matrices of the different constituent BNps is not an appropriate strategy for context-sensitive PBNs with low switching probabilities. For that situation, we can consider the other extreme scenario, where $q = 0$. Then the context-sensitive PBN reduces to a single BNp. A natural question that arises in trying to estimate the underlying PBN from the state transition data is which form of the transition probability matrix should be used. A reasonable answer for $q = 0$ would be to use the individual transition probability matrix for each BNp. This significantly increases the cardinality of the model space A and leads to Algorithm 2. For instance, if we have 4 constituent BNps for each PBN as in Fig. 6, then the cardinality of the model space A will be increased by a factor of 4. Algorithm 2 assumes no context switching and uses the set of constituent BNps as the model set A. This set is used to estimate the model number and the stationary control policy is determined using the policy iteration algorithm. Using simulations it will be shown that Algorithm 2 works better than Algorithm 1 for small values of q. This is quite intuitive because we estimate the model number only after a time interval of m, and if the switching probability q is low, then the number of context switchings inside one estimation time window is expected to be quite low. So, our assumption about the BNp not changing within an estimation window is reasonable. In the next section we will discuss the simulation results for two different sets of data and compare the performance of the two algorithms for three different values of q.

C. Examples

In this section, we present simulations to demonstrate the efficacy of the proposed adaptive intervention strategies. Such simulation studies are especially important since the theoretical results in [26] guarantee only almost sure convergence and, that in a Cesaro sense[4]. We will consider two different examples of genetic regulatory networks. The first will be an artificial example and the second will be a network derived from gene expression data collected in a metastatic melanoma study. In each case, we will carry out simulation studies using the previously discussed algorithms.

1. Artificial example

We consider a 4-gene network modeled by an unknown member of a known family of context-sensitive PBNs. We assume that the cardinality of this family is 7, for each member in this family we have 4 constituent BNps, and $p = 0.01$. The value of q will be chosen differently for various simulations. Since gene values are binary, the cardinality of the state space is 16. Without loss of generality, we assume that the first gene, i.e, the gene corresponding to the most significant bit (MSB) in the gene activity profile, is the gene that needs to be down-regulated, i.e, set to 0. We assume that the second gene is the control gene that can be flipped, with $u = 1$ and $u = 0$ denoting the flipping and no flipping actions, respectively. To adaptively intervene in the network, we choose $m = 32$. The cost of control is assumed to be 0.5 and the states are assigned penalties as follows:

$$\tilde{g}(u,j) = \begin{cases} 5 & \text{if } u = 0 \text{ and MSB is } 1 \text{ for state } j \\ 5.5 & \text{if } u = 1 \text{ and MSB is } 1 \text{ for state } j \\ 0.5 & \text{if } u = 1 \text{ and MSB is } 0 \text{ for state } j \\ 0 & \text{if } u = 0 \text{ and MSB is } 0 \text{ for state } j \end{cases}$$

[4] Roughly speaking, convergence in the Cesaro sense formalizes the notion of convergence of the time average of a signal. This clearly doesn't imply pointwise convergence.

Since our objective is to down-regulate the MSB gene, a higher penalty is assigned for destination states having the MSB gene up-regulated. Also for a given MSB gene status for the destination state, a higher penalty is assigned when the control is active versus when it is not. We want to examine how algorithm 1 performs when the true model is deterministically switched. Accordingly, we set up the simulation with the actual model being switched from PBN2 (model number 2) to PBN6 (model number 6) at the 10th estimation window (time = 320). The switching probability (q) is 0.01. This emulates a context-sensitive PBN. Fig. 7 shows the convergence results. Each of the following figures shows model and cost comparisons between the non-adaptive regular controller (with complete model information) and the adaptive controller. The top plot shows the estimated and actual models as functions of the estimation time steps. The x-axis is calibrated in terms of the number of estimation windows with each window being 32 time steps long. Similarly, the bottom plot in each of the convergence figures shows the comparison of the cumulative adaptive average cost and the cumulative non-adaptive average cost (assuming perfect knowledge about the true model).From Fig. 7, it is clear that the estimated model converges to the true model and the cumulative adaptive average cost goes towards the cumulative non-adaptive average cost. Fig. 8 shows the simulation results obtained using algorithm 2 on the same simulation set up as above with $q = 0.01$. Clearly, the estimated model converges to the true model and the cumulative adaptive average cost converges to the cumulative non-adaptive average cost for the true model. The estimated model convergence in the case of algorithm 2 is much faster than that obtained using algorithm 1. This is to be expected since, with $q = 0.01$, the underlying assumptions for algorithm 2 are a better fit to the real scenario. We next study the effect of the value of q on the performance of the two algorithms. To compare the two algorithms, we cannot rely on just one simulation. Moreover, we are more interested in achieving controlled cost convergence rather than model convergence as our sole aim in intervention is to minimize the long term average cost. Accordingly, we run

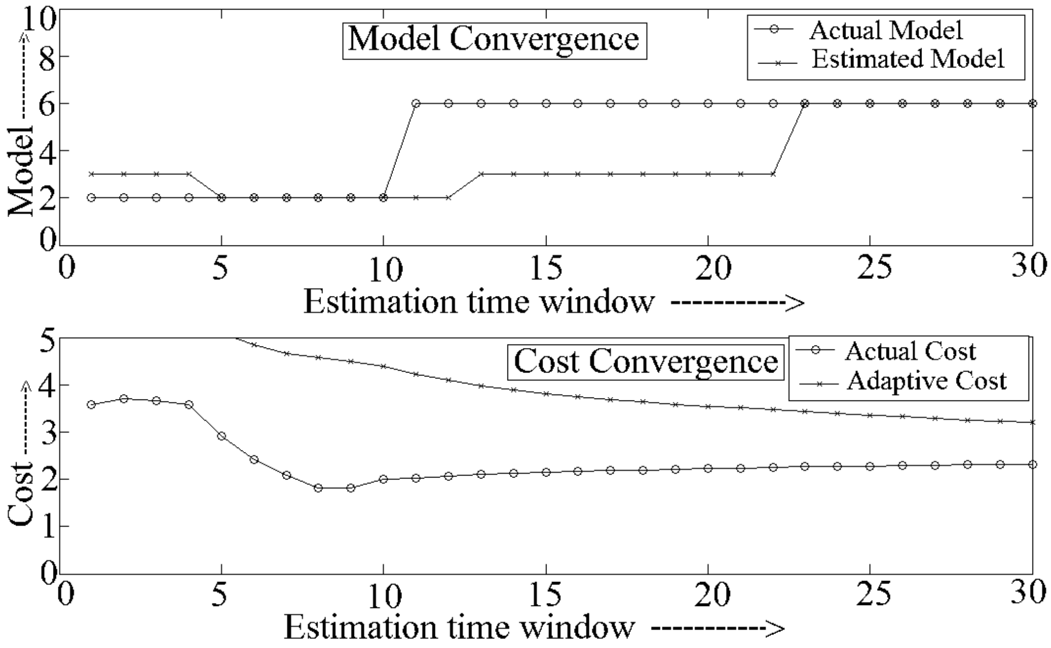


Fig. 7. Artificial Example:Algorithm 1.

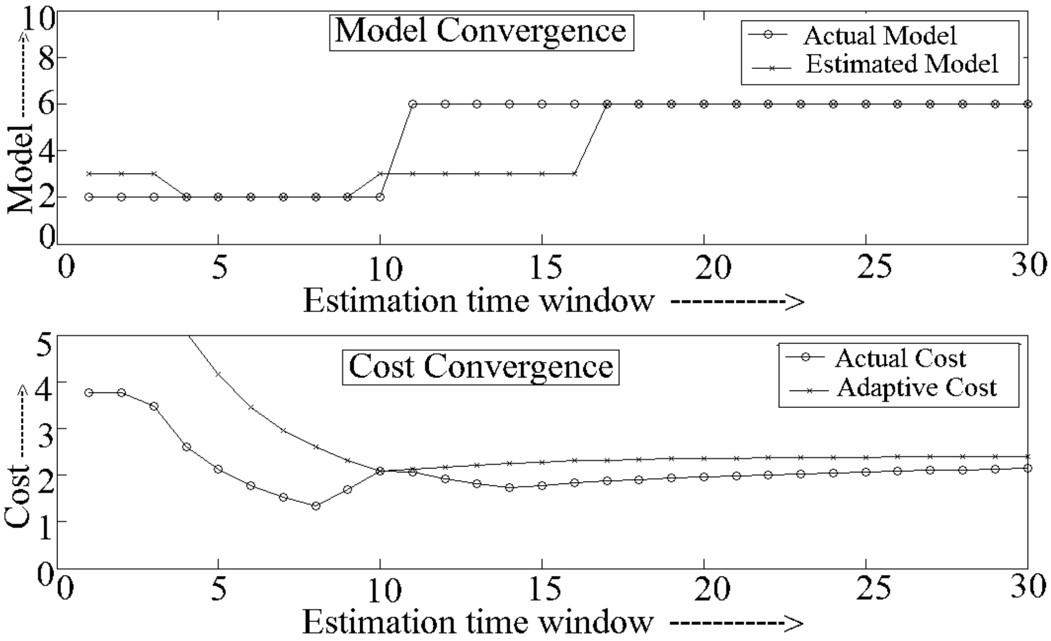


Fig. 8. Artificial Example: Algorithm 2.

the same simulation one hundred times and calculate the difference between the cumulative adaptive and cumulative non-adaptive average costs in each case. We then average the difference sequence over the 100 simulations. Fig. 9 shows the results for 30 estimation windows (time = 960) for three different values of q.From Fig. 9, we see that the results match our intuition: algorithm 1 works well for $q = 1$ (instantaneously random PBN) whereas when q is low or 0, algorithm 2 works better.

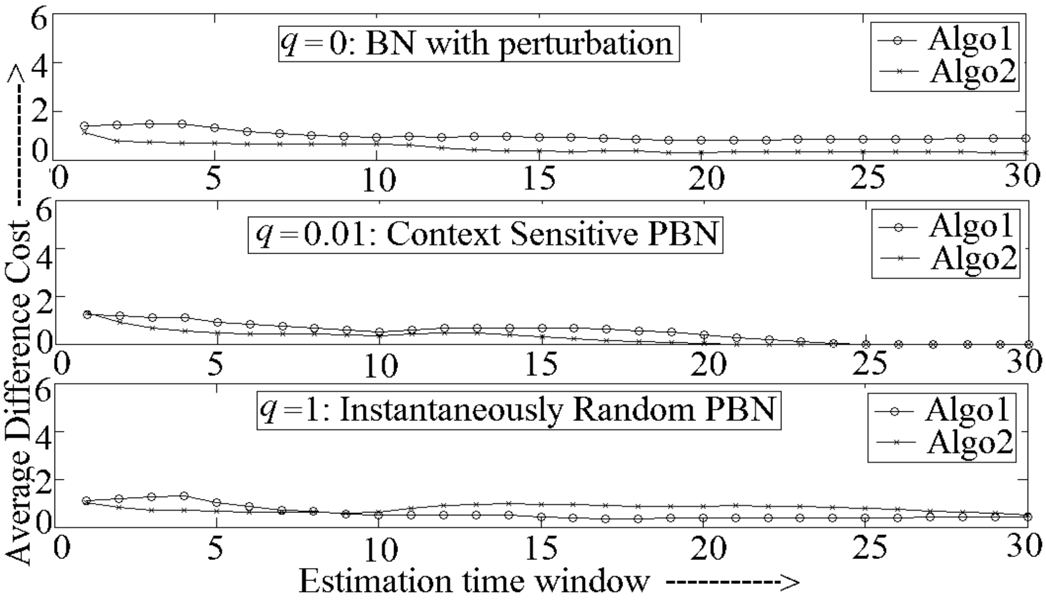


Fig. 9. Artificial Example: Cost Difference Comparison of the two Algorithms for different values of *q*.

2. Melanoma application

In a study of metastatic melanoma it was found that experimentally increasing the levels of the Wnt5a protein secreted by a melanoma cell line via genetic engineering methods directly altered the metastatic competence online as measured by the standard *in vitro* assays for metastasis [35]. Furthermore, it was found that an intervention that blocked the Wnt5a protein from activating its receptor, the use of an antibody that binds the Wnt5a protein, could substantially reduce Wnt5a's ability to induce a metastatic phenotype. This suggests a control strategy that reduces the WNT5A gene's action in affecting biological regulation, since the data suggest that disruption of this influence could reduce the chance of a melanoma metastasizing, a desirable outcome. PBNs derived from the same expression data have been used in [15, 16, 17, 34] for demonstrating earlier non-adaptive intervention strategies. We consider 7-gene PBNs containing the genes WNT5A, pirin, S100P, RET1, MART1, HADHB and STC2 obtained via the algorithms described in [27]. The states are ordered as above, with WNT5A as the most significant bit (MSB) and STC2 as the least significant bit (LSB).

We have constructed 7 PBNs with four constituent BNs in each. The adaptive intervention strategy has been applied to the family of PBNs with pirin as the control gene ($u = 1$, state of pirin is reversed, and $u = 0$, no intervention), $m = 256$, and $p = 0.01$. The value of q varies between simulations. The cost of control is assumed to be 0.5 and the states are assigned penalties as follows:

$$\tilde{g}(u,j) = \begin{cases} 5 & \text{if } u = 0 \text{ and WNT5A is } 1 \text{ for state } j \\ 5.5 & \text{if } u = 1 \text{ and WNT5A is } 1 \text{ for state } j \\ 0.5 & \text{if } u = 1 \text{ and WNT5A is } 0 \text{ for state } j \\ 0 & \text{if } u = 0 \text{ and WNT5A is } 0 \text{ for state } j \end{cases}$$

Since our objective is to down-regulate the WNT5A gene, a higher penalty is assigned for destination states having WNT5a up-regulated. Also, for a given WNT5A status for the destination state, a higher penalty is assigned when the control is active versus when it is not. Figs. 10 and 11 show the performance of the adaptive intervention schemes using algorithms 1 and 2, respectively. In each case, the genetic regulatory network is initially described by PBN4 (model number 4) and at estimation window number 10 (corresponding to time $=2560$), the underlying model is deterministically switched to PBN2 (model number 2). The switching probability (q) is assumed to be 0.01. From the model convergence plots in Figs. 10 and 11, it is clear that the estimated models track the actual model quite well. Furthermore, the model tracking using algorithm 2 is better than with algorithm 1. This is consistent with our expectation since for the small q, the underlying assumption for algorithm 2 represents a closer approximation to reality. The cumulative adaptive average costs also appear to converge to the non-adaptive ones. To see if these results are representative, we ran the same simulation one hundred times and calculated the differences between the cumulative adaptive and non-adaptive costs for each of the two algorithms.We then averaged the difference sequence over the one hundred simulations. Fig. 12 shows the plots of the average difference sequence over 30 estimation windows (time = 7680) for three different values of the switching probability q. From the figure, we see that the results match our intuition: algorithm 1 works well for $q = 1$ (instantaneously random PBN)

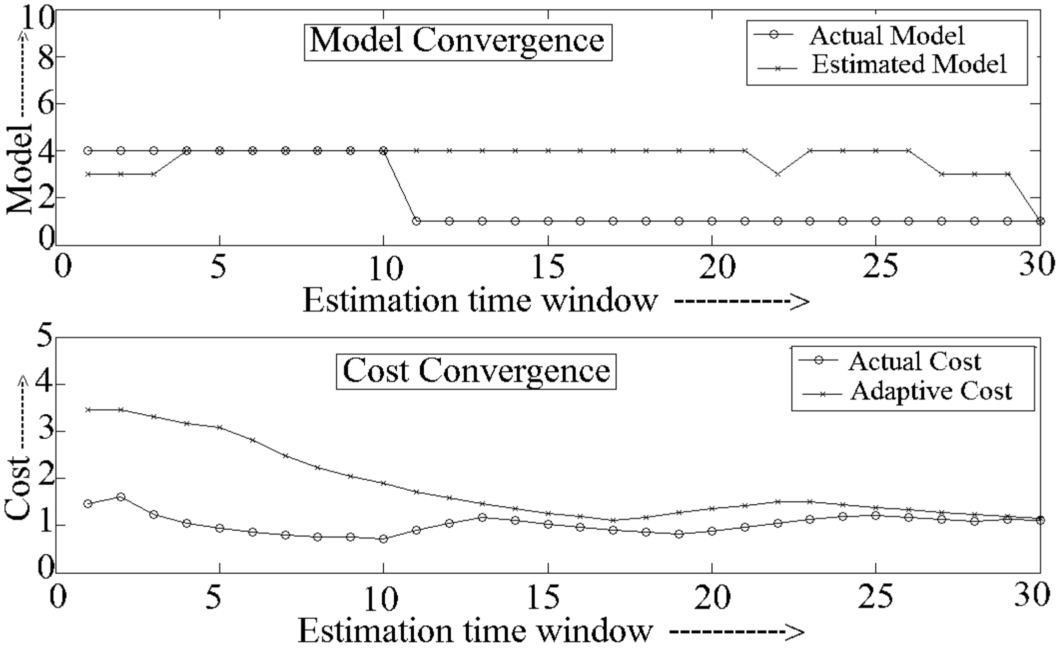


Fig. 10. Melanoma Application: Algorithm 1.

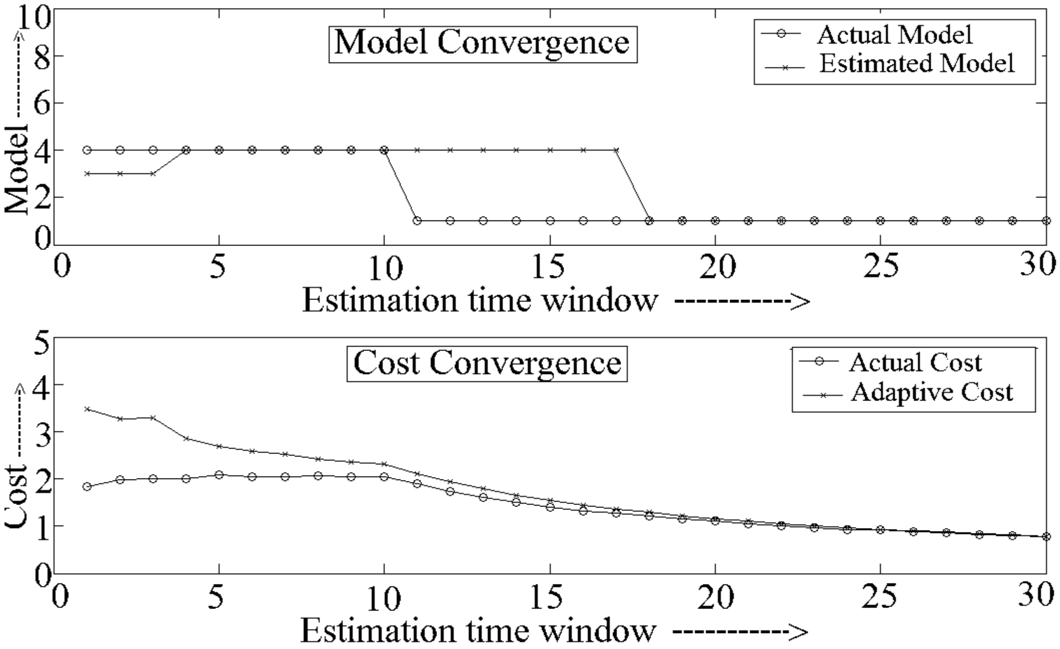


Fig. 11. Melanoma Application: Algorithm 2.

whereas when q is low or 0, algorithm 2 works better.

D. Concluding remarks

We have demonstrated the feasibility of applying adaptive intervention to improve intervention performance in genetic regulatory networks modeled by PBNs. Specifically, we have shown via simulations that when the network is modeled by a member of a known family of PBNs, one can use adaptation and carry out a certainty equivalence design that leads to improved performance in terms of the average cost. These simulation studies are important since the theoretical results in the literature guarantee only almost sure convergence and, that too, in the Cesaro sense. We have presented two different algorithms for model estimation, and argued that while one of the algorithms is well suited for instantaneously random PBNs, the other is much better for context-sensitive PBNs with low switching probability between the constituent BNs. Our simulation results confirm these intuitive expectations.

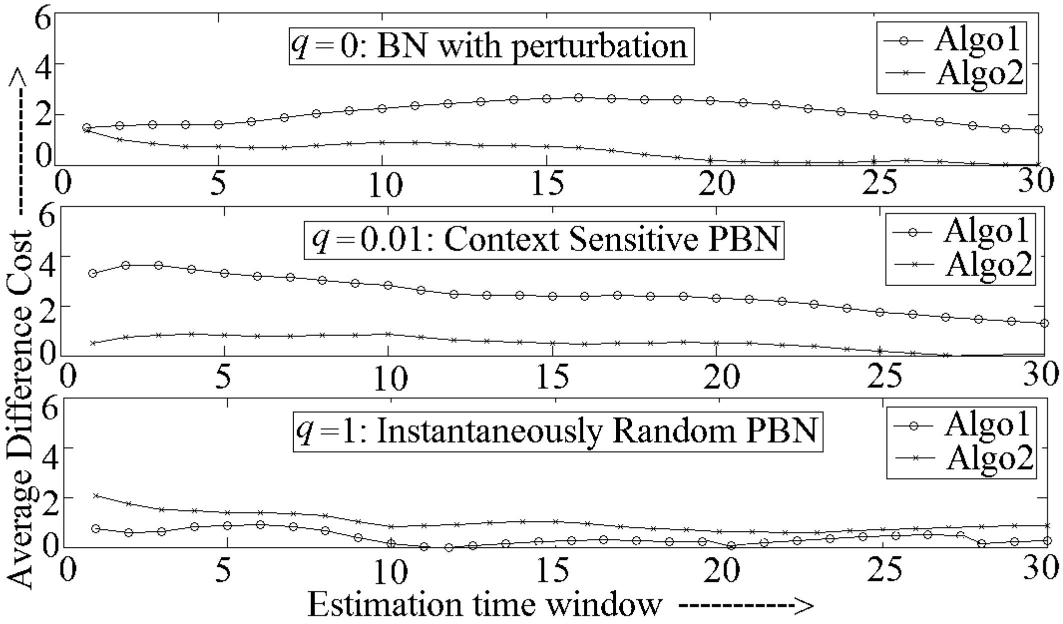


Fig. 12. Melanoma Application: Cost difference comparison of the two algorithms for different values of q.

Though mathematically the intervention strategy is close to optimal, there are some serious problems associated with this approach. Estimating the transition probability matrix (TPM) of a probabilistic Boolean network and practical feasibility of switching control are two of the major impediments. This motivates the introduction in CHAPTER III of a new approach for constructing networks consistent with prior biological knowledge. We will focus mostly on Boolean Networks because the parameters in a stochastic model are difficult to estimate, given the paucity of biological data. In CHAPTER IV, some real world examples are used to introduce practically feasible intervention strategies.

CHAPTER III

FROM PATHWAYS TO NETWORKS*

This chapter develops a general theoretical framework for arriving at genetic regulatory networks whose state transitions realize a set of given biological pathways or minor variations thereof. Often in biology, the a priori or expert knowledge is presented in the form of signaling pathways. Although such pathway information can be useful, it fails to capture the multivariate interactions between the genes. Interventions based on univariate gene interactions captured in pathways often fail to achieve the intended effects. In addition, it is quite common to have information on multiple pathways that may share some common nodes. In such a case, each pathway attempts to capture the intergene relationships when restricted to the genes in that pathway, but provides no information about the global interaction between the genes involved in the different pathways other than putting the constraint that the global interactions, when restricted to a particular pathway, must satisfy the relationships mandated by that pathway.

