CUSTOMIZATION OF TREATMENT FOR CANCER PATIENTS: AN ENGINEERING APPROACH

A Dissertation

by

BIBHU PRASAD MISHRA

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

DOCTOR OF PHILOSOPHY

Chair of Committee,	Aniruddha Datta
Committee Members,	V. S. Venkatraj
	P. R. Kumar
	Byung-Jun Yoon
Head of Department,	Miroslav M. Begovic

August 2018

Major Subject: Electrical Engineering

ABSTRACT

Cancer is a disease associated with uncontrolled cell proliferation or reduced cell death, either of which can lead to tumorigenesis. A possible route through which cancer can develop is by breakdowns in the signaling cascade of proteins at the cellular level. Since there are many ways in which such breakdowns can occur, anti-cancer chemotherapeutic drugs show varying degrees of efficacy in different patients. Thus, there is an urgent need to personalize the drug treatment regimen for better response to treatment while trying to reduce the side effects of these drugs. One way to meet this need would be to try every possible drug combination on cell lines extracted from a patient and find the combination with the least number of drugs in the mix but providing the best possible output. Although this method may work it is tedious and time consuming as the number of combinations increase exponentially with every new drug that is introduced into the repertoire.

First, we consider the problem where the tumor is homogeneous in nature but the mutations within the mutated cells are unknown. We use Boolean network models with monotonicity properties to reduce the number of test cases, while still getting the best possible combination with the least number of drugs in the mix. This approach is efficient both in terms of time required and the costs involved. This method has also been applied to both simulated and real-world data collected from fibroplasts using qPCR to demonstrate the usefulness of the method.

Another important area of study in cancer research concerns the heterogeneous nature of tumors. The clonal evolution of tumors is the driving force leading to heterogeneity in cancer tissues. Thus, in order to customize the treatment of cancer we need to be able to better model the heterogeneous subpopulations in the tumor. This can be done by estimating the impact of the various sub-populations and by modeling the interplay of various sub-populations within the heterogeneous tumor. Prior works in the literature have already addressed the problems of estimating the proportion of the sub-populations within a tumor and of modeling the interaction between the various sub-populations. In this work we present a way to improve the accuracy of the Bayesian

hierarchical model which helps in estimating the proportional breakup of the tumor population. Additionally, it looks at ways to use the knowledge of the proportional breakup of tumor subpopulations and the interplay between the various subpopulations to help customize the treatment for the patient by making use of evolutionary game theory. We demonstrate the improvement of the presented methods as compared to the existing Bayesian hierarchical model by applying these techniques to qPCR and fluorescent data.

Finally, the problem becomes more challenging when the nature and the number of the subpopulations are variable and difficult to estimate. In this work, we present a feasible way to find the best possible drug combination for such a scenario by training two neural network models on synthetic and real-world cancer data. Then we test each model, to verify their effectiveness and to demonstrate their usefulness in choosing the appropriate combination therapy. The models were evaluated on synthetic qPCR data and fluorescent data obtained from experiments.

The results obtained from these methods take us a step closer to the realization of customized treatment for cancer patients. This will not only make the treatment more effective but also help reduce the side effects of the drug treatment.

DEDICATION

To my family without whose unconditional love and support this would not have been possible

ACKNOWLEDGMENTS

First and foremost, I thank and bow to God for helping me through all these wonderful people in my life, for guiding me in more ways than I can fathom and whose blessings helped me achieve my PhD. I express my deepest gratitude to my advisor Dr. Aniruddha Datta who gave me the opportunity to work on the topic of customization of cancer therapy for patients. I would also like to thank him for his patient guidance, understanding, valuable time and support during my stay at Texas A&M University which made this dissertation possible. I would like to thank Dr. Aniruddha Datta and Dr. Ana Goulart for supporting me financially as a research assistant, a teaching assistant and as a lecturer during the course of my doctoral program. I thank Dr. Vijayanagaram Venkatraj for his time and guidance he provided for conducting the wet-lab experiments and also helping me as a committee member. I sincerely thank Dr. P.R. Kumar and Dr. Byung-Jun Yoon for their useful suggestions and for taking time out of their busy schedule to serve on my committee. I am very thankful to Dr. Chao Sima, Dr. Jianping Hua, Dr. Rosana Lopes and Dr. Michael Bitner for providing me with the experimental data for carrying out my research. I would like to convey my sincere gratitude to Dr. N. Sivakumar, Dr. Peter Howard, Dr. Bojan Popov and my doctoral committee for being extremely supportive in helping me to obtain a MS in Mathematics during the course of my doctoral program. I would also like to thank all the professors, whose classes I took during my stay at Texas A&M University, for providing me with a valuable and exciting learning experience. I sincerely thank all my teachers, since my childhood, whose guidance, knowledge and support has always propelled me further in life.

I remember my late paternal grandfather, Nilamber Mishra and my late maternal grandfather, Dibyasingh Mishra, in gratitude for their blessings, for being a reason for me to pursue a PhD and who I know would have been the happiest people on earth if they were here today. I also remember my late paternal grandmother, Shantilata Mishra, for her love and blessings and also thank my maternal grandmother, Kuntala Mishra, for her constant support and blessings. I am eternally indebted to my parents, Jagat Ballav Mishra and Swarnaprava Mishra, for their unconditional

love, unflinching support, valuable guidance and countless blessing without which my dissertation would have been impossible. I thank my maternal uncle, Dr. Kailash Misra, and aunt, Suprava Misra, for their constant support and guidance during my stay in the US. I would also like to thank my parents-in-law, Rama Krushna Nayak and Snehalata Nayak for their blessings and support, my brother-in-law Srivatsa Nayak for his good wishes and suggestions during these final years of my PhD. I would like to thank my wife, Sharmistha Nayak, for coming into my life as a loving companion and patiently supporting my endeavor during these final years of my PhD. I thank my younger sister, Ipsa Mishra, for always being a cheerful and supportive presence in my life.

I thank my fellow students, friends and seniors, Dr. Anwoy Kumar Mohanty, Dr. Ritwik Layek, Sanat Pandey, Nityendra Singh, Dibyendu Rana, Dr. Sriram Sridharan, Dr. Osama Arshad and Dr. Priyadarshini Venkat, for the interesting discussions and memories. I also thank all my students whom I had the opportunity to teach, as a teaching assistant and as a lecturer at Texas A&M University, whose positive reviews also encouraged me to better myself. Last but not the least, I would like to take this opportunity to thank the current and past staff members at the Department of Electrical and Computer Engineering (Tammy Carda, Katie Bryan, Anni Brunker and Melissa Sheldon), at the Engineering Technology and Industrial Distribution (Rosanne Gueguen, Emma Carrigan and Annalyse Escalante), at the Department of Mathematics (Monique Stewart) and all the other staff members associated with Texas A&M University who have helped or will help me in dealing with the various administrative processes during my stay at Texas A&M University and beyond that.

CONTRIBUTORS AND FUNDING SOURCES

Contributors

This work was supported by a dissertation committee consisting of Dr. Aniruddha Datta [advisor], Dr. P. R. Kumar and Dr. Byung-Jun Yoon of the Department of Electrical and Computer Engineering and Dr. V. S. Venkatraj of the Department of Veterinary Integrative Biosciences.

The qPCR data used for the dissertation was obtained with the help of Dr. V. S.Venkatraj and Dr. Anwoy Kumar Mohanty. The fluorescence data utilized in this work was provided by Dr. Chao Sima, Dr. Jianping Hua, Dr. Rosana Lopes and Dr. Michael Bitner.

All other work conducted for the dissertation was completed by the student independently.

Funding Sources

Graduate study was supported by a research assistantship from the National Science Foundation under grant ECCS-1404314 and by teaching assistantship, graduate assistant lectureship at the Department of Engineering Technology & Industrial Distribution.

NOMENCLATURE

MCMC	Markov Chain Monte Carlo
EGT	Evolutionary Game Theory
MSE	Mean Squared Error
CEE	Cross-Entropy Error
TP	True Positives
FP	False Positives
TN	True Negatives
FN	False Negatives
ROC	Receiver Operating Characteristic
MCC	Matthews Correlation Coefficient
DCA	Drug Combination Accuracy
DPA	Drug Prediction Accuracy
MAPK	Mitogen-Activated Protein Kinase
GF	Growth Factor
qPCR	Quantitative Polymerase Chain Reaction
FBS	Fetal Bovine Serum
RNA	Ribonucleic Acid
DNA	Deoxyribonucleic Acid
mRNA	Messenger Ribonucleic Acid
EGF	Epidermal Growth Factor
HBEGF	Heparin-Binding EGF-like Growth Factor
IGF	Insulin-like Growth Factor

NRG1	Neuregulin 1
PTEN	Phosphatase and Tensin Homolog
IGFR	Insulin-like Growth Factor 1 Receptor
IRS1	Insulin Receptor Substrate 1
SP1	Specificity Protein 1
SRF	Serum Response Factor
GAPDH	Glyceraldehyde 3-Phosphate Dehydrogenase

TABLE OF CONTENTS

Page

LIST OF FIGURES

FIGURE Page

LIST OF TABLES

TABLE Page

1. INTRODUCTION

1.1 Background

In multi-cellular organisms different cells work in harmony to sustain life. Each cell serves a specific purpose in the organism after which, in most cases, it dies via the process of programmed cell death. The coordination between different types of cells in an organism is brought about by various types of extrinsic and intrinsic signaling in the cells. Extrinsic signaling or communication between cells can be mechanical, electrical or biochemical in nature. Mechanical signaling is due to forces exerted on the cell or produced by the cell which in turn can be sensed and responded to by the other cells. Electrical signaling is usually carried out by nerve cells via transmission of action potentials whereas biochemical signaling involves various biomolecules such as proteins, lipids, ions and gases. Proteins for example, by virtue of their amino acid sequence and spatial conformation, transmit highly specific signals to the desired target. Generally, the somatic cells in our body depend on proteins for signaling. The chain of sequential biochemical protein interactions within the cell, usually triggered by an extrinsic signal received by the cell receptor, forms what is known as a signaling pathway. A breakdown in signaling can lead to various diseases. Such breakdowns can occur due to mutations in the exon region of the genes, chromosomal rearrangements within the DNA sequence or due to viral infections. These breakdowns can in turn lead to the production of faulty proteins which can inhibit proper signal transduction. Not all breakdown in signaling would manifest itself as a change in the phenotypic behavior of the cell, due to the inherent redundancy present in the cell's signaling mechanism such as feedbacks, alternate signaling pathways, etc.. However, when these breakdowns cause the cell to divide rapidly and/or inhibit the apoptosis of cells, it leads to growth of tumor. Thus, in cancer patients, the tumor cells tend to have a faulty signaling which causes them to rapidly divide and/or not follow the path of programmed cell death known as apoptosis [1, 2]. Hence this leads to cancer cells using up the resources, earmarked for other cells, to grow in a limitless fashion, ultimately leading to the death of the organism. There

are various anti-cancer drugs which target different proteins along the signaling pathway but as the nature of cancer varies from one person to another the same drug combination may perform well for a particular patient and not at all in the case of another. Thus, there is a critical need to tailor the drug treatment regimen to the individual patient, leading to what is commonly referred to as personalized therapy. The goal in this work has been to identify the best possible drug combination which helps to control the growth rate of the tumor in general and with reduced side effects. To achieve reduced side effects, we simply try and find the smallest subset of drugs which is most efficient in slowing down tumor growth. We primarily focus on cancer caused by mutations on well-known signaling pathways in this work, but the specific nature of the problem dealt with are described in greater detail [3, 4, 5] in the sections 2, 3 and 4. In subsection 1.2 we look at the organization of this work.

1.2 Organization

In order to help customize the treatment for cancer patients we start by breaking up the problem into various subproblems. Thus, we consider the four different scenarios with respect to the nature of mutations within the tumor cells. First, we consider the scenario where the tumor consists of a homogeneous subpopulation of cancer cells and the nature of mutations are known. In such a case the problem of identifying the drugs for treatment becomes trivial as we simply use the drug combination that gives us the best results by making use of signaling pathway knowledge. Thus, we do not invest our efforts on this problem but rather focus on solving the remaining three scenarios.

Second, we consider the scenario where the tumor is homogeneous in nature [3] but the nature of the mutations is unknown. This is dealt with in section 2. In this case we look at signaling pathways of a particular type. The pathways dealt with should satisfy the condition of monotonicity. To solve this problem, we make use of the properties of monotonic Boolean networks which are described in the next section.

Third, we consider the subproblem where the tumor is heterogeneous in nature [4] but the nature of the various subpopulations is known. In this case we look at the observation data and

estimate the relative proportion of the various subpopulations that are present within the cell. We make use of a Bayesian hierarchical model to achieve this. Once we have the relative ratios of the different subpopulations we identify the target subpopulation for drug therapy by making use of evolutionary game theory. The problem is described and dealt with in section 3.

Fourth, in section 4 we look at the situation where the total number of subpopulations within the heterogeneous tumor is unknown and/or the nature of the mutations in some or all of the subpopulations is unknown [5]. This is dealt with by making use of feedforward neural networks which are trained on previously known patient data.

Finally, in the section 5 we conclude this work by summarizing our work and looking at possible avenues for future research so as to bring the dream of customizing cancer treatment for patients closer to reality.

2. BOOLEAN NETWORKS WITH MONOTONIC PROPERTIES AND THEIR UTILIZATION IN THE EFFICIENT SELECTION OF KINASE-INHIBITOR COMBINATION THERAPIES FOR THE TREATMENT OF CANCER

2.1 Introduction

Designing proper drug treatment for cancer requires the understanding of the overall multivariate nature of protein-protein interactions instead of focusing on their simple pair wise interactions. The extreme complexity of studying multivariate protein-protein interactions has historically motivated biologists to look only at the marginal relationships between interacting proteins. Thus, to develop a proper understanding of cellular behavior, we need to develop multivariate models consistent with known prior marginal relationships between interacting proteins. There are various techniques in the literature [6] which are used to model the signaling behavior of cells. Differential equations based on the law of mass action [6, 7, 8], for example, provides a general framework to model signaling pathways. By making certain assumptions one can model the cellular signaling by a system of non-linear ordinary differential equations (ODEs) in variables representing protein and mRNA concentrations. However, for a signaling pathway, which involves a large number of proteins, to write down such equations using the mass action models we need to be able to predict the rate constants for the various biochemical reactions which occur in the pathway. This requires us to obtain a lot of time course data for the various proteins in the pathway. Also, as we are dealing with mutated pathways in cancer cells some portion of the pathway may behave unexpectedly due to mutations and thus the system would not be deterministic as is normally assumed for such models. To deal with this we can make use of probabilistic methods which makes use of the Langevin approach [6] which also adds a noise term to the differential equations to account for the non-deterministic nature of the pathways. We can also make use of the Fokker-Planck approach [6, 9] which involves equations describing the changes in the probabilistic distribution of protein states (i.e. concentration of proteins) as a function of time. These PDEs are extremely difficult to

solve as compared to ODEs [6]. Also, as these are probabilistic models, lots of data points would be needed so as to reduce the error in prediction.

Before choosing the model, we need to ask ourselves a few questions such as whether we are interested in the time course behavior of the pathway or whether we are interested in the equilibrium state of the pathway. In our case we are concerned with the equilibrium state of the outputs when drugs are applied. Another goal in our case is to predict the behavior of pathway with a minimal number of data points so that the process is time efficient and cost efficient. These requirements rule out the use of statistical or differential models for modeling the signaling pathway. What we need to predict is if a reporter protein for the signaling pathway is up regulated or down regulated and thus there are two possible states for a protein/gene. This is nicely represented by a discrete Boolean model. A method using such a Boolean model was presented in [10]. That paper used digital logic to understand the mechanics of a signaling pathway and to map that information into a Boolean network. Various other works on cellular signaling also have used Boolean networks [11, 12, 13, 14] to model the protein signaling within the cells. The reason behind using Boolean network models is that many proteins/genes inside the cell exhibit an ON/OFF switch like behavior [15] which can be easily modeled using the binary states 0 and 1. Subsequently, in [16] such a network was used to categorize mutations in genes/proteins of the corresponding pathway and predict the effect of different kinase-inhibitor combination therapies administered to the patient. The work in [16] considered only the single mutation case but in most cancers, the signaling pathway could have multiple mutations. This motivates the topic of this section, which is to try and develop a method to efficiently arrive at effective combination therapies when there are multiple mutations in the signaling pathway. The main objective of the section is to help reduce the labor associated with testing the effect of drug combinations on tumor tissue, in the process of identifying the best drug combination for the patient. Thus, our primary emphasis is on reducing the number of drug combinations which needs to be manually tested before the best drug combination can be identified. The approach makes use of certain monotonicity properties that can be

associated with Boolean networks used to model signaling pathways. [1] In this section we focus on the well-known MAPK signaling pathway and the mutations of the proteins/genes associated with it. We do not consider the behavior of the MAPK signaling pathway in the presence of viral infections or in the case of chromosomal rearrangement.

In the following subsections we start off by first investigating the monotonic properties of fixed Boolean networks. Then, we show how these properties can be utilized to make an efficient selection of customized combination therapy. This is followed by the presentation of an algorithm to infer the best possible drug combination. To bring out the usefulness of our method, it has been applied to synthetic data generated from the prior knowledge of MAPK network. The method has also been applied to experimental data collected by applying some of the drug combinations on normal adult fibroplast cell lines. To test the success of our method we simply see if our method is able to identify the best possible drug combination with least number of drugs as mentioned earlier, which it does. We then discuss the results and talk about the possibilities for the improvement of the algorithm in the future followed by a concluding statement.

2.2 Fixed monotonic Boolean networks

To arrive at Boolean networks with monotonicity properties, we first represent signaling pathways as digital networks following the approach introduced in [16]. Here, a protein which is active is represented as having state 1 while one which is inactive is represented as having a state 0. Mutations can either cause a protein to become constitutively active or render a protein inactive. Such anomalies can be represented as stuck-at-faults [17] when the pathway is mapped into a Boolean network. Signaling breakdowns which introduce new signaling mechanisms to existing pathways can be represented as bridging faults [17] in the corresponding Boolean network. In this section, we will consider only signaling pathways without any feedback and the only faults considered will be of the stuck-at fault type. Furthermore, we will assume that only one particular combination of mutations has occurred in the cells making up the cancerous tissue; in other words, the cancerous

[1] Note that in the MAPK signaling diagram shown in this section, there are various proteins such as PTEN, AKT, mTOR etc. which are not explicitly part of MAPK pathway but have been shown. This is because they are known to interact with the MAPK pathway.

tissue is homogeneous.

These assumptions serve as a starting point for dealing with the issue of identifying the best possible drug combination for a patient. Now, in terms of digital logic, we can think of cancer as being caused by a combination of various faults in the signaling pathway. In reality a signaling pathway can have more than one cancer causing stuck-at-fault or bridging faults with various sub-populations of cells harboring different combinations of mutations. Thus, in reality, a cancerous tissue will be composed of a heterogeneous population of cancer cells. However, as mentioned earlier, we will consider the case where there is only one kind of sub-population of cancer cells with multiple stuck-at-faults. Our aim is to arrive at a combination of cancer drugs from an existing list of protein kinase inhibitors, which are known to target the signaling pathway at various locations, such that any particular patient's cancer can be treated optimally both in terms of reduced side effects [18, 19] and maximal suppression of the effect of disruptive mutations. In other words, our goal is to find a drug combination with the best possible output and the least number of drugs. For illustrative purposes, we will be considering the Mitogen-activated protein kinase (MAPK) signaling pathway. The MAPK pathway has been considered here as a mutation along that pathway is normally a necessary step in the development of tumors [20, 21].