The problem of piecing together an overall underlying genetic regulatory network structure given (partial) pathway information is, therefore, very important in all areas of biology. However, to our knowledge, thus far the problem has not even been formulated properly, let alone be solved. Perhaps, one reason for this is the absence of a wide enough realization that pathway knowledge, no matter how appealing it may be, constitutes only partial knowledge restricted to a particular context. We next further motivate the work presented here by using a specific application area and its research needs.

In recent years, there has been considerable interest in the area of Genomic Signal Pro-

*Part of this chapter is reprinted with permission from “From biological pathways to regulatory networks ” by Ritwik K. Layek , Aniruddha Datta and Edward R. Dougherty, 2011, Mol. BioSyst., vol. 7, pp. 843-851, Copyright [2011], Royal Society of Chemistry. (http://pubs.rsc.org/en/content/articlelanding/2011/mb/c0mb00263a) cessing [36, 37] which seeks to mathematically model the multivariate interactions between the genes and utilize these models to not only differentiate between normal and abnormal (diseased) behavior but also to suggest appropriate therapeutic interventions in the case of the latter. The principal motivation for this is the growing realization that in the case of complex diseases such as cancer, therapeutic approaches based on simplistic marginal modeling, as in the case of biological pathways, can at best achieve modest success. To capture the holistic behavior of the genes, one can use genetic regulatory networks instead of working with only pathway knowledge. To date, genetic regulatory network modeling has been carried out using various approaches such as differential equations and their discrete-time counterparts [38, 39, 40, 41], Bayesian networks [42, 43, 44, 45], Boolean networks [46, 47, 18],[48, 49, 50, 51, 52, 45, 53, 54, 55], and their probabilistic generalizations, the probabilistic Boolean networks (PBNs) [19, 20, 21]. PBNs have also found extensive use in the design of intervention approaches that seek to slow down or halt disease progression [37, 56, 57, 58, 59, 60, 61, 32, 62, 63, 17, 64].

Most of the intervention approaches developed thus far for PBNs make use of the fact that the state transitions in a PBN can be modeled as a Markovian process. Estimating the transition probabilities for such a process, which is by no means a straight forward task, is an essential pre-requisite for the successful application of most of these intervention approaches. Although a handful of these schemes [61, 32] are able to bypass the need for estimating the transition probability matrix, none of them are capable of incorporating prior biological pathway information into the network design. This is a significant drawback since most of the prior biological knowledge in the literature resides in the form of biological pathways, gleaned as empirical observations across different experiments. Indeed, the accuracy of genetic regulatory networks and the data requirements for their inference could be greatly improved by developing a mechanism to incorporate pathway knowledge into the network itself. This chapter develops a systematic procedure for doing precisely that for the case of Boolean networks. Here, it is appropriate to point out that earlier work has focussed on generating Boolean networks satisfying principally attractor constraints [27, 65]. The results presented here are more general and essentially subsume the earlier ones.

This chapter is organized as follows. In section A, we introduce some notation and present the basics of digital design. In section B, we present a simple example to demonstrate how one can use pathway knowledge and Karnaugh maps to generate a family of BNs whose trajectories realize the given pathways. In section C, the general procedure for synthesizing Boolean network from a set of pathways is presented. In section D, the simple example of section B is revisited and solved using the algorithm developed in section C. In section E, we impose additional attractor constraints on the family of BNs to facilitate the choice of a particular BN. In section F, we apply the results of this chapter to the widely studied p53 pathway and demonstrate that the resulting network exhibits dynamic behavior consistent with experimental observations from the published literature. Finally, section G contains some concluding remarks.

A. Notation and digital design basics

1. Boolean networks

A *Boolean Network* (*BN*), $\Upsilon = (V, F)$, on n genes is defined by a set of nodes/genes $V = \{x_1, ..., x_n\}$, $x_i \in \{0, 1\}$, $i = 1, ..., n$, and a list $F = (f_1, ..., f_n)$, of Boolean functions, $f_i : \{0, 1\}^n \to \{0, 1\}$, $i = 1, ..., n$ [18]. The expression of each gene is quantized to two levels, and each node x_i represents the state/expression of the gene x_i, where $x_i = 0$ means that gene i is OFF and $x_i = 1$ means that gene i is ON. The function f_i is called the *predictor function* for gene i . Updating the states of all genes in Υ is done synchronously at every time step according to their predictor functions. At time t, the network state is given by $x(t) = (x_1(t), x_2(t), ..., x_n(t))$, called a *gene activity profile* (*GAP*).

2. Karnaugh map representation of Boolean networks

The subsequent development in this chapter relies heavily on the *Karnaugh Map (K-map)*[66] representation of a Boolean function. Consequently, let us now briefly introduce Karnaugh Maps and demonstrate their utility in digital design. Consider an arbitrary Boolean Network on three genes A, B and C with the following three Boolean update rules:

$$\begin{aligned} A_{next} &= B + C \\ B_{next} &= A\bar{C} \\ C_{next} &= A + \bar{B}. \end{aligned} \tag{3.1}$$

Here A_{next}, B_{next} and C_{next} denote the values of A, B and C at the next time step. Although the above rules represent the Boolean network in a compact form, they do not permit ready visualization of the state transitions or the attractors. Such ready visualization can be achieved by equivalently representing Equation (3.1) using the truth table shown in Table IV or the state transition diagram shown in Fig. 13.

Note, however, that the information contained in the truth table or the state transition diagram would not allow one to directly arrive at the Boolean update rules in Equation (3.1) which is what would be required if one were trying to realize the network using logic gates. This synthesis of Boolean functions from the truth table is facilitated by Karnaugh maps. In a Karnaugh Map, each current state is represented by a square and two neighboring squares have a Hamming distance of unity. This is crucial because this Hamming distance separation enables us to cluster large blocks of size 2^m in the maps. For each current state (represented by a square), the value of the particular gene in the next state is written inside the square. As an example, the three Karnaugh Maps for the Boolean Network corresponding to the three update rules in Equation (3.1) are shown in Fig 14. Since we have three genes with expressions which can only be binary, there is a total of eight states and hence eight squares in each Karnaugh map. For a moment, let us focus attention on the K-map for

Table IV. Truth Table of the Boolean Network (Eqn. 3.1).

A(n)	B(n)	C(n)	A(n+1)	B(n+1)	C(n+1)
0	0	0	0	0	1
0	0	1	1	0	1
0	1	0	1	0	0
0	1	1	1	0	0
1	0	0	0	1	1
1	0	1	1	0	1
1	1	0	1	1	1
1	1	1	1	0	1

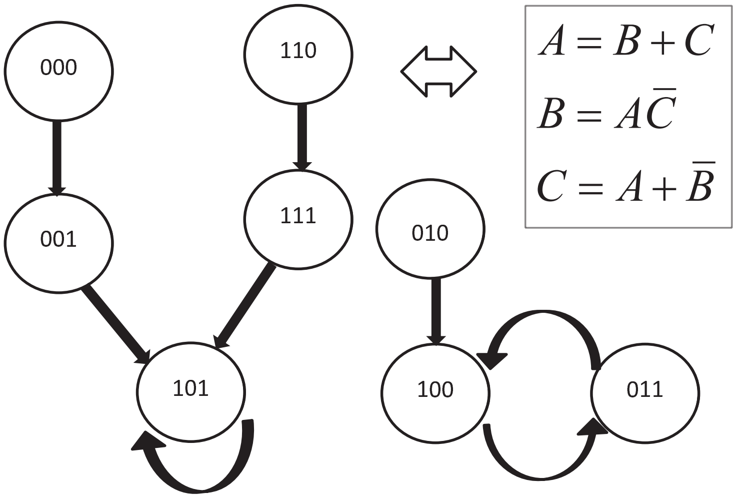


Fig. 13. State transition diagram of the Boolean Network (Eqn. 3.1).

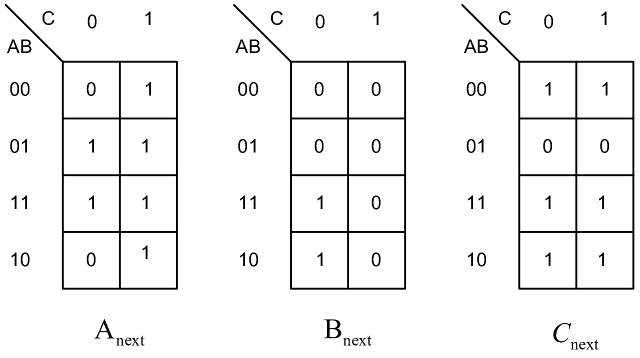


Fig. 14. Karnaugh map representation of Table IV.

gene A, denoted by A_{next} in Fig. 14. The possible gene value combinations for the current time step are shown to the left (genes A and B) and to the top (gene C) of the K-map. Also it should be noted that the bottom two rows of the K-map correspond to $A = 1$, the middle two rows correspond to $B = 1$ and the rightmost column corresponds to $C = 1$, all in the current state.

Define a *minterm* as a Boolean product (‘AND’ function) where each gene or its complement occurs exactly once. A three gene network, having exactly eight states, will have eight possible minterms, and each square in the Karnaugh map is the support of a unique minterm. For instance the square corresponding to the state 010 is the support of the unique minterm $\bar{A}B\bar{C}$ (the Boolean product $\bar{A}B\bar{C} = 1$ if and only if the state is 010).

We next use this three gene example to show how the Karnaugh map representation can help us in arriving at the Boolean functions for the update rules. Let us focus on the K-map for gene B, i.e. B_{next} in Fig. 14. In this K-map, two of the minterms giving $B_{next} = 1$ are $AB\bar{C}$ and $A\bar{B}\bar{C}$. By summing (‘OR’ing) the minterms having functional value 1 (the value inside the squares), we can generate the network functions. For example, $B_{next} = AB\bar{C} + A\bar{B}\bar{C} = A\bar{C}$. In the K-map, this can be done geometrically. As, the two neighboring states (squares) have a hamming distance of 1, we can remove the variable that differs between the corresponding minterms to more compactly represent the set of two squares. In the K-map of B_{next}, the states corresponding to $B_{next} = 1$ are 110 and 100 and their hamming distance is 1. So, the product term representing the two states is simply $A\bar{C}$ (We remove the variable B as both B and its complement $\bar{B}$ appear in the two minterms $AB\bar{C}$ and $A\bar{B}\bar{C}$ and, therefore, B is a non-essential variable). The idea that we have just illustrated for clustering two minterms in the K-map can be extended to cluster additional minterms and obtain a minimal realization of the Boolean function in question. Indeed, this procedure is used extensively in computer architecture and digital design [67].

In this chapter we will follow the clustering of minterms approach which will give us the minimal SoP or Sum of Products (‘OR’ of ‘AND’s) form of the Boolean functions [67]. The prior knowledge presented in the form of signalling pathways will furnish us with partially filled Karnaugh Maps for updating each of the genes. Clearly, such a partially filled Karnaugh Map will not yield a unique Boolean function, even in the Sum-of-Products form, so that instead of arriving at a unique Boolean network, we may end up with a family of Boolean networks. On the other hand, different pathways may introduce conflicts in the Karnaugh Map describing the update of a particular gene, in which case it would be impossible to arrive at a Boolean network to simultaneously satisfy all the pathway constraints. Fortunately, in such a case, the pathway constraints can be relaxed since (i) pathways only represent empirical observations across different experiments; and (ii) there is no accurate timing information to go with the pathways, which means that the initially assumed timing information in the pathways can be slightly altered to facilitate a solution. In this chapter, we will formally develop these ideas and present a systematic solution to the problem of generating a family of Boolean networks whose trajectories satisfy given pathways or minor variations thereof. For clarity of presentation, we first begin with a simple example which can be handled in an intuitive way without having to invoke the complete machinery to be developed for the general case.

B. From pathways to a family of BNs: a simple example

For clarity of presentation, consider a Boolean Network (BN) on 4 genes A,B,C & D so that each state (or GAP) is given by a binary vector of the form $V = abcd$, where a, b, c, d are either 0 or 1. Define the term *pathway segment* $A \xrightarrow{t:a,b} B$ to mean that if gene A assumes the value a then gene B transitions to b in no more than t subsequent time steps. A *pathway* is defined to be a sequence of pathway segments of the form $A \xrightarrow{t_1:a,b} B \xrightarrow{t_2:b,c} C$. In the above pathway, there are two pathway segments $A \xrightarrow{t_1:a,b} B$ and $B \xrightarrow{t_2:b,c} C$. We define a *trajectory* to be a sequence of states $V_0 \rightarrow V_1 \rightarrow V_2 \rightarrow V_3 \rightarrow V_4$ resulting from the network rules beginning at some initial state . Clearly, a trajectory provides a more complete picture of the dynamic evolution of the BN resulting from the multivariate interactions between the genes. Pathway information, on the other hand, is neither regulatory nor state space knowledge; it is marginal and incomplete.

Given the wide prevelance of apriori biological knowledge in the form of pathway information, an important problem to consider is how to generate a BN whose trajectories are consistent with a given set of pathways. This is an ill-posed inverse problem that could have multiple solutions or perhaps none. Therefore, our objective will be to investigate and devise an algorithm to generate the set of all possible Boolean networks and to find out the minor required timing or functional perturbation of the pathways if no Boolean network can be found to satisfy the set of pathway constraints. We will do a structural analysis for the Boolean Network synthesis problem. This is a brute force exhaustive procedure but can be used to generate the complete set of admissible BNs. Later the BN set can be shrunk by imposing various realistic constraints such as (i) an upper bound on the number of predictors per gene; (ii) an upper bound on the number of attractors; (iii) steady state distribution of the attractors from actual experiments (e.g, Microarray Experiments); (iv) concordance with experimentally measured time series dynamics (e.g, those obtained using Green Fluorescent Protein based techniques), and so on. We next use a simple four-gene network to illustrate the key ideas behind the exhaustive search procedure.

We have chosen an example with four genes since four is the largest number for which the Karnaugh map can be visualized in two dimensions. For larger networks, the underlying philosophy is the same although one would have to resort to computer programming. Now let us assume that we are given three pathways: $A \xrightarrow{1:1,1} B \xrightarrow{1:1,1} D$, $A \xrightarrow{1:1,0} C$ and $C \xrightarrow{1:1,0} D$. First, we solve the inverse problem for pathway 1; thereafter, we add pathways 2 and 3, respectively, and shrink the solution space.

State space Realization of pathways

Pathway 1 ($A \xrightarrow{1:1,1} B \xrightarrow{1:1,1} D$): There are two segments to this pathway. The first segment $A \xrightarrow{1:1,1} B$ mandates that if the current state has $A = 1$, then it will transition to a state with $B = 1$ in one time step. So the state transition consistent with the pathway information is 1xxx→x1xx[1]. Similarly, $B \xrightarrow{1:1,1} D$ translates to the state transition x1xx→xxx1. These are the only state transition constraints mandated by pathway 1. These state transition constraints can be represented in the Karnaugh Map for the individual genes as follows.

A^{pw1}_{next}:

		D	C D	C
	x	x	x	x
B	x	x	x	x
A B	x	x	x	x
A	x	x	x	x

B^{pw1}_{next}:

		D	C D	C
	x	x	x	x
B	x	x	x	x
A B	1	1	1	1
A	1	1	1	1

C^{pw1}_{next}:

		D	C D	C
	x	x	x	x
B	x	x	x	x
A B	x	x	x	x
A	x	x	x	x

D^{pw1}_{next}:

		D	C D	C
	x	x	x	x
B	1	1	1	1
A B	1	1	1	1
A	x	x	x	x

Here, the 4 Karnaugh Maps correspond to the truth tables for genes A, B, C and D in the next time step as a function of the current state. As before, the bottom two rows correspond to $A = 1$, the middle two rows correspond to $B = 1$, the right two columns correspond to

[1] Here 'x' denotes a gene value that could be either 0 or 1.

$C = 1$ and the middle two columns correspond to $D = 1$, all in the current state. Also, the superscript $pw1$ indicates that these K-maps correspond to pathway 1.

Clearly we see that the value of gene A at the next time step does not depend on what the current state is. In the case of gene B, if the current state is 1xxx (meaning $A = 1$), then the next state will be x1xx (meaning $B = 1$). This is shown in the K-map for gene B where the bottom two rows are filled with $1s$ and the remaining 8 entries are either 0 or 1. Similarly the value of gene C at the next time step does not depend on the current state while the value of gene D at the next time step depends only on the current value of B. The above K-maps characterize the entire family of BNs satisfying the constraints mandated by pathway 1.

Pathway 2 ($A \xrightarrow{1:1,0} C$): If we solve separately for pathway 2, we get another set of K-maps for each of the genes A, B, C and D. The K-maps are shown below.

A^{pw2}_{next}:

		D	C D	C
	x	x	x	x
B	x	x	x	x
A B	x	x	x	x
A	x	x	x	x

B^{pw2}_{next}:

		D	C D	C
	x	x	x	x
B	x	x	x	x
A B	x	x	x	x
A	x	x	x	x

C^{pw2}_{next}:

		D	C D	C
	x	x	x	x
B	x	x	x	x
A B	0	0	0	0
A	0	0	0	0

D^{pw2}_{next}:

		D	C D	C
	x	x	x	x
B	x	x	x	x
A B	x	x	x	x
A	x	x	x	x

Next we would like to merge the two sets of K-maps to obtain K-maps consistent with both the pathways 1 and 2. The solution set is shown below.

A_{next}:

		D	C D	C
	x	x	x	x
B	x	x	x	x
A B	x	x	x	x
A	x	x	x	x

B_{next}:

		D	C D	C
	x	x	x	x
B	x	x	x	x
A B	1	1	1	1
A	1	1	1	1

C_{next}:

		D	C D	C
	x	x	x	x
B	x	x	x	x
A B	0	0	0	0
A	0	0	0	0

D_{next}:

		D	C D	C
	x	x	x	x
B	1	1	1	1
A B	1	1	1	1
A	x	x	x	x

Pathway 3 ($C \xrightarrow{1:1,0} D$): If we solve separately for pathway 3, we get another set of K-maps for each of the genes A, B, C and D. The K-maps are shown below.

A^{pw3}_{next}:

x	x	x	x
x	x	x	x
x	x	x	x
x	x	x	x

B^{pw3}_{next}:

x	x	x	x
x	x	x	x
x	x	x	x
x	x	x	x

C^{pw3}_{next}:

x	x	x	x
x	x	x	x
x	x	x	x
x	x	x	x

D^{pw3}_{next}:

x	x	0	0
x	x	0	0
x	x	0	0
x	x	0	0

Clearly, for genes A, B and C, there is no conflict between the two sets of K-maps and we can easily merge them to get the K-maps shown below.

A_{next}:

x	x	x	x
x	x	x	x
x	x	x	x
x	x	x	x

B_{next}:

x	x	x	x
x	x	x	x
1	1	1	1
1	1	1	1

C_{next}:

x	x	x	x
x	x	x	x
0	0	0	0
0	0	0	0

On the other hand, the two K-maps for gene D are in conflict as evident from the K-maps given below.

D^{pw1}_{next}:

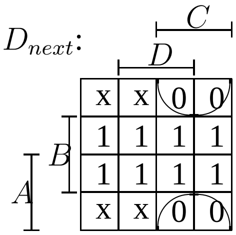
x	x	x	x
1	1	1	1
1	1	1	1
x	x	x	x

D^{pw3}_{next}:

x	x	0	0
x	x	0	0
x	x	0	0
x	x	0	0

The conflict occurs when the current state is x11x (see the entries inside the two circles). This conflict is not at all surprising: if the current state is x11x then $B = 1$ will try to force $D = 1$ at the next time step as per pathway 1 while $C = 1$ will try to simultaneously force $D = 0$ in accordance with pathway 3. One way to resolve the conflict would be to decide in favor of one of the two requirements. Let us assume (without loss of generality) that A

has higher priority than B and B has higher priority than C, and so on. In that case, gene B will affect the state transition earlier than gene C. Accordingly, we decide that in the above conflict, x11x will transition to xxx1 in the next time step so that the K-map of D gets modified as according to the K-map below.



x	x	0	0
1	1	1	1
1	1	1	1
x	x	0	0

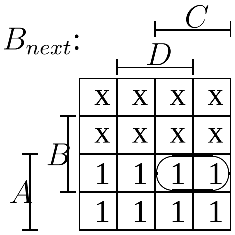
However, we still need to satisfy pathway 3. Although it is not possible to meet both pathway constraints in the same time step, we can relax the timing of the third pathway as $C \xrightarrow{2:1,0} D$, which means that $C = 1$ will lead to $D = 0$ in no more than 2 time steps. Accordingly, x11x will transition to xxx0 in no more than two time steps. Thus, the complete transition will be x11x$\rightarrow$ xxx1 $\rightarrow$ xxx0. However, we know from the merged K-map of D (after conflict elimination), that only x01x leads to xxx0 in one time step (see the two semicircles in the above K-map where the value is 0). Hence, the second state in the above state transition becomes x011, leading to the actual state transitions x11x$\rightarrow$ x011 $\rightarrow$ xxx0. This set of two state transitions yields two new pathways: pathway 4 : $BC \xrightarrow{1:1,0} B$ and pathway 5: $BC \xrightarrow{1:1,1} C$. The introduction of these two new pathways will lead to the iterative update of the K-maps until the K-maps converge to a stable set of BNs.

1. Iterative update of K-maps

Now, pathways 4 and 5 mandate that whenever the current state becomes x11x, the next state will be x011 which means that gene $B = 1$ and gene $C = 1$ will lead to gene $B = 0$ and gene $C = 1$ in the next time step. This again conflicts with the earlier K-maps of genes B and C.

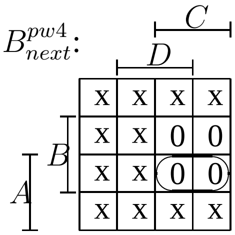
Conflict in K-map of gene B: If the current state is 111x, then a conflict arises in the K-map of gene B. Specifically, $A = 1$ in the current state mandates $B = 1$ in the next state, whereas $BC = 1$ in the current state mandates $B = 0$ in the next state.The conflicting Karnaugh Maps for these two cases are shown below along with the marked conflict zone.

B_{next}:



x	x	x	x
x	x	x	x
1	1	1	1
1	1	1	1

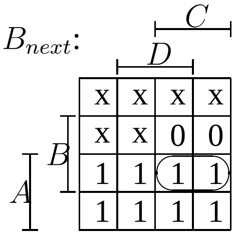
B^{pw4}_{next}:



x	x	x	x
x	x	0	0
x	x	0	0
x	x	x	x

As before, we once again apply the conflict resolution rule. We set $B = 1$ in the next state when $BC = 1$ in the current state to obtain the modified K-map for gene B shown below.

B_{next}:

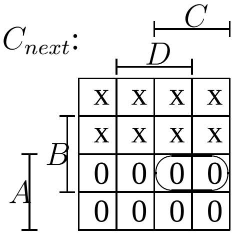


x	x	x	x
x	x	0	0
1	1	1	1
1	1	1	1

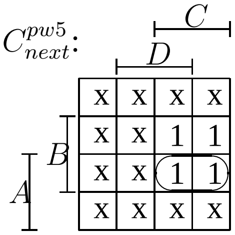
This mandates the state transition: 111x→x1xx. As before, we have to relax the timing constraint of pathway 4: $BC \xrightarrow{1:1,0} B$ is changed to $BC \xrightarrow{2:1,0} B$. Consequently, the state x1xx has to be followed by the state x0xx. However, we know that a necessary condition for $B = 0$ in the next state is that $A = 0$ in the current state. So, we get another new pathway, pathway 6: $ABC \xrightarrow{1:1,0} A$. It is clear that although the original three pathways did not yield any update rules for gene A, the conflict resolution rules that we have applied have given rise to a new reverse pathway which imposes an update rule on gene A.

Conflict in K-map of gene C: If the current state is 111x, then a conflict arises in the

K-map of gene C as well. Specifically, $A = 1$ in the current state mandates $C = 0$ in the next state, whereas $BC = 1$ in the current state mandates $C = 1$ in the next state. The conflicting K-maps for these two cases are shown below along with the marked zone of conflict.

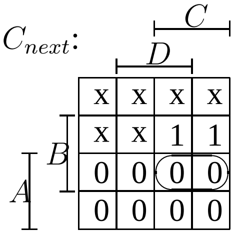


x	x	x	x
x	x	x	x
0	0	0	0
0	0	0	0



x	x	x	x
x	x	1	1
x	x	1	1
x	x	x	x

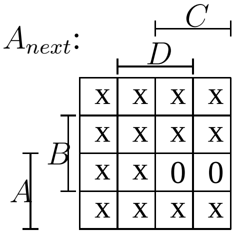
As before, we once again apply the same conflict resolution rule. We set $C = 0$ in the next state when $A = 1$ in the current state to obtain the modified K-map for gene C shown below.



x	x	x	x
x	x	1	1
0	0	0	0
0	0	0	0

This mandates the state transition: 111x→xx0x. As before, we have to relax the timing constraint of pathway 5: $BC \xrightarrow{1:1,1} C$ is changed to: $BC \xrightarrow{2:1,1} C$. Consequently, the state xx0x has to be followed by the state xx1x. However, we know that a necessary condition for $C = 1$ in the next state is that $A = 0$ in the current state. This leads to the same pathway 6 :$ABC \xrightarrow{1:1,0} A$, as before. Fortunately, in this simple example, the conflict resolutions in the K-maps of B and C both lead to the same pathway 6. This means that whenever $ABC = 1$, the next state will be 0xxx. Accordingly, the K-map of gene A is modified as shown below.

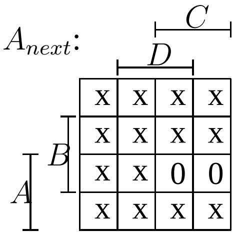
A_{next}:



x	x	x	x
x	x	x	x
x	x	0	0
x	x	x	x

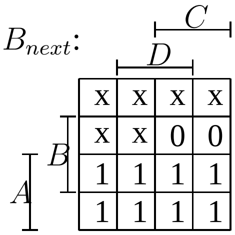
Finally we have reached a stage where there are no more conflicts in the K-maps and the final K-maps are shown below.

A_{next}:



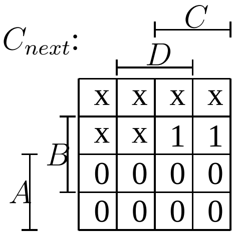
x	x	x	x
x	x	x	x
x	x	0	0
x	x	x	x

B_{next}:



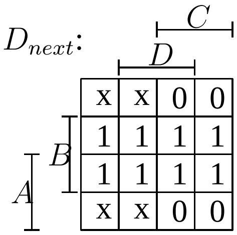
x	x	x	x
x	x	0	0
1	1	1	1
1	1	1	1

C_{next}:



x	x	x	x
x	x	1	1
0	0	0	0
0	0	0	0

D_{next}:



x	x	0	0
1	1	1	1
1	1	1	1
x	x	0	0

Thus, with minor modifications of the third original pathway, we have solved the inverse problem of finding the class of Boolean Networks. The procedure can be extended to find the complete set of BNs consistent with any number of given pathways. In this problem, we can see the shrinkage in the number of possible Boolean networks. To start with, the search space had a cardinality of 2^{64}. After incorporating the pathway knowledge, the cardinality of the search space shrinks to 2^{30}.

As we will see later, the cardinality of the search space can be further reduced by imposing constraints on the number and relative significance of the attractors, the connectivity of the network, etc.

C. From pathways to a family of BNs: the general procedure

1. Definitions and preliminary observations

In section B we defined a pathway and pathway segments only in terms of single genes. The solution to the simple example presented there was also based on an intuitive procedure. Our objective here is to develop a systematic general procedure which can yield a Boolean network consistent with an arbitrary number of pathways. Towards this end, we next introduce some definitions and make some preliminary observations.

For any Boolean function f, define the support of f, denoted by $supp(f)$, to be the set of all argument values that make f assume the value of 1. Also, for any Boolean function S, and for a Boolean value $s \in \{0,1\}$, define

$$S_s = \begin{cases} S & \text{if } s = 1 \\ \bar{S} & \text{if } s = 0. \end{cases} \tag{3.2}$$

The function S_s defined in Eqn. 3.2 can be thought of as being the indicator function of the set $\{x : S(x) = s\}$.

Let us now generalize the pathway segment definitions presented earlier:

1. A *Simple Pathway Segment* is defined as $Y \overset{t_1:a,b}{\longrightarrow} B$, where B is a single gene, Y can be an arbitrary Boolean function and $a, b \in \{0,1\}$. Unless otherwise indicated, the term *pathway segment* in this chapter will refer to a simple pathway segment. A simple pathway segment can be implemented using only the K-map of the target gene, i.e. gene B in this case.

2. A *composite pathway segment* is defined as $Y \overset{t_2:y,z}{\longrightarrow} Z$ where both Y and Z are arbitrary Boolean functions and $y, z \in \{0,1\}$. We next develop the theory for decomposing a given composite pathway segment into a number of simple pathway segments since only the latter are directly implementable using K-maps.

Without any loss of generality, let us assume $t_2 = 1$ in the above composite pathway definition. Using Equation (3.2), the composite pathway segment $Y \xrightarrow{1:y,z} Z$ can be alternatively written as $Y \xrightarrow{1:y,1} Z_z$.

Furthermore, Z_z can be expressed in the minimal SOP (Sum of Products) form [67]:

$$Z_z = P_1 + P_2 + \cdots + P_k \tag{3.3}$$

where each P_i, $i = 1, 2, \cdots, k$ is a Boolean product term of the form

$$P_i = A_1^i A_2^i \cdots A_{l_i}^i \tag{3.4}$$

and each A_j^i is either a gene or its complement.

From the above analysis it is evident that $Y \xrightarrow{1:y,1} P_i \Longrightarrow Y \xrightarrow{1:y,1} Z_z$ (because $P_i = 1 \Longrightarrow Z_z = 1$). So, any P_i can replace Z_z producing the desired pathway effect. For the sake of simplicity we will choose the product term P_m having the least number of genes. The resulting composite pathway segment $Y \xrightarrow{1:y,1} P_m$ can be decomposed into l_m simple pathway segments that have to be simultaneously satisfied:

$$\begin{aligned} Y &\xrightarrow{1:y,1} A_1^m \\ Y &\xrightarrow{1:y,1} A_2^m \\ Y &\xrightarrow{1:y,1} A_3^m \\ &\vdots \\ Y &\xrightarrow{1:y,1} A_{l_m}^m. \end{aligned} \tag{3.5}$$

Thus the K-map implementation of these l_m simultaneous simple pathway segments will provide a non-unique realization of the original composite pathway segment: $Y \xrightarrow{1:y,z} Z$.

3. A *pseudo pathway* is defined to be any pathway that can be inferred from a given

Table V. Priority Ordering

A	$\bar{\mathrm{A}}$	B	$\bar{\mathrm{B}}$	C	$\bar{\mathrm{C}}$	D	$\bar{\mathrm{D}}$

Boolean network. The update rules for a Boolean network mandate that the state (or GAP) transitions occur in a particular sequence. By marginally focusing on the transitioning of particular components of the GAP, one can come up with inherent pathway relationships, which we refer to as pseudo pathways.

2. Priority ordering between Boolean functions

In section B, we loosely introduced the notion of priority among genes for deciding which gene would preferentially act on a target. Since different gene combinations, and not necessarily individual genes, could be acting on a target, it is necessary to generalize the notion to Boolean functions of genes. Such a generalization is carried out in this subsection by the introduction of what we refer to as a *priority index*.

From biological understanding, we know that all genes do not influence a particular target gene to the same extent. As an example, suppose genes A and B both influence the status of target gene C in some way but with different relative abilities. Define *priority* as the power of each gene to influence others in the pathway. Priority is a qualitative term and cannot be used for conflict resolution unless we quantify it in some sense. Accordingly, we next introduce a *priority index* which will be employed as the decision making parameter in times of conflict resolution. Suppose that from our qualitative knowledge of genes and pathways we can make a list of all the genes according to their powers. This is called the *priority list*. As an example, suppose the priority list for the four genes A, B, C, D and their complements $\bar{A}, \bar{B}, \bar{C}, \bar{D}$ are as shown in Table V:

Here A has the highest priority followed by $\bar{A}$ and so on. Symbolically we write

$A > \bar{A} > B > \bar{B} > C > \bar{C} > D > \bar{D}$. We assume that the priority ordering is transitive which means that if $A > B$ and $B > C$, then $A > C$.

We next extend the notion of priority ordering between genes to that between product terms and ultimately to that between two arbitrary Boolean functions. To do so, define the *priority index* between two genes A and B by:

$$\rho(A, B) = \begin{cases} 1 & \text{if } A > B \\ 0 & \text{if } A < B. \end{cases}$$

Next we define the priority index between a product term $Y = Y_1Y_2Y_3Y_4..Y_l$ and a gene B as:

$$\rho(Y, B) = 1/l \sum_{i=1}^{l} \rho(Y_i, B). \tag{3.6}$$

Finally we define the priority index between two product terms $Y = Y_1Y_2Y_3Y_4..Y_l$ and $Z = Z_1Z_2Z_3Z_4..Z_k$ as:

$$\begin{aligned} \rho(Y, Z) &= \frac{1}{k} \sum_{j=1}^{k} \rho(Y, Z_j) \\ &= \frac{1}{kl} \sum_{i=1}^{l} \sum_{j=1}^{k} \rho(Y_i, Z_j). \end{aligned} \tag{3.7}$$

We next extend the priority index definition to the case of arbitrary Boolean functions S_1 and S_2. To do so, we make use of the well known fact from Boolean algebra that any Boolean function can be represented in a minimal Sum of Products (SoP) form. Suppose the two functions S_1 and S_2 are expressed in such a form as:

$$\begin{aligned} S_1 &= P_1 + P_2 + P_3 + \cdots + P_n \\ S_2 &= Q_1 + Q_2 + Q_3 + \cdots + Q_n. \end{aligned} \tag{3.8}$$

Furthermore, suppose that P_i is the minimal product term in S_1, i.e. the term having the minimum number of genes or gene's complements. Consequently, P_i corresponds to the maximum number of minterms (unit squares) in the K-map of S_1. Similarly, assume that Q_j is the minimal product term in S_2. The priority index between S_1 and S_2 is defined by

$$\rho(S_1, S_2) = \rho(P_i, Q_j). \tag{3.9}$$

The priority indices defined above satisfy the following properties:

$$\begin{aligned} 0 \leq \rho(S_1, S_2) \leq 1 \\ \rho(S_1, S_2) = 1 - \rho(S_2, S_1). \end{aligned} \tag{3.10}$$

Based on the priority indices just defined, the priority ordering between two Boolean functions S_1 and S_2 can be made as follows:

$$S_1 \begin{cases} > S_2 & \text{if } \rho(S_1, S_2) > 0.5 \\ < S_2 & \text{otherwise.} \end{cases}$$

Before concluding this section on priority ordering, we mention that, as a general rule, if we come across a composite pathway segment, then the highest priority will be given to accommodate that. This is because a composite pathway segment gives rise to several simple pathway segments that have to be simultaneously satisfied and, therefore, it is reasonable to give it the highest priority.

3. Conflict and its resolution

In this subsection we generalize the conflict resolution procedure, introduced in section B, to the case where we have an arbitrary number of genes and an arbitrary number of pathways. Define a *conflict* as a situation when for a new pathway $Y \xrightarrow{1:y,b} B$, where Y is a Boolean function and B is a gene, there already exists a Boolean function Ψ with $supp(\Psi) \subset supp(Y_y)$, $supp(\Psi) \neq supp(Y_y)$ such that $\Psi \xrightarrow{1:1,\bar{b}} B$. We next explain how such a conflict can arise.

We have seen in section B that if the truth table mandated by a pathway segment does not contradict the existing truth tables for the network (i.e, $supp(\Psi) = \emptyset$), then there is no problem in incorporating such a pathway. However, if the demands of a new pathway segment contradict the already existing truth table values (i.e, $supp(\Psi) \neq \emptyset$), then a conflict arises. An adhoc procedure for resolving such a conflict for the simple example was demonstrated in section B. Here we develop a systematic procedure for handling the general case.

Suppose at a particular stage while determining an n-gene Boolean network satisfying pathway information, we come to a new pathway segment $Y \xrightarrow{1:y,b} B$ where $y, b \in \{0, 1\}$. This pathway segment can be implemented in the K-map for gene B. Specifically, we would try to insert the value b in every minterm $\in supp(Y_y)$ in the K-map of gene B. While doing so we may discover a set of minterms in $supp(Y_y)$ whose values are already $\bar{b}$ in the K-map of gene B. We can combine the entire set of such minterms and sum them up to obtain a Boolean function Ψ such that $supp(\Psi) \subset supp(Y_y)$. The situation is graphically illustrated in Fig 15.

Clearly, in the above K-map of gene B, the minterms where $Y_y\bar{\Psi} = 1$ can be unambiguously assigned the value of b. The conflict will arise in the subset $supp(\Psi)$. Resolution of this conflict will require us to determine the priority ordering between Y_y and Ψ which can be carried out by evaluating the priority norm between these two Boolean functions. Depending on the priority ordering, we will adopt one of the two following options:

1. $Y_y > \Psi$: In this case, $Y \xrightarrow{1:y,b} B$ is given higher priority and all the minterms $\in supp(Y_y)$ will be assigned the value b. However, to satisfy the inherent pathway segment $\Psi \xrightarrow{2:1,\bar{b}} B$, in the next time step, we generate another constraint. Clearly, to get to the value of $\bar{b}$ in the truth table of gene B, the trajectory would have to traverse to the states which can lead to $B = \bar{b}$ in one time step. So, the additional pathway we obtain is a composite pathway segment: $\Psi \xrightarrow{1:1,1} S(\bar{b})$, where $S(\bar{b})$ is a minimal

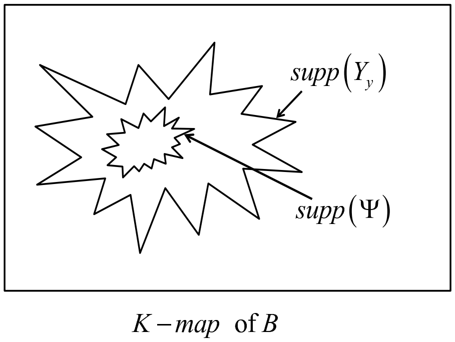


Fig. 15. Conflict in K-map of gene B.

SoP set in the K-map of B having value $\bar{b}$. That means this set inherits an already implemented pseudo pathway $S(\bar{b}) \xrightarrow{1:1,\bar{b}} B$ which gets rid of the conflict. If, however, no such $S(\bar{b})$ can be found in the K-map of B, then the problem is unsolvable and the algorithm is terminated.

2. $\Psi > Y_y$: In this case, $\Psi \xrightarrow{1:1,\bar{b}} B$ is implemented first. The minterms $\in supp(\Psi)$ will be assigned the value $\bar{b}$. However, to satisfy the other pathway segment in the next time step, i.e. $Y \xrightarrow{2:y,b} B$, we generate the other constraint. Clearly, to get to the value b in the truth table of gene B, the trajectory has to traverse to the states in $supp(Y_y) \cap supp(\bar{\Psi})$. So, the additional pathway that we obtain out of this reasoning is: $\Psi \xrightarrow{1:1,1} Y_y\bar{\Psi}$. This being a composite pathway segment, one would have to decompose it into simple pathway segments before proceeding further.

4. Total conflict and cyclic total conflict

In this subsection we wish to demonstrate conflict resolution in the extreme case when $supp(\Psi) = supp(Y_y)$. This situation is called a *total conflict* and we denote it by the notation

$Y \overset{1:y,b}{\twoheadrightarrow} B$. Since in the case of a total conflict $supp(Y_y) \cap supp(\bar{\Psi}) = \emptyset$, only the first method presented in the previous subsection can be used. This is demonstrated in the following example.

Example: Consider a 4 gene network with genes A, B, C, D so that each state (or GAP) is given by a binary vector of the form $V = abcd$ where $a, b, c, d \in \{0, 1\}$. Suppose that gene B currently updates according to the Karnaugh map shown below.

B_{next}:

$$\begin{array}{c|cccc} & & D & D\,C & C \\ \hline & 1 & 1 & 0 & 0 \\ B & 1 & 1 & x & x \\ A\,B & x & x & 0 & 0 \\ A & 1 & 1 & 0 & 0 \end{array}$$

Let us now introduce the new pathway segment $AC \xrightarrow{1:1,1} B$. In the notation of section 3, $Y = AC$, $b = 1$ and $Y_1 = AC$. Clearly the pseudo pathway we get from the Karnaugh map corresponding to the minterms having $AC = 1$ is $AC \xrightarrow{1:1,0} B$ so that $\Psi = AC$. This is a good example of a total conflict which cannot be resolved in a direct fashion. However, examining the truth table of gene B, we get some useful information that could facilitate a solution.

Without any loss of generality let us assume that whenever a total conflict arises, the new pathway segment always gets the highest priority. Using this priority ordering, the truth table of gene B is modified according to the K-map below.

B_{next}:

			D	C D	C
		1	1	0	0
	B	1	1	x	x
A	B	x	x	1	1
A		1	1	1	1

The minterms in this new truth table having value 0 are $\bar{A}\bar{B}CD$ and $\bar{A}\bar{B}C\bar{D}$ which can be 'OR'ed to get the product term $\bar{A}\bar{B}C$. So, the new composite pathway segment which will resolve the total conflict is given by $AC \xrightarrow{1:1,1} \bar{A}\bar{B}C$. This is a composite pathway segment and will lead to three simple pathway segments: $AC \xrightarrow{1:1,1} \bar{A}$, $AC \xrightarrow{1:1,1} \bar{B}$ and $AC \xrightarrow{1:1,1} C$ which are to be implemented simultaneously.

Now it may happen that while trying to remove a total conflict $A \overset{1:a,b}{\twoheadrightarrow} B$, we may arrive at some other total conflicts like $C \overset{1:c,d}{\twoheadrightarrow} D$, $E \overset{1:e,f}{\twoheadrightarrow} F$, .. so on. These total conflicts and their derivative total conflicts can be represented in a directed graph structure. The total conflicts arising from the ordinary conflicts are the starting nodes of the graph. Then all the derivative total conflicts will become the children of the corresponding parent nodes. We can attempt to solve the total conflict problem in an iterative way by growing the graph. This iteration will continue until either the graph is *acyclic* and converges to a solution when there is no child node left, or the graph becomes cyclic, when we say that there is a *cyclic total conflict* in the problem and the problem cannot be solved. These two situations are depicted in Fig 16.

D. Simple example revisited

The adhoc solution to the small 4 gene BN synthesis example presented in section B was mainly based on our intuition. We now apply the systematic procedure of the previous section to the same example and highlight the differences. First, we point out that the systematic procedure begins to deviate from the earlier adhoc one only from section C.

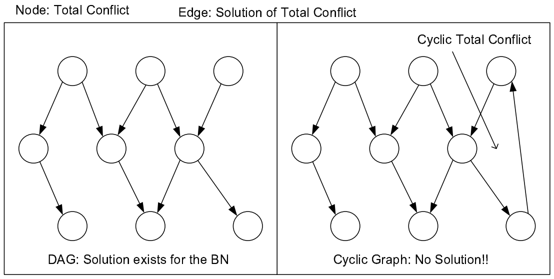

Node: Total Conflict
Edge: Solution of Total Conflict
Cyclic Total Conflict
DAG: Solution exists for the BN
Cyclic Graph: No Solution!!

Fig. 16. Solvability of the conflicting pathway problem.

Hence it is appropriate to start our analysis from there.

The adhoc treatment prior to section C tells us that the two new pathways 4 and 5 have to be satisfied simultaneously as otherwise the state transitions may not be the same as desired. To ensure that, we assign the highest priority to the incorporation of the two new pathways.

Iterative update of the truth tables: Recall that the K-maps for this example prior to section B.1 are given by the K-maps below.

A_{next}:

x	x	x	x
x	x	x	x
x	x	x	x
x	x	x	x

B_{next}:

x	x	x	x
x	x	x	x
1	1	1	1
1	1	1	1

C_{next}:

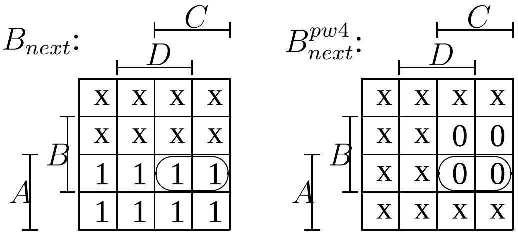
x	x	x	x
x	x	x	x
0	0	0	0
0	0	0	0

D_{next}:

x	x	0	0
1	1	1	1
1	1	1	1
x	x	0	0

Also, the two pathways yet to be incorporated are $pathway4 : BC \xrightarrow{1:1,0} B$ and $pathway5 : BC \xrightarrow{1:1,1} C$. These two pathways are to be solved simultaneously and with the highest priority.

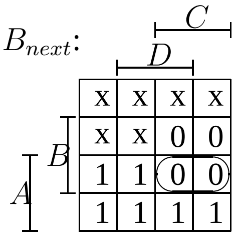
Conflict in the truth table of gene B: While trying to incorporate the simple pathway segment 4, we get the conflict shown below for gene B:



x	x	x	x
x	x	x	x
1	1	1	1
1	1	1	1

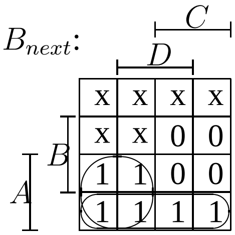
x	x	x	x
x	x	0	0
x	x	0	0
x	x	x	x

Using the notation introduced in section C.3, we have $Y_0 = BC$ and the new pathway segment is $Y_0 \xrightarrow{1:1,0} B$. The conflicting function Ψ is given by $\Psi = ABC$. Clearly, $supp(\Psi) \subset supp(Y_0)$. Since $Y_0 > \Psi$, we put 0 in every minterm $\in supp(Y_0) = supp(BC)$. The resulting K-map is shown below.



x	x	x	x
x	x	0	0
1	1	0	0
1	1	1	1

Next, we have to find a minimal set $S(1)$ in the K-map which will suggest a new pathway segment for resolving the conflict. To do this, the 1s in the K-map of B can be clustered as shown below.



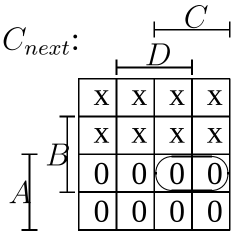
x	x	x	x
x	x	0	0
1	1	0	0
1	1	1	1

From this clustering, we obtain the set $S(1) = A\bar{B} + A\bar{C}$ which suggests the composite pathway segment: $\Psi \xrightarrow{1:1,1} S(1)$.

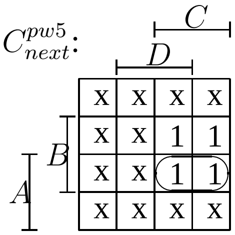
$$\begin{aligned} \text{Now } \Psi \xrightarrow{1:1,1} S(1) \quad &\Leftrightarrow \quad ABC \xrightarrow{1:1,1} A\bar{B} + A\bar{C} \\ &\Leftarrow \quad ABC \xrightarrow{1:1,1} A\bar{B} \text{ (choosing one)} \\ &\Leftrightarrow \quad ABC \xrightarrow{1:1,1} A \\ &\& \quad ABC \xrightarrow{1:1,0} B. \end{aligned} \tag{3.11}$$

Thus, we get two new pathway segments : pathway 7 :$ABC \xrightarrow{1:1,1} A$ and pathway 8 :$ABC \xrightarrow{1:1,0} B$.