Before plunging full swing into the solution of the problem we first take an intuitive look at it from the point of view of signaling pathway knowledge. We know certain mutations, for example Ras mutation in MAPK [22], are associated with cancer. In terms of digital logic this means that a stuck-at-one fault at a point in the MAPK digital diagram corresponding to Ras will always try to promote cancer which means activating downstream proteins and transcription factors regardless of whether the upstream proteins are activated or not. A protein kinase inhibitor at a proper target on the other hand would always try to suppress the cancer by preventing downstream proteins and transcription factors from being activated by the mutated protein.

Looking at this from a digital standpoint, we see that if we drive Ras from state 0 to state 1 (i.e. mutation) it would cause downstream transcription factors such as SP1 to be driven from 0 to 1 or stay at the same state but never allow them to go from state 1 to state 0. Similarly applying a drug

such as U0126 which targets location MEK1 will drive MEK1 from state 1 to state 0 which in turn will always cause SP1 to go from state 1 (active) to state 0 (inactive) or stay unchanged but never allow it to go from state 0 to state 1. Based on this line of reasoning, we see that there is a kind of monotonic relationship between fixed upstream and downstream proteins/transcription factors regardless of the activation status of other proteins. This motivates us to take a deeper look into the structure of Boolean networks, which we do next.

When representing a signaling pathway as a Boolean network, all the interactions in the pathway will be represented using only primary gates, universal gates and buffers. We now proceed to study the properties of networks constructed out of these modules.

2.2.1 Properties of fixed monotonic Boolean networks

The concept of monotonicity in Boolean circuits has been utilized in the context of different topics [23, 24] such as reliability theory, theory of complexity and computation models, cancer diagnosis [25], etc.. Authors working on different topics tend to use a variety of ways to define monotonicity [24] to suit their needs. Thus, to enhance the clarity of presentation, in the context of cellular signaling, we begin by introducing some terminology and notation of our own. Let $\aleph$ be a Boolean network. Define the network class $\aleph\zeta$ to be made up of all possible networks that can be obtained by introducing stuck-at-faults at various locations in network $\aleph$, with the no-fault case included in the class. Note that stuck-at zero and stuck-at-one faults lead to different networks in the network class $\aleph\zeta$. We see that $\aleph \, \epsilon \, \aleph\zeta$ and we call it the *primary network* of class $\aleph\zeta$ as it has no variables with fixed values unlike other networks of the same class which have at least one variable forcibly held at a fixed value. For any variable V we represent a change of value of that variable from 0 to 1 as $0 \rightarrow 1$ and from 1 to 0 as $1 \rightarrow 0$. So, if V goes from state 0 to 1 we write it as $V : 0 \rightarrow 1$. Similarly, we write $V : 1 \rightarrow 0$ if the value of V goes from state 1 to 0. A *network scenario* is simply defined as a network belonging to the network class $\aleph\zeta$ with fixed values (0 or 1) at the locations corresponding to the inputs of the primary network $\aleph$. Different combination of input values for a given network in $\aleph\zeta$ lead to different network scenarios. For a network $\aleph_i$ belonging to $\aleph\zeta$ we say a location is a *clean location* if there are no stuck-at-faults in the location

or else the location is a *faulty location*.

We consider logic variables X and Y with X upstream and Y downstream in the network $\aleph$. If forcing $X : 0 \to 1$ causes $Y : 0 \to 1$ in at least one network scenario, while letting Y to stay unchanged or go from 0 to 1 in other network scenarios, then we represent it by $X \uparrow Y$. If the same transition in X causes $Y : 1 \to 0$ in at least one network scenario while letting Y remain unchanged or go from 1 to 0 in other network scenarios, we represent it by $X \downarrow Y$. If a change in the value of X does not cause a change in the value of Y in any of the network scenarios, then we represent it by $X|Y$ which means Y is independent of changes in X. This is the case when Y does not lie downstream of X. Note that when Y is downstream of X then it is also possible to have $X|Y$ depending on the structure of the network.

For a network $\aleph$ and a pair of variables X and Y in the network which are upstream and downstream respectively of each other, if there exists either a monotonic relationship $X \uparrow Y$ or $X \downarrow Y$, when Y is not independent of X, we call it a *fixed monotonic relationship* between X and Y. If upstream variable X and downstream variable Y are related either as $X \uparrow Y$ or as $X|Y$ then we represent it by $X \Uparrow Y$. Similarly, we define $X \Downarrow Y$ as when X and Y are related either as $X \downarrow Y$ or as $X|Y$. Thus $X \Uparrow Y$ and $X \Downarrow Y$ can be thought of as weaker versions of fixed monotonic relationship between X and Y where the requirement that for at least one network scenario Y changes appropriately in response to $X : 0 \to 1$ is relaxed.

If all upstream-downstream variable pairs (X, Y) where Y is dependent on X share a fixed monotonic relationship, then the network $\aleph$ is called a *fixed monotonic network*. Note that in a fixed monotonic network if upstream-downstream variable pairs (X, Y) does not satisfy $X \uparrow Y$ then $X \Downarrow Y$. Similarly if (X, Y) does not satisfy $X \downarrow Y$ then $X \Uparrow Y$.

All throughout the section Σ will denote the logical 'OR' operation involving two or more variables while Π will denote the logical 'AND' operation.

Property 1:

Statement: *The primary gates (including buffer) and universal gates introduce fixed monotonic relationships between input variables and the output variable as shown in Table 2.1.*

Table 2.1: Relationship between input and output for primary gates and universal gates.

Gates	Relationship between input X and output Y
OR	$\uparrow$
AND	$\uparrow$
Buffer	$\uparrow$
NOT	$\downarrow$
NOR	$\downarrow$
NAND	$\downarrow$

Proof: Consider a primary gate such as 'AND' or 'OR'. For one of the inputs X and the output Y we see that $Y = X.\prod_{w \neq X} w$ or $Y = X.\sum_{w \neq X} w$ depending on whether it is an 'AND' or 'OR' respectively, where w represents inputs to the gate. For an 'AND' gate, if $\prod_{w \neq X} w = 1$ then for $X : 0 \rightarrow 1$ we have $Y : 0 \rightarrow 1$ and for $\prod_{w \neq X} w = 0$ we have Y remains unaltered at state 0. Thus, we can say $X \uparrow Y$ when X is one of the inputs of the 'AND' gate and Y is the output. Similarly for the 'OR' gate, if we have $\sum_{w \neq X} w = 0$ then for $X : 0 \rightarrow 1$ we have $Y : 0 \rightarrow 1$ and for $\sum_{w \neq X} w = 1$ we have Y remains unaltered at state 1. Thus, again we can say that $X \uparrow Y$ when X is one of the inputs of the 'OR' gate and Y is the output. Now consider a 'NOT' gate or a simple buffer. For a 'NOT' gate we can see that for $X : 0 \rightarrow 1$ we have $Y : 1 \rightarrow 0$. Hence $X \downarrow Y$ when X is the input of a 'NOT' gate and Y is the output. In a buffer, for $X : 0 \rightarrow 1$ we have $Y : 0 \rightarrow 1$. Therefore $X \uparrow Y$ when X is the input of a buffer and Y is the output. Using an argument similar to that used for 'AND' and 'OR' we can see that for universal gates 'NAND' and 'NOR' we have $X \downarrow Y$. Table 2.1 presents the relationship between input X and output Y for various primary gates and the universal gates. Note that such a fixed monotonic relationship would not hold if 'XOR' (exclusive or) was used instead of primary gates or the universal gates.

Definition: *The gates for which $X \uparrow Y$, given input X and output Y, will be referred to as the* positive gates. *Similarly the gates for which $X \downarrow Y$ will be referred to as the* negative gates.

Property 2:

Let there be three variables X, Y and Z in network $\aleph$ such that X is upstream of Y and Y is upstream of Z. Also, all the paths from X to Z pass through Y. Now we consider five possible

cases.

Statement (a): *If* $X \uparrow Y$ *and* $Y \uparrow Z$ *then* $X \Uparrow Z$.

Proof: If $X : 0 \rightarrow 1$ then $Y : 0 \rightarrow 1$ or Y is unaltered. When $Y : 0 \rightarrow 1$ then $Z : 0 \rightarrow 1$ or Z is unaltered. If Y is unaltered then Z remains unaltered. Thus, we can see that either $X \uparrow Z$ or $X|Z$. Note that $X|Z$ can occur when no common network scenarios exist where $X : 0 \rightarrow 1$ leads to $Y : 0 \rightarrow 1$ and $Y : 0 \rightarrow 1$ leads to $Z : 0 \rightarrow 1$.

Statement (b): *If* $X \uparrow Y$ *and* $Y \downarrow Z$ *then* $X \Downarrow Z$.

Proof: If $X : 0 \rightarrow 1$ then $Y : 0 \rightarrow 1$ or Y is unaltered. When $Y : 0 \rightarrow 1$ then $Z : 1 \rightarrow 0$ or Z is unaltered. If Y is unaltered then Z remains unaltered. Thus, we can see that either $X \downarrow Z$ or $X|Z$. Note that $X|Z$ can occur when no common network scenarios exist where $X : 0 \rightarrow 1$ leads to $Y : 0 \rightarrow 1$ and $Y : 0 \rightarrow 1$ leads to $Z : 1 \rightarrow 0$.

Statement (c): *If* $X \downarrow Y$ *and* $Y \downarrow Z$ *then* $X \Uparrow Z$.

Proof: If $X : 0 \rightarrow 1$ then $Y : 1 \rightarrow 0$ or Y is unaltered. When $Y : 1 \rightarrow 0$ then $Z : 0 \rightarrow 1$ or Z is unaltered. If Y is unaltered then Z remains unaltered. Thus, we can see that either $X \uparrow Z$ or $X|Z$. Note that $X|Z$ can occur when no common network scenarios exist where $X : 0 \rightarrow 1$ leads to $Y : 1 \rightarrow 0$ and $Y : 1 \rightarrow 0$ leads to $Z : 0 \rightarrow 1$.

Statement (d): *If* $X \downarrow Y$ *and* $Y \uparrow Z$ *then* $X \Downarrow Z$.

Proof: If $X : 0 \rightarrow 1$ then $Y : 1 \rightarrow 0$ or Y is unaltered. When $Y : 1 \rightarrow 0$ then $Z : 1 \rightarrow 0$ or Z is unaltered. If Y is unaltered then Z remains unaltered. Thus, we can see that either $X \downarrow Z$ or $X|Z$. Note that $X|Z$ can occur when no common network scenarios exist where $X : 0 \rightarrow 1$ leads to $Y : 1 \rightarrow 0$ and $Y : 1 \rightarrow 0$ leads to $Z : 1 \rightarrow 0$.

Statement (e): *If* $X|Y$ *or* $Y|Z$ *then* $X|Z$.

Proof: If $X|Y$ then $X : 0 \rightarrow 1$ does not affect Y. If Y is unaltered then Z is unaltered regardless of the relationship between Y and Z. Similarly, if $Y|Z$ then $Y : 0 \rightarrow 1$ does not affect Z. Thus, any change in X does not effect any changes in Z. Hence, we have $X|Z$ in either of the cases.

Property 3:

Suppose that we are given $n + 2$ variables X, Y_i and Z, where $i = 1, 2, ..., n$, in network $\aleph$ such

that X is upstream of Y_i and Y_i is upstream of Z. Furthermore, suppose that Y_i is neither upstream nor downstream of Y_j for $i \neq j$ and all paths from X to Z pass through one of the Y_i's. Here again we consider three cases.

Statement (a): *For each Y_i, if $X \uparrow Y_i$ and $Y_i \uparrow Z$ or $X \downarrow Y_i$ and $Y_i \downarrow Z$ or $X|Y_i$ or $Y_i|Z$ then $X \Uparrow Z$.*

Proof: Let y_i^k represent the value of Y_i when $X = k$. As $X : 0 \to 1$ we have $Y_i : y_i^0 \to y_i^1$. To understand how all changes in Y_i affect the value of Z we check how each Y_i affects Z. We can think of changing each value of Y_i individually rather than simultaneously because the final value of Z is determined by the final values of Y_i. For $i = 1$ fix $Y_j = y_j^0$ for $j \neq 1$. If $X \uparrow Y_1$ then we will have $Y_1 \uparrow Z$ or if $X \downarrow Y_1$ then we will have $Y_1 \downarrow Z$. In either case $Z : 0 \to 1$ or Z remains unaltered (using property 2). Also, if $X|Y_i$ or $Y_i|Z$ then Z remains unaltered (using property 2). Next for each $i \,\epsilon\, 2, ..., n$ we fix $Y_j = y_j^1$ for $j < i$ and $Y_j = y_j^0$ for $j > i$ and observe how changes in Y_i affect Z. Applying a similar argument as for Y_1 we see that either Z remains unaltered or $Z : 0 \to 1$. So, after considering all the Y_k's we see that either $Z : 0 \to 1$ or Z remains unaltered in response to $X : 0 \to 1$. Hence $X \uparrow Z$ if there is at least one scenario where $Z : 0 \to 1$ in response to $X : 0 \to 1$ or else $X|Z$.

Statement (b): *For each Y_i, if $X \uparrow Y_i$ and $Y_i \downarrow Z$ or $X \downarrow Y_i$ and $Y_i \uparrow Z$ or $X|Y_i$ or $Y_i|Z$ then $X \Downarrow Z$.*

Proof: Making a similar argument as in the case of the proof of Statement (a) and using property 2 we can see that for $X : 0 \to 1$ we have either $Z : 1 \to 0$ or Z remains unaltered. Hence $X \downarrow Z$ or $X|Z$.

Statement (c): *For each Y_i, if $X|Y_i$ or $Y_i|Z$ then $X|Z$.*

Proof: Making a similar argument as in the case of the proof of Statement (a) and using property 2 we can see that for $X : 0 \to 1$, Z remains unaltered. The reason is that any signaling due to a change in the value of X does not reach Z as either $X|Y_i$ or $Y_i|Z$. Hence $X|Z$.

Remark: Note that property 3 is a generalization of property 2. Also, in the property 2 and property 3 statements if we replace the $\uparrow$ by $\Uparrow$ and/or $\downarrow$ by $\Downarrow$ in the 'If' statements then the results still hold,

as the arguments remain the same.

Definition: *In a network* $\aleph$*, suppose* $X_1, X_2, ..., X_n$ *are upstream of the variable* Z*. Then, we say* $(X_1, X_2, ..., X_n) \uparrow Z$ *if two conditions are satisfied. First, changing any subset of the variables* $X_1, X_2, ..., X_N$ *from* 0 *to* 1 *causes* Z *to either change from* 0 *to* 1 *or remain unaltered in any network scenario. Second, for at least one network scenario in* $\aleph\zeta$ *there exists a subset of the variables* $X_1, X_2, ..., X_N$ *which when changed from* 0 *to* 1 *forces* Z *to change from* 0 *to* 1*. Similarly we define* $(X_1, ..., X_n) \downarrow Z$*. If in all possible network scenarios for all possible combinations of changes in values of* X_i *the value of* Z *is unchanged then we write* $(X_1, ..., X_n)|Z$*. We can define the weaker version of the statement by simply relaxing the second condition for* $(X_1, X_2, ..., X_n) \Uparrow Z$ *and* $(X_1, X_2, ..., X_n) \Downarrow Z$*.*

Property 4:

Here we have three cases to consider.

Statement (a): $X_i \uparrow Z$ *or* $X_i|Z$ *for all* $i = 1, ..., n$ *with* $X_i \uparrow Z$ *for at least one* i *if and only if* $(X_1, ..., X_n) \uparrow Z$*.*

Proof: This 'only if' part of the statement is easily proved by using an argument similar to that used in proving case (a) of Property 3. Given $X_i \uparrow Z$ for all $i = 1, ..., n$ we change each variable X_i individually from value 0 to 1 or leave them unchanged and then we see that in each case $Z : 0 \rightarrow 1$ or Z remains unaltered. Since all these changes are concordant with each other, we will see either $Z : 0 \rightarrow 1$ or Z remains unaltered. Thus as $X_i \uparrow Z$ for at least one i we have $(X_1, ..., X_n) \uparrow Z$.

Now given $(X_1, ..., X_n) \uparrow Z$, for a chosen i if Z is unchanged regardless of any change in the value of X_i in any network scenario then we say that $X_i|Z$ otherwise, we say $X_i \uparrow Z$. Note that for some i we must have $X_i \uparrow Z$ or else $(X_1, ..., X_n) \uparrow Z$ would not hold.

Statement (b): $X_i \downarrow Z$ *or* $X_i|Z$ *for all* $i = 1, ..., n$ *with* $X_i \downarrow Z$ *for at least one* i *if and only if* $(X_1, ..., X_n) \downarrow Z$*.*

Proof: This proof is exactly the same as that for case (a) with the only difference that the direction of change for Z is reversed.

Statement (c): $X_i|Z$ *for all* $i = 1, ..., n$ *if and only if* $(X_1, ..., X_n)|Z$*.*

Proof: Again this proof is exactly the same as that for case (a) with Z remaining unchanged all throughout the different cases.

Property 5:

Statement: *For an upstream-downstream pair (X, Y), if the parity of the number of negative gates along any path from X to Y in network $\aleph$ is fixed, then X has a fixed monotonic relationship with Y or $X|Y$. Also, if the parity is even then $X \Uparrow Y$ and for odd parity we have $X \Downarrow Y$. (sufficiency)*

Proof: The proof is very simple. We start from X and move downstream towards Y analyzing the relationship of X and the output of each gate G along any path from X to Y. We do not analyze the relationship of the gate output with X unless all the gate inputs which are downstream of X have their relationship analyzed with X. Observe that we can sequentially go through all the gates as there are a finite number of such gates which lie on the paths between X and Y and also as mentioned earlier the networks we are dealing with here does not have any feedback loops. We can safely say that $X \downarrow X$. This becomes our starting point.

Let the output of G be represented by Z. Let the inputs of G which lie on the path from X be represented by W_i where $i = 1, ..., k$. Note that the parity of the number of negative gates along any of the paths from X to each W_i must be the same or else we will have paths from X to Y with different parity of the number of negative gates which contradicts our initial assumption. Also, the parity number for the negative gates for all W_i's must be the same. As we are sequentially analyzing the gates and we know from property 1 and property 2 that negative gates will flip the monotonicity (also weak monotonicity) direction, we can safely conclude that for all $i = 1, ..., k$ we either have $X \Uparrow W_i$ or $X \Downarrow W_i$. Note that as the gate has multiple inputs, using property 1 we have $W_i \uparrow Z$ for positive gate G or we have $W_i \downarrow Z$ for negative gate G. Thus $X \Uparrow W_i$ for positive gate G and $X \Downarrow W_i$ for negative gate G implies $X \Uparrow Z$. Similarly $X \Downarrow W_i$ for positive gate G and $X \Uparrow W_i$ for negative gate G implies $X \Downarrow Z$. These follow from property 3. At the end of any signaling path we will have $Z = Y$. Thus, we have monotonic relationship between X and Y. Finally, we can see from the proof that if there is an even number of negative gates, then there is an even number of flips in the monotonicity direction and using property 3 to combine the

relationship along different paths, we have $X \Uparrow Y$. If there is an odd number of negative gates, then there is an odd number of flips in the monotonicity direction and hence $X \Downarrow Y$. This proves the sufficiency condition.