Conflict in the truth table of gene C: While trying to incorporate the simple pathway segment 5, we get the conflict shown below for gene C.

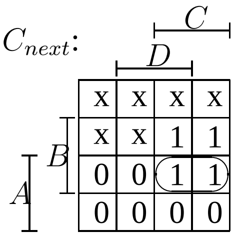


x	x	x	x
x	x	x	x
0	0	0	0
0	0	0	0



x	x	x	x
x	x	1	1
x	x	1	1
x	x	x	x

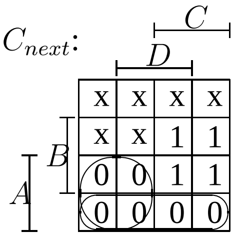
Following the same procedure as we did for pathway 4, we see that $Y_1 = BC$ and the new pathway segment is $Y_1 \xrightarrow{1:1,1} C$. The conflicting function Ψ is given by $\Psi = ABC$. As before, $supp(\Psi) \subset supp(Y_1)$. Since $Y_1 > \Psi$, we put a 1 in every minterm $\in supp(Y_1) = supp(BC)$. The resulting K-map is shown below.



x	x	x	x
x	x	1	1
0	0	1	1
0	0	0	0

Next we have to find a minimal set $S(0)$ in the K-map which will suggest a new pathway segment for resolving the conflict. To do so, the 0s in the K-map of C can be clustered below.

C_{next}:



x	x	x	x
x	x	1	1
0	0	1	1
0	0	0	0

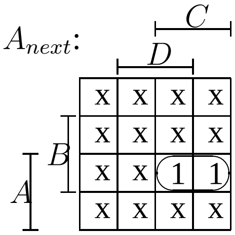
From this clustering, we obtain the set $S(0) = A\bar{B} + A\bar{C}$ which suggests the composite pathway segment: $\Psi \xrightarrow{1:1,1} S(0)$.

$$\begin{aligned} \text{Now } \Psi \xrightarrow{1:1,1} S(0) \quad &\Leftrightarrow \quad ABC \xrightarrow{1:1,1} A\bar{B} + A\bar{C} \\ &\Leftarrow \quad ABC \xrightarrow{1:1,1} A\bar{B} \text{ (choosing one)} \\ &\Leftrightarrow \quad ABC \xrightarrow{1:1,1} A \qquad (3.12) \\ &\& \quad ABC \xrightarrow{1:1,0} B. \end{aligned}$$

Thus, we arrive at the same two pathway segments as before, pathway 7 :$ABC \xrightarrow{1:1,1} A$ and pathway 8 :$ABC \xrightarrow{1:1,0} B$.

Solution of pathways 7 and 8: There are no additional conflicts introduced while trying to incorporate these two pathway segments. The implementation for pathway6 is shown below.

A_{next}:



x	x	x	x
x	x	x	x
x	x	1	1
x	x	x	x

Pathway 7 :$ABC \xrightarrow{1:1,0} B$ has already been implemented in the K-map for gene B as shown below and so no additional conflicts are created.

B_{next}:

x	x	x	x
x	x	0	0
1	1	0	0
1	1	1	1

Thus, the final set of K-maps, devoid of all conflicts, is shown below.

A_{next}:

x	x	x	x
x	x	x	x
x	x	1	1
x	x	x	x

B_{next}:

x	x	x	x
x	x	0	0
1	1	0	0
1	1	1	1

C_{next}:

x	x	x	x
x	x	1	1
0	0	1	1
0	0	0	0

D_{next}:

x	x	0	0
1	1	1	1
1	1	1	1
x	x	0	0

E. Network design to satisfy additional constraints

The pathway information constitutes prior biological knowledge and in the previous sections we have shown how to generate a family of Boolean networks consistent with the given pathway information. However, the cardinality of this family is still quite large and it is reasonable to incorporate other available knowledge and experimental results to further shrink the size of this family. One relevant piece of information that can aid in this is the number, location and relative significance of the attractors. Since the procedure developed earlier provides us with the final Karnaugh maps for each gene, one can easily check to see if the attractor constraints can be satisfied. This is most readily demonstrated using our earlier example.

1. Imposition of attractor constraints

Consider the same example that we considered in section B and section D to construct a family of BNs from pathways. Now, let us additionally assume that experimental data have given us a steady state distribution. For a BN, for the steady state behaviour, one would expect zero probability mass in the transient states and non-zero mass only for the attractors. However, due to the fact that the system is not an ideal BN (possibly a more general Probabilistic Boolean Network which is equivalent to an ergodic Markov Chain), there could be some non-zero mass in the transient states too. Also there can be some noise in the data as well. Therefore, for inferring a BN from experimental data, a threshold should be established for extracting the attractor states. Suppose that experimental data gives us the steady state distribution shown in Fig 17 for our four gene network. Furthermore, suppose that for this example the threshold is chosen to be 0.08. This yields the attractor states: 0010, 0101, 1011 and 1101. Next we need to check whether these attractors are consistent with the family of BNs that we determined in section D. For an attractor to be consistent

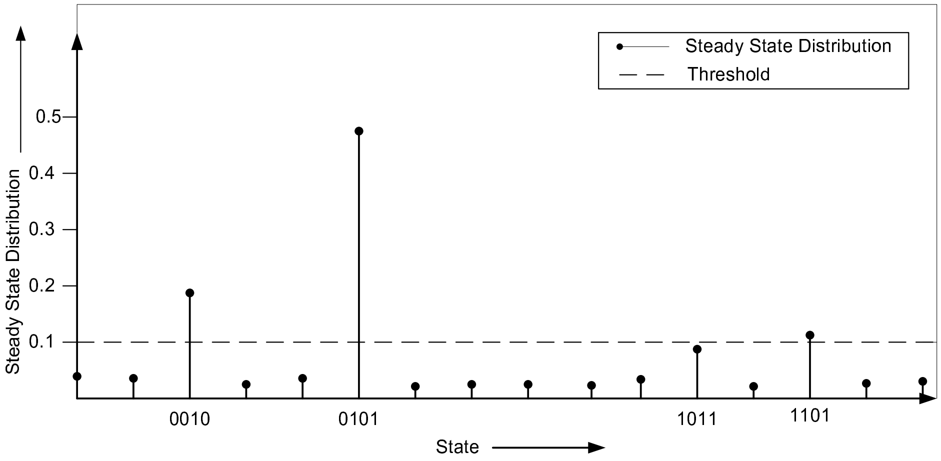
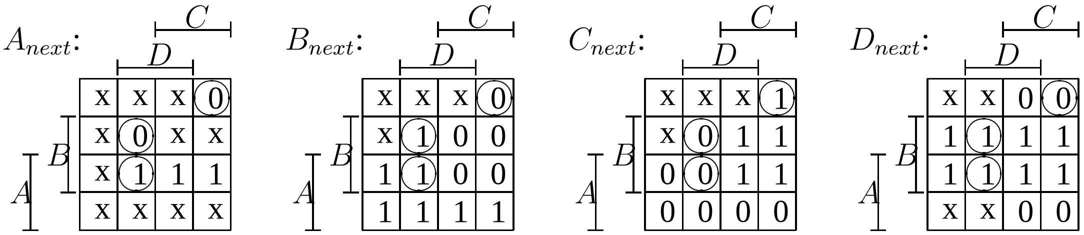


Fig. 17. Steady state distribution and threshold.

with a family of BNs, the rules of regulatory interaction between the genes of each network should guarantee that an attractor transitions only to itself. This can be easily verified from the truth table for the update of each gene. We see that three of the attractors, namely 0010, 0101 and 1101 are consistent with the truth tables shown below.



A_{next}:

x	x	x	⓪
x	⓪	x	x
x	①	1	1
x	x	x	x

B_{next}:

x	x	x	⓪
x	①	0	0
1	①	0	0
1	1	1	1

C_{next}:

x	x	x	①
x	⓪	1	1
0	⓪	1	1
0	0	0	0

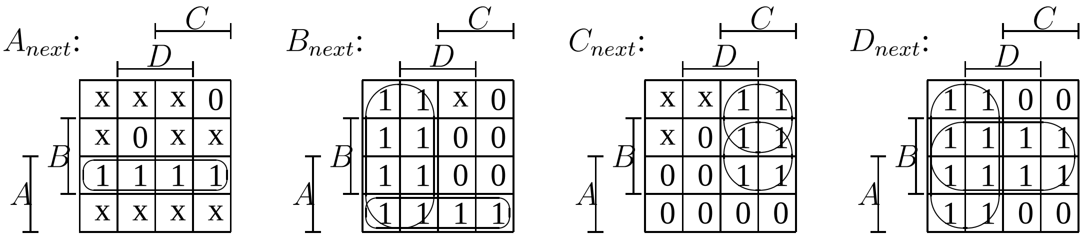
D_{next}:

x	x	0	⓪
1	①	1	1
1	①	1	1
x	x	0	0

However, the attractor 1011 is not consistent with these truth tables, thereby suggesting that it may not be a valid attractor. To remove the state 1011 from the set of attractors obtained from the data, we can increase the threshold to 0.11, say. Thus, the family of BNs that we have constructed based on pathway information provides a useful way to eliminate attractors whose steady state mass is near the threshold value. On the other hand, if we get some attractor state whose steady state mass is very high (say 0.9) and it still contradicts the truth tables obtained from the pathway knowledge, then we have every reason to question the validity of the pathway information that has been provided to us. So, in that case, the steady state distribution data can be used to assess the accuracy of our pathway information.

2. Boolean network from predictors

Suppose that in the above example, one imposes the additional constraint that the maximum number of predictors allowed for each gene is 3. Such an upper limit on the number of predictors per gene could be motivated from the biological consideration that the promoter region for a gene only has enough room for at most only a few transcription factors to bind. We currently do not have a systematic procedure for imposing such a predictor constraint. However, arbitrarily putting in some 1's and 0's for the x's, it is possible to use the four truth tables derived in section E.1 to arrive at a reasonable Boolean Network by using a Karnaugh Map. For instance, by considering the truth tables (K-maps) as shown below, taking the circled *minterms* and filling up all the x's by 0, we obtain the the Boolean rules (Eqn.(3.13)).



A_{next}:

x	x	x	0
x	0	x	x
1	1	1	1
x	x	x	x

B_{next}:

1	1	x	0
1	1	0	0
1	1	0	0
1	1	1	1

C_{next}:

x	x	1	1
x	0	1	1
0	0	1	1
0	0	0	0

D_{next}:

1	1	0	0
1	1	1	1
1	1	1	1
1	1	0	0

$$\begin{aligned} A_{next} &= AB \\ B_{next} &= A\bar{B} + \bar{C} \\ C_{next} &= (\bar{A} + B)C \\ D_{next} &= B + \bar{C} \end{aligned} \tag{3.13}$$

This is a Boolean network with at most three predictors per gene and it satisfies the original pathway constraints, after some minor timing modifications. As discussed earlier, such minor timing modifications are inconsequential since biological pathway information usually does not come with strict timing. Next we can determine the attractor and attractor basins for the generated network. For the network given in Equation (3.13), the state transition diagram and the attractors are shown in Fig 18. From the state transition diagram in Fig 18, it is easy to verify that the state trajectories obey the original pathway constraints, of course with the timing possibly altered. For instance, consider the state trajectory $1010 \rightarrow 0100 \rightarrow 0101$ marked in purple in Fig. 18. The red numbers in this trajectory show that the

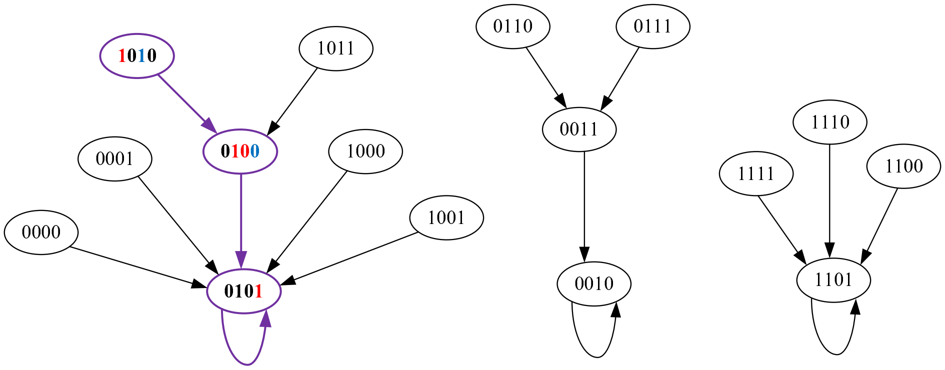


Fig. 18. State transition diagram for the Boolean network described by Equation (3.13).

following pathway relationships are realized: $A = 1$ implies $B = 1$ and $C = 0$ after one time step. $B = 1$ implies $D = 1$ after one time step. Similarly, the blue numbers show the realization of the pathway relationships: $C = 1$ implies $D = 0$ in one time step.

F. Modeling pathways involving the p53 gene

Some of the most widely studied pathways in molecular biology involve the tumor suppressor gene $p53$. In fact, p53 is the “Master Guardian” gene[8] which plays a very important role in cancer. Indeed, it has been observed that p53 is mutated in $30\% - 50\%$ of commonly occurring human cancers [8] and more importantly some parts of the p53 pathways are altered in almost all types of human cancer. Thus, the dynamical behavior of p53 and its tight regulation has become one of the most widely studied problems in cancer biology [68, 69, 70, 71, 72, 73]. Unlike many other important regulated genes, p53 is constitutively expressed in the cell. However, the p53 protein concentration is low under normal conditions. This constitutive but low expression is maintained by the Mdm2 protein: p53 being a transcription factor expresses Mdm2 which in turn binds to p53 and promotes its ubiquitination and degradation [5]. Since the protein-protein interactions occur on a much faster time scale than transcription and translation, the presence of active p53 is usually not detected in a normal cell under normal conditions. The biological reason for the constitutive expression of p53 is that it facilitates a fast response in the face of extreme stress: it is much easier and faster to stop the degradation of p53 protein (by blocking the negative regulators) than turning on the un-expressed p53 gene. The primary role of p53 in mammalian genomes is its function as a transcription factor for hundreds of downstream genes. The expression of these downstream genes can modulate cell cycle progression, repair damaged DNA, induce senescence and apoptosis. A detailed discussion of the cellular processes mediated by p53 can be found in [8]. Although there are hundreds of genes that are downstream of p53, our main goal here is to model the dynamics of p53 itself. So, we focus on the pathways known to be important for p53's regulation. From [3], we get some major pathways involving p53 which are activated in the presence of double strand DNA breaks. These pathways are shown in Fig. 19. In the following subsections we develop the Boolean Network from the pathways of Fig. 19 using the method of this chapter. Thereafter, we simulate the dynamic behavior of the resulting BN. Finally we validate our model by matching our model's time course behavior with the p53-related experimental results reported in [74, 5].

1. Boolean network modeling of the p53 pathways

The pathway segments from the pathways in Fig.19 are: $1.dna\_dsb \xrightarrow{1:1,1} ATM$, $2.ATM \xrightarrow{1:1,1} p53$, $3.p53 \xrightarrow{1:1,1} Wip1$, $4.p53 \xrightarrow{1:1,1} Mdm2$, $5.ATM \xrightarrow{1:1,0} Mdm2$, $6.Mdm2 \xrightarrow{1:1,0} p53$, $7.Wip1 \xrightarrow{1:1,1} Mdm2$, $8.Wip1 \xrightarrow{1:1,0} ATM$.

Here the external signal is dna_dsb, the DNA damage input. The state space is defined as $[ATM, p53, Wip1, Mdm2]$. Using the methodology developed in earlier sections we get the following Boolean update functions for the 4 genes:

$$\begin{aligned} ATM_{next} &= \overline{Wip1}(ATM + dna\_dsb) \\ p53_{next} &= \overline{Mdm2}(ATM + Wip1) \\ Wip1_{next} &= p53 \\ Mdm2_{next} &= \overline{ATM}(p53 + Wip1). \end{aligned} \tag{3.14}$$

This Boolean network will have two different contexts based on the value of the external signal dna_dsb.

If $dna\_dsb = 0$, we get the state transition diagram of Fig. 20.

We can see the state transition diagram of the Boolean network has only one attractor 0000. Now our prior biological knowledge [8] indicates that in absence of any stress, all four proteins are required to be inactive in the steady state. The presence of the singleton attractor 0000 is consistent with the biological information. Next let us see what happens if $dna\_dsb = 1$ i.e, the DNA damage input turns on. In this case, we arrive at the transition diagram shown in Fig. 21 which corresponds to the other interesting context. Notice that here there is a single cyclic attractor involving cyclic variation in the expression patterns of all the four genes.

2. Model validation using the published literature

To further understand the functionality of the context sensitive Boolean network of Eqn. 3.14, we carried out the simulation described next. Suppose that initially the network state is evolving in the absence of the DNA damage signal and that at a certain time (say, t = 25 time steps), the DNA damage signal dna_dsb is activated. Let us further assume that the DNA damage signal dna_dsb returns to 0 at time = 75 time steps. The simulated time course behavior of the expression patterns of the different genes is shown in Fig. 22.

From this simulation we can see that the proteins initially reach the steady state of

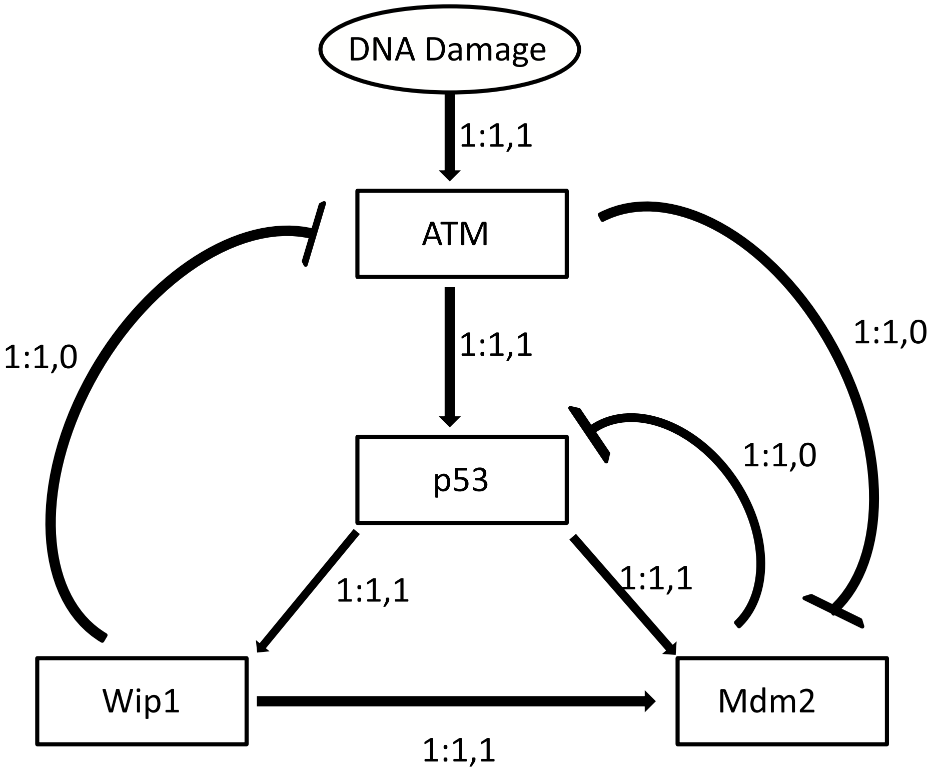


Fig. 19. ATM-p53-Wip1-Mdm2 pathways (From [3]).

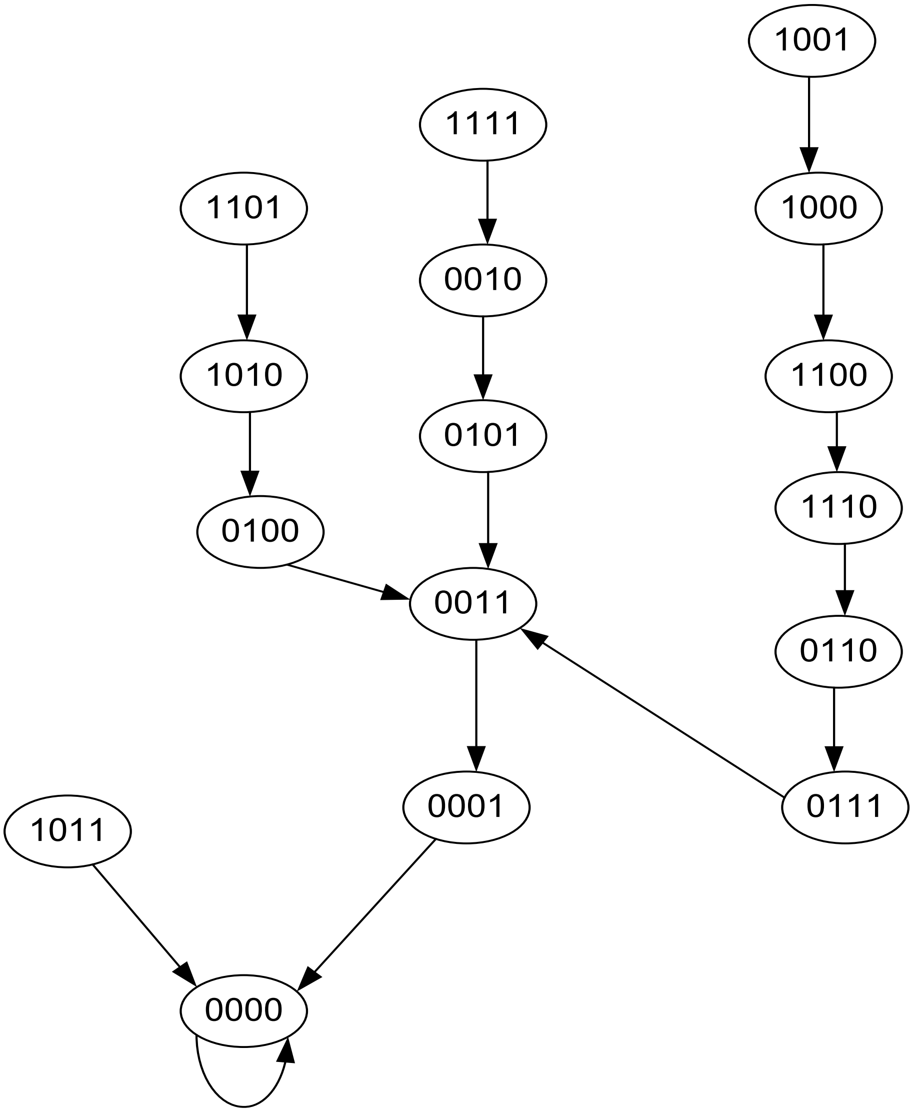


Fig. 20. State transition diagram for the Boolean Network of the p53 pathways under normal conditions.

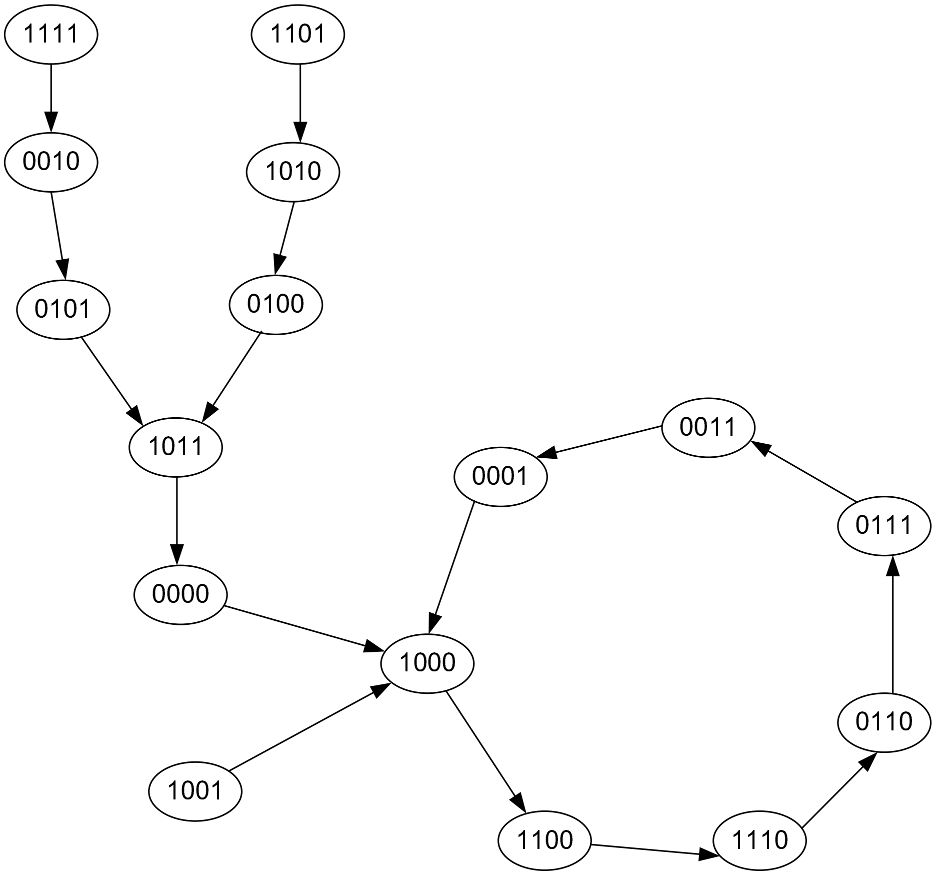


Fig. 21. State transition diagram for the Boolean Network of the p53 pathways in the presence of DNA damage.

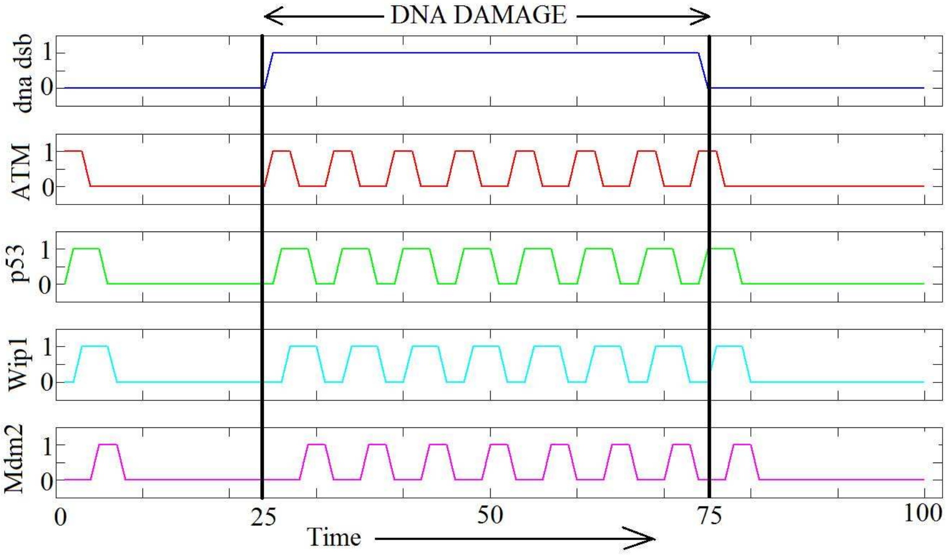


Fig. 22. Oscillation of the proteins in the presence of the DNA damage signal.

deactivation if the cell doesn't receive any stress causing DNA damage. However, once the onset of DNA damage occurs, the oscillation starts. Furthermore, the oscillation continues until the DNA damage is repaired (or the cell dies). The pattern of the oscillation is also unique. ATM leads the oscillation followed by $p53$, $Wip1$ and $Mdm2$ in that order. This dynamic behavior of the four proteins is consistent with published experimental results from the p53 literature.

Indeed [74] discusses the experimentally observed oscillations between $p53$ and $Mdm2$ in the presence of external stress. In that paper it is also reported that the $Mdm2$ protein response lags behind the $p53$ response. [4] reports an interesting time series experiment of $p53$ and $Mdm2$ oscillation and the results are shown in Fig. 23. Similarly [5] reports the DNA damage induced oscillation patterns of ATM, $p53$, $Wip1$ and $Mdm2$ along with some other proteins. Fig. 24 demonstrates that the $p53$ response lags behind the ATM response; Fig. 25 demonstrates that the $Mdm2$ response lags behind the $p53$ response; and

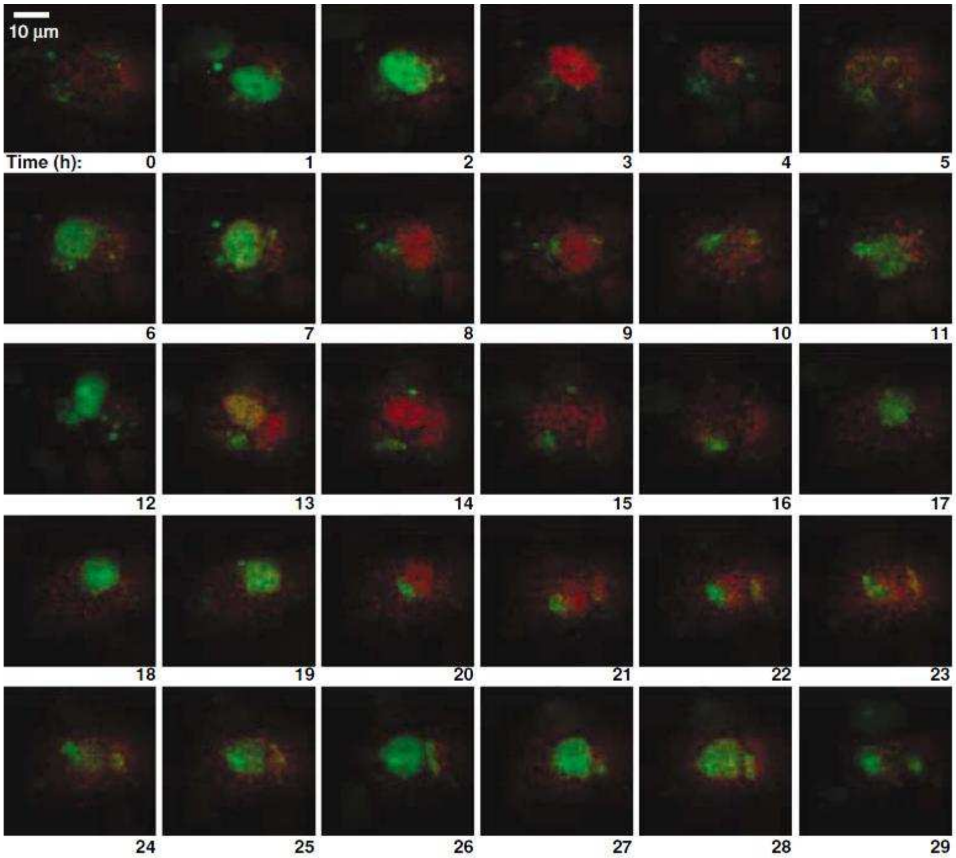


Fig. 23. Timelapse fluorescence images of one cell over 29 h after 5 Gy of gamma irradiation. Nuclear p53-CFP and Mdm2-YFP are imaged in green and red, respectively. Time is indicated in hours. Adapted by permission from Macmillan Publishers Ltd: [Molecular Systems Biology] [4], copyright (2006)

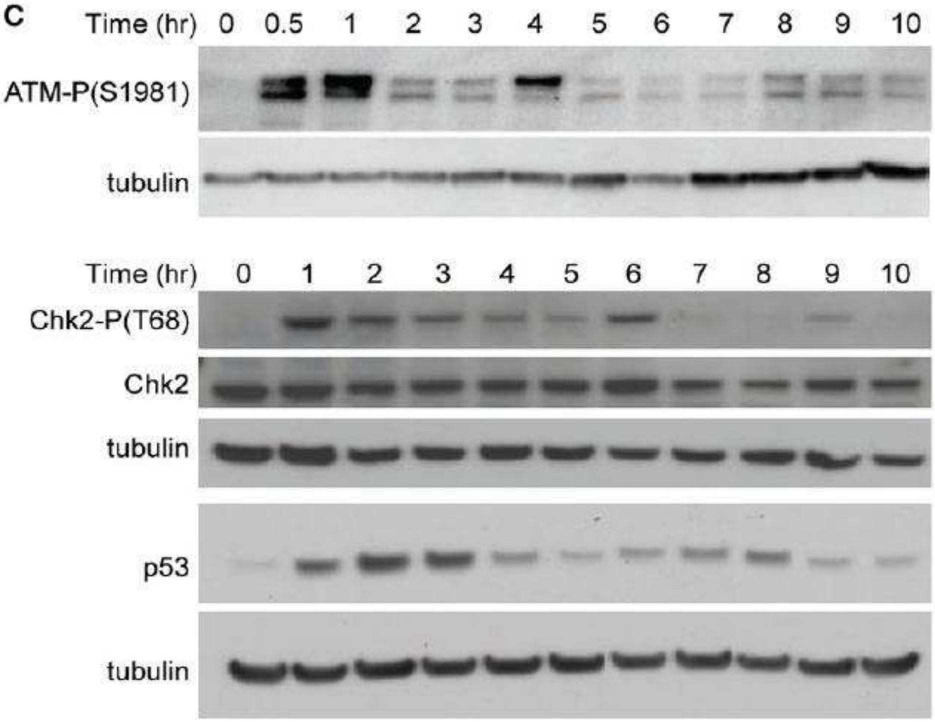

C
Time (hr) 0 0.5 1 2 3 4 5 6 7 8 9 10
ATM-P(S1981)
tubulin
Time (hr) 0 1 2 3 4 5 6 7 8 9 10
Chk2-P(T68)
Chk2
tubulin
p53
tubulin

Fig. 24. Immunoblots of ATM-P(S1981), Chk2-P(T68), and p53 kinetics in MCF7 cells irradiated with 10Gy of gamma-irradiation. Reprinted from [5], Copyright (2008), with permission from Elsevier.

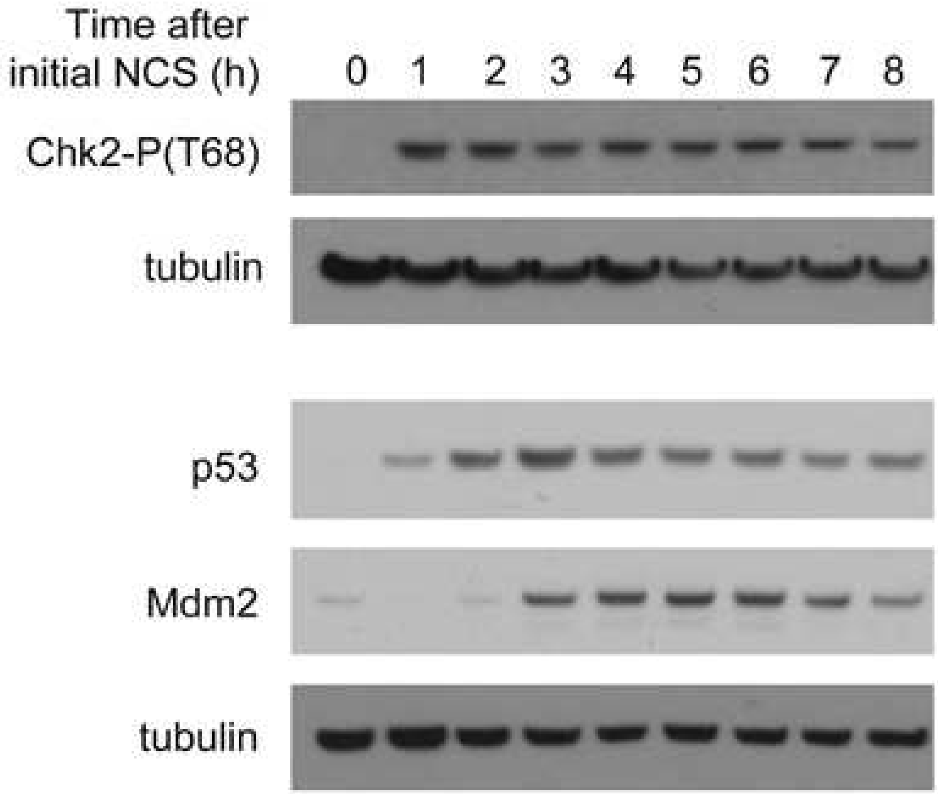


Fig. 25. Immunoblots of Chk2-P(T68), p53, and Mdm2 kinetics in MCF7 cells treated with 400 ng/ml NCS every hour. Blots are representative of triplicate experiments. Reprinted from [5], Copyright (2008), with permission from Elsevier.

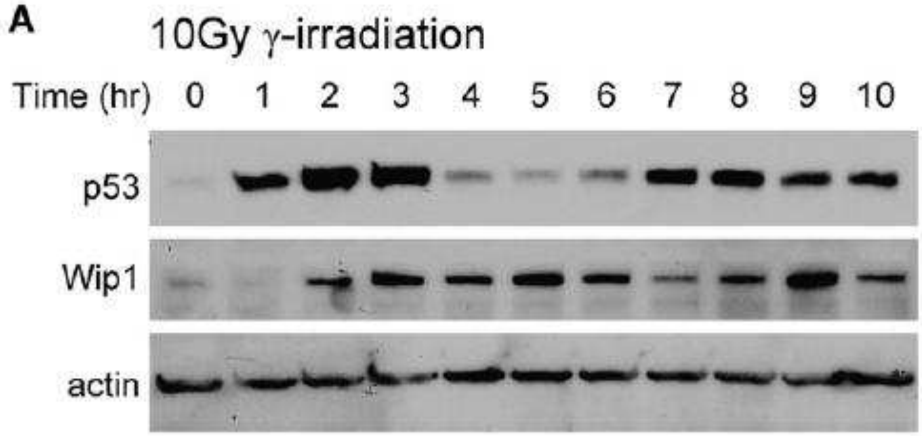


Fig. 26. Immunoblots of p53 and Wip1 kinetics in MCF7 cells irradiated with 10 Gy of gamma-irradiation. Reprinted from [5], Copyright (2008), with permission from Elsevier.

Fig. 26 demonstrates that the $Wip1$ response lags behind the $p53$ response. Thus the network that we have generated based on only p53 pathway information is able to qualitatively reproduce experimentally observed p53 behavior from published literature. This is a very positive development which suggests that the full potential of the approach presented here remains to be explored.

G. Concluding remarks

In this chapter, we have presented a complete solution to the problem of determining a family of Boolean networks that can generate trajectories consistent with given pathway information. The solution makes use of the Karnaugh map realization of a Boolean function. In the case where the different pathways can be implemented without any conflicts in the associated Karnaugh maps, the generation of the family of Boolean networks is straightforward. When a conflict does arise, a systematic procedure is presented to resolve it by slightly perturbing the original pathway information. The resolution of a particular conflict may lead to the emergence of additional conflicts further downstream, and the resolution of these conflicts would require repeated use of the same conflict resolution procedure. When the resolution of the progressively downstream conflicts leads back to one of the original conflicts further upstream, the problem is not solvable. In all other cases, the procedure presented here converges to a family of BNs whose trajectories are consistent with the given pathway information or minor variations thereof. As demonstrated here, further reduction in the cardinality of the family of networks can be achieved by imposing additional constraints such as the number and relative significance of the attractors, upper bounds on network connectivity, etc. The approach developed in this chapter has been applied to the well studied p53 pathway and it has been shown that the resulting network exhibits dynamic behavior consistent with experimental observations from the published literature.

We believe that the results presented here and their future extensions will significantly impact all areas of biology where prior knowledge is present in the form of signalling pathways and where genetic regulatory networks are used to model multivariate gene relationships. In particular, all of the currently available results in the genomic signal processing area pertaining to inference and intervention in genetic regulatory networks will have to be revisited to permit the incorporation of valuable pathway information. For cancer genomics, this would mean that in future, intervention design would be carried out with more accurate and more easily inferred models thereby greatly enhancing the likelihood of these methods succeeding in practice.

CHAPTER IV

FAULT DETECTION AND INTERVENTION IN BNS*

In this chapter, our goal is to go a few steps further from where we left in CHAPTER III. Here, we are not content with just producing a Boolean network model from given pathway information. Instead our objective is to utilize such a model to (i) enumerate all the possible fault scenarios; (ii) use the response of the model to a test input to determine which fault or class of faults has occurred; and (iii) finally use this information to prescribe an appropriate therapeutic action. To keep the discussion biologically focussed, we will consider two biological examples, one for a combinatorial Boolean network and the other for a sequential Boolean network. The chapter is organized as follows. In section A, cancer is modeled as an ensemble of faulty Boolean Networks. In section B, drug therapies are modeled as interventions to alter aberrant network behavior emanating from a fault. Section C and D gives the first biological example (growth factor mediated signalling pathways) showing the power of our methodology. Specifically, fault classification and intervention results for our example are presented. Section E gives the second example (p53 mediated DNA-damage pathways) showing the effectiveness of our approach for sequential Boolean networks as well. Finally section F contains some concluding remarks.

A. Modeling cancer as faults in the signaling network

In molecular biology, the marginal behavior of the normal cell is described using signaling pathways. Boolean networks represent a paradigm that can be used to incorporate this information to model the overall dynamic behavior of the cell, consistent with the pathway

*Part of this chapter is reprinted with permission from “Cancer therapy design based on pathway logic” by R. Layek, A. Datta, M. Bittner, and E. R. Dougherty, 2011, Bioinformatics, vol. 27, no. 4, pp. 548555, Copyright [2011], Oxford University Press. (http://bioinformatics.oxfordjournals.org/content/27/4/548.short) knowledge. However, the translational motivation behind this type of dynamical modeling is to facilitate corrective intervention when the cell behaves abnormally. Cancer is actually a disease of several faults in the network. A ‘fault’ is defined by any structural error of the physical system, such that the dynamics become aberrant. For example, the accumulation of point mutations in the genomic DNA may cause the signaling pathways to behave erratically leading to proliferation. On the other hand, sometimes the fault may not be in the genetic code of a particular protein, but rather it is in the protein synthesis factory ribosome, or in some control mechanism of alternative splicing. The fault could also be in the chromosomal spindle resulting in unequal splitting of the chromosomal DNA between the two daughter cells during cell division. Any of these different kinds of errors could cause structural changes in the regulatory network, thereby changing its dynamics and steady-state behavior. In this section, we try to model different types of biological errors within the Boolean network (digital electronics) framework. In a Boolean Network, the faults can be broadly divided into two types.

- Stuck-at Fault: A stuck-at fault means that a point in the network circuitry is stuck to a particular value. As a result, the incoming information is no longer communicated beyond the faulty point; instead, only the stuck-at value is passed on to the outgoing port. Clearly stuck-at faults can commonly be of two types: ‘stuck-at-1’ faults and ‘stuck-at-0’ faults with obvious interpretations. We next present an example to show that modeling via stuck-at faults makes biological sense.

  In the Mitogen Activated Protein Kinase (MAPK) pathways, an important signaling protein kinase is the Ras protein. Ras is phosphorylated by many upstream proteins (by Growth factor mediated pathways). Once activated, Ras activates downstream proteins which have transcriptional control on cyclin D1 and hence cell cycle progression. However, the inherent enzymatic GTPase activity of Ras hydrolyzes the active Ras-GTP complex into the inactive Ras-GDP complex, so that Ras activity ceases after some time delay. However, if due to some mutations in the Ras gene, the GTPase activity of the Ras protein is lost, the once activated Ras protein will be constitutively active and will signal the downstream transcription causing proliferation and cancer [8]. This constitutive activation of Ras can be modeled as a ‘stuck-at-1’ fault in the Ras node of the Boolean network model of the cell signal transduction. Indeed, the “stuck-at” fault is a very common one in cancer biology. One of the earliest findings of a very prevalent mutation in cancer was the identification of the Ras oncogene family members, HRAS, KRAS and NRAS. These genes play a critical role in the signaling that drives proliferation. KRAS genes constitutively activated by mutations are found at the very high rate of 17-25% in human cancers [75].

- Bridging Fault: As the name suggests, a bridging fault refers to the disruption of old interconnections and incorporation of new interconnections in the network. Bridging faults also make biological sense. The molecular signal transduction relies on the sequences and 3 dimensional conformations of the molecules involved. So, any variation in the sequence and 3 dimensional conformation of a molecule (mainly protein) will alter its functionality. As a result, many pathways involving that molecule will become inactive while the altered molecule may open up new ones. Without any loss of generality, this kind of aberrant behavior could be modeled as a bridging fault in the Boolean network.

  Indeed, the “bridging” fault is also a common occurrence in human cancers. A wide variety of tumor types carry chromosomal translocations, where parts of different chromosomes have been joined together. The first such event to be associated with a specific cancer is the Philadelphia chromosome, a translocation joining chromosomes 9 and 22 [76] and fusing the BCR and ABL genes. The event makes the action of the Abl kinase constitutive in its stimulation of proliferation and inhibition of DNA repair and, if this happens in early blood cell progenitors in the bone marrow, can cause chronic myelogenous leukemia. A variety of drugs that inhibit this kinase activity can produce remission of the disease.

Stuck-at faults and bridging faults are illustrated in Fig. 27, where a fault free Boolean network is shown in Fig. 27a while the corresponding faulty network is shown in Fig. 27b.

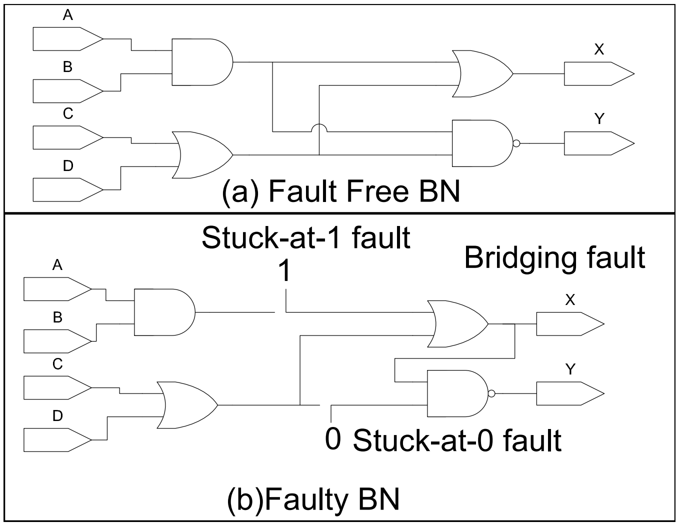


Fig. 27. Stuck-at faults and bridging faults in a digital circuit.

Based on the preceding discussion, it is clear that cancer can be broadly modeled as multiple stuck-at and bridging faults in the Boolean networks corresponding to the normal signaling pathways. In [77], extensive theoretical work on digital system testing and fault modeling is presented which engineers have been successfully using for digital circuit testing for quite some time now. One of the goals of this chapter is to use a similar approach for the prediction of fault locations in cancerous networks and the design of intervention policies to compensate for the effect of these faults. For the sake of simplicity, we will focus only on single stuck-at faults. The more general case of cancer modeling involving multiple stuck-at and bridging faults will be taken up in future research.