Property 6:

Statement: *The fixed monotonic nature of a network is not affected by stuck-at-faults.*

Proof: Note that introduction of stuck-at-faults can cause a path from an upstream variable X to a downstream variable Y to be disconnected but the parity of the number of negative gates along any path would still be the same. Hence, using property 5, the monotonic relationship between X and Y either remains the same (i.e. $X \Uparrow Y$ or $X \Downarrow Y$) or $X|Y$ in case all paths are disconnected due to stuck-at-faults. Thus, the monotonic nature of the network is generally unaffected. Thus, the monotonic nature of the relationship between X and Y remains the same or they satisfy $X|Y$ in the networks belonging to the class $\aleph\zeta$.

Property 7:

Statement: *If the parity of the number of negative gates for all existing paths from any particular input of network $\aleph$ to any particular output of network $\aleph$ is constant, then the network $\aleph$ is a fixed monotonic network.*

Proof: Consider any upstream-downstream variable pair (X, Y), then they must have a constant parity number for the negative gates along all paths; otherwise, we can always find a path P_1 from an input variable to the upstream variable and a path P_2 from the downstream variable to an output variable of the network. These paths P_1 and P_2 combined with the paths between (X, Y) give us paths between an input and an output variable with different parity numbers of negative gates, which is not possible. So all upstream-downstream variable pairs (X, Y) have a single parity number of negative gates along all paths from X to Y. Thus using property 5 we see all upstream downstream variable pairs (X, Y) have a fixed monotonic relationship. Hence by definition, network $\aleph$ is a fixed monotonic network.

Remark: Note that to prove strict monotonic relationship, given fixed monotonic relationship between upstream variable X and downstream variable Y, all we need to do is to show one network

scenario where a change in X causes a change in the value of Y. On the other hand, if it can be proven that no such network scenarios exist then we have $X|Y$.

2.3 Application to MAPK signaling

The purpose of applying the concept of fixed monotonic Boolean networks to signaling pathways is to reduce the activity of cancerous cells. We will attempt to achieve that by regulating the proteins and transcription factors using protein kinase inhibitors (or drugs) in such a way so as to reduce the proliferation of cancerous cells. In our work, we use the MAPK signaling pathway which plays a key role in the process of cell division. Before proceeding any further, we establish the role of fixed monotonic Boolean networks with regard to the MAPK signaling pathway (Figure 2.1). For the sake of brevity henceforth the term signaling pathway and its corresponding network equivalent $\aleph$ will be used interchangeably. Note that as defined earlier, $\aleph$ is also the primary network of class $\aleph\zeta$ and hence has no stuck-at-faults. Hence a protein/transcription factor in the signaling pathway corresponds to a Boolean variable in the network $\aleph$.

Observation 1: *The Boolean network corresponding to the MAPK signaling pathway is a fixed monotonic network.*

Proof: We form a table to see whether the parity values of negative gates between each input and output variable pair, with existent paths between them, are fixed or not. We indicate 'even' parity with 0, 'odd' parity with 1, varying parity with x and n indicates input-output pairs with no paths between them. This is shown in Table 2.2.

We see from Table 2.2 and property 7 that the MAPK network is indeed a fixed monotonic network.

Observation 2: *The drug targets D shown in Figure 2.1 and the output O of the MAPK pathway are related either as $D \Uparrow O$ or as $D|O$.*

Proof: If we trace the path(s) from any drug target location D to the output O we see that there is always an even parity in the number of negative gates between the drug location to the output. This is shown in Table 2.3, utilizing the same notation that was established earlier. Hence, using property 5, we conclude that any single drug target D is linked to an output O as $D \Uparrow O$ or $D|O$.

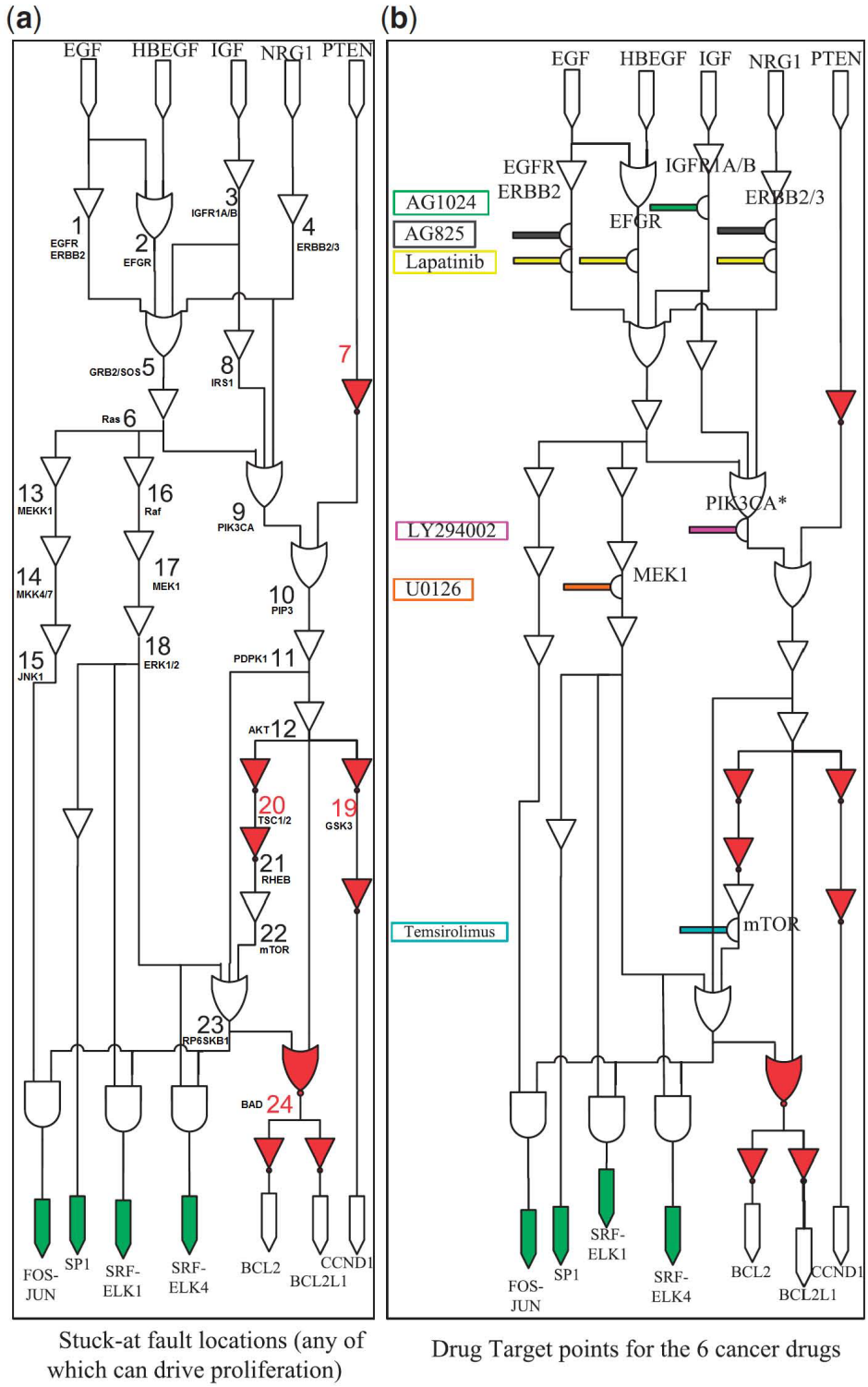

Figure 2.1: Boolean network corresponding to MAPK signaling with (a) representing the fault locations and (b) showing the drug target locations (adapted and reprinted with permission of Oxford University Press from [16]). (NOTE: The outputs marked in green are transcription factors and the outputs marked in white are the reporter proteins)

Table 2.2: Parity of the number of negative gates between each input and output of MAPK network.

Input	Output						
	FOS-JUN	SP1	SRF-ELK1	SRF-ELK4	BCL2	BCL2L1	CCND1
EGF	0	0	0	0	0	0	0
HBEGF	0	0	0	0	0	0	0
IGF	0	0	0	0	0	0	0
NRG1	0	0	0	0	0	0	0
PTEN	1	*n*	1	1	1	1	1

The case of drugs with multiple target locations can be handled using property 4.

Here observation 2 can be generalized to state that the drugs drive all outputs to an acceptable value. In MAPK as the acceptable value for the outputs are 0 we have $D \Uparrow O$. If the acceptable value for any of the outputs were 1 (i.e. if the output proteins inhibited cell division) then we would require the signaling pathway to have $D \Downarrow O$ for all the drug target locations w.r.t the output location O in order to have a property similar to that of observation 2.

Definition: *An acceptable value for a protein is simply the desirable value of the protein state which depends on the particular nature of the fixed monotonic network. This value can be 0 or 1.*

A faulty value for a protein is the state opposite to that of the acceptable value. Thus, if the acceptable value of a protein is 0 then the faulty value of the protein is 1 and vice versa.

In the context of MAPK signaling it is well known that the output transcription factors such as FOS-JUN, SP1, SRF-ELK1, SRF-ELK4 stimulate cell proliferation. Furthermore, the output reporter proteins BCL2, BCL2L1 report on apoptosis suppression while CCND1 reports on cell proliferation. Hence in the absence of proliferation stimuli, the acceptable values for the outputs are all 0 (i.e. inactive or downregulated state). It can be seen from Table 2 that the parity value of the negative gates from the input proteins EGF, HBEGF, IGF, NRG1 to the output transcription factors and reporter proteins is even but for the input PTEN it is odd. Thus, this agrees with the fact that the input protein PTEN is a well-known cell division inhibitor and hence has an acceptable value of 1 (active or upregulated state). The remaining input proteins encourage cell division and hence have an acceptable value of 0. It is worth noting that the drug target locations as shown

Table 2.3: Parity of the number of negative gates between each drug location and output of MAPK network.

Drug	Output						
	FOS-JUN	SP1	SRF-ELK1	SRF-ELK4	BCL2	BCL2L1	CCND1
Lapatinib	0	0	0	0	0	0	0
AG825	0	0	0	0	0	0	0
AG1024	0	0	0	0	0	0	0
U0126	0	0	0	0	0	0	*n*
LY294002	0	*n*	0	0	0	0	0
Temsirolimus	0	*n*	0	0	0	0	*n*

in Figure 2.1 have acceptable values of 0. Hence when a protein kinase inhibitor targets these positions it forces them to have an acceptable value by rendering the target protein inactive.

Observation 3: *If a drug d_i drives an output o_j to 0 by itself, then it would still drive the output to zero (acceptable value) even in the presence of other drugs.*

Proof: If we have other drugs we keep on adding them one by one after applying drug d_i. Note that as d_i has already driven the relevant output to zero and the relationship between the drug target and the output is given by $D \Uparrow O$ or $D|O$, we can see that adding other drugs and thereby forcing the drug targets to zero would not cause the output to go from 0 to 1 as this would violate the relationship $D \Uparrow O$ given by observation 2.

Observation 4: *Consider combination of drugs $\hat{d}$ and $\check{d}$ such that $\hat{d} \subseteq \check{d}$. Then any output driven to zero by $\hat{d}$ will also be driven to zero by $\check{d}$.*

Proof: This basically follows from observation 3 since, if we add extra drugs to the cocktail of drugs $\hat{d}$ to get $\check{d}$, the outputs which were already driven to zero by $\hat{d}$ will stay at zero and hence the result.

These observations will help us in reducing the number of drug combinations to be tested to find the best possible drug combination which gives best possible output along with reduced side effects. Here we assume that the number of drugs corresponds to the extent of side effects due to chemotherapeutic treatment. In the next subsection, we will first see how these observations, especially the second observation, can be used to reduce the number of test cases in our quest to

find the best therapy while using the minimal number of drugs.

Remark: Note that observation 4 can be generalized, for fixed monotonic networks which satisfy generalized observation 2, by using the term 'acceptable value' instead of 'zero' in the statement and proof.

Definition: *A clean friendly location (transcription factor/protein) in the network $\aleph$ is a location where the Boolean variable assumes an acceptable value if at least one of its immediate upstream variables has an acceptable value.*

A fault friendly location in a signaling pathway is a location where the Boolean variable assumes a faulty value if at least one of its immediate upstream variables has a faulty value.

For example, in the MAPK network, the location 23 which represents the protein RPS6KB1 has an acceptable value of zero and so do the immediate upstream proteins at locations 11, 18 and 22. Thus, if one of the locations 11, 18 or 22 has a faulty value then the protein RPS6KB1 is also upregulated or has a faulty value of 1. Thus, RPS6KB1 is a faulty friendly location. Similarly, FOS-JUN is clean friendly as it will have an acceptable value (downregulated or 0) if either JNK1 has an acceptable value (downregulated or 0) or RPS6KB1 has an acceptable value (downregulated or 0). Note that the proteins which have only a single immediate upstream protein like GSK3 at location 19 or RHEB at location 21 can be categorized as both clean friendly and fault friendly.

Now keeping these definitions in mind, we look forward to using the properties of monotonic networks in general and the signaling structure of MAPK pathway in particular to find the best possible drug combinations using a limited number of test cases.

2.4 Drug intervention on MAPK signaling pathway

In this subsection we will show how the observations made in the previous subsection can be used along with the MAPK pathway information to reduce the number of test cases. The demonstration will be carried out using both synthetic data and experimental data.

Table 2.4: Entries under drug combinations are output strings. TC1, etc. represents the extra test cases needed to find the best drug combination d_{final}.

Index number	Scenario type	Fault locations	Drug combinations used for testing								d_{final}
			d_ϕ	d_1	d_2	d_3	d_4	d_5	d_6	d_Σ	
1	1	18	0111110	0111110	0111110	0111110	0111110	0111110	0111110	0111110	No drugs
2	1	19,23	0000111	0000111	0000111	0000111	0000111	0000111	0000111	0000111	No drugs
3	2	1,5,7,16,17	1111111	1111111	1111111	1111111	1000111	1111111	1111111	1000111	4
4	2	2,7,21,22,24	1111111	0000111	1111111	1111111	1000111	1111111	1111111	0000111	1
5	3	3,8,10,16,20	1111111	1111111	1111111	0111111	1000111	1111111	1111111	0000111	3,4
6	3	2,11,12,17	1111111	0111111	1111111	1111111	1000111	1111111	1111111	0000111	1,4
7	4	2,8,16,20,24	1111111	0111111	1111111	1111111	1000111	1111110	1111111	0000110	TC1
8	4	3,9,17,22,24	1111111	1111111	1111111	0111111	1000111	1111110	1111111	0000110	TC2
9	5	5,8	1111111	1111111	1111111	1111111	1000111	1111110	1111111	0000000	TC3
10	5	16,20,21	0111110	0111110	0111110	0111110	0000110	0111110	0111110	0000000	TC4

2.4.1 Synthetic Data

We assume in this test that all the inputs corresponding to growth factors i.e. EGF, HBEGF, IGF, NRG1 are absent and the tumor suppressor protein PTEN is active (refer to Fig. 2.1). Thus, a normal cell would force all the outputs to be 'OFF' or deactivated as the transcription factors (such as FOS-JUN, SP1, SRF-ELK1 and SRF-ELK4) and reporter proteins (such as BCL2, BCL2L1 and CCND1) are expected to be inactive or downregulated in such a scenario. If there is a breakdown, i.e. a mutation which causes a breakdown in the MAPK signaling pathway, it should be appropriately reflected as a stuck-at-fault in the network. Note that in Figure 2.1, for locations $X = \{7, 19, 20, 24\}$ we have $X \downarrow O$ where O represents any one of the 7 outputs. For the rest of the locations we have $X \uparrow O$. We consider only stuck-at-faults which force the output to be activated, thus leading to cancer. Hence for locations $X = \{7, 19, 20, 24\}$ we consider stuck-at-zero faults (marked in red in Fig. 1 (a)) and for the rest of the fault locations corresponding to values of $X = \{1, 2, ..., 24\} \backslash \{7, 19, 20, 24\}$, we consider stuck-at-one faults. So, if we say location $X = 20$ has a fault, it would imply that the value at the location $X = 20$ is stuck at 0.

For representing outputs, we simply use a string of binary numbers with 1 representing the active state and 0 representing the inactive state. Thus the value 1000110 means that the output vector [FOS-JUN, SP1, SRF-ELK1, SRF-ELK4, BCL2, BCL2L1, CCND1]=1000110. In this case, only FOS-JUN, BCL2 and BCL2L1 are activated and the rest of the outputs are downregulated or inac-

tive. If only SP1 is active then we write the output string as 0100000.

For the sake of convenience, the drugs Lapatinib, AG825, AG1024, U0126, LY294002 and Temsirolimus are numbered from 1 through 6 in that order. For our testing purposes we represent the case when no drugs are applied by d_ϕ. When all the six drugs are applied we represent it by d_Σ. For cases when at least one drug is applied but not all of them, we use the notation $d_{a,b,...,k}$ where $\{a, b, ..., k\}$ is a listing of the drugs in the combination used. In our first round of tests we only check the outputs of the network for the cases $\{d_\phi, d_1, d_2, d_3, d_4, d_5, d_6, d_\Sigma\}$. In other words, we are initially considering the cases: no drugs used, all drugs used simultaneously, and all possible drugs singly used. For any drug combination d_Ω we represent the output as Od_Ω. Hence Od_Ω is a binary string of length 7. We perform the combination $\oplus$ of two or more such output strings such that the resulting string has an acceptable value in the i^{th} bit if at least one of the original strings had an acceptable value in the i^{th} bit or else the i^{th} bit has a faulty value. In case of MAPK this is implemented simply by using the 'AND' operator on the individual string positions. Thus we have $0011001 \oplus 0000111 = 0000001$. For any two output sequences Od_a and Od_b if Od_b has acceptable values at all the locations Od_a has acceptable values in, and acceptable values in a few more positions where Od_a has faulty values, we write $Od_a < Od_b$. We can use the symbol $\leq$ instead of $<$ where a possibility of equality exists. Now a total of 5 scenarios can occur when we test the output for the cases $d_\phi, d_1, d_2, d_3, d_4, d_5, d_6, d_\Sigma$. They are;

Scenario 1: $Od_\phi = Od_\Sigma$

Scenario 2: $Od_\phi \neq Od_\Sigma$ *&* $Od_i = Od_\Sigma$ *for some* $i \,\epsilon\, \{1, 2, 3, 4, 5, 6\}$

Scenario 3: $Od_i \neq Od_\Sigma$ *for any* $i \,\epsilon\, \{1, 2, 3, 4, 5, 6\}$ *&* $Od_i \oplus Od_j = Od_\Sigma$ *for some* $i, j \,\epsilon\, \{1, 2, ..., 6\}$

Scenario 4: $\oplus_{i\epsilon S} Od_i = Od_\Sigma$ *only for one or a few* $S \subseteq \{1, 2, 3, 4, 5, 6\}$ *where* $|S| > 2$

Scenario 5: $\oplus_{i\epsilon S} Od_i < Od_\Sigma$ *where* $S = \{1, 2, 3, 4, 5, 6\}$

Note that for any signaling pathway, which is a fixed monotonic Boolean network and satisfies the generalized observation 2, we will have these five scenarios assuming that we are only using the drug combination d_ϕ, d_Σ and d_i where $i \,\epsilon\, \{1, 2, ..., k\}$ where $k > 2$ is the number of drugs available to target the pathway. The reasoning is as follows. On basis of generalization of observations 4

and definition of the operation $\oplus$ we can say that $Od_\phi \leq Od_i \leq Od_\Sigma$ and $\oplus_{i\epsilon S} Od_i \leq Od_S$ where $S \subseteq \{1, 2, ..., k\}$. Thus we obtain $Od_\phi \leq Od_i \leq Od_i \oplus Od_j \leq \oplus_{k\epsilon S} Od_k \leq Od_\Sigma$ where i, $j\epsilon S$ and $|S| > 2$. Thus, looking at this inequation with four inequalities and five expressions we come up with the five scenarios. In scenario 1 we have $Od_\phi = Od_\Sigma$ thus forcing all the intermediate inequalities to become equality. In scenario 2 we have $Od_i = Od_\Sigma$ for some i and $Od_\phi < Od_\Sigma$. In scenario 3 we have $Od_i \oplus Od_j = Od_\Sigma$ for some i, j whereas $Od_i < Od_\Sigma$ for all i. In scenario 4 we have $\oplus_{i\epsilon S} Od_i = Od_\Sigma$ for some S where $|S| > 2$, $S \subseteq \{1, 2, ..., k\}$ and $Od_i \oplus Od_j < Od_\Sigma$ for all i and j. Finally in scenario 5 we have $\oplus_{i\epsilon S} Od_i < Od_\Sigma$ for all $S \subseteq \{1, 2, ..., k\}$.