1. Test inputs and fault detectability

Consider the BN of Fig. $27a$ which has 4 inputs and 2 outputs as shown. Now suppose that the only possible fault in this network is the stuck-at-1 fault shown in Fig. $27b$. Following [77], for a combinatorial circuit (i.e, non-feedback BN) N, let $Z(x)$ denote the output vector for the input vector x. The presence of a fault f transforms N into N_f with output function $Z_f(x)$ for the same input vector x. We say that a test vector t detects the fault f iff $Z_f(t) \neq Z(t)$. Clearly, for the stuck-at-1 fault in Fig. $27b$, the test input vector $ABCD = 0000$ can detect the fault because, $Z(0000) = 01$ while $Z_f(0000) = 11$. However, the test input vector 1111 cannot detect the fault since $Z(1111) = Z_f(1111) = 10$. These ideas about fault detectability will be applied to the biological examples in section C.3 and section E.1.

B. Modeling drug intervention

In a cancerous network, identification of the fault locations is only a part of the task. The major challenge lies in finding the best possible drug or drug combinations with which to intervene. From a theoretical perspective, we can consider the non-cancerous and cancerous (faulty) networks as two different Boolean networks. In general, it will be impossible to make a cancerous network revert to the original non-cancerous one using any sort of drug intervention, because the mutations leading to cancer are usually irreversible. Instead, what the best drug combination could do is to nullify some of the deadly effects (like constitutive cell division) of the cancerous faulty system and try to kill the cell by inducing apoptosis.

The following modeling of drug intervention is inspired by the biological effect of the drug on the pathways. A drug goes into the cell to bind a particular kinase to deactivate its phosphorylating capability. This means that the drug can cut the effect of that particular kinase on molecules further downstream. Hence, the drug can be modeled as an inverted input to an 'AND' gate at the target point of the Boolean network. This schematic modeling of drug intervention is shown in Fig. 28.

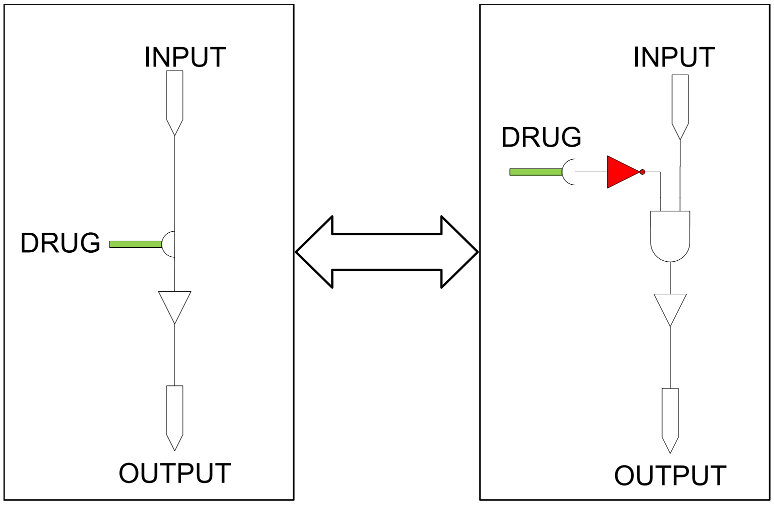


Fig. 28. Drug intervention modeling.

In this chapter, our goal is not to derive the mathematical expression for the optimal drug intervention policy, since most mathematically derived policies may be difficult, or impossible, to biochemically implement. Instead, our objective is to model known and well tested cancer drugs separately and then to find the best sub-optimal combination of drugs for a particular cancerous network. The method is described in detail in section C.4, where it is applied to a biological example.

C. Biological example: growth factors and cellular signal transduction

In multicellular eucaryotic organisms, the cell numbers are very tightly controlled, and cells divide to form more cells only when they receive signals from other cells directing them to do so. The external signals that stimulate a cell to divide are usually called *growth factors* or *mitogens*. Normally these are protein or steroid ligands. The external signal directing a cell to divide is usually communicated to the cell division machinery inside the cell through a transmembrane protein called a *growth factor receptor*. These transmembrane proteins contain the amino acid tyrosine and activate the cell division machinery inside the cell by phosphorylating some key proteins; hence, they are also sometimes referred to as *receptor tyrosine kinases*. Each growth factor binds to its membrane bound receptor with great specificity and when that happens, an intracellular signaling cascade occurs that can result in enhanced cell proliferation, enhanced protein synthesis or inhibition of apoptosis. In this chapter, we will focus on the signaling pathways associated with a number of growth factors. One of the reasons for this choice is that these signaling pathways have not only been widely studied in the context of cancer but also different cancer drugs, known to affect different parts of the pathways, are currently available.

Before presenting a detailed schematic diagram of the components involved in these pathways and their interactions, it is appropriate to first briefly review the eucaryotic cell-cycle and point out how malfunctions in the associated control system can lead to cancer.

1. Cell cycle control, DNA mutation and cancer

In a multicellular organism, cell growth and proliferation are tightly controlled by the cell cycle control system. The typical eucaryotic cell-cycle has four phases called G_1(Gap 1), S(Synthesis), G_2(Gap 2) and M(Mitosis) as shown in Fig. 29. The resting phase G_0 is a phase where the cell has made a decision (in the G_1 phase) to temporarily withdraw from the cell cycle. The G_0 and G_1 phases are in equilibrium with each other so that a resting cell in the G_0 phase can readily re-enter the cell cycle, if the external conditions require additional cells to be produced. In the G_1 phase, the cell processes all the extra-cellular

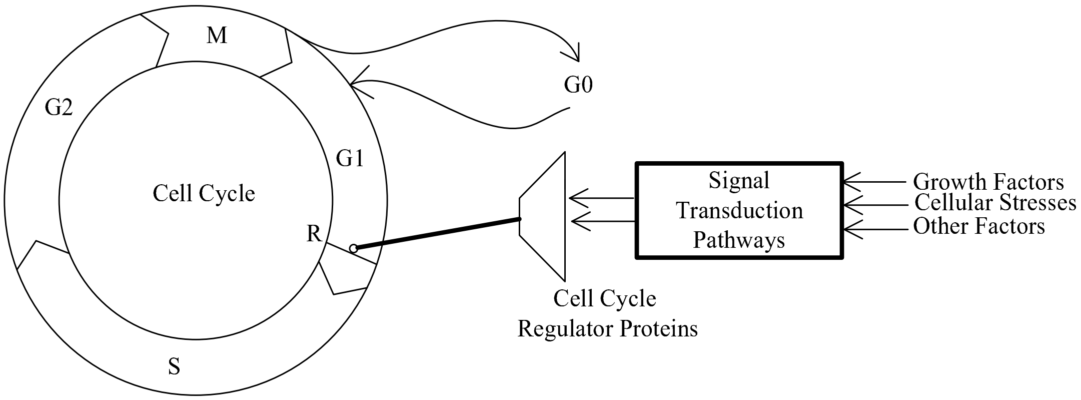


Fig. 29. The eucaryotic Cell Cycle(G0: Quiscence, G1: Gap 1, R: Restriction point, S: Synthesis, G2: Gap 2 and M: Mitosis) and the signal transduction pathways controlling the cell cycle.

signals (through different pathways) and decides whether to go back to G_0 or proceed towards the S or DNA synthesis phase. The R checkpoint (see Fig. 29) is very critical in the cell cycle regulation. Once the cell goes past the R checkpoint, the progression of the cell cycle no longer depends on the mitogens (the growth factors or the inputs of the transduction pathways). Cyclin-CDK(Cyclin dependent kinase) complexes play major roles in the regulation of the cell cycle dynamics. The growth factor activation of the receptor tyrosine kinases results in rapid accumulation of Cyclin D1. Similarly in normal cells, removal of growth factors results in rapid decline in the Cyclin D1 level. This Cyclin D(1 or 2) and CDK4/6 complexes carry the cell past the R checkpoint. Beyond this checkpoint, although there are mechanisms to check for correct DNA replication and proper apportioning of the chromosomes between the two daughter cells, there are no more decisions made between whether to remain in quiescence or to proceed to proliferation. Thus, after the R checkpoint, the cell cycle is more or less automated and independent of the extracellular inputs. In normal cells, if there is no mitogen during the $G_0 \longleftrightarrow G_1$ transition, the cell will not enter the S phase. However, in cancerous cells, the proto-oncogenes can get mutated to become oncogenes. The translated oncoproteins have 3 dimensional conformations which are quite different from that of the corresponding normal protein and can behave differently. For instance, if Ras proto-oncogene mutates to Ras oncogene, the encoded Ras oncoprotein can become constitutively active and start perpetually signaling to the downstream proteins. In that case, even if there is no mitogenic signaling from the outside, the cell will be stimulated to divide. Similarly mutation in pro-apoptotic genes can stop apoptosis resulting in tumorigenesis. Since almost all the genes/proteins along the important proliferation/apoptosis pathways are prone to mutation, the number of possibilities for mutation leading to cancer is quite large.

D. Growth factor mediated pathways: combinatorial network

The particular set of signaling pathways that we will focus on in this first example are the so called Growth Factor (GF) Pathways. Our goal is to model these signaling pathways as an input-output Boolean circuit and to use the latter for (i) enumerating the different fault (or malfunction) possibilities, (ii) carrying out fault classification and (iii) designing the appropriate corrective action (or therapy). Such modeling must necessarily be preceded by a biological understanding of the different components of this pathway and their interactions. Fig. 30 is a schematic diagram showing the different components of this pathway and their interactions, as currently understood by biologists, (for example the Kegg collection of pathways (http://www.genome.jp/kegg/pathway.html) and the NIH BioCarta collection of pathways (http://cgap.nci.nih.gov/Pathways/$BioCarta_Pathways$). The input nodes in the diagram are the growth factors (shown in the rhombuses in Fig. 30). The external signals corresponding to the growth factors are transmitted through the kinase cascades and finally activate the appropriate transcription factors. The black and red lines in the diagram indicate relationships which are known to be activating and inhibitory, respectively. Fig. 30 also shows six different cancer drugs (red boxes) and the points in the pathway where they are believed to intervene.

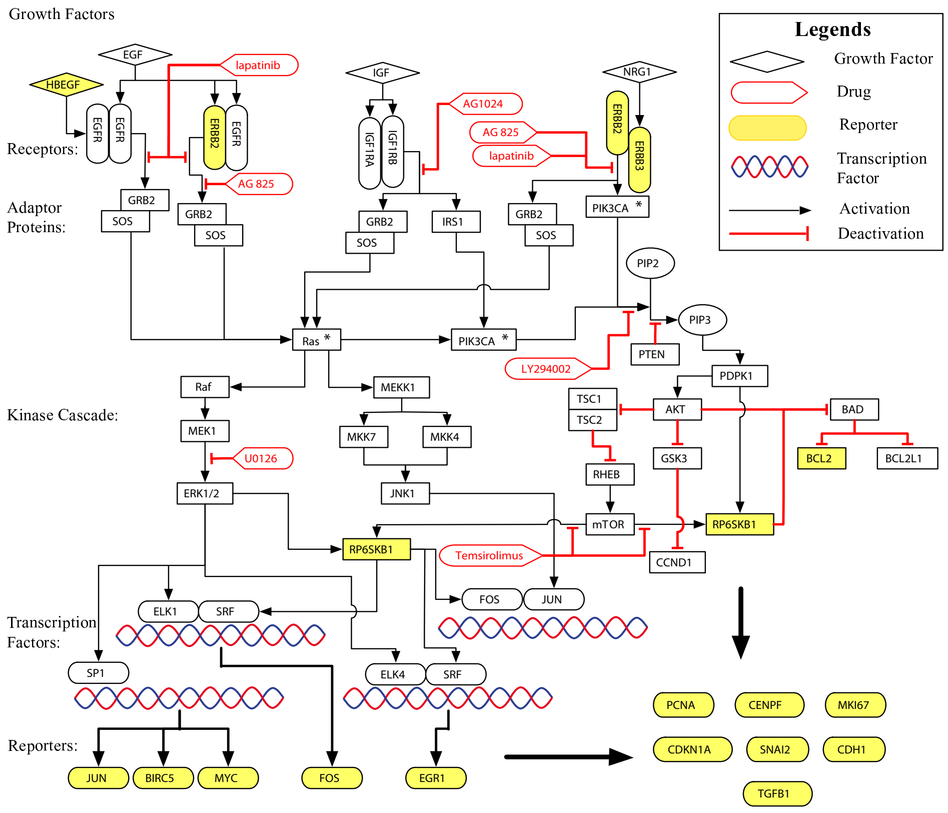


Fig. 30. A schematic diagram of the growth factor signaling pathways (the yellow color is used for the reporter proteins which will be measured in future experiments).

These signal transduction pathways in Fig. 30 constitute a module in a larger tightly controlled network of cell growth, cell division and metabolism.

Believing these pathways to be true, we can develop another level of abstraction by modeling using Boolean Networks (BNs). For most of the pathways the modeling is trivial.

Using the methodology of CHAPTER III, the modeling approach is quite intuitive and logical and can be applied to the pathways in Fig. 30 to arrive at Fig. 31.

This module can be easily modeled using the Boolean circuit shown in Fig. 31, where the seven outputs of interest, shown at the bottom of the figure, are transcription factors (marked in green) and the activation status of some key proteins (not colored). As we will see, such a Boolean circuit model can play an important role in understanding the proliferation versus quiescence decision for a cell.

1. Input-output simulation of the BN

Since, there is no feedback path in the BN of Fig. 31, the current states are independent of previous states. Also we are not concerned about the entire state vector, rather we are primarily interested in the output response of the network. Hence, the complete input-output mapping is essential for understanding the dynamics of this BN. This mapping is shown in Table VI.

It is evident from the simulation that only the input of 00001 provides the ‘all-zero output response’. This is again intuitive because 00001 means all the growth factors EGF, HBEGF, IGF and NRG1 are inactive and the negative regulatory protein PTEN is high. This input condition is crucial for investigating the fault scenario inside the network.

2. Modeling faults and therapeutic interventions using the Boolean network

Any mutation of any gene or post transcriptional modification of the corresponding protein can constitutively turn ‘ON’ or ‘OFF’ that particular protein. This fits in precisely within the stuck-at fault paradigm considered in section A. For the sake of simplicity, in our growth factor pathways case study, we will consider only single faults of the stuck-at type. In addition, we will only consider the stuck-at faults which can lead to cancer. For the Boolean circuit shown in Fig. 31 the possible locations for the different stuck-at errors,

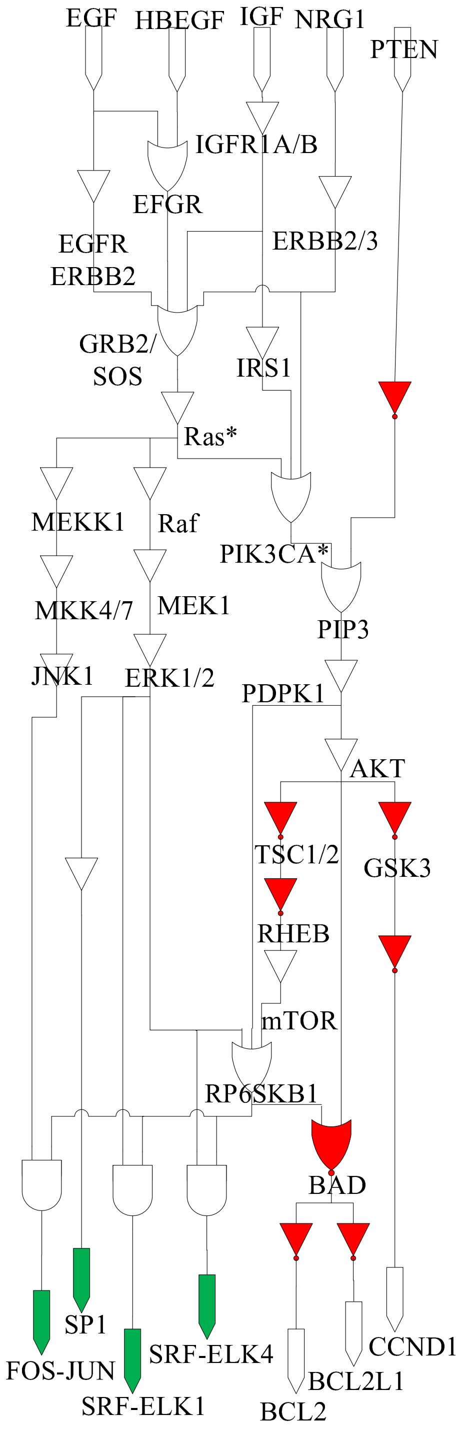


Fig. 31. An input output Boolean network model of the signalling pathways of Fig. 30.

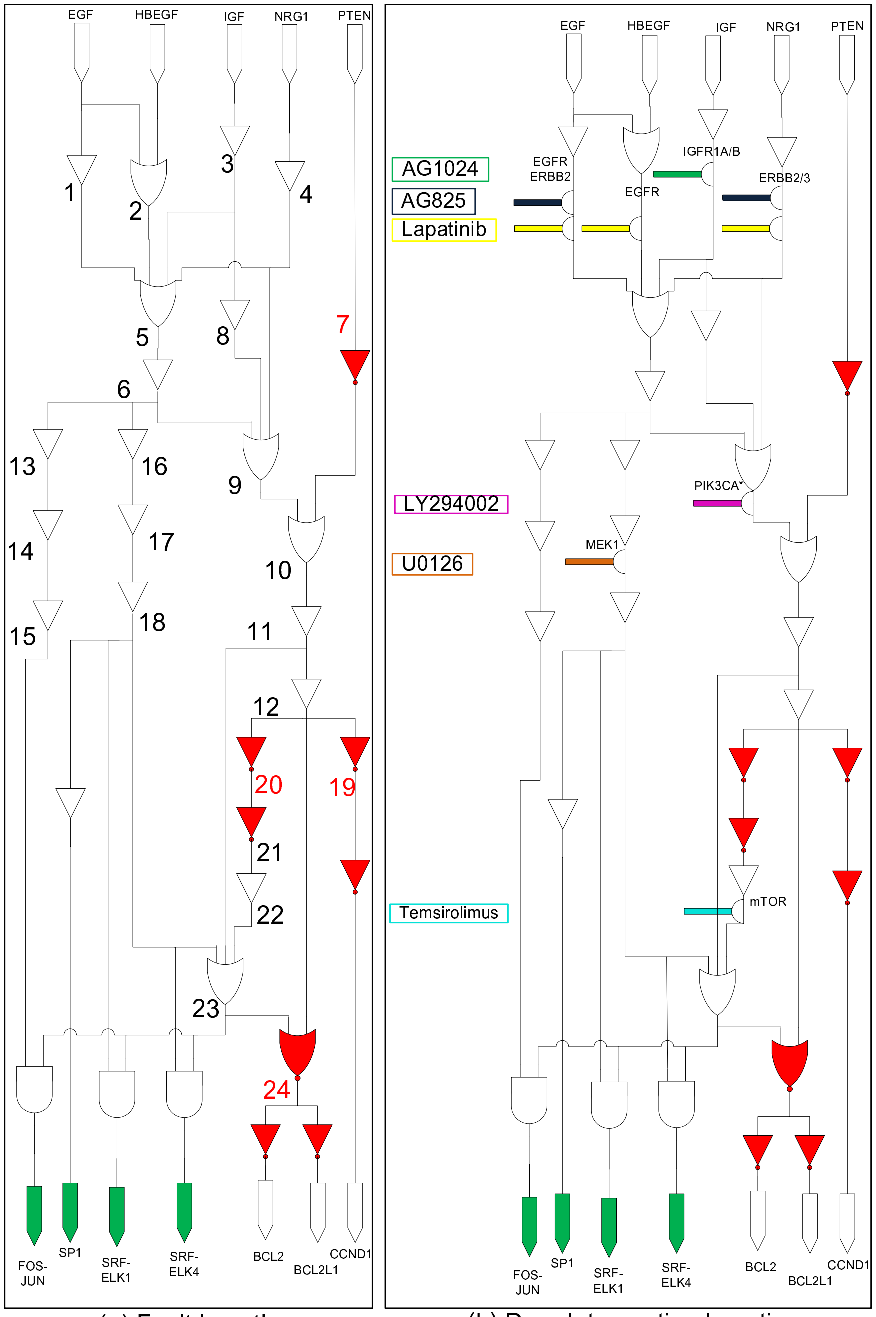
Table VI. Input-output Mapping of the Boolean Network of Fig. 31.

Input	Output	Input	Output
00000	0000111	10000	1111111
00001	0000000	10001	1111111
00010	1111111	10010	1111111
00011	1111111	10011	1111111
00100	1111111	10100	1111111
00101	1111111	10101	1111111
00110	1111111	10110	1111111
00111	1111111	10111	1111111
01000	1111111	11000	1111111
01001	1111111	11001	1111111
01010	1111111	11010	1111111
01011	1111111	11011	1111111
01100	1111111	11100	1111111
01101	1111111	11101	1111111
01110	1111111	11110	1111111
01111	1111111	11111	1111111

which can induce proliferation and stop apoptosis, are shown in Fig. 32a. The numbers are color coded to distinguish between the ‘stuck-at-1’ and ‘stuck-at-0’ faults. Specifically, the black numerals refer to the stuck-at-1 faults while the red numerals refer to the stuck-at-0 faults.

As discussed in section B, a drug targets particular enzymes along the pathways and cuts off the connectivity of that enzyme to the downstream proteins. This connection cleavage can be achieved via various mechanisms. For instance, the drug may have the capability to bind a target protein and inhibit it from undergoing phosphorylation. For our case study, we consider six potent cancer drugs. Our objective here is not to study their detailed mechanisms of action. Instead, we are interested in using the knowledge from biologists to mark in their intervention locations and corresponding activities on the Boolean circuit of Fig. 31. This leads to the effects shown in Fig. 32b. Such pictorial representation of the drug activity information is useful.

For instance, let us consider the drug ‘*lapatinib*’ which is known to work on $EGFR$, $ERBB2$ or $ERBB3$ by inhibiting the signaling capabilities of these receptor tyrosine kinases. From Fig. 32b, one can conclude that the drug ‘lapatinib’ will likely be responsive for the treatment of cancers caused by mutations in the receptor tyrosine kinases although it will probably be ineffective against cancers caused by mutations in the Ras protein, which lies further downstream. Two central objectives of this chapter are: (i) to use the information contained in Fig. 32a to group the numbered faults into different classes; and (ii) to use the information in Fig. 32b to predict which set of drugs/drug combinations would be most effective against a particular fault. These objectives are pursued in the next two subsections.



(a) Fault Locations (b) Drug Intervention Locations

Fig. 32. Possible fault locations and drug intervention locations: (a) proliferative stuck-at fault locations and (b) intervention locations for the available cancer drugs.

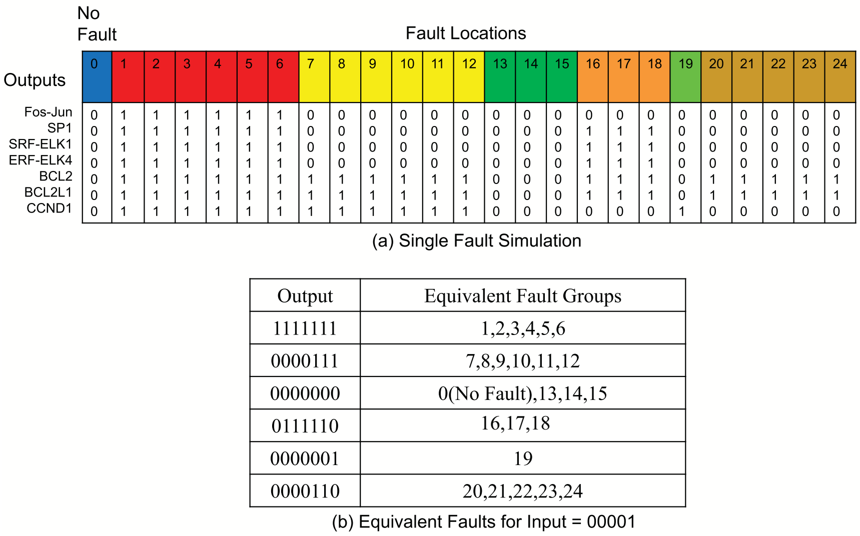
3. Fault analysis and classification

From Fig. $32a$, we see that there are 24 possible fault locations. Alternatively, we could have arrived at the fault locations based on our biological understanding. As already indicated, in this chapter we will be confining ourselves to the analysis of single faults only. So, for our purposes, the fault can be any one of the 24 faults in the figure. Carrying forward the discussion from section A, we use f_i^1 to denote the fault at the i^{th} location. Then the sample space for the single fault modeling can be defined as $F^1 = \{f_1^1, f_2^1, f_3^1, \cdots, f_{24}^1\}$. Here the superscript 1 refers to the fact that we are considering only single faults. Now if $f_i^1 \in F^1$ occurs, an input vector t detects the fault iff the output vector Z in the faultless system differs from the output vector Z_{f_1} in the faulty system. Mathematically $Z(t) \neq Z_{f_1}(t)$. If we cannot find such an input t, we say the fault is undetectable. In the circuit shown in Fig. $32a$, the only input vector which can detect any $f_i^1 \in F^1$ for this particular network is $V = 00001$ which is achieved with $EGF = 0$, $HBEGF = 0$, $IGF = 0$, $NRG1 = 0$ and $PTEN = 1$. This is due to the fact that for any other binary input V, all the outputs are equal to 1, regardless of whether a stuck-at fault is present or not. This result is not at all surprising. Indeed, when there is no growth factor outside the cellular membrane and also the tumor suppressor protein $PTEN$ is active, we expect to see all the proliferative transcription factors and anti-apoptotic factors deactivated or turned ‘OFF’. However, if there are faults (mutations) in the signal transduction pathways, we could see proliferation even in the absence of active input signals (mitogens).

a. Single fault simulation

In this subsection, computer analysis for the single fault model of the circuit in Fig. 31 is presented. The single fault model of the Boolean circuit is shown in Fig. $32a$. The input vector is $V = [EGF, HBEGF, IGF, NRG1, PTEN]$. Each input can take binary values.

For this simulation we take $V = 00001$. The output vector is Z = [$FOS - JUN$, $SP1$, $SRF - ELK1$, $SRF - ELK4$, $BCL2$, $BCL2L1$, $CCND1$]. For the fault-free circuit we get the output $Z(00001) = 0000000$. Now for the 24 different faults which may induce cancer in the given circuit, the outputs are tabulated in Fig. $33a$.



	No Fault	Fault Locations																							
Outputs	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
Fos-Jun	0	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
SP1	0	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	1	1	1	0	0	0	0	0	0
SRF-ELK1	0	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	1	1	1	0	0	0	0	0	0
ERF-ELK4	0	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	1	1	1	0	0	0	0	0	0
BCL2	0	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0	1	1	1	0	1	1	1	1	1
BCL2L1	0	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0	1	1	1	0	1	1	1	1	1
CCND1	0	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	1	0	0	0	0	0

(a) Single Fault Simulation

Output	Equivalent Fault Groups
1111111	1,2,3,4,5,6
0000111	7,8,9,10,11,12
0000000	0(No Fault),13,14,15
0111110	16,17,18
0000001	19
0000110	20,21,22,23,24

(b) Equivalent Faults for Input = 00001

Fig. 33. Single fault simulation: (a) output simulation in presence of all proliferative single stuck-at faults for input V = 00001 and (b) equivalent faults for input V = 00001.

b. Fault classification

From the outputs shown in Fig. $33a$, we can classify the faults into different groups of equivalent faults. Faults which generate the same output vectors for a particular test input vector are called *'equivalent faults'* with respect to that input test vector. The information in Fig. $33a$ leads us to sets of equivalent faults for the test input vector $V = 00001$. The equivalent fault groups along with their corresponding outputs are shown in Fig. $33b$.

From Fig. $33b$, it is clear that any fault in the locations $13, 14, 15$ cannot be detected from the output since the corresponding output is the same as that for the fault-free case.

Hence, this class of faults is said to be ‘undetectable’. It is true that ‘undetectable faults’ cannot be compensated for based on observations of the output. Assuming that the outputs are true indicators of the processes being monitored, there is no reason why we should be concerned with faults that do not manifest themselves in the outputs. Hence, this is not a major concern especially if we are only interested in the behavior of the outputs.

4. Simulation results for drug intervention

Since we have only the 6 available drugs, we define a drug vector of length 6 as follows. If a particular drug is applied it is assigned the value 1, otherwise it is assigned the value 0. Consequently, the drug vector space has cardinality $2^6 = 64$. The simulation is carried out for all of the possible faults, taken one at a time, and for each of the 64 different drug vectors, and the corresponding outputs are computed. The drug vector is defined by $[lapatinib, AG825, AG1024, U0126, LY294002, Temsirolimus]$.

a. Continuous real mapping of the output vector

To avoid introducing any possible ambiguity about the origin of the proliferative signaling, we take the same input vector (00001) that we have previously used for the fault analysis. In the no fault case, with the drug vector 000000 we get the output 0000000 which is certainly non-proliferative. However, in the presence of faults, the outputs will be different. The objective of this simulation is to determine the best possible drug sequence which can nullify the effect of the fault, i.e, produce an output close to 0000000 or away from the proliferative output 1111111. We note that although all the output vectors are represented as binary numbers, assigning the usual binary weights to the digits here does not make any biological sense. In other words, 1111111 here does not really mean 127 or 0000111 does not really mean 7. Consequently, we need to determine some transformation which will map these $128 = 2^7$ output vectors to a continuous real number scale in a biologically meaningful way. One way to do this is to proceed as follows.

If we examine the components of the output vector, we see that out of the 7 components, 4 are transcription factors which express (turn ON) the important genes leading to proliferation. The remaining 3 components capture the activation status of some key proteins in the cytoplasm. So, these two groups of outputs have different biological significance and should be encoded separately. A possible mathematical transformation on the output vectors is described next. The output vector is $OUTPUT$ =[$FOS - JUN$, $SP1$, $SRF - ELK1$, $SRF - ELK4$, $BCL2$, $BCL2L1$, $CCND1$].

Now suppose we take the number of active transcription factors as the first variable(F) and the number of active remaining outputs as the second variable(S). The mathematical transformation makes use of these two variables as described in Eqn. 4.2 below:

$$\begin{aligned} Output &= [a, b, c, d, e, f, g] \\ F &= a + b + c + d \\ S &= e + f + g \\ P &= F \times S \\ S &= F + S \\ \psi(Output) &= \alpha P + (1 - \alpha)S, \end{aligned} \tag{4.1}$$

where $\alpha \in (0, 1)$ is a design parameter. The above encoding scheme counts the number of active transcription factors and the number of active key proteins, and combines these two counts via a nonlinear many-to-one map, the idea being to quantify the degree of abnormal behavior , e.g. proliferation in the absence of growth factors, etc. With α chosen as 0.5, the function ψ's values over the full sweep on the drug vectors and faults are shown in Fig. 34. Here the fault numbers and drug vectors are listed along the horizontal and vertical directions respectively. The results are color coded for easier visualization, and the color codes used are tabulated on the right side in Fig 34 .

Different Fault Locations

Drug Vector	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
000000	0	9.5	9.5	9.5	9.5	9.5	9.5	1.5	1.5	1.5	1.5	1.5	1.5	0	0	0	5.5	5.5	5.5	0.5	1	1	1	1	1
000001	0	9.5	9.5	9.5	9.5	9.5	9.5	1.5	1.5	1.5	1.5	1.5	1.5	0	0	0	5.5	5.5	5.5	0.5	0	0	0	1	1
000010	0	7	7	7	7	7	7	1.5	0	0	1.5	1.5	1.5	0	0	0	5.5	5.5	5.5	0.5	1	1	1	1	1
000011	0	7	7	7	7	7	7	1.5	0	0	1.5	1.5	1.5	0	0	0	5.5	5.5	5.5	0.5	0	0	0	1	1
000100	0	3.5	3.5	3.5	3.5	3.5	3.5	1.5	1.5	1.5	1.5	1.5	1.5	0	0	0	0	0	5.5	0.5	1	1	1	1	1
000101	0	3.5	3.5	3.5	3.5	3.5	3.5	1.5	1.5	1.5	1.5	1.5	1.5	0	0	0	0	0	5.5	0.5	0	0	0	1	1
000110	0	0	0	0	0	0	0	1.5	0	0	1.5	1.5	1.5	0	0	0	0	0	5.5	0.5	1	1	1	1	1
000111	0	0	0	0	0	0	0	1.5	0	0	1.5	1.5	1.5	0	0	0	0	0	5.5	0.5	0	0	0	1	1
001000	0	9.5	9.5	0	9.5	9.5	9.5	1.5	1.5	1.5	1.5	1.5	1.5	0	0	0	5.5	5.5	5.5	0.5	1	1	1	1	1
001001	0	9.5	9.5	0	9.5	9.5	9.5	1.5	1.5	1.5	1.5	1.5	1.5	0	0	0	5.5	5.5	5.5	0.5	0	0	0	1	1
001010	0	7	7	0	7	7	7	1.5	0	0	1.5	1.5	1.5	0	0	0	5.5	5.5	5.5	0.5	1	1	1	1	1
001011	0	7	7	0	7	7	7	1.5	0	0	1.5	1.5	1.5	0	0	0	5.5	5.5	5.5	0.5	0	0	0	1	1
001100	0	3.5	3.5	0	3.5	3.5	3.5	1.5	1.5	1.5	1.5	1.5	1.5	0	0	0	0	0	5.5	0.5	1	1	1	1	1
001101	0	3.5	3.5	0	3.5	3.5	3.5	1.5	1.5	1.5	1.5	1.5	1.5	0	0	0	0	0	5.5	0.5	0	0	0	1	1
001110	0	0	0	0	0	0	0	1.5	0	0	1.5	1.5	1.5	0	0	0	0	0	5.5	0.5	1	1	1	1	1
001111	0	0	0	0	0	0	0	1.5	0	0	1.5	1.5	1.5	0	0	0	0	0	5.5	0.5	0	0	0	1	1
010000	0	0	9.5	9.5	0	9.5	9.5	1.5	1.5	1.5	1.5	1.5	1.5	0	0	0	5.5	5.5	5.5	0.5	1	1	1	1	1
010001	0	0	9.5	9.5	0	9.5	9.5	1.5	1.5	1.5	1.5	1.5	1.5	0	0	0	5.5	5.5	5.5	0.5	0	0	0	1	1
010010	0	0	7	7	0	7	7	1.5	0	0	1.5	1.5	1.5	0	0	0	5.5	5.5	5.5	0.5	1	1	1	1	1
010011	0	0	7	7	0	7	7	1.5	0	0	1.5	1.5	1.5	0	0	0	5.5	5.5	5.5	0.5	0	0	0	1	1
010100	0	0	3.5	3.5	0	3.5	3.5	1.5	1.5	1.5	1.5	1.5	1.5	0	0	0	0	0	5.5	0.5	1	1	1	1	1
010101	0	0	3.5	3.5	0	3.5	3.5	1.5	1.5	1.5	1.5	1.5	1.5	0	0	0	0	0	5.5	0.5	0	0	0	1	1
010110	0	0	0	0	0	0	0	1.5	0	0	1.5	1.5	1.5	0	0	0	0	0	5.5	0.5	1	1	1	1	1
010111	0	0	0	0	0	0	0	1.5	0	0	1.5	1.5	1.5	0	0	0	0	0	5.5	0.5	0	0	0	1	1
011000	0	0	9.5	0	0	9.5	9.5	1.5	1.5	1.5	1.5	1.5	1.5	0	0	0	5.5	5.5	5.5	0.5	1	1	1	1	1
011001	0	0	9.5	0	0	9.5	9.5	1.5	1.5	1.5	1.5	1.5	1.5	0	0	0	5.5	5.5	5.5	0.5	0	0	0	1	1
011010	0	0	7	0	0	7	7	1.5	0	0	1.5	1.5	1.5	0	0	0	5.5	5.5	5.5	0.5	1	1	1	1	1
011011	0	0	7	0	0	7	7	1.5	0	0	1.5	1.5	1.5	0	0	0	5.5	5.5	5.5	0.5	0	0	0	1	1
011100	0	0	3.5	0	0	3.5	3.5	1.5	1.5	1.5	1.5	1.5	1.5	0	0	0	0	0	5.5	0.5	1	1	1	1	1
011101	0	0	3.5	0	0	3.5	3.5	1.5	1.5	1.5	1.5	1.5	1.5	0	0	0	0	0	5.5	0.5	0	0	0	1	1
011110	0	0	0	0	0	0	0	1.5	0	0	1.5	1.5	1.5	0	0	0	0	0	5.5	0.5	1	1	1	1	1
011111	0	0	0	0	0	0	0	1.5	0	0	1.5	1.5	1.5	0	0	0	0	0	5.5	0.5	0	0	0	1	1
100000	0	0	0	9.5	0	9.5	9.5	1.5	1.5	1.5	1.5	1.5	1.5	0	0	0	5.5	5.5	5.5	0.5	1	1	1	1	1
100001	0	0	0	9.5	0	9.5	9.5	1.5	1.5	1.5	1.5	1.5	1.5	0	0	0	5.5	5.5	5.5	0.5	0	0	0	1	1
100010	0	0	0	7	0	7	7	1.5	0	0	1.5	1.5	1.5	0	0	0	5.5	5.5	5.5	0.5	1	1	1	1	1
100011	0	0	0	7	0	7	7	1.5	0	0	1.5	1.5	1.5	0	0	0	5.5	5.5	5.5	0.5	0	0	0	1	1
100100	0	0	0	3.5	0	3.5	3.5	1.5	1.5	1.5	1.5	1.5	1.5	0	0	0	0	0	5.5	0.5	1	1	1	1	1
100101	0	0	0	3.5	0	3.5	3.5	1.5	1.5	1.5	1.5	1.5	1.5	0	0	0	0	0	5.5	0.5	0	0	0	1	1
100110	0	0	0	0	0	0	0	1.5	0	0	1.5	1.5	1.5	0	0	0	0	0	5.5	0.5	1	1	1	1	1
100111	0	0	0	0	0	0	0	1.5	0	0	1.5	1.5	1.5	0	0	0	0	0	5.5	0.5	0	0	0	1	1
101000	0	0	0	0	0	9.5	9.5	1.5	1.5	1.5	1.5	1.5	1.5	0	0	0	5.5	5.5	5.5	0.5	1	1	1	1	1
101001	0	0	0	0	0	9.5	9.5	1.5	1.5	1.5	1.5	1.5	1.5	0	0	0	5.5	5.5	5.5	0.5	0	0	0	1	1
101010	0	0	0	0	0	7	7	1.5	0	0	1.5	1.5	1.5	0	0	0	5.5	5.5	5.5	0.5	1	1	1	1	1
101011	0	0	0	0	0	7	7	1.5	0	0	1.5	1.5	1.5	0	0	0	5.5	5.5	5.5	0.5	0	0	0	1	1
101100	0	0	0	0	0	3.5	3.5	1.5	1.5	1.5	1.5	1.5	1.5	0	0	0	0	0	5.5	0.5	1	1	1	1	1
101101	0	0	0	0	0	3.5	3.5	1.5	1.5	1.5	1.5	1.5	1.5	0	0	0	0	0	5.5	0.5	0	0	0	1	1
101110	0	0	0	0	0	0	0	1.5	0	0	1.5	1.5	1.5	0	0	0	0	0	5.5	0.5	1	1	1	1	1
101111	0	0	0	0	0	0	0	1.5	0	0	1.5	1.5	1.5	0	0	0	0	0	5.5	0.5	0	0	0	1	1
110000	0	0	0	9.5	0	9.5	9.5	1.5	1.5	1.5	1.5	1.5	1.5	0	0	0	5.5	5.5	5.5	0.5	1	1	1	1	1
110001	0	0	0	9.5	0	9.5	9.5	1.5	1.5	1.5	1.5	1.5	1.5	0	0	0	5.5	5.5	5.5	0.5	0	0	0	1	1
110010	0	0	0	7	0	7	7	1.5	0	0	1.5	1.5	1.5	0	0	0	5.5	5.5	5.5	0.5	1	1	1	1	1
110011	0	0	0	7	0	7	7	1.5	0	0	1.5	1.5	1.5	0	0	0	5.5	5.5	5.5	0.5	0	0	0	1	1
110100	0	0	0	3.5	0	3.5	3.5	1.5	1.5	1.5	1.5	1.5	1.5	0	0	0	0	0	5.5	0.5	1	1	1	1	1
110101	0	0	0	3.5	0	3.5	3.5	1.5	1.5	1.5	1.5	1.5	1.5	0	0	0	0	0	5.5	0.5	0	0	0	1	1
110110	0	0	0	0	0	0	0	1.5	0	0	1.5	1.5	1.5	0	0	0	0	0	5.5	0.5	1	1	1	1	1
110111	0	0	0	0	0	0	0	1.5	0	0	1.5	1.5	1.5	0	0	0	0	0	5.5	0.5	0	0	0	1	1
111000	0	0	0	0	0	9.5	9.5	1.5	1.5	1.5	1.5	1.5	1.5	0	0	0	5.5	5.5	5.5	0.5	1	1	1	1	1
111001	0	0	0	0	0	9.5	9.5	1.5	1.5	1.5	1.5	1.5	1.5	0	0	0	5.5	5.5	5.5	0.5	0	0	0	1	1
111010	0	0	0	0	0	7	7	1.5	0	0	1.5	1.5	1.5	0	0	0	5.5	5.5	5.5	0.5	1	1	1	1	1
111011	0	0	0	0	0	7	7	1.5	0	0	1.5	1.5	1.5	0	0	0	5.5	5.5	5.5	0.5	0	0	0	1	1
111100	0	0	0	0	0	3.5	3.5	1.5	1.5	1.5	1.5	1.5	1.5	0	0	0	0	0	5.5	0.5	1	1	1	1	1
111101	0	0	0	0	0	3.5	3.5	1.5	1.5	1.5	1.5	1.5	1.5	0	0	0	0	0	5.5	0.5	0	0	0	1	1
111110	0	0	0	0	0	0	0	1.5	0	0	1.5	1.5	1.5	0	0	0	0	0	5.5	0.5	1	1	1	1	1
111111	0	0	0	0	0	0	0	1.5	0	0	1.5	1.5	1.5	0	0	0	0	0	5.5	0.5	0	0	0	1	1

Output Vector
Color Codes

🟥	$\psi^{-1}(9.5)=1111111$
🟧	$\psi^{-1}(7)=1111110$
🟧	$\psi^{-1}(5.5)=0111110$
🟧	$\psi^{-1}(3.5)=1000111$
🟨	$\psi^{-1}(1.5)=0000111$
🟨	$\psi^{-1}(1)=0000110$
🟩	$\psi^{-1}(0.5)=0000001$
🟩	$\psi^{-1}(0)=0000000$

Fig. 34. Drug vector response in the presence of a single fault: (Left) output responses of the circuit for all drug vectors in presence of all single stuck-at faults and (Right) the map between the color codes and the output vectors.

b. Interpretation of the result

From the output tables and the color codes we see that the color green corresponds to non-proliferation while the color red corresponds to a high chance of proliferation even in the absence of mitogenic signals. So, the best drug vector will be the one which can drive the largest number of faulty circuits towards non-proliferative (green) outputs. For example, the drug vector 000110 drives all of the faults $1 - 6$ to green and most of the remaining boxes along that row away from red. So, the drug combination of U0126 and LY294002 will likely be effective in producing a non-proliferative output. Another point to note is that there can be faults (like fault 18 in Fig. 34) whose output cannot be altered using any drug sequence. This is not at all surprising and is consistent with the pathway information that we have. Indeed, the fault location 18 is at the $ERK1/ERK2$ protein and there is no available drug in our list downstream of that protein. Consequently, no drug in this particular case study would be able to block the effect of a mutated $ERK1/ERK2$ protein.

E. p53 mediated DNA damage pathways: sequential network

Since we have already discussed the modeling of p53 mediated DNA damage pathways in CHAPTER III, we will not unnecessarily repeat it here. For further analysis, it sufficeth to only recall the Boolean Network update equations.

$$\begin{aligned}
ATM_{next} &= \overline{Wip1}(ATM + dna\_dsb) \\
p53_{next} &= \overline{Mdm2}(ATM + Wip1) \\
Wip1_{next} &= p53 \qquad\qquad (4.2) \\
Mdm2_{next} &= \overline{ATM}(p53 + Wip1).
\end{aligned}$$

There are two contexts for this Boolean Network depending on the external signal dna_dsb. The state transition diagrams for both the contexts are given in Figs. 20 and 21. Clearly, in the presence of DNA damage the activity of the 4 relevant proteins will oscillate to normalize the behavior of the cell. However, if mutation strikes either of the 4 genes, the question of interest becomes what the behavior of the transformed Boolean network will be. The relevant fault analysis and possible intervention strategies are discussed below.

1. Fault analysis

We simulated the network for all the possible single stuck-at fault scenarios. The resulting steady state behavior of the BN is shown in Table VII. The most interesting observation is that the oscillations have ceased to exist. Analyzing the steady state (assuming the state is observable), it is evident that the steady state singleton attractor corresponding to each fault is unique. So, complete identification of all the single stuck-at faults for this network is possible using just steady-state data. Recall that the state for this network is defined as [ATM, p53, Wip1, Mdm2].

2. Intervention design

If the steady state of the BN of Eqn. 4.2 enters a singleton attractor with $p53 = 0$, the cell loses the capability to repair DNA damage and this increases the risk of acquiring genetic diseases including cancer. So, the objective of intervention design is to stop the replication of mutated DNA. One way to do that would be to induce apoptosis especially if the DNA cannot be repaired. Assuming that the therapeutic interventions utilize the kinase blocking mechanism (section B), the simulation results of Table VIII show the possible corrective actions.

Table VII. Steady State Attractors in the presence of Single Stuck-at Faults.

Fault	DNA_DSB = 0	DNA_DSB = 1
ATM:s-a-0	0000	0000
ATM:s-a-1	1110	1110
p53:s-a-0	1000	1000
p53:s-a-1	0111	0111
Wip1:s-a-0	1100	1100
Wip1:s-a-1	0011	0011
Mdm2:s-a-0	0110	0110
Mdm2:s-a-1	1001	1001

Table VIII. Intervention Design for the Critical Faults in ATM-p53-Mdm2-Wip1 Boolean Network.

Fault	steady state(no control)	control	steady state(control)
ATM:s-a-0	0000	No solution	-
p53:s-a-0	1000	No Solution	-
Wip1:s-a-1	0011	Block Wip1/Mdm2	1100/0110
Mdm2:s-a-1	1001	Block Mdm2	0110

F. Concluding remarks

In this chapter, we have presented a new approach for designing therapeutic intervention policies based on available pathway information and the manner in which drugs target specific pathway connections. Relevant pathway information is first used to produce Boolean networks whose state transitions are consistent with the given pathway information, or minor variations of it. The Boolean network is then used to (i) enumerate all the possible fault scenarios; (ii) classify the faults into different classes based on their responses to a particular test input; and (iii) prescribe an appropriate course of therapeutic action, tailored to the fault or set of faults that has occurred.