Now we describe the procedure which automates most of steps to identify the best possible drug combination(s) given the initial round of tests. After the initial round of tests in some cases a second round of tests might be needed, which forms the manual part of the procedure, as we shall see later. The steps have been depicted by a flowchart which is shown in Figure 2.2. Here, we will focus on and use examples based on the MAPK signaling pathway shown in Figure 2.1 to find the best possible drug combination d_{final}. We will see that the d_{final} is indeed the best possible combination satisfying our requirements by looking at the outputs for all the possible drug combinations.

Please note that these steps can be applied to any signaling pathway with any input values as long as it satisfies the condition for fixed monotonic Boolean network and the generalized observation 2 in the previous subsection.

Steps to identify the best drug combination

Step 1: *If the test cases belong to scenario 1, 2 or 3 then the best possible drug combination d_{final} can be computed only by using the general properties of fixed monotonic Boolean network otherwise proceed to step 2.*

Implementation and Explanation: Here we assume that the signaling pathway information along with the drug data for the relevant network is already available. The observed output values are manually entered.

Clearly drug combination d_Σ gives the best output on the basis of observation 4. Scenarios

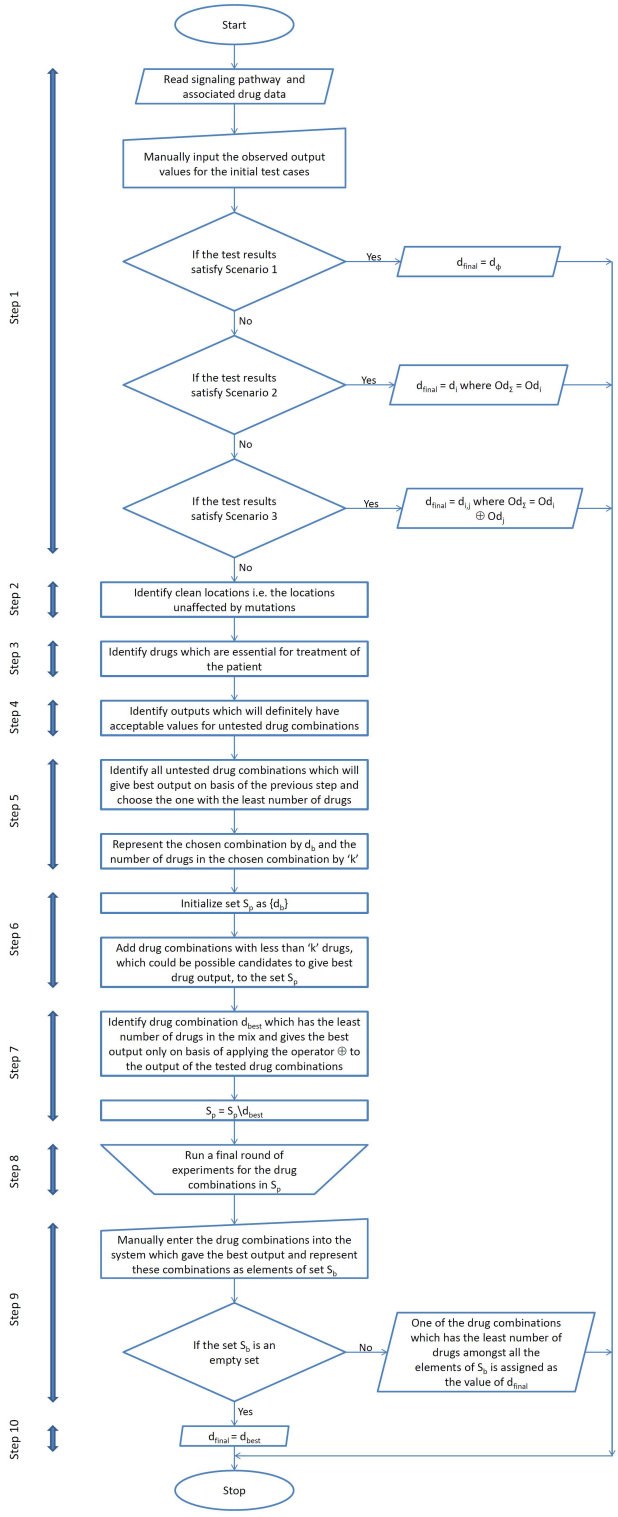

Figure 2.2: Flowchart depicting the method to find the best drug combination

1, 2 and 3 offer the best drug combination in a straightforward manner. For scenario 1, the best combination is giving no drugs. On the basis of observation 4, clearly any subset of drugs cannot further downregulate the active outputs; hence, it is best not to give any drugs in this case and thus there is no possible chance for any side effects. Thus for scenario 1, we have $d_{final} = d_{\phi}$.

If the test results satisfy scenario 2, then we need at least one drug to achieve the best output. Hence the drug 'i' is the best possible drug combination since it gives the best output and also requires the least number of drugs. Thus, for scenario 2, we have $d_{final} = d_i$. In the case of scenario 3, we can clearly see that no single drug is able to give the best result in terms of output downregulation but the two drugs individually combined downregulate all the outputs to give the best result. Hence the best possible combination of drugs is the combination of the two drugs d_i and d_j. Thus $d_{final} = d_{i,j}$ for scenario 3. We can see examples of these three scenarios in Table 2.4, by looking at the d_{final} for the first six mutated networks.

Now for scenarios 4 and 5, we are unable to obtain a generic result, that holds for any fixed monotonic Boolean network satisfying a generalized form of observation 2. Instead, we will also have to exploit the particular network structure along with the fixed monotonic nature of the pathway.

Step 2: *Identify the clean locations (i.e. the proteins which are unaffected by cancerous mutations). Then proceed to step 3.*

Implementation and Explanation: The identification of clean locations is carried out as follows. We assume a fault in the location x (say) and then trace down the network to the terminal outputs. Since the location x has a faulty value this causes the downstream locations to assume faulty values. This may contradict with the actual test results where some of the outputs may have acceptable values. In this case, we would have a contradiction and hence our initial assumption about location x would be wrong. Thus, we deduce that the location x is clean. Using this method, we identify the clean locations. This idea is utilized in two ways as follows:

a) We take into account a test case (A single test case comprises of all the observed outputs for a single drug combination) out of the tests already performed on the cell lines (we have a

total of eight test results in our case for the synthetic MAPK data) and see which of the output nodes have acceptable values. Assume that a location x has a mutation and then use the Boolean network (MAPK in this case) to see if it predicts a faulty value in the output which in reality has an acceptable value. If it does, then the actual test does not agree with the prediction which means that our assumption was wrong. This implies that location x is a clean location and does not have any mutation.

For example, in mutated network numbered 7 in Table 2.4, let us look at the response of MAPK cell lines to the drug combination d_5. Here the output protein CCND1 has an acceptable value of 0. Hence, we can use this information to deduce that location 10, 11, 12 and 19 must be clean or otherwise they would lead to a faulty value in the output CCND1 which actually has an acceptable value.

b) Let us assume that in a test using the drug combination d_α we observe an acceptable value in the output node o_i but a faulty value in the output node o_j while for the test using the drug combination d_β we observe a faulty value in o_i but an acceptable value in o_j. This would mean that out of all the locations in the signaling pathway targeted by the drug combination d_β but not targeted by the drug combination d_α, we must have some locations which have faulty values when we are testing using the drug combination d_α. These faulty values would lead some of the signal paths to have faulty values. We could eliminate these faulty signal paths from the list of paths between target locations of the drug combination d_α and the output o_i while determining clean locations.

Let us look at an example to better understand this idea. Consider the mutated network numbered 7 in Table 2.4. In this case, for the drug combination d_1 output node FOS-JUN (o_1) has an acceptable value but output node CCND1 (o_7) has a faulty value. On the other hand, for the drug combination d_5, the output node o_1 has a faulty value and the output node o_7 has an acceptable value. This would imply that when we are testing using the drug combination d_1 the target location of d_5 (not targeted by d_1) i.e. location 9 must have a faulty value which leads o_7 to have a faulty value. Faulty value at the location 9 implies a faulty value at the location 23 when we are applying

the drug combination d_1. This would lead us to the conclusion that the location 15 must have an acceptable value or else FOS-JUN would be upregulated (i.e. have a value 1) which is not the case. Only a clean location can have an acceptable value hence location 15 does not have any cancerous mutations. Similarly, locations 3, 5, 6, 13 and 14 must be clean or be mutation free.

Step 3: *Identify drugs which are essential to be considered for treatment of patient. Then proceed to step 4.*

Implementation and Explanation: We can eliminate the drugs which we deem unnecessary in two situations. This algorithmic step is explained below.

a) In this case we will show how a drug or a combination of drugs can be eliminated once it is determined that the target locations of the drug never has a faulty value due to mutations in the signaling pathway.

For example, again looking at mutated network 7 we saw in the previous step that location 3 is a clean location. Upstream of this location we only have input values which are already set to acceptable values. Hence location 3 always has an acceptable value. Thus, the drug AG1024 which targets only location 3 is unnecessary and can be dropped from the list of drugs which are to be considered for treating the patient.

b) In the second situation if a drug completely overlaps the targets of another drug then the overlapped drug can be removed from the list of drugs to be considered for treating the patient as the activity of the overlapping drug subsumes that of the overlapped drug. This follows from observation 4 in the previous subsection.

As an example, consider the drug AG825 which targets location 1 and 4 and consider the drug Lapatinib which targets 1, 2 and 4. We can clearly drop the drug AG825 from the list of drugs to be considered for treatment of the patient.

Step 4: *Calculate which outputs should have acceptable value for each available drug combination (the ones not tested involving the essential drugs from step 3). Then proceed to step 5.*

Implementation and Explanation: There are two cases which need to be considered in this step. Each case is illustrated by an example below.

a) Suppose a tested drug combination d_α drives the output node o_i to an acceptable value. Now consider an untested drug combination d_β. Represent all the target location(s) of drug combination d_α which does not overlap with target location(s) of d_β by t_α. We know from the signaling pathway information that the target locations of d_β cut off all paths between t_α and output node o_i. If we apply a drug combination which includes all the drugs in d_α and d_β then definitely output node o_i will be driven to an acceptable value on basis of generalized observation 4. Removing the drugs from the pathway location t_α should not have any effect on output node o_i as there is no available path between t_α and the output node o_i through which any changes in signal value at t_α can be transmitted. Thus applying the drug combination d_β only we can say that output node o_i is driven to an acceptable value.

For example assume drug AG1024 drives output o_5 (i.e. BCL2) to an acceptable value but the drugs U0126 and LY294002 do not drive o_5 to an acceptable value when used individually. This can actually happen if the MAPK pathway has a single fault at the location 3 (target of AG1024). In this case looking at the structure of MAPK we can easily deduce that the drug combination involving the drugs U0126 and LY294002 together will drive o_5 to an acceptable value. The reasoning is that these two drugs force the locations 9 and 17 to an acceptable value and also cut off all possible paths between the location 3 and the output o_5. Thus, based on observation 2 and the fact that the two locations through which the acceptable value would be transmitted from the location 3 to the output o_5 have been forced to 0 we can safely deduce that o_5 is driven to 0 (acceptable value) by the drug combination U0126 and LY294002 together.

b) In this case using observation 4 and the operation $\oplus$ defined earlier we can determine which outputs have acceptable values (0 in case of MAPK). Based on this we can determine which of the clean locations determined in step 2 must have acceptable values. This determination of clean locations with acceptable values is done by identifying fault friendly locations which have acceptable values. This implies that immediate upstream locations must have acceptable values by definition of fault friendly location. We repeat the process over and over again by identifying new fault friendly locations with acceptable values and stop when we cannot find any new fault friendly

locations with acceptable values. We assume that the rest of the locations have faulty values. Then using the structure of the fixed monotonic network, we compute which outputs have acceptable values. As this is a worst-case scenario where we are forcing maximum locations to have faulty values the outputs which are computed to have acceptable values must have acceptable values (0 in our case of MAPK).

Consider for example the same drug combination U0126 and LY294002 described in the previous case (a). An acceptable value at the location o_5 forces the location 23 in MAPK to have an acceptable value when the two drugs are applied which would mean the output o_1 (i.e. FOS-JUN) must be 0 (an acceptable value) because o_1 is a clean friendly location.

Step 5: *Identify the combination with the least number of drugs which gives the best output on basis of step 4. Then proceed to step 6.*

Implementation and Explanation: We check if any untested drug combinations considered in step 4 attains the best possible output. This is done by comparing outputs obtained in step 4 with the best possible output i.e. Od_Σ and then choosing the one combination involving the least number of drugs which provides the best possible output. Let us represent this combination by 'd_b' and the total number of drugs in the combination by 'k'.

Step 6: *Identify the possible drug combination candidates with less than k drugs which could give the best output. Then proceed to step 7.*

Implementation and Explanation: For this step we look at all untested combinations with less than 'k' number of drugs where 'k' is defined in step 5. Once the drug combinations have been identified let us represent the set of these combinations by S_t. For each combination in S_t we see which output nodes have faulty values, on basis of step 4, but have acceptable values in the best output Od_Σ. Then we try to see if there is a possibility of the drug combination achieving an acceptable value in those output nodes.

The implementation is explained using the mutated network numbered 8 in Table 2.4 as follows. We know drug combination (let's say d_{45}) composed of LY294002 (d_5) and U0126 (d_4) would give an output 1000110 based on step 4. Best possible output is 0000110. So, the output

Table 2.5: Second round of experiments. Here S_p represents the set of drug combinations to be tested. d_{best} represents the drug combination identified in step 7 of the algorithm which will certainly give the best output. d_{final} is the drug combination which is finally identified as the best combination to be used for the cancer patient following the protocol for treatment.

Extra Test Cases	S_p	Experimental Output Values	d_{best}	d_{final}
TC1	4,5	1000110	1,4,5	1,4,5
TC2	4,5	1000110	3,4,5	3,4,5
TC3	1,3	1111111	1,2,3,4,5,6	4,5
	4,5	0000000		
	4,5,6	0000000		
TC4	4,6	0000000	1,2,3,4,5,6	4,6

node o_1 (i.e. FOS-JUN) is supposed to have an acceptable value but has faulty value based on step 4. Our aim is to see if drug LY294002 and U0126 combined can drive o_1 to an acceptable value. In this case we see that downstream paths from d_5 and d_4 combine at location 23 which is upstream of o_1 and is a fault friendly node. Thus it is entirely possible that the drug combination $d_{4,5}$ forces location 23 to have an acceptable value whereas d_4 and d_5 could not force location 23 to have an acceptable value themselves due to the way fault friendly nodes behave. As location 23 is upstream of o_1, an acceptable value at location 23 would lead o_1 to have an acceptable value. In general, networks having an acceptable value at a fault friendly location where paths from two or more drugs first combine could lead the downstream output node to assume an acceptable value. Thus, we basically assume an acceptable value in the output node being considered if there is a fault friendly node upstream of it where paths from the two or more drugs in the drug combination being considered combine (eg. paths from LY294002 and U0126 combine at fault friendly node at location 23). Now with the updated output list for the combinations in S_t we see which of the outputs are same as the best possible combination. These drug combinations which are possible candidates to have best possible outputs along with the drug combination d_b obtained in step 5 are represented by the set S_p.

Step 7: *Identify the drug combination d_{best} with the least number of drugs which is predicted to*

produce the best output just on the basis of the results of applying the operator $\oplus$ on the tested drug combinations. Remove this drug combination from the set S_p or simply put $S_p = S_p \backslash d_{best}$ (popularly used notation for set subtraction in set theory).

Implementation and Explanation: In order to find d_{best} for scenario 4 we can find a drug combination with the least number of drugs which gives the best output on basis of the operator $\oplus$. For scenario 5 we simply use the drug combination d_Σ.

Clearly mutated network numbered 8 in Table 2.4 is an example of scenario 4. Here we can see that d_{best} is the drug combination involving $d_{3,4,5}$. After step 6 of this algorithm $S_p = \{d_{3,4,5}, d_{4,5}\}$ for test case 8. As $d_{best} = d_{3,4,5}$ we reevaluate S_p by subtracting d_{best} from S_p. Finally we have $S_p = d_{4,5}$ and $d_{best} = d_{3,4,5}$.

Similarly, we can see that the mutated network numbered 10 in Table 2.4 is an example of scenario 5. Hence the drug combination d_{best} is simply the combination d_Σ. After step 6 of the algorithm $S_p = d_{4,6}$. Finally after evaluating this step we have $S_p = d_{4,6}$ and $d_{best} = d_\Sigma$.

Note that the value of d_{best} found in this step of the algorithm need not be the same as the value of d_b found in step 5. This is because, computation of d_{best} makes use of the operator $\oplus$ and the experimental results only unlike d_b, whose computation also makes use of the signaling structure of the network.

Step 8: *Run a final round of experiments involving the drug combination identified in S_p. Proceed to step 9.*

Implementation and Explanation: This is the manual part of the procedure which involves testing the drug combinations in set S_p on the mutated network.

We can see in Table 2.5 the output strings for the mutated network 7, 8, 9 and 10 (refer to Table 2.4) in the second round of testing.

Step 9: *If some combinations in S_p actually achieve the best output in the testing then choose the one with least number of drugs d_{final} and Stop. If not proceed to step 10.*

Implementation and Explanation: Here we can see in Table 2.5 that for mutated network numbered 9 and 10 (refer to Table 2.4), corresponding to test cases TC3 and TC4 respectively, some drug

combinations in S_p achieve the best output thus providing us the d_{final} values.

Step 10: *If none of the drug combinations in S_p actually produce the best output then we set $d_{final} = d_{best}$ where we computed d_{best} in step 7 of the algorithm. Stop.*

Implementation and Explanation: Here we can see in Table 2.5 that for mutated network numbered 7 and 8 (refer to Table 2.4) , corresponding to test cases TC1 and TC2 respectively, we find the d_{final} values in this step.

The synthetic data set (as shown in Table 2.4 and Table 2.5) has been generated on basis of the prior knowledge about the MAPK pathway. The drug combination d_{final} (as shown in Table 2.4 and Table2.5) decided following the steps mentioned above can then be administered to a cancer patient. The results of the steps above are shown in Table 2.4 and Table 2.5. Note that in Table 2.4 for the first six mutated networks we can immediately identify the best drug combination in the first step of the procedure as they belong to one the scenarios 1, 2 or 3. For the remaining mutated networks (i.e. 7, 8, 9 and 10) we have to go through all of the steps described above. Thus, for the last four mutated networks the drug combinations, S_p (shown in second column of Table 2.5) which are possible candidates for best drug combination are identified. In fact, the goal of the steps 2 through 7 is to help identify the smallest possible set S_p which will help us in identifying the best drug combination with certainty. The actual experimental results for the drug combinations S_p are provided in the third column of Table 2.5. The fourth column of Table 2.5 lists the combination which is supposed to give the best output as computed in the seventh step.

Thus, we see that for handling scenarios 4 and 5, we need to use the structure of the MAPK signaling pathway along with the properties of the fixed monotonic Boolean network class. It is worth noting that the drug combination d_b which was identified in step 5 of the algorithm should definitely give the best possible output if signaling pathway behaves as a perfect Boolean network. However, we do include d_b in the set of drug combinations S_p to undergo a final round of testing to see which combinations actually give the best output. This is because the cell signaling pathways are less than perfect examples of Boolean circuits, even though using logic gates are an efficient way to represent various protein-protein interactions. The properties of fixed monotonic

Table 2.6: Expression values of reporter genes and transcription factors for initial test case.

	Housekeeping gene	SRF-ELK reporter gene	SP1 reporter genes		
	GAPDH	EGR1	cMYC	JUN	IRF3
No drugs	1	1	1	1	1
LY294002	1	0.795536	0.411796	0.946058	0.806642
LY294002+U0126	1	0.386891	0.291183	0.31864	0.353553

Table 2.7: Logical values of reporter genes and transcription factors for initial test case.

	Housekeeping gene	SRF-ELK reporter gene	SRF-ELK	SP1 reporter genes			SP1
	GAPDH	EGR1		cMYC	JUN	IRF3	
No drugs	1	1	1	1	1	1	1
LY294002	1	1	1	0	1	1	1
LY294002+U0126	1	0	0	0	0	0	0

Boolean network still hold for the Boolean analogues of the signaling pathways. Hence, we can be much more certain that the drug combination d_{best} computed in step 7 of the algorithm will definitely provide the best output in an experimental set up. In the next subsection we will further explain the above statements as we deal with real world data from actual cell lines in the following subsubsection.

The d_{final} obtained for the 10 mutated networks are actually the best possible drug combination. This can be easily verified by simply checking that no other drug combination with lesser number of drugs gives the best output. This is immediately clear for the networks 1, 2, 3, 4, 5, 6, 9 and 10 from Table 2.4 and Table 2.5. For networks 7 and 8 we simply check the outputs corresponding to the drug combinations with two drugs and find that none of these combinations actually give the best output.

2.4.2 Experimental Data

To illustrate the usefulness of the idea of fixed monotonic Boolean networks we perform a simple experiment [26] on actual cell lines. For this purpose, we choose normal adult fibroblast cell lines. These cell lines were grown in Fibroblast Basal Medium (ATCC) in 60 mm tissue cul-

ture petri dishes till confluence. After this the cells were exposed to Dulbeccos modified Eagles medium-F12 (DMEM/F12) (Atlanta Biologicals) which was supplemented with 0.2% fetal bovine serum (Atlanta Biologicals) for 4 days. Every day after washing with phosphate buffer solution (PBS) the medium was changed. The cell cultures were incubated in a 5% CO_2 incubator at 37^0C. After this they were exposed to DMEM/F12 further supplemented with 0.2% FBS and $100\mu M$ Anisomycin for a period of 30 minutes. Anisomycin is a protein synthesis inhibitor which activates the MAPK signal transduction network and keeps it responsive to kinase specific inhibitors [27, 28]. Further it also cuts any feedback path that has a protein synthesis step in it. The tissue culture petri dishes were then grouped into 4 groups. After initially exposing them for 30 minutes to Anisomycin each group was exposed to DMEF/F12 supplemented with 20% FBS, $100\mu M$ Anisomycin, $50\mu M$ of LY294002, and/or $10\mu M$ of U0126. [2] Genes which have response elements to transcription factors SP1 and SRF-ELK1/4 were quantified using real time PCR and the delta-delta method [29] with GAPDH being used as a reference gene or the housekeeping gene. The group which has no drugs applied is used as the control. EGR1 the reporter gene for transcription factor SRF-ELK [30] is measured. For the transcription factor SP1 the reporter genes measured are cMYC, JUN, IRF3 [31, 32, 33]. Hence, we have a total of 4 observed reporters genes whose expression values are computed as shown in Table 2.6 and 2.8. In our case we use a simple threshold for expression values to decide if a gene is active or inactive. The threshold is set at 0.5. Any value more than 0.5 is assigned a value of 1 or else it is assigned a value of 0. In a more realistic scenario, this value could also be set by taking input from doctors treating cancer patients. As SP1 has three reporter genes, to decide the Boolean state of SP1 we pick the state which appears most frequently for its reporter genes. In this case, the MAPK pathway has all growth factors present and PTEN is active. Hence all inputs to the network are set to 1.

As for the present experiment, we assume we have only two drugs U0126 and LY294002. In the first set of tests, we run the tests with no drug case, LY294002 only and the best output

[2]Note that since we have only two drugs here, if we proceeded by testing with no drugs, then with one drug at a time, and then finally with both the drugs, we would have exhaustively covered all possible cases, in the first round itself. Accordingly, to keep the problem interesting, we initially consider only one of the cases where a single drug has been applied.

Table 2.8: Expression values of reporter genes and transcription factors for the second round of tests.

	Housekeeping gene	SRF-ELK reporter gene	SP1 reporter genes		
	GAPDH	EGR1	cMYC	JUN	IRF3
U0126	1	0.812252	0.397768	0.68302	0.45376

Table 2.9: Logical values of reporter genes and transcription factors for the second round of tests.

	Housekeeping gene	SRF-ELK reporter gene	SRF-ELK	SP1 reporter genes			SP1
	GAPDH	EGR1		cMYC	JUN	IRF3	
U0126	1	1	1	0	1	0	0

case involving LY294002 and U0126 as shown in Table 2.6 and Table 2.7. Now using the [3] steps described in the previous subsection we see that, in Table 2.7 the best output is not achieved in either the no drug case or when LY294002 is applied. Thus, it does not satisfy scenario 1 or 2. As we do not have the output data for application of U0126, we can say that this test case does not satisfy scenario 3 either. It thus belongs to scenario 4 or 5. Now going through the steps described earlier we have in step 5, $d_b = d_{U0126}$. Finally, in step 7 we have $S_p = \{d_{U0126}\}$ and $d_{best} = d_{LY294002,U0126}$. Then we run the experiment on the drug combinations of S_p. These results are shown in Table 2.8 and Table 2.9. As d_{U0126} does not give the best output we will finally choose d_{best} i.e. $d_{LY294002,U0126}$ as the best drug combination to be used for treatment.

The d_{final} obtained for the cell lines is also the best possible. This is easily seen by looking at Table 2.10 observed values column.

[3]The automated portion of the steps described in the previous subsection under subsubsection titled “synthetic data" is implemented in a manner such that it can handle experimental data as long as we have the experimental results from the application of d_ϕ, d_Σ and some of the d_i's. If results for some d_i's are missing it would not impede the working of the procedure. Also, the input values to the signaling network should not affect the working of the algorithm as long as they are known.

Table 2.10: Comparison of Expected and Observed Values.

	Expected values		Observed values	
	SRF-ELK	SP1	SRF-ELK	SP1
No drugs	1	1	1	1
LY294002	1	1	1	1
U0126	0	0	1	0
LY294002+U0126	0	0	0	0

2.5 Discussion

The primary goal of the section is to use minimal number of test cases to identify the best possible drug combination for the patient thus making the whole process more cost efficient and time efficient. Hence to achieve the goal of reduction in the number of test cases, the properties of fixed monotonic networks is exploited to help identify the best drug combination using a smaller subset of test cases. As the general monotonic properties associated with the network is not enough to help identify the best drug combination in some scenarios, we further exploit the signaling structure of the individual pathway as shown in the procedure in the previous subsection to obtain the minimal set of test cases S_p in order to fulfill our purpose.

In simulated data we have used a total of six kinase inhibitors on the MAPK pathway. Hence the total number of test cases involved in brute force experimentation is $2^6 = 64$. In the first three scenarios, clearly, we just need only a total of 8 cases to give us the best drug combination which will not only produce the best output but also be the one with the least possible side effects. The remaining two scenarios could require a few more test cases depending on the actual mutation and the signaling network structure. This is clearly seen in the previous subsection. Thus, the properties of the fixed monotonic Boolean network class is the crucial ingredient that permits a reduction in the number of test cases, no matter what the actual network structure is as long as the network belongs to that class. This is especially evident for the first three scenarios considered where we can clearly see that the number of test cases for n drugs is simply $n + 2$ which is linear w.r.t. n as opposed to the total number of available drug combinations 2^n which is exponential in

nature. However, as scenario 4 and 5 need us to exploit the signaling structure of the pathway this method sometimes can be limited by the accuracy of representation of signaling pathways using Boolean networks as we see in the experimental data set. Also, this adds a varying number of drug combinations to the test set depending on where the mutations are located in the network. Thus, one scope of improvement in the future would be to further exploit the monotonic nature of networks and reducing the utilization of signaling pathway structure while deducing the best possible drug combination like we did for scenarios 1, 2 and 3.

From the experimental results in the previous subsection, we see that the properties of fixed monotonic Boolean networks hold even in real life scenarios and can be exploited as such. Indeed, from Table 2.7 and Table 2.9 it is clear that a subset of drugs drives fewer components of the output to zero as compared to a superset of drugs. Also looking at the expression values in Table 2.6 and Table 2.8 we can see that expression values of the output is always lower for a superset of drugs as compared to a subset of it. Thus, the advantage of using properties of fixed monotonic Boolean networks instead of just using Boolean representation is that it is robust to the choice of threshold for determining the binary value from the expression data. The reason behind this is that the MAPK signaling network which is derived from the MAPK signaling pathway diagram still satisfies property 7 given in an earlier subsection. The negative gates are introduced only where a signaling protein inhibits the immediate downstream protein. Hence, if we look at any signaling path from any input to any output, the parity of the number of inhibitory signaling steps is exactly the same as the parity of the number of negative gates in its corresponding digital signaling diagram. So, our method based on properties of fixed monotonic Boolean network still holds good even though we introduce loss of information when we quantize continuous expression data into digital data.

Using the threshold 0.5, we computed the observed states for the outputs SP1 and SRF-ELK. In the Table 2.10, the expected values have been calculated by taking into account the fact that all the inputs corresponding to growth factors in the MAPK signaling network are present and the tumor suppressor input PTEN is also present. The observed and expected values diverge for SRF-ELK

when the testing is done using only drug U0126. One reason for observed values for SRF-ELK not agreeing with the expected value could be that signaling pathways do not behave exactly as Boolean networks. To better understand this, let us look at the behavior of SRF-ELK and SP1 when the drug U0126 is applied. The median value for the expression levels of the SP1 reporter genes lies close to 0.45. Thus, looking at the signaling pathway we can say that, in all probability expression level of ERK1/2 also lies close to 0.45 as SP1 depends only on ERK1/2. Now the protein RPS6KB1 is expected to have a higher expression level as the input growth factors are present. SRF-ELK is determined by the combined effect of both RPS6KB1 and ERK1/2. Thus, as expression level of ERK1/2 is close to 0.5 albeit less than it and RPS6KB1 has a high expression value we would expect a higher level of expression for SRF-ELK. However, since we are converting expression values into binary values simply by using a threshold, the information that the value of ERK1/2 is close to 0.5 is not utilized here and hence our observed and expected values do not match up. Perhaps such an issue can be prevented by using a slightly stronger concentration of drugs which would perhaps downregulate the outputs further instead of the current expression values which lie close to the transition value of 0.5. The strength of drugs applied in such a context can be a topic of interest for research. Another reason for the observed value and expected value not matching up might have to do with biological noise inherent in such experiments. Such problems due to noise can be avoided by looking at expression values of few more reporter genes for SRF-ELK or by taking multiple reading of the same reporter gene to help reduce biological noise present in the readings. Note that in the case of SP1 for which expression value readings from multiple reporter genes were taken, the expected and observed values match up completely. Nevertheless, we can see that prediction made on basis of assumptions of a perfect Boolean circuit might lead to faulty results in certain cases. This is also the reason why the drug combination d_b identified in step 5 of the procedure to give the best output is experimentally tested as a part of the set S_p. However, the drug combination d_{best} computed in step 7 only using the monotonic properties of MAPK still works and hence it need not be tested to guarantee its outcome in terms of the pathway output. This is one of the major advantages of utilizing the properties of fixed monotonic Boolean network.

The computational complexity of the automated portion of the procedure (step 1 - step 7) has not been considered in the current work, as our primary focus has been to reduce the number of tested drug combinations needed to identify the best drug combination. This makes sense as testing of drug combinations is much more time consuming compared to the computational time associated with running the automated process for a single signaling pathway. However, as we try to create a unified picture of the signaling in various cellular pathways to better target cancer it will become much more pertinent to try and find ways to reduce computational complexity.

In our work the proposed method has been applied to pathways which satisfy the properties of fixed monotonic network and thus can be modeled as such. There may be many pathways which do not behave as a fixed monotonic network. In such cases we could make use of our method by looking at a portion of the pathway which might behave as a fixed monotonic network. This will allow us to apply our method and identify the best possible drug combination which might be used for that sub pathway. Another way to deal with this problem, would be to extend our method to monotonic networks where we do not simplify the network as a digital circuit but rather make use of the monotonic relationship between inputs and outputs of the pathway. This would definitely increase the scope of our method.

Some other work such as [16, 26] have considered the problem of identifying the best possible drug combination for a cancer patient. In [16] they have considered networks with single mutations only and best drug combination is identified by categorizing different mutations into different classes on basis of their response to a test input and then mapping each fault locations onto the appropriate drug combination. In this section the idea is not to identify mutations or categorize mutated networks but rather use results from a smaller number of test cases to try and predict output for other drug combinations and utilize that to find the best drug combination. Also, our method is designed to deal with signaling pathways with multiple mutations as compared to [16].

The method as developed here can be used for treating cancer where a single type of mutated signaling pathway is present in all the cells making up the cancerous tissue. However, more often than not, and especially in the case of advanced disease, the cancerous tissue is heterogeneous in

nature. In other words, they have different kinds of mutated signaling networks. Work on how to identify best possible drug combination for such heterogeneous tissue has been considered in [26] and a refined version is presented in the section 3 [4]. Although this section deals with heterogeneous networks but it requires a prior knowledge of the mutated networks present in the heterogeneous mix. In this section we do not assume any prior knowledge of the mutation location whereas we deal with homogeneous cancer tissue. Thus, a possible direction of research is to develop methods to deal with heterogeneous cancerous tissue when the mutations are apriori unknown. This approach has been dealt with in section 4 [5].

Another aspect which needs to be considered is when mutations introduce new signaling connections into the existing pathway structure. This is better represented by bridging faults [17]. Finally, we have considered only networks where there are no feedback loops but in certain pathways feedback is present. Preliminary work on behavior prediction of such feedback networks has been considered in [34]. In the next two sections, this aspect is implicitly taken care of. Making substantial progress on each of the problems mentioned above should go a long way towards bringing our results to the domain of practical implementation.

2.6 Conclusion

In this section, we have investigated the properties of fixed monotonic Boolean networks and tried to exploit the monotonic nature of such pathways. These properties were applied to digital equivalents of cellular signaling pathways. We have shown how these properties can be utilized in combination therapy design for cancer to efficiently find the best possible drug combination with minimal side effects from the available spectrum of drugs. Another interesting feature of this method is that we do not need to actually identify exact mutations but can instead focus on the output of the signaling pathways which is what matters while treating cancer. The advantage of looking at outputs rather than exact mutations is that even if there is an unknown secondary signaling mechanism through which the mutation might affect the outputs our method would intrinsically take that into account as we are trying to modify the outputs instead of targeting particular mutations. As we also saw in the previous subsection there are some issues when Boolean network

models are used for signaling pathways. If certain protein-protein interactions are unknown then Boolean network model is unable to accurately capture the essence of the pathway and some steps in the presented algorithm which requires a perfect knowledge of the Boolean structure would not provide us with accurate results. Thus, one of our goals for future improvements would be to make our method more dependent on monotonic properties of the network rather than knowing the exact details of the protein-protein interactions to avoid this problem. Also, another issue involving this method is with regards to choosing a threshold. As was mentioned earlier the quantization of expression values to two states leads to loss of information in some cases, especially in the cases where the expression value lies close to the threshold. Perhaps in the future to avoid this issue instead of a sharp threshold we could define a buffer region where expression values are not mapped to either the state 0 or 1 and the method only utilizes the outputs which have a clearly defined state.

As we keep on building upon this method we might need to take into account more biomarkers (including DNA, RNA, protein and metabolomic profile, etc.) associated with various kinds of tumor before we have a viable treatment method. In the next two sections, we extend the scope of this method to the domain of heterogeneous cancerous tissue which implicitly deals with bridging faults and pathways with feedback.

3. UNDERSTANDING CANCER TISSUE HETEROGENEITY USING BAYESIAN HIERARCHICAL MODELS ALONG WITH EVOLUTIONARY GAME THEORY

3.1 Introduction

Cancer is generally a result of the introduction of various mutations within cells, which in turn make the cells neoplastic and give them a selective proliferative advantage over their normal neighbors [35]. To target cancer, we need a methodology to understand and represent the various biochemical regulatory processes that occur within the cell. Information collected by biologists over a long period of time has resulted in a wealth of available information about cellular biochemical processes and their regulation. Thus, it is apt to use such knowledge, also known as signaling pathways, while devising methods for the treatment of diseases such as cancer which result from a breakdown of cellular regulatory processes. A method making use of prior pathway knowledge to build a Boolean signaling model was presented in [16]. In this work the proteins within a cell which are in an active state are represented by the Boolean state 1 while the proteins which are in an inactive state are represented by the Boolean state 0. The direct interaction of the various proteins in a cell has been represented by the use of logic gates. In a normal cell, the state of the upstream protein dictates the state of the downstream proteins but that may no longer hold in the case of a mutated cell. In many of these cases, the mutations can thus be rightly represented by stuck-at-faults which means a protein is either perpetually active (stuck-at-one) or rendered completely inactive (stuck-at zero) regardless of what states the upstream protein may be in. In certain cases, the mutations can also be represented by a bridging fault.

Tumors can be such that the nature of mutations in all the cells are similar. In such cases the tumors are said to be homogeneous in nature. The work in [16] showed how to design therapy for tumors where there is a single stuck-at mutation along the signaling pathway of the cancerous cells. This work has been further extended in the work [3], as shown in section 2, where tumors of a homogeneous nature but with multiple mutations, are handled by making use of the concept

of monotonicity in Boolean networks. More often than not, however, tumors tend to have heterogeneous subpopulations i.e. different cells of the tumor tend to have a different set of mutations [35, 22] which can make the problem of treating cancer more complex. The work done in [26] takes a step in this direction and shows how to deal with such heterogeneous tumors assuming that the nature of mutations in the various subpopulations are known beforehand. The accuracy of compositional breakup prediction achieved in that work requires improvement and accomplishing that will be a major focus of this section. Another factor which also needs to be taken into account for cancer therapy design is the interaction between the various subpopulations of the tumor [36]. In this section, we will also try to combine the knowledge of proportional breakup of various subpopulations and the interactive relationship between them to identify the subpopulation which needs to be targeted so as to improve and customize cancer treatment for patients.