CHAPTER V

CONCLUSION

In this dissertation, it has been shown that modeling and controlling the cellular dynamical system is a non-trivial task. The biologist's idea of cellular interactions in the form of signalling pathways can be modeled as Boolean Networks which can serve as the starting point for further systematic research. For instance, if we know the underlying network structure of a particular kind of gene-protein interaction experiment, we can predict the dynamics associated with it. If the experimental result matches the prediction, the degree of confidence in the model will be enhanced. Likewise, if the experimentation refutes the prediction, we need to either increase the accuracy of the experiment or update the knowledge base of the pathways. As we have already seen in the drug efficacy simulation in CHAPTER IV, the simulation can enable us to predict the possibility of success of a certain drug combination for a particular pathway. Experimental design based on the predictive model can save us time and effort and the experiment can be focussed towards a certain goal. Any mismatch between the drug's observed response and the prediction result will again lead to an update of our understanding of the drug as well as the model network. The resulting iterative paradigm for systems biology is sketched in Fig. 35. Intuitively, this update mechanism is the key for success in systems biology. Model based design of experiments will eventually lead to more accurate estimation of the model.

The idea of personalized medicine for diseases like cancer can also be viewed in this context. Genetic profiling of a new patient will decide the first level of diagnosis of the genetic mutations. The systems biologist will then decide the relevant pathways in the tumor from the marginal gene profiling data. Thereafter, the network can be constructed and in vitro experiments can be performed on the tumor cells to get the right drug combination from the drug database. This procedure which can be iterated upon has the potential to

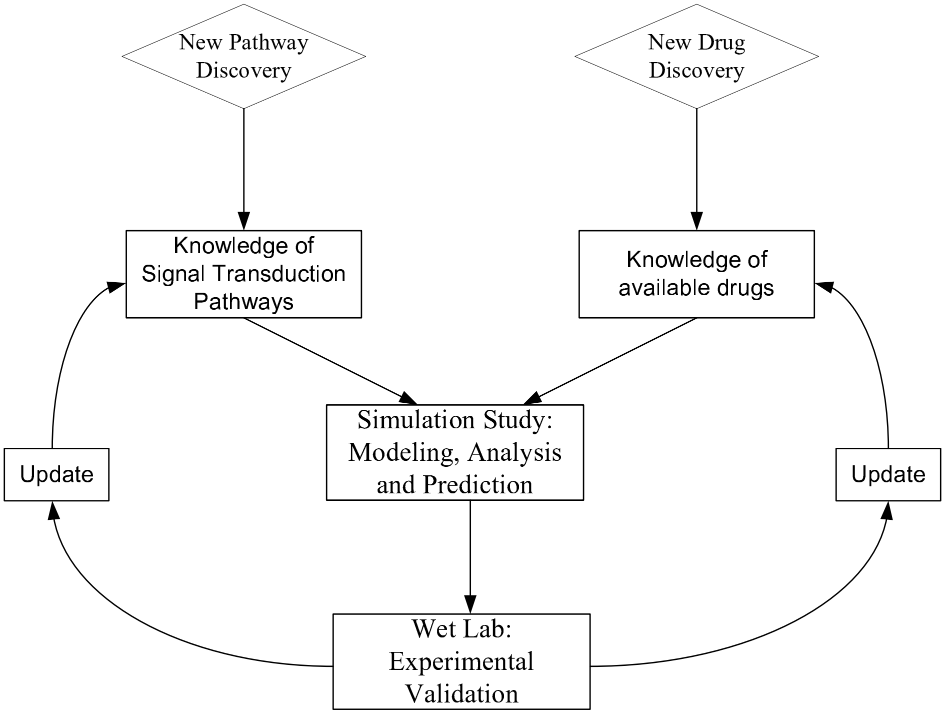

New Pathway Discovery
New Drug Discovery
Knowledge of Signal Transduction Pathways
Knowledge of available drugs
Simulation Study: Modeling, Analysis and Prediction
Update
Update
Wet Lab: Experimental Validation

Fig. 35. Iterative update scheme of pathways and therapeutic target point knowledge in systems biology.

treat a cancer patient in a much better way than any traditional approach. The schematic diagram for this personalized approach to medicine is given in Fig. 36.

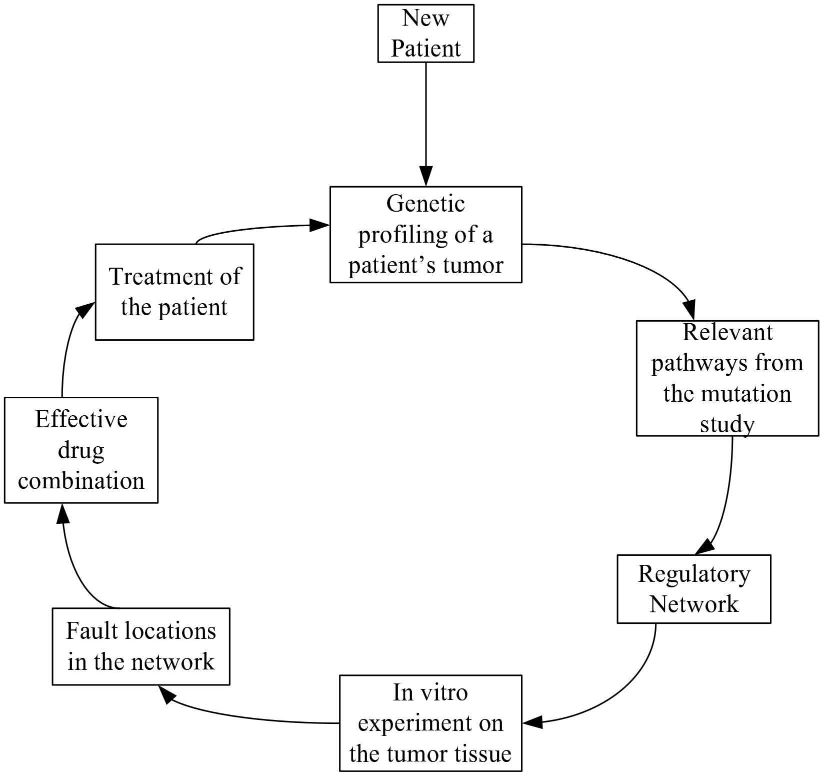

New Patient
Genetic profiling of a patient's tumor
Relevant pathways from the mutation study
Regulatory Network
In vitro experiment on the tumor tissue
Fault locations in the network
Effective drug combination
Treatment of the patient

Fig. 36. Personalized medicine using systems biology.

The list of future research topics is essentially unending. There are so many pathways in so many cell types, that a complete solution of the problem is probably not possible. The subject of systems biology will also evolve as more inventions and discoveries are made. However, for the sake of completeness of the current dissertation we mention a few immediate future research problems:

- Analysis of the multi-fault scenario in Boolean networks.

- Modeling of bridging faults in Boolean networks.
- Developing models for higher level cellular events like cell cycle, apoptosis, differentiation etc.
- Modeling of metabolic regulation in the cell.
- Validating the mathematical models in vitro and in vivo.

REFERENCES

[1] Wikipedia, "http://en.wikipedia.org/wiki/timeline_of_biology_and_organic_chemistry," 2012.

[2] B. Alberts, A. Johnson, J. Lewis, M. Raff, K. Roberts, and P. Walter, *Molecular Biology of the Cell*, 4th ed. New York City, NY, USA: Garland Science, 2002.

[3] E. Batchelor, A. Loewer, and G. Lahav, "The ups and downs of p53: Understanding protein dynamics in single cells," *Nature Reviews: Cancer*, vol. 9, no. 5, pp. 371–377, May 2009.

[4] N. Geva-Zatorsky, N. Rosenfeld, S. Itzkovitz, R. Milo, A. Sigal, E. Dekel, T. Yarnitzky, Y. Liron, P. Polak, G. Lahav, and U. Alon, "Oscillations and variability in the p53 system," *Mol. Syst. Biol.*, vol. 2, pp. 1–13, 2006.

[5] E. Batchelor, C. S. Mock, I. Bhan, A. Loewer, and G. Lahav, "Recurrent initiation: A mechanism for triggering p53 pulses in response to dna damage," *Mol. Cell*, vol. 30, no. 3, pp. 277–289, May 2008.

[6] NIH, "http://www.nigms.nih.gov/research/featuredprograms/sysbio/," 2012.

[7] B. Lewin, *Genes*, 6th ed. New York City, NY, USA: Oxford Univ. Press, 2002.

[8] R. A. Weinberg, *The Biology of Cancer*. Princeton, NJ, USA: Garland Science, 2006.

[9] R. Layek, A. Datta, and E. R. Dougherty, "From biological pathways to regulatory networks," *Mol. BioSyst.*, vol. 7, pp. 843–851, 2011.

[10] R. Layek, A. Datta, R. Pal, and E. R. Dougherty, "Adaptive intervention in probabilistic boolean networks," *Bioinformatics*, vol. 25, no. 16, pp. 2042–2048, 2009.

[11] R. Layek, A. Datta, M. Bittner, and E. R. Dougherty, “Cancer therapy design based on pathway logic,” *Bioinformatics*, vol. 27, no. 4, pp. 548–555, 2011.

[12] F. Jacob and J. Monod, “Genetic regulatory mechanisms in the synthesis of proteins,” *J. Mol. Biol.*, vol. 3, pp. 318–356, 1961.

[13] B. Loriferne, *Analog-Digital and Digital-Analog Conversion*. Philadelphia, PA, USA: Heyden, 1982.

[14] B. C. Kuo, *Digital Control Systems*, 2nd ed. New York City, NY, USA: Oxford Univ. Press, 1995.

[15] A. Datta, A. Choudhary, M. L. Bittner, and E. R. Dougherty, “External control in markovian genetic regulatory networks,” *Machine Learning*, vol. 52, no. 1-2, pp. 169–191, 2003.

[16] R. Pal, A. Datta, M. L. Bittner, and E. R. Dougherty, “Intervention in context-sensitive probabilistic boolean networks,” *Bioinformatics*, vol. 21, no. 7, pp. 1211–1218, 2005.

[17] R. Pal, A. Datta, and E. R. Dougherty, “Optimal infinite horizon control for probabilistic boolean networks,” *IEEE Trans. on Sig. Proc.*, vol. 54, pp. 2375– 2387, 2006.

[18] S. Kauffman, *The Origins of Order: Self-Organization and Selection in Evolution*. New York City, NY, USA: Oxford Univ. Press, 1993.

[19] I. Shmulevich, E. R. Dougherty, S. Kim, and W. Zhang, “Probabilistic boolean networks: A rule-based uncertainty model for gene regulatory networks,” *Bioinformatics*, vol. 18, pp. 261–274, 2002.

[20] I. Shmulevich, E. R. Dougherty, and W. Zhang, “From boolean to probabilistic boolean networks as models of genetic regulatory networks,” *Proc. of IEEE*, vol. 90, no. 11, pp. 1778–1792, 2002.

[21] M. Brun, E. R. Dougherty, and I. Shmulevich, “Steady-state probabilities for attractors in probabilistic boolean networks,” *Sig. Proc.*, vol. 85, no. 10, pp. 1993–2013, 2005.

[22] B. Faryabi, G. Vahedi, J.-F. Chamberland, A. Datta, and E. R. Dougherty, “Intervention in context-sensitive probabilistic boolean networks revisited,” *EURASIP J. Bioinfo. and Syst. Biol.*, p. 13, 2009.

[23] D. P. Bertsekas, *Dynamic Programming and Optimal Control*. Bellmont, MA, USA: Athena Scientific, 1995.

[24] P. A. Ioannou and J. Sun, *Robust Adaptive Control: A Unified Approach*. Englewood Cliffs, NJ, USA: Prentice Hall, 1996.

[25] P. R. Kumar and P. Varaiya, *Stochastic Systems Estimation, Identification and Adaptive Control*. Englewood Cliffs, NJ, USA: Prentice Hall, 1986.

[26] P. R. Kumar and W. Lin, “Optimal adaptive controller for unknown markov chains,” *IEEE Trans. on Automatic Control*, vol. 27, pp. 765–774, 1982.

[27] R. Pal, I. Ivanov, A. Datta, M. L. Bittner, and E. R. Dougherty, “Generating boolean networks with a prescribed attractor structure,” *Bioinformatics*, vol. 21, pp. 4021–4025, 2005.

[28] X. Zhou, X. Wang, and E. R. Dougherty, “Binarization of microarray data based on a mixture model,” *Mol. Cancer Therapy*, vol. 2, pp. 679–684, 2003.

[29] E. R. Dougherty and Y. Xiao, “Design of probabilistic boolean networks under the requirement of contextual data consistency,” *IEEE Trans. on Sig. Proc.*, vol. 54, pp. 3603– 3613, 2006.

[30] A. Choudhary, A. Datta, M. L. Bittner, and E. R. Dougherty, “Intervention in a family of boolean networks,” *Bioinformatics*, vol. 22, pp. 226–232, 2006.

[31] B. Faryabi, G. Vahedi, J.-F. Chamberland, A. Datta, and E. R. Dougherty, “Constrained intervention in a cancerous mammalian cell-cycle network,” *IEEE GENSIPS*, June 2008.

[32] X. Qian, I. Ivanov, N. Ghaffari, , and E. R. Dougherty, “Intervention in gene regulatory networks via greedy control policies based on long-run behavior,” *BMC Syst. Biol.*, vol. 3, no. 61, pp. 1–16, 2009.

[33] A. Faure, A. Naldi, C. Chaouiya, and D. Theiffry, “Dynamical analysis of a generic boolean model for the control of the mammalian cell cycle,” *Bioinformatics*, vol. 22, pp. 124–131, 2006.

[34] R. Pal, A. Datta, and E. R. Dougherty, “Robust intervention in probabilistic boolean networks,” *IEEE Trans. on Sig. Proc.*, vol. 56, no. 3, pp. 1280–1294, 2008.

[35] A. T. Weeraratna, Y. Jiang, G. Hostetter, K. Rosenblatt, P. Duray, M. Bittner, and J. M. Trent, “Wnt5a signalling directly affects cell motility and invasion of metastatic melanoma,” *Cancer Cell*, vol. 1, pp. 279–288, 2002.

[36] E. R. Dougherty and A. Datta, “Genomic signal processing: Diagnosis and therapy,” *IEEE Sig. Proc. Mag.*, vol. 22, pp. 107–112, 2005.

[37] A. Datta and E. R. Dougherty, *Introduction to Genomic Signal Processing with Control*. New York City, NY, USA: CRC Press, 2007.

[38] J. M. Bower and H. Bolouri, *Computational Modeling of Genetic and Biochemical Networks*. Boston, MA, USA: MIT Press, 2001.

[39] R. Sharan and T. Ideker, “Modeling cellular machinery through biological network comparison,” *Nat. Biotech.*, vol. 24, pp. 427–433, 2006.

[40] E. D. Conrad and J. J. Tyson, *System Modeling in Cell Biology from Concepts to Nuts and Bolts*. Cambridge, MA, USA: MIT Press, 2006.

[41] D. S. Douglas and S. Jones, *Differential Equations and Mathematical Biology*. London, UK: Chapman & Hall/CRC, 2003.

[42] N. Friedman, M. Linial, I. Nachman, and D. Pe’er, “Using bayesian networks to analyze expression data,” *Comp. Biol.*, vol. 7, no. 3-4, pp. 601–620, 2000.

[43] A. Hartemink, D. Gifford, T. Jaakkola, and R. Young, “Using graphical models and genomic expression data to statistically validate models of genetic regulatory networks,” *Pac. Symp. on Biocomp.*, pp. 422–433, 2001.

[44] K. Murphy and S. Mian, “Modelling gene expression data using dynamic bayesian networks,” 1999, (Technical Report, Computer Science Division, University of California, Berkeley, CA).

[45] K. Sachs, D. Gifford, T. Jaakkola, P. Sorger, and D. A. Lauffenburger, “Bayesian network approach to cell signaling pathway modeling,” *Sci. Signaling*, vol. 148, p. pe38, 2002.

[46] S. Kauffman, “Metabolic stability and epigenesis in randomly constructed genetic nets,” *J. Theor. Biol.*, vol. 22, pp. 437–467, 1969.

[47] L. Glass and S. Kauffman, “The logical analysis of continuous, nonlinear biochemical control networks,” *J. Theor. Biol.*, vol. 39, pp. 103–129, 1973.

[48] R. Thomas, “Boolean formalization of genetic control circuits,” *J. Theor. Biol.*, vol. 42, no. 3, pp. 563–585, 1973.

[49] R. Thomas and R. D'ari, *Biological Feedback*. Boca Raton, Fl, USA: CRC, 1990.

[50] R. Thomas, "Regulatory networks seen as asynchronous automata: A logical description," *J. Theor. Biol.*, vol. 153, pp. 1–23, 1991.

[51] R. Thomas, A.-M. Gathoye, and L. Lambert, "A complex control circuit: Regulation of immunity in temperate bacteriophages," *Euro. J. Biochem.*, vol. 71, pp. 211–227, 1976.

[52] R. Thomas, "Logical analysis of systems comprising feedback loops," *J. Theor. Biol.*, vol. 73, pp. 631–656, 1978.

[53] M. K. Morris, J. Saez-Rodriguez, P. K. Sorger, and D. A. Lauffenburger, "Logic-based models for the analysis of cell signaling networks," *Biochemistry*, vol. 49, pp. 3216–3224, 2010.

[54] D. Thieffry and D. Romero, "The modularity of biological regulatory networks," *Biosystems*, vol. 50, pp. 49–59, 1999.

[55] E. Dubrova and M. Teslenko, "A sat-based algorithm for computing attractors in synchronous boolean networks," *ArXive ePrint Tech. Report*, vol. 0901.4448, pp. 1–9, 2009.

[56] A. Datta, R. Pal., A. Choudhary, and E. R. Dougherty, "Control approaches for probabilistic gene regulatory networks," *IEEE Sig. Proc. Mag.*, vol. 24, no. 1, pp. 54–63, 2007.

[57] G. Vahedi, B. Faryabi, J.-F. Chamberland, A. Datta, and E. R. Dougherty, "Optimal intervention strategies for cyclic therapeutic methods," *IEEE Trans. on Biomed. Engg.*, vol. 56, pp. 281–291, 2009.

[58] B. Faryabi, G. Vahedi, J.-F. Chamberland, A. Datta, and E. R. Dougherty, “Optimal constrained intervention in genetic regulatory networks,” *EURASIP J. Bioinfo. and Syst. Biol.*, vol. 2008, pp. 1–10, 2008.

[59] B. Faryabi, J.-F. Chamberland, G. Vahedi, A. Datta, and E. R. Dougherty, “Optimal intervention in asynchronous genetic regulatory networks,” *IEEE J. of Selected Topics in Sig. Proc.*, vol. 2, pp. 412–423, 2008.

[60] B. Faryabi, G. Vahedi, J. F. Chamberland, A. Datta, and E. R. Dougherty, “Intervention in context-sensitive probabilistic boolean networks revisited,” *EURASIP J. Bioinfo. and Syst. Biol.*, vol. 2009, p. 13 pages, 2009.

[61] G. Vahedi, B. Faryabi, J.-F. Chamberland, A. Datta, and E. R. Dougherty, “Intervention in gene regulatory networks via a stationary mean-first-passage-time control policy,” *IEEE Trans. on Biomed. Engg.*, vol. 55, no. 10, pp. 2319–2331, 2008.

[62] T. Akutsu, M. Hayashida, W. K. Ching, and M. K. Ng, “Control of boolean networks: Hardness results and algorithms for tree structured networks,” *J. Theor. Biol.*, vol. 244, no. 4, pp. 670–679, 2007.

[63] M. K. Ng, S. Q. Zhang, W. K. Ching, and T. Akutsu, “A control model for markovian genetic regulatory networks,” *Trans. on Comp. Syst. Biol.*, vol. 4070, pp. 36–48, 2006.

[64] W. K. Ching, S. Q. Zhang, Y. Jiao, T. Akutsu, N. K. Tsing, and A. S. Wong, “Optimal control policy for probabilistic boolean networks with hard constraints,” *IET Syst. Biol.*, vol. 3, pp. 90–99, 2009.

[65] Y. Xiao and E. R. Dougherty, “The impact of function perturbations in boolean networks,” *Bioinformatics*, vol. 23, pp. 1265–1273, 2007.

[66] M. Karnaugh, “The map method for synthesis of combinational logic circuits,” *AIEE Sum. Gen. Meet.*, pp. 593–599, 1953.

[67] J. Millman, H. Taub, and M. S. P. Rao, *Pulse, Digital and Switching Waveforms*, 2nd ed. New Delhi, India: McGraw-Hill, 2007.

[68] W. Abou-Jaoud, D. Ouattara, and M. Kaufman, “From structure to dynamics: Frequency tuning in the p53-mdm2 network. i. logical approach,” *J. Theor. Biol.*, vol. 258, pp. 561–577, 2009.

[69] D. Ouattara, W. Abou-Jaoude, and M. Kaufman, “From structure to dynamics: Frequency tuning in the p53mdm2 network. ii. differential and stochastic approaches,” *J. Theor. Biol.*, vol. 264, pp. 1177–1189, 2010.

[70] C. Bakkenist and M. Kastan, “Dna damage activates atm through intermolecular autophosphorylation and dimer dissociation,” *Nature*, vol. 421, pp. 499–506, 2003.

[71] R. Bar-Or, R. Maya, L. Segel, U. Alon, A. Levine, and M. Oren, “Generation of oscillations by the p53-mdm2 feedback loop: A theoretical and experimental study,” *Proc. Natl. Acad. Sci.*, vol. USA 97, pp. 11 250–11 255, 2000.

[72] S. Bottani and B. Grammaticos, “Analysis of a minimal model for p53 oscillations,” *J. Theor. Biol.*, vol. 249, pp. 235–245, 2007.

[73] A. Ciliberto, B. Novak, and J. Tyson, “Steady states and oscillations in the p53/mdm2 network,” *Cell Cycle*, vol. 4, pp. 488–493, 2005.

[74] G. Lahav, N. Rosenfeld, A. Sigal, N. G. Zatorsky, A. J. Levine, M. B. Elowitz, and U. Alon, “Dynamics of the p53-mdm2 feedback loop in indivisual cells,” *Nature Genetics*, vol. 36, no. 2, pp. 147–150, 2004.

[75] O. Kranenburg, “The kras oncogene: past, present, and future.” *Biochem. Biophys. Acta.*, vol. 1756, pp. 81–82, 2005.

[76] P. C. Nowell and D. A. Hungerford, “A minute chromosome in human chronic granulocytic leukemia,” *Science*, vol. 132, no. 3438, p. 1497, 1960.

[77] M. Abramovici, M. A. Breuer, and A. D. Friedman, *Digital Systems Testing and Testable Design*. New York City, New York, USA: IEEE Press, 1990.

VITA

RITWIK KUMAR LAYEK

ADDRESS:

Genomic Signal Processing Laboratory
9 Zachry Engineering Center, TAMU 3128,
College Station, TX-77843-3128, USA.

EMAIL:

ritwik@neo.tamu.edu

INTERESTS:

Systems biology, cancer biology, linear control theory, digital systems modeling, stochastic processes, experimental genomics, digital image processing.

EDUCATION:

- PhD Student in Electrical Engineering, Texas A&M University, College Station, USA (2007-Present). current CGPA: 3.78 (out of 4).
- Master of Technology, Indian Institute of Technology, Kharagpur, India (2006-2007) . M.Tech CGPA: 9.19 (out of 10), Specilization: Automation and computer vision.
- Bachelor of Technology(Hons.), Indian Institute of Technology, Kharagpur, India (2002-2006). Total CGPA: 8.45 (out of 10).

Publications:

Journal publications: 3

Conference publications: 7

The typist for this dissertation was Ritwik Kumar Layek.