In the subsections 3.2, 3.3 we design two Bayesian hierarchical models which built upon the work in [26] and are shown to improve the accuracy. We then move on to make use of evolutionary game theory to exploit the dynamics at play amongst the various sub-populations and use it along with the knowledge of proportional breakup of the subpopulations to identify the target sub-population. Finally, we present a brief discussion of the results and a concluding subsection.

3.2 Bayesian hierarchical model I

In this subsection we will present a modified Bayesian hierarchical model [26] which will allow us to estimate the proportional breakup of the various tumor subpopulations. In case of a heterogeneous tissue there are different subpopulations each with a different set of mutations in the cell. Thus, it can be clearly seen that due to difference in the nature of mutations in these different subpopulations they will respond differently to the application of various drug stimuli. This difference in response is what will help us in identifying the proportional breakup of the various sub populations in the tumor tissue. One thing we should note here is that the observed values could be gene/protein expression or it could be something else which allows us to differentiate between the various subpopulations.

The authors in [16] presented a Boolean model of the growth factor signaling pathway which

has been reproduced here in Figure 3.1. This model represents the signaling within a normal cell which is free of mutations. In case of mutated cells there will be stuck-at-faults at various locations. To represent a heterogeneous population, we will have to use an ensemble of Boolean networks where each network describes a different subpopulation. Next, we describe the model below.

3.2.1 Intuition for the model

Let us consider a hypothetical scenario where there are three subpopulations in a tumor. Let us suppose that the cells in the first subpopulation have a mutated PTEN gene. Thus, in this case the PTEN protein is downregulated as it is supposed to be a tumor suppressor gene [37]. This can also be seen from Figure 3.1 where an inactive PTEN will lead to an upregulated output. In the second subpopulation let us assume that the Ras gene [22] has been mutated and in the third subpopulation, suppose the mTOR gene [38] has been mutated. Our goal is to identify the proportional breakup of these subpopulations in the tumor.

Suppose we observe the expression levels of BCL2 protein in the signaling pathway model shown in Figure 3.1(a) in the two scenarios when the first drug combination (U0126, LY294002) is applied and the second drug combination (U0126, LY294002, Temsirolimus) is applied. The usual method to compute normalized expression values is by using the delta-delta method [29] with GAPDH being used as the housekeeping (reference) gene for the first normalization step and the second normalization step involving a control experiment. In our case for the control experiment we simply observe the expression levels of BCL2 without the application of any of the drugs.

A useful way to model the normalized expression values [26] is to represent them as ratios between two normally distributed variables with a standard deviation value which is proportional to the mean. The constant of proportionality between the standard deviation and the mean is called the coefficient of variation. This coefficient of variation is assumed to be the same for all such normally distributed variables. A justification for making this assumption is provided in [39].

In order to get an intuitive feel for how the model can be used, we initially assume that the coefficient of variation is 0. Then we assume that when a protein is downregulated in a subpopula-

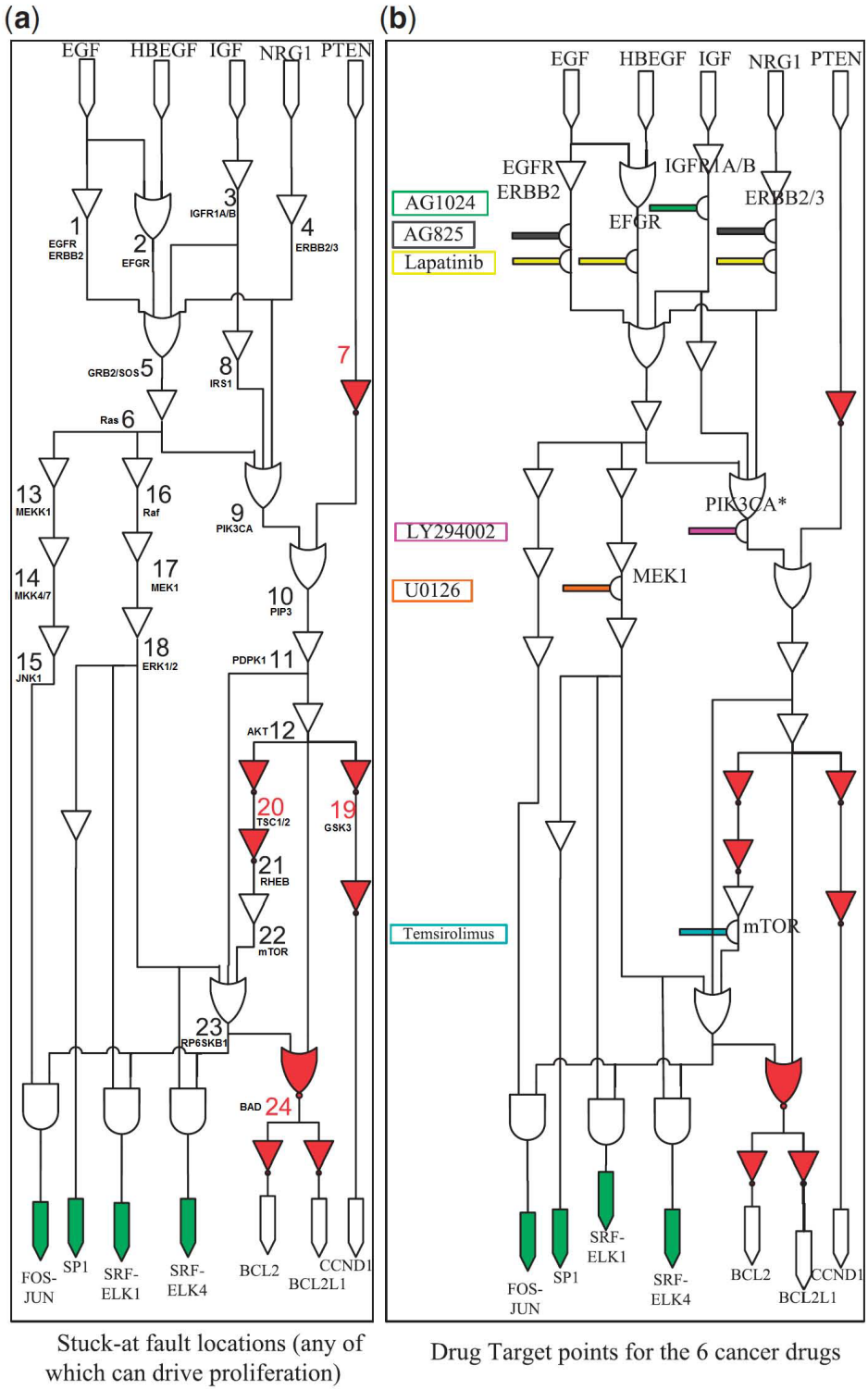

Figure 3.1: (a) The Boolean model of the growth factor signaling pathway along with the proteins which are involved in the signaling (b) shows the protein kinase inhibitors and their target proteins along the signaling pathway. (Reproduced from [3])

Table 3.1: The upregulated states are represented by 1 and the down regulated states are represented by a 0.

	No Drugs Applied	U0126 + LY294002	U0126 + LY294002 + Temsirolimus
Subpopulation 1	1	1	1
Subpopulation 2	1	0	0
Subpopulation 3	1	1	0

tion the contribution of that subpopulation to the expression level of that protein is θ and when the protein is upregulated the contribution of that subpopulation to the expression level of that protein is 1. Let us represent the impact of subpopulation i on the expression level of protein p normalized with respect to the housekeeping gene by $\alpha_{p,i}$. Similarly, when the protein is downregulated its impact is $\alpha_{p,i} \times \theta$. The table 3.1 shows whether the protein BCL2 is upregulated or downregulated for the three defined subpopulations in the presence of the drug combinations being considered. This can also be predicted on the basis of the model presented in figure 3.1. This response profile of the observable (i.e the protein expression) will now be referred to as the expression profile. For example, when drugs U0126 and LY294002 are applied the expression profile is $[1\,0\,1]$ across the three subpopulations.

Let us represent the final normalized expression value for the protein p under the impact of drug combination j on the heterogeneous tumor sample as $\gamma_{p,j}$. Now we can obtain an expression for the observed normalized expression values in terms of α's. The value will be the summation of the impact of the various subpopulations under the effect of the drug combination and then normalized by values obtained by the control experiment.

$$\gamma_{BCL2,1} = \frac{\alpha_{BCL2,1} + \alpha_{BCL2,3} + \theta \times \alpha_{BCL2,2}}{\alpha_{BCL2,1} + \alpha_{BCL2,2} + \alpha_{BCL2,3}} \tag{3.1}$$

where U0126+LY294002 is the drug combination 1.

$$\gamma_{BCL2,2} = \frac{\alpha_{BCL2,1} + \theta(\alpha_{BCL2,2} + \alpha_{BCL2,3})}{\alpha_{BCL2,1} + \alpha_{BCL2,2} + \alpha_{BCL2,3}} \tag{3.2}$$

where U0126+LY294002+Temsirolimus is the drug combination 2.

As we are interested in the proportional breakup of the various subpopulations we can safely assume that:

$$\alpha_{BCL2,1} + \alpha_{BCL2,2} + \alpha_{BCL2,3} = 1 \tag{3.3}$$

Thus, we can obtain the equations 3.4 and 3.5 as below;

$$\alpha_{BCL2,1} + \theta \times \alpha_{BCL2,2} + \alpha_{BCL2,3} = \gamma_{BCL2,1} \tag{3.4}$$

$$\alpha_{BCL2,1} + \theta(\alpha_{BCL2,2} + \alpha_{BCL2,3}) = \gamma_{BCL2,1} \tag{3.5}$$

Hence, from equations 3.3, 3.4 and 3.5 we can easily compute the proportional breakups of the various subpopulations in the tumor as we have three independent equations in three variables. This is the intuition which forms the basis of the model presented in [26] assuming that $\theta = 0$, that is ignoring θ. However, we will update this by taking into account the fact that when a protein in downregulated it does not have zero impact as assumed. In fact, we can see from the experimental data in [3] and [26] that even in the downregulated state, the expression values for a protein need not be zero. Hence let us assume that the average value of a downregulated observable (protein/gene/etc.) is non-zero and is stored in θ. The benefit of assuming a non-zero θ is that we can also account for the variation in the expression values when the observable is downregulated, which is impossible for a zero valued θ due to the use of coefficient of variation to obtain the standard deviation. This non-zero value for θ is obtained from prior knowledge.

Another important thing to keep in mind is that the contribution of the various subpopulations on the various observables need not be the same as the relative ratio of the subpopulations [26]

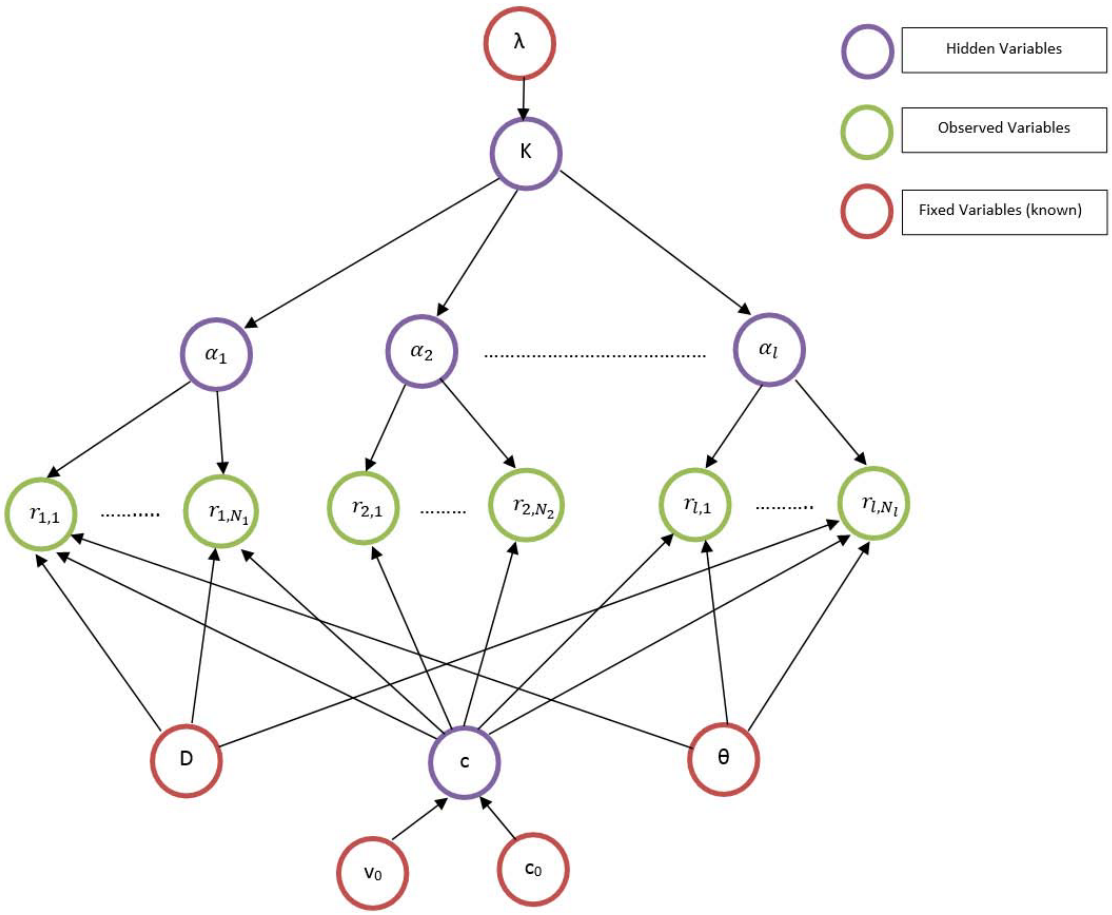

Figure 3.2: Hierarchical Model I showing all the conditional dependencies of the latent variables and observed variables. The observed variables are shown in green, the latent variables are shown in purple and the fixed variables which are either used to fine tune the model or represent some prior knowledge in the model are shown in red.

themselves. Thus, we need to have varying relative ratio parameters for each observable while making use of the information obtained on the basis of observation of these observables. Keeping this in mind we move on to describing the model.

3.2.2 Model Description

The hierarchical model is presented in Figure 3.2. Here we can see the conditional dependencies amongst the variables. First, we define the variables in the model. In the model let us assume that there are l observables and N subpopulations. The observable could refer to gene/protein or it could be something else too as we shall see later in this section. Let K denote the concentration

parameter of the Dirichlet prior on the distribution of subpopulations within the tumor and also the distribution of the contribution of the subpopulations to the various observables. The relative ratio parameter $\alpha_i = [\alpha_{i,1}, \alpha_{i,2}, \cdots, \alpha_{i,N}]^T$ represents the gene expression ratio for each gene i across all the subpopulations. θ, which was first defined in the previous subsubsection 3.2.1, represents the mean value associated with the absence of activity across all the observables. $d_{i,j}$ is an N dimensional expression profile representing the presence or the absence of activity for the j^{th} observation of the i^{th} observable across all the subpopulations. For example $d_{i,j} = [1\,0\,1]^T$ means that for the j^{th} observation the observable i is active for the first and third subpopulations but inactive for the second subpopulation. D is the collection of all such $d_{i,j}$'s. $r_{i,j}$ is the j^{th} normalized observation for the i^{th} observable. The normalization to obtain $r_{i,j}$ is done against the value obtained from the control experiment where the observable is active in all the subpopulations. c is the coefficient of variation associated with the observations made for the various observables as defined in the previous subsubsection 3.2.1. λ is the parameter which controls the exponential prior distribution of K. ν_0, c_0 are the parameters of the inverse-gamma distribution which forms the prior on the hidden variable c.

Now we present the mathematical relationship between the child variables and their parent variables in this hierarchical model. Equations 3.6 - 3.8 show the distributions from which the random variables are derived. In equation 3.6, the Gamma distribution has a shape parameter $\frac{\nu_0}{2}$ and a rate parameter equal to $\frac{\nu_0 c_0^2}{2}$. In equation 3.7, λ is the rate parameter of the exponential distribution. In equation 3.8, as mentioned earlier, K is the concentration parameter of the Dirichlet distribution.

$$\frac{1}{c^2} \sim \Gamma(\frac{\nu_0}{2}, \frac{\nu_0 c_0^2}{2}) \tag{3.6}$$

$$K_i \sim Exp(\lambda) \tag{3.7}$$

$$\alpha_i \sim Dir(K) \tag{3.8}$$

Next we define a variable $b_{i,j}$ in equation 3.9. $b_{i,j}$ is a modified expression profile obtained from $d_{i,j}$ and represents the fact that an inactive observable need not have zero expression level.

$$b_{i,j} = d_{i,j} + \theta \times (\mathbf{1} - d_{i,j}) \tag{3.9}$$

where **1** represents a vector of ones.

Equations 3.10 - 3.12 tells us how the variable $r_{i,j}$ is sampled given the parent variables shown in the hierarchical model presented in Figure 3.2. Note that $\mathcal{N}(x, y)$ represents a normal distribution with mean x and standard deviation y.

$$O_{i,j} \sim \mathcal{N}(b_{i,j}^T \alpha_i, c b_{i,j}^T \alpha_i) \tag{3.10}$$

$$T_{i,j} \sim \mathcal{N}(u^T \alpha_i, c u^T \alpha_i) \tag{3.11}$$

where u is simply a vector of ones

$$r_{i,j} = \frac{o_{i,j}}{t_{i,j}} \tag{3.12}$$

Note that in equation 3.11 the vector of ones is used instead of $b_{i,j}$ in order to represent the control case where the observable is active for all the subpopulations. As α_i is derived from a Dirichlet distribution all its elements add up to a one. Thus in equation 3.11 it is interesting to note that the variable $t_{i,j}$ is obtained from a normal distribution centered around the mean 1 with a standard deviation c. Now we need to establish a direct relationship between $r_{i,j}$ and its parent variables c, θ, α_i and $d_{i,j}$. Using a method similar to that shown in [26] we try to find the probability density function over the space $(R_{i,j}, T_{i,j})$ given the density function over the space $(O_{i,j}, T_{i,j})$. Note that the random variables $O_{i,j}$ and $T_{i,j}$ are independent of one another so the joint

distribution is simply a product of the marginal distributions of $O_{i,j}$ and $T_{i,j}$. In order to obtain the distribution for the transformed space, we need to compute the Jacobian J as shown in equation 3.13.

$$J = \begin{vmatrix} T_{i,j} & R_{i,j} \\ 0 & 1 \end{vmatrix} = |T_{i,j}| \simeq T_{i,j} \tag{3.13}$$

The reason for approximating the modulus of $T_{i,j}$ is that normally in such scenarios the distribution has a very thin tail in the negative region due to the values of the coefficient of variation c being small [39]. This gives us equation 3.14 and integrating over $T_{i,j}$ we obtain the density function for $R_{i,j}$ in equation 3.15.

$$P_{R_{i,j},T_{i,j}}(r_{i,j}, t_{i,j}) = P_{O_{i,j}}(o_{i,j}) \times P_{T_{i,j}}(t_{i,j}) \times J \tag{3.14}$$

$$\begin{aligned} P(r_{i,j}|c, \alpha_i; D, \theta) \simeq & \frac{m_{i,j}(r_{i,j} + m_{i,j})}{\sqrt{(}2\pi c)(r_{i,j}^2 + m_{i,j}^2)^{3/2}} \\ & \times exp(-\frac{1}{2c^2}\frac{(r_{i,j} - m_{i,j})^2}{r_{i,j}^2 + m_{i,j}^2}) \end{aligned} \tag{3.15}$$

where $m_{i,j} = b_{i,j}^T \alpha_i$.

Note that $exp(k)$ refers to e^k, where as $Exp(k)$ refers to an exponential distribution with rate parameter k. Once we have defined the relationship of variables with their parent variables the primary goal is now to estimate the posterior distribution of K which in turn would give us an idea of the distribution of the relative ratio of the subpopulations within the tumor. In Bayesian modeling one way to achieve this is to make use of Markov chain Monte Carlo (mcmc) techiniques. In order to implement the mcmc methods we first need to compute the full conditionals of the unknown hidden variables. The derivation of full conditionals is similar to those shown in [26]. The full conditionals are shown in the equations 3.16 through 3.18.

$$P(\frac{1}{c^2}|\alpha, K, r; D, \theta) = \Gamma(\frac{\nu_0 + \Sigma_{i=1}^{l} n_i}{2}, \frac{\nu_0 c_0^2 + \Sigma_{i,j} \frac{(r_{i,j} - m_{i,j})^2}{(r_{i,j}^2 + m_{i,j}^2)}}{2}) \tag{3.16}$$

$$P(K|\alpha, c, r; D, \theta) \propto P(K) \times (\frac{\Gamma(\Sigma_{q=1}^{N} K_q)}{\Pi_{q=1}^{N} \Gamma(K_q)})^l \times \Pi_{q=1}^{N} (\Pi_{i=1}^{l} \alpha_{i,q})^{K_q - 1} \tag{3.17}$$

$$P(\alpha_i|K, c, \alpha_{-i}, r; D, \theta) \propto \frac{m_{i,j}(r_{i,j} + m_{i,j})}{\sqrt{(2\pi c)}(r_{i,j}^2 + m_{i,j}^2)^{3/2}} \times exp(-\frac{1}{2c^2} \frac{(r_{i,j} - m_{i,j})^2}{r_{i,j}^2 + m_{i,j}^2}) \times \Pi_{q=1}^{N} \alpha_{i,q}^{K_q - 1} \tag{3.18}$$

where α_{-i} represents all the α_j's such that $j \neq i$ and $m_{i,j}$ are as defined earlier in equation 3.15. Now before we proceed further we need to assign values for fixed variables like D, θ, v_0, c_0 and λ. D is derived from the prior knowledge about an observable being active or inactive in a subpopulation for each of the observations. For example, in the case where protein expression is the observable we can predict whether the protein is active or inactive using the signaling pathway knowledge as shown in Figure 3.1. θ is obtained from prior knowledge as mentioned in the previous subsubsection 3.2.1. v_0, c_0 and λ are chosen such that the prior distribution for K and c are flat compared to the posterior distribution. The reason we choose such parameter values is so as to represent our ignorance about the distribution of the hidden parameters K and c.

3.2.3 Model Implementation

We will be making use of the Metropolis-Hastings algorithm [40, 41, 42] to sample from the posterior distribution of the hidden variables K, c and α. The reason for using Metropolis-Hastings is that the full conditionals obtained in equations 3.16 - 3.18 are not standard distributions from which we can easily sample data points as required for Gibbs sampling [43, 44] are not easily invertible as required by slice sampling [45] and we need to make use of proposal distributions which might not be symmetric as required by the Metropolis algorithm [46]. The use of full

conditionals in the Metropolis-Hastings algorithm or any other mcmc algorithm for that matter allows us to sample from the true posterior distribution of each of the hidden unknowns in the hierarchical model.

Although c can be generated from its full conditional as it involves the use of inverse-gamma distribution but K and α have full conditionals which are not standard distributions. Thus, in Metropolis-Hastings we make use of the proposal distributions to generate new samples. In the case of K we use a symmetric proposal distribution. We sample an element K_i^* from the uniform distribution over the closed interval $[K_i - U_K, K_i + U_K]$ where K_i is the i^{th} element of K and U_K is a tuning parameter which needs to be chosen carefully so as to allow for good mixing and ensuring that the value of K is also updated often. We set the value of U_K as 10. If $K_i^* < 0$ then update the value of K_i^* to $-K_i^*$. Using the method described in [42] we calculate the acceptance ratio AR_K as shown in equation 3.19. Then we update K to K^* with a probability of $min(1, AR_K)$.

$$AR_K = \frac{P(K^*|\alpha, c, r; D, \theta)}{P(K|\alpha, c, r; D, \theta)} \tag{3.19}$$

In order to sample from the full conditional of α, we follow a similar procedure as mentioned for K. Here we choose a new value of α_i^* from a Dirichlet distribution (i.e. the proposal distribution) with the concentration parameter $U_\alpha \times \alpha_i$. We chose a value of 100 for the tuning parameter U_α during the course of our simulation. Then we compute the acceptance ratio AR_{α_i} as shown in equation 3.20 where $Dir(x|C_p)$ refers to the probability of x given a Dirichlet distribution with concentration parameters C_p. We update α_i to α_i^* with a probability of $min(1, AR_{\alpha_i})$.

$$AR_K = \frac{P(\alpha_i^*|K, c, \alpha_{-i}, r; D, \theta) \times Dir(\alpha_i|U_\alpha \times \alpha_i^*)}{P(\alpha_i|K, c, \alpha_{-i}, r; D, \theta) \times Dir(\alpha_i^*|U_\alpha \times \alpha_i)} \tag{3.20}$$

In the prior distributions of c we use a value of 1 for v_0 and a value of 0.1 for c_0 in equation 3.6. We set λ as 0.01 for the prior distribution of K as defined in equation 3.7 so as to provide a large range for K but also to make sure that the value of K does not blow up during the mcmc process. The mcmc method requires us to sample iteratively from the full conditionals and letting the pro-

cess run till it attains stationarity. We use a burn-in period of 100000 and drop the initial 100000 samples so that the Markov chain attains stationarity and the samples are more representative of the posterior distribution of the latent variables. We ran the Markov chain for 10000000 iterations after the burn-in period and used a thinning factor of 1000 i.e. picked 1 sample out of each 1000 samples in order to reduce the correlation between consecutive samples and make sure that the data were well mixed. Thus, we obtained a total of 10000 samples. Finally, in order to plot the marginal posterior distribution of the relative ratio parameter representing the proportion of various subpopulations we use the samples of K generated during the mcmc process and use it as concentration parameters of the Dirichlet distribution as mentioned earlier in the model to sample α. Then, we used the inbuilt function 'ksdensity' in MATLAB which uses a Gaussian kernel and logarithmic transformation for boundary correction [47, 48] in order to plot the posterior probability density function (pdf) of the hidden variables (primarily for the relative ratio parameters). Finally, we use the mode of the pdf for the relative ratio parameter to give the final estimate of the proportion of the various subpopulations.

3.2.4 Application on qPCR data

In this dataset, normal adult fibroblast cell lines are used and normalized gene expression ratios are obtained using qPCR. The details of the experimental process undertaken to obtain this data are given in [3, 26]. We look at the reporter genes that have the response elements for the transcription factors SP1 and SRF-ELK (please see to Figure 3.1). EGR1 is a reporter for SRF-ELK [49] and cMYC, JUN, IRF3 are reporters for the transcription factor SP1 [31, 32, 50]. Some more genes which report on SP1 are BIRC5, Decorin, VEGFA [51, 52, 53, 54]. The normalization to obtain the final gene expression values is done with respect to the control case where no drugs were applied [3, 26].

In order to obtain the value of θ we take a simple average over all the expression data presented in [3] for which a gene is inactive. This value of θ came out to be 0.37337. The observation dataset used was the complete qPCR gene expression dataset presented in [55] and also has been presented here in Table 3.2 for the convenience of the reader. The table shows the expression profile $d_{i,j}$ and

the observation data $r_{i,j}$ obtained from the experiments. As the data obtained is based on a single type of cell line which corresponds to a normal subpopulation without mutations we assume two fictitious mutated subpopulations. The second subpopulation is assumed to have a stuck-at-one fault at ERK1/2 and the third subpopulation is assumed to have a stuck-at-one fault at the SRF-ELK1 and SRF-ELK4 locations. Now given the mutation location, type of mutation and drug applied we can predict the output for the reporter genes on the basis of Figure 3.1 and obtain the expression profile $d_{i,j}$ for the subpopulations. The drug combinations applied are only LY294002 for group 1 and LY294002 with U0126 for group 2 which would result in the $d_{i,j}$s shown in table 3.2.

The dataset was fed to the model and as mentioned earlier we expect that subpopulation 1 (i.e. the normal cells) should have a relative ratio equal to 1 and the other two subpopulations (i.e. the mutated cells) should have a relative ratio equal to 0 each. Our calculations give us results which are very close to the actual values. We also ran the method presented in [26] using the same parameter values as described in our model while ignoring the variable θ. The results for our model and the one in [26] (i.e. without θ) are given in table 3.3. Figure 3.3 shows the posterior distribution of the elements of the relative ratio parameter as derived from the data using our model. We can see that the model presented in this section clearly outperforms the model presented in [26] and is very close to the actual relative ratio parameter values.

3.2.5 Application on Fluorescence data

In this dataset, cells from three different human cancer cell lines were used. The cell lines are HCT116 (colon carcinoma), A2058 (metastatic melanoma) and SW480 (colorectal adenocarcinoma). The cell lines are marked with specific fluorophores so that they emit specific wavelengths of light under excitation which helps in identifying the cells [56]. The HCT116 cell lines were marked with red wavelength emitting fluorophores and A2058 was marked with green wavelength emitting fluorophores. Two different mixtures using these three cell lines in varying proportions were considered and no drugs were applied to these mixtures. The general procedures for preparing the samples are described in [56].

Table 3.2: This table shows the expression profile (i.e. the expected state of the observable (gene) for the first, second and third subpopulations) and the observation data (i.e. the normalized gene expression values). (Reprinted from [55] © 2016, IEEE.)

Observables	Group 1		Group 2	
↓	Expression profile ($d_{i,j}$)	Observation ($r_{i,j}$)	Expression profile ($d_{i,j}$)	Observation ($r_{i,j}$)
EGR1	1 1 1	0.5987 0.7320 0.5586 0.6199	0 1 1	0.4796 0.2892 0.2535 0.2698
JUN	1 1 1	0.4931 0.6736 0.6598 0.7792	0 1 0	0.1550 0.2793 0.3015 0.3415
BIRC5	1 1 1	0.5799	0 1 0	0.3842
cMYC	1 1 1	0.3209 0.2852 0.3439 0.2994	0 1 0	0.2570 0.2059 0.2717 0.2679
Decorin	1 1 1	0.0819 0.0728 0.3345 0.4353 0.4601 0.4147 0.4323	0 1 0	0.0087 0.0242 0.1661 0.2793 0.3789 0.3737 0.3536
IRF3	1 1 1	0.5176 0.4204 0.3560 0.3345	0 1 0	0.2624 0.2553 0.2253 0.2031
VEGFA	1 1 1	0.4444 0.4989 0.5176 0.5105	0 1 0	0.3164 0.4623 0.4633 0.3660

The red/green emission level for each of the individual cells in each of the samples was measured but we do not make use of that information in our model. The average green wavelength emission level and red wavelength emission level were the observable attributes for this experiment. Each mixture had a total of 18 replicates for which emission level was measured and thus we had a total of 18 average emission values for each of the green and red wavelengths. The activity of each single cell line was separately measured too but we only make use of the average emission level for green/red wavelengths when they are supposed to be active and inactive (for

Table 3.3: Comparison of the estimated relative ratio values using our model and model presented in [26] with the actual relative ratios using qPCR data.

Relative Ratio Parameters ↓	Subpopulation 1	Subpopulation 2	Subpopulation 3
Actual	1	0	0
Estimated (with θ)	0.9705	0.0092	0.0203
Estimated (without θ)	0.6513	0.2824	0.0663

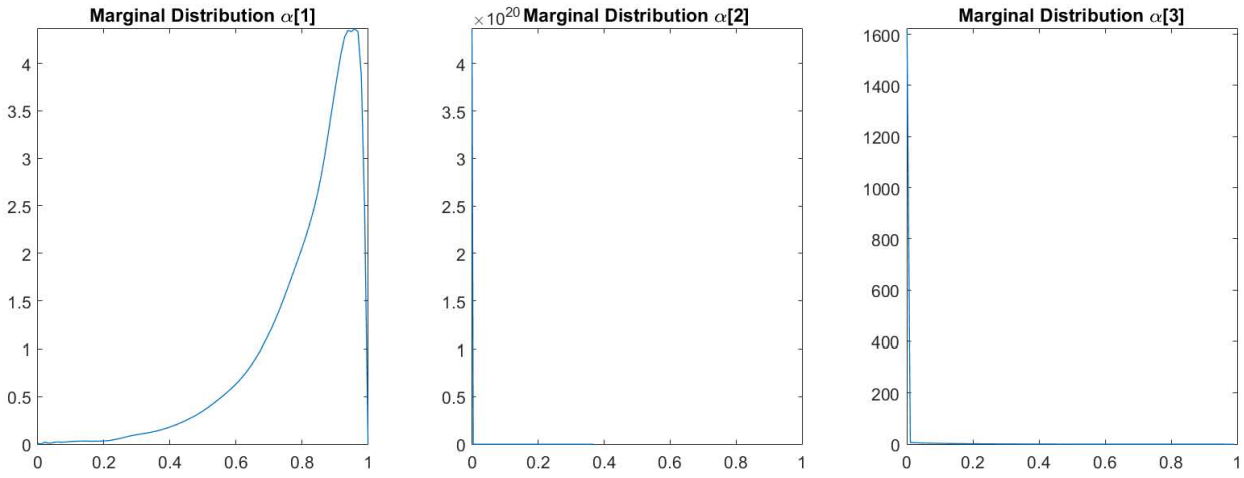

Figure 3.3: The marginal posterior distribution for each of the relative ratio parameters are shown here. This shows that the normal subpopulation consists of almost the entirety of the population.

example in the case of the A2058 cell line, green wavelength fluorescence is supposed to be active and red wavelength spectrum is supposed to be inactive) separately over all cell lines and all replicates.

The average activity level for an active green/red emission wavelength is used to normalize all our observations for the mixtures for green/red emission values respectively. The value of θ is calculated as follows. First, the green signals associated with the cell lines where the green is supposed to be inactive are averaged out. Thereafter, the same procedure is repeated for the cell lines where the green is active. Then dividing the inactive average by the active average, we obtain

Table 3.4: This table shows the expression profile (i.e. the expected state of the observable (green/red emission) for the first, second and third subpopulations) and the observation data (i.e. the normalized emission values).

Mixtures →	Mixture # 1				Mixture # 2			
Observables →	Green		Red		Green		Red	
Replicate Number ↓	$d_{i,j}$	$r_{i,j}$	$d_{i,j}$	$r_{i,j}$	$d_{i,j}$	$r_{i,j}$	$d_{i,j}$	$r_{i,j}$
1	[0 1 0]	0.36	[1 0 0]	0.41	[0 1 0]	0.18	[1 0 0]	0.43
2		0.44		0.40		0.28		0.19
3		0.43		0.29		0.12		0.55
4		0.48		0.38		0.25		0.31
5		0.31		0.50		0.17		0.36
6		0.44		0.41		0.12		0.40
7		0.52		0.35		0.19		0.28
8		0.52		0.29		0.10		0.33
9		0.53		0.22		0.18		0.26
10		0.41		0.37		0.19		0.47
11		0.46		0.31		0.11		0.68
12		0.43		0.45		0.37		0.20
13		0.38		0.31		0.21		0.33
14		0.43		0.47		0.21		0.35
15		0.46		0.40		0.28		0.42
16		0.27		0.32		0.31		0.66
17		0.42		0.27		0.13		0.42
18		0.40		0.32		0.21		0.42

the θ_{green}. In a similar way, we compute θ_{red} and θ is obtained as the arithmetic mean of θ_{green} and θ_{red}. The value of θ came out to be 0.00289, which is very close to zero. Thus, the individual cell data are only used to contribute to the prior knowledge contained in θ and as the normalizing factor for the average green/red emission levels for the two mixtures so that the value of r can be determined. Table 3.4 shows the expression profile $(d_{i,j})$ and the observation data $(r_{i,j})$ for mixtures 1 and 2.

This dataset was fed to our model and the model described in [26] (without using θ) and the results are compared in table 3.5 for both the mixtures. We can see that for the fluorescent dataset our method performs slightly better than the model in [26].

One of the reasons for the minor difference in the performance is that the value of θ is close to

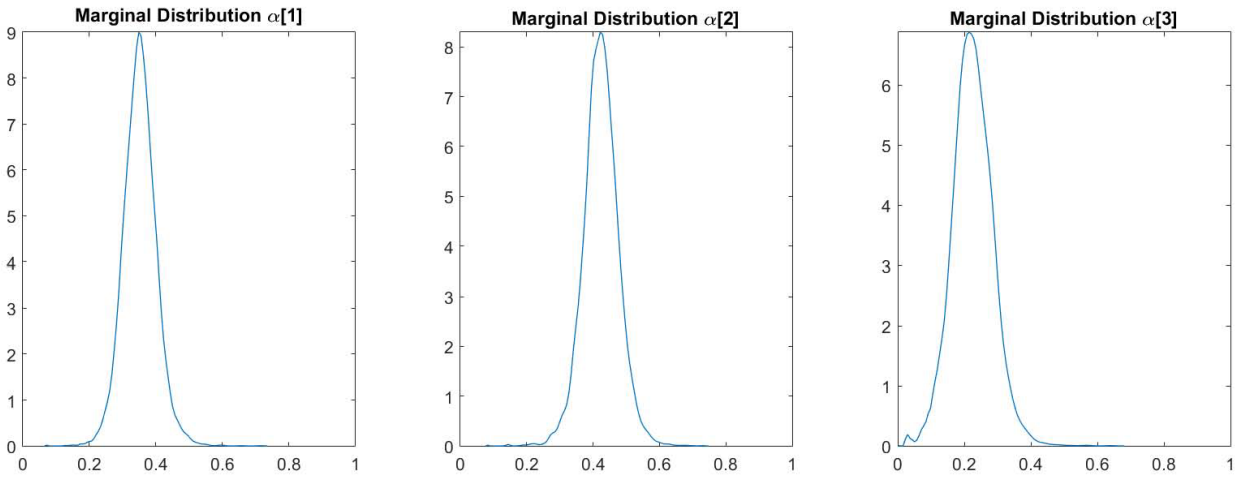

Figure 3.4: The marginal posterior distribution for each of the relative ratio parameters are shown here for mixture 1 of the fluorescent dataset.

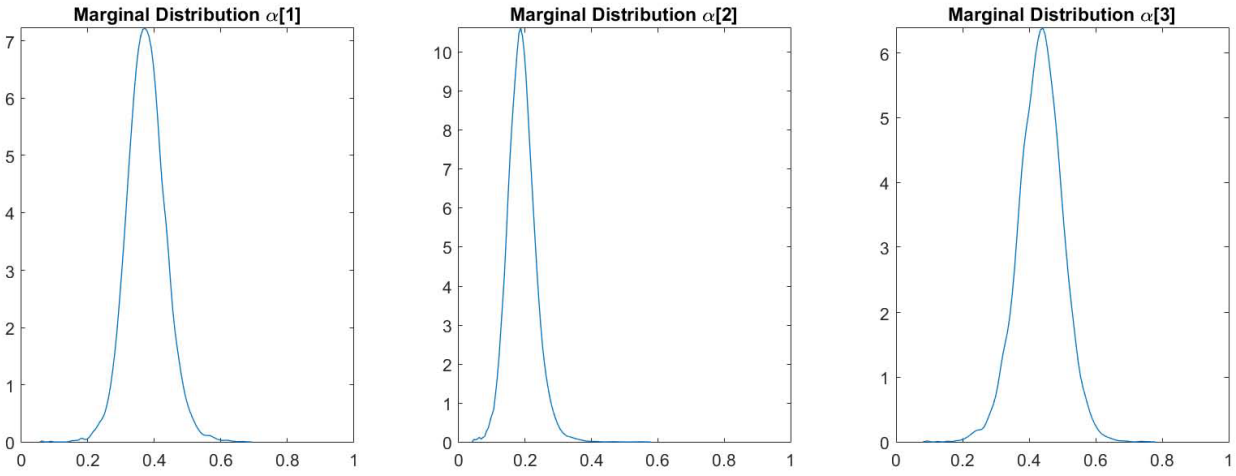

Figure 3.5: The marginal posterior distribution for each of the relative ratio parameters are shown here for mixture 2 of the fluorescent dataset.

Table 3.5: Comparison of the estimated relative ratio values using our model and model presented in [26] to the actual relative ratios using fluorescent data.

Mixture # 1			
Relative Ratio Parameters ↓	HCT116	A2058	SW480
Actual	0.364	0.408	0.228
Estimated (with θ)	0.3535	0.4229	0.2236
Estimated (without θ)	0.3527	0.4245	0.2228
Mixture # 2			
Relative Ratio Parameters ↓	HCT116	A2058	SW480
Actual	0.388	0.200	0.412
Estimated (with θ)	0.3739	0.1872	0.4389
Estimated (without θ)	0.3770	0.1831	0.4399

zero and hence it is as good as setting a θ at zero i.e. a model without θ. We discuss further about this in subsection 3.5. Figures 3.4 and 3.5 show the posterior distribution of the elements of the relative ratio parameter as derived from the fluorescent data using our model.

We can see in the figures how the peaks also closely correspond to the actual proportion of the various subpopulations in the dataset. In the next subsection we will see how we can update the value of θ using the mcmc method when we do not have any prior knowledge of θ.

3.3 Bayesian hierarchical model II

In the previous subsection we saw how we can make use of the prior knowledge of average inactivity levels over all the observables to further improve detection accuracy of the Bayesian hierarchical model. In this subsection, we try to answer the question about what could be done if we do not have the prior knowledge about θ. This scenario can also be resolved using the mcmc method subject to certain conditions. Now let us see what we might need to be able to achieve similar levels of accuracy as model I when θ is unknown.

3.3.1 Brief intuition for the Model

Let us revisit at the example described using equations 3.1-3.5 and table 3.1. Equations 3.3-3.5 are the final equations which were used to solve for the αs. However, the value of θ was known. If θ were unknown then we would need one more equation to solve for 4 unknowns. Hence, this means that for three subpopulations to compute the three relative ratio parameters along with the value of θ we need to have data pertaining to at least four unique expression profiles so that we have four equations to solve for the four unknowns. Thus, if we have n different subpopulations then we would need $n + 1$ independent equations to solve for the relative ratio parameters and θ simultaneously which means we would need the data to have at least $n + 1$ independent expression profiles.

3.3.2 Model Description and Implementation

The model is shown in Figure 3.6. In this case the variable θ is a hidden variable controlled by the shape parameters s and t of a Beta distribution. This relationship is described in equation 3.21.

$$\theta \sim Beta(s, t) \tag{3.21}$$

The conditional dependence of the rest of the variables on their parent variables remain the same as described in equations 3.6-3.15. Now we need to compute the full conditionals for all the hidden variables. Luckily, for the hidden variables c, K and α the equation for the full conditionals are the same as described in equations 3.16-3.18. Thus, we need only compute the full conditional for θ. This is derived and described in equations 3.22 and 3.23 respectively.

$$P(\theta|K, c, \alpha, r; D) \propto P(\theta) \times P(r|\alpha, c, \theta; D) \tag{3.22}$$

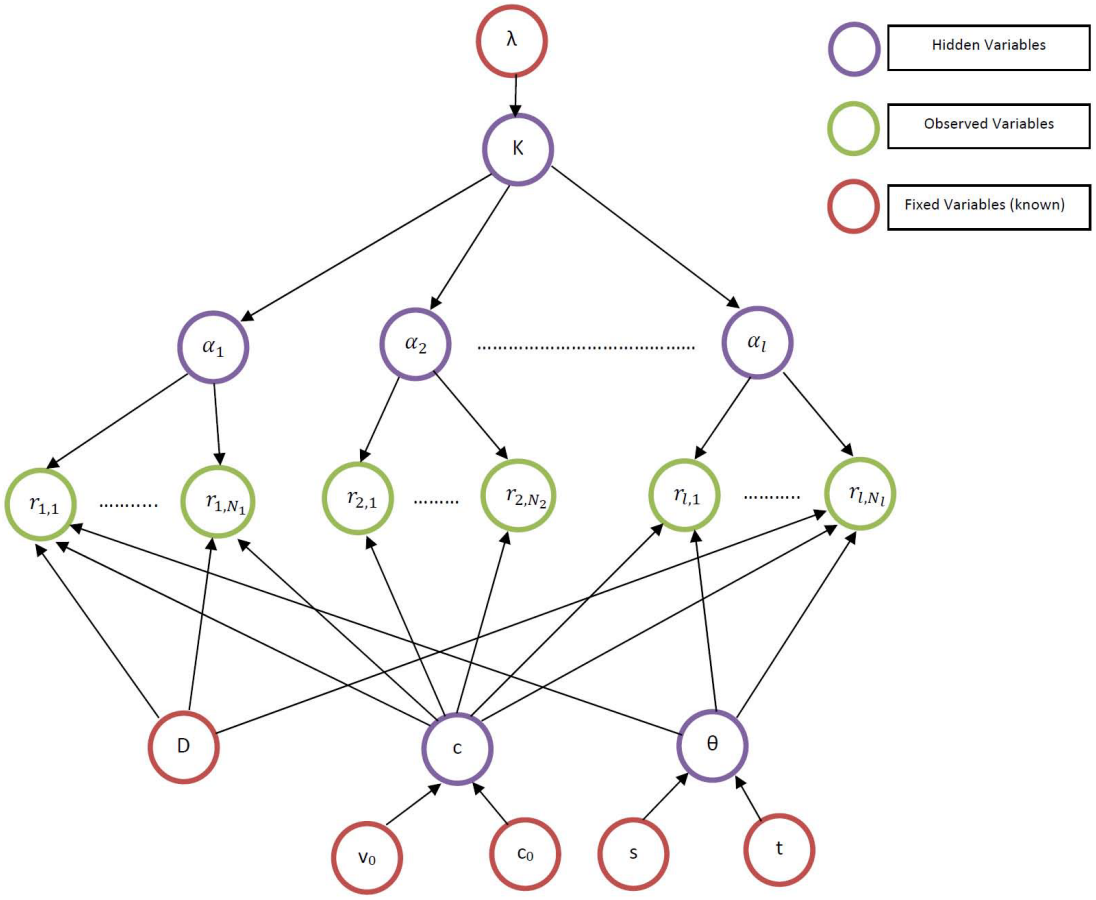

Figure 3.6: Hierarchical Model II showing all the conditional dependencies of the latent variables and observed variables.

$$P(\theta|K,c,\alpha,r;D) \propto \frac{m_{i,j}(r_{i,j}+m_{i,j})}{\sqrt{2\pi c}(r_{i,j}^2+m_{i,j}^2)^{3/2}} \times exp(-\frac{1}{2c^2}\frac{(r_{i,j}-m_{i,j})^2}{r_{i,j}^2+m_{i,j}^2}) \times \theta^s \times (1-\theta)^t \tag{3.23}$$

We used the value $[s,t] = [1,9]$ for the prior parameters. This represents the belief that θ is closer to zero. As the full conditional of θ is a non-standard distribution we need to make use of proposal distribution as required by the Metropolis-Hastings algorithm for implementing the model. Given a current value of θ we choose a new value of θ^* from the Beta distribution with the shape parameters $U_\theta \times \theta$ and $U_\theta \times (1-\theta)$. We used a value of 100 for the tuning parameter U_θ.

Table 3.6: This table gives the updated expression profiles to be used instead of the ones shown in table 3.2 for testing the model II.

Expression profile in table 3.2	Updated expression profile
1 1 1	1 1
0 1 1	0 1
0 1 0	0 0

We update θ to θ^* with a probability of $min(1, AR_\theta)$ where the acceptance ratio AR_θ is defined in equation 3.24.

$$AR_\theta = \frac{P(\theta^*|K, c, \alpha, r; D)}{P(\theta|K, c, \alpha, r; D)} \times \frac{Beta(\theta|U_\theta \times \theta^*, U_\theta \times (1-\theta^*))}{Beta(\theta^*|U_\theta \times \theta, U_\theta \times (1-\theta))} \tag{3.24}$$

The rest of the fixed parameters apart from D (which varies on basis of data) and mcmc process variables are kept the same as in Model I.

3.3.3 Application to qPCR data

We apply the model II described in this subsection to qPCR data. We make use of the same observation data as described in table 3.2. However, as mentioned earlier we need to have data pertaining to enough expression profiles so as to be able to estimate the relative ratio parameters and the value of θ. For testing this model, we assume only two subpopulations, one being the normal subpopulation and the other being a mutated subpopulation with mutations in the locations SRF-ELK1 and SRF-ELK4 in the signaling pathway. Hence, the expression profiles in table 3.2 are updated as shown in table 3.6.

The mcmc process was run and the results are shown in table 3.7 and the posterior distribution of the marginals of the relative ratio parameters are shown in figure 3.7. We see that even when θ is unknown the estimated values are equal to the actual relative ratio parameters up to four decimal places.

Now we move on to the next subsection where we look at how to combine the estimated

Table 3.7: Compares the estimated relative ratio values (up to 4 decimal places) obtained using Model II and the actual relative ratio values.

Relative Ratio Parameters ↓	Subpopulation (Normal)	Subpopulation (Mutated)
Actual	1	0
Estimated	1.0000	0.0000

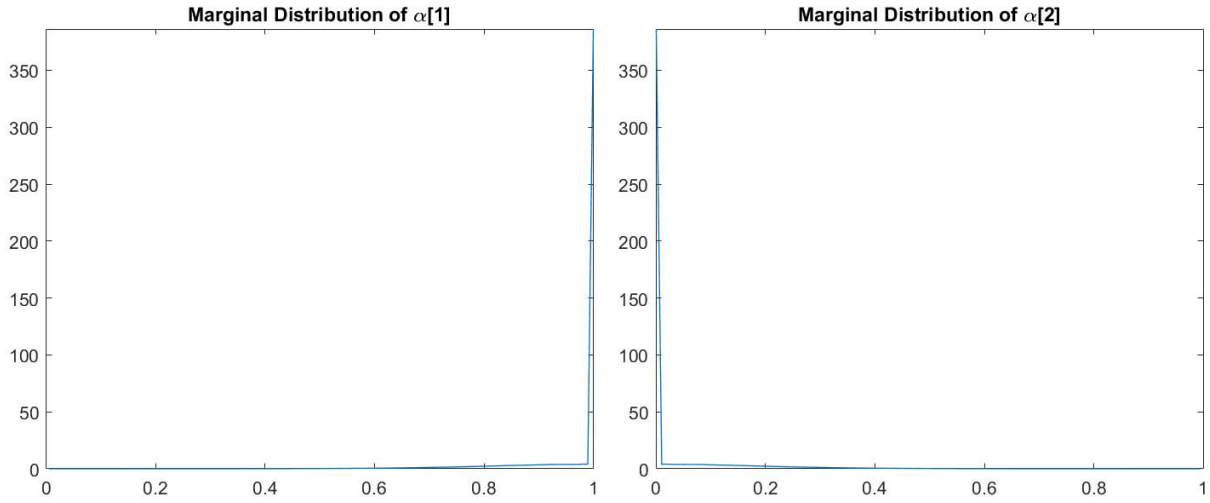

Figure 3.7: Plot of the marginals of the posterior distribution of the relative ratio parameters obtained from applying Model II to the qPCR data.

relative ratio parameters for the subpopulations and the information on interactions between subpopulations to identify the target subpopulation.

3.4 Identifying the Target subpopulation

Estimating the proportion of the various subpopulations is one step in the process of finding the subpopulation which needs to be targeted. Simply targeting the largest subpopulation [26] regardless of other considerations may not be the best solution [36] for the patient. The next step is to actually identify the target subpopulation given the relative proportion of the subpopulations within the tumor. In order to correctly identify a target subpopulation, we need to clearly define what we want to achieve by targeting the subpopulation. Hence, our primary goal is to treat or manage the metastatic cells. The reason is that the metastatic cells are the primary cause of mortality [57] in

Table 3.8: Payoff Matrix when two players are pitted against each other in a game (any situation can be posed as a game).

Players → ↓		Player II	
	Strategies → ↓	c	d
Player I	a	$(p_{a,c}, q_{c,a})$	$(p_{a,d}, q_{d,a})$
	b	$(p_{b,c}, q_{c,b})$	$(p_{b,d}, q_{d,b})$

cancer patients. If there are no metastatic cells in the population then our aim would be to maximize reduction in the growth rate of the tumor. In order to find out the target subpopulation we will make use of the existing knowledge regarding the interactions between various cell types using the framework of evolutionary game theory (EGT). There is a body of literature [36, 58, 59, 60] on how to exploit the knowledge of interactions of different subpopulations while designing therapies for cancer. Here, we describe briefly the basics of evolutionary game theory and then look into a few interesting scenarios on how to make use of evolutionary game theory.

3.4.1 Evolutionary Game Theory

A very good introduction to evolutionary game theory (EGT) is provided in [61, 62]. In game theory, the idea is to look at the payoff when a particular strategy is paired off with another strategy. By looking at these payoffs we can predict what strategies might fare better in a particular situation. So normally the payoffs between all pairs of strategies amongst two players can we written in a matrix form. Hence, in each entry of the matrix we have the payoff for a strategy when paired off with another strategy and vice versa. Therefore, each entry has two values as shown in the example in table 3.8. The p's are payoffs player I receives when paired off with player II strategies and similarly, q's are the payoffs for strategies of player II against strategies of player I. For example, player I will receive a payoff $p_{a,c}$ when player I is using a strategy 'a' and player II is using a strategy 'c'.

However, in the case of evolutionary game theory (EGT) we are looking at the interactions within a population. Hence in a population, players refer to the population as a whole, strategies

Table 3.9: Payoff table for the population types '1', '2' and '3'. Each row gives the fitness of a particular individual type.

Population type $\rightarrow$ $\downarrow$	1	2	3
1	$f_{1,1}$	$f_{1,2}$	$f_{1,3}$
2	$f_{2,1}$	$f_{2,2}$	$f_{2,3}$
3	$f_{3,1}$	$f_{3,2}$	$f_{3,3}$

correspond to different types of individuals (or behavioral phenotypes) [34] and the payoffs corresponds to fitness (or reproductive success) [59] of the various types within the population. Thus, we have the same strategies or population types along both the rows and columns as we are looking for interactions between the various population types. Thus, we can write the fitness matrix a bit differently as shown in table 3.9 where $f_{x,y}$ refers to the reproductive success or fitness of type 'x' when interacting with individual type 'y'.

Thus, given the payoff matrix F (for example like the one shown in table 3.9 and the proportion of the individual types within a population (vector P) we can calculate the fitness of each population type (T) by simply using equation 3.25 below. For example, consider the three population types '1', '2' and '3' as shown in table 3.9. Suppose the relative ratio of these population types is $P = [p_1,\, p_2,\, p_3]^T$ where p_1, p_2 and p_3 are the proportion of type '1', type '2' and type '3' respectively. Then fitness of type '1' is $t_1 = f_{1,1}p_1 + f_{1,2}p_2 + f_{1,3}p_3$ and similarly we can calculate t_2 and t_3 easily by using the vector form of the equation as shown in equation 3.25. The average fitness 'A' of the whole population can be easily derived by using equation 3.26.

$$T = FP \tag{3.25}$$

$$A = P^T T = P^T F P \tag{3.26}$$

This gives the replicator equation (RE) [61, 62] shown in equation 3.27 for population type

Table 3.10: Payoff table for the population types '1', '2' and '3'. The fitness of each subpopulation type is the same as the others regardless of the interacting cell types.

Population type $\rightarrow$ $\downarrow$	1	2	3
1	1	1	1
2	1	1	1
3	1	1	1

'i' which describes the evolution of the types within the population. This basically shows that the difference between the fitness of type 'i' and the average fitness of the population controls the growth rate of population type 'i'. Note that although p_i is not a continuous variable but for a large enough population [60, 61] this is a good approximation.

$$\frac{dp_i}{dt} = p_i(T_i - A) \tag{3.27}$$

Note that as $\Sigma_i p_i = 1$ we are not looking at the total size of the population but rather just the relative ratio of the various types within the population.

3.4.2 A Few Interesting Examples

Here, we will simply look at a few interesting examples involving different cell types in cancer and show how they would be handled within this framework. This subsubsection simply serves to show instances of how useful EGT can be when used in conjunction with our model to estimate the relative ratio parameters. A general solution to the problem is beyond the scope of the current work. We show four examples here.

3.4.2.1 Example 1

Consider the case when we have three subpopulations and none of them is metastatic. Their payoff matrix is given in table 3.10. Suppose their relative ratio is given as $[0.33,\ 0.4,\ 0.27]$ and we have drugs to target each cell type.

Using equation 3.25 the fitness for the individual types comes out to be $[1,\ 1,\ 1]^T$ and using

Table 3.11: Payoff table for the population types '1', '2' and '3'. The fitness of each type is fixed regardless of its interaction.

Population type $\rightarrow$ $\downarrow$	1	2	3
1	3	3	3
2	2	2	2
3	4	4	4

equation 3.26 the average fitness comes out to 1. Now using the RE in equation 3.27 we get equation 3.28 showing the growth rates of each cell types frequency.

$$[\frac{dp_1}{dt},\ \frac{dp_2}{dt},\ \frac{dp_3}{dt}]^T = [0,\ 0,\ 0]^T \tag{3.28}$$

From this we can see that all the subpopulation types have a zero-growth rate of cellular frequency within the tumor. Hence in this case we end up targeting cell type '2' which has the highest relative ratio among all the cell types in the tumor.

3.4.2.2 Example 2

Consider the case when we have three subpopulations and none of them is metastatic. Their payoff matrix is given in table 3.11. Suppose their relative ratio is given as $[0.33,\ 0.4,\ 0.27]$ and we have drugs to target each cell type.

Using equation 3.25 the fitness for the individual types comes out to be $[3,\ 2,\ 4]^T$ and using equation 3.26 the average fitness comes out to 2.87. Now using the RE in equation 3.27 we get equation 3.29 showing the growth rates of each cell types frequency.

$$[\frac{dp_1}{dt},\ \frac{dp_2}{dt},\ \frac{dp_3}{dt}]^T = [0.0429,\ -0.348,\ 0.3051]^T \tag{3.29}$$

From this we can see that the cell type '3' has the maximum growth rate of cellular frequency within the tumor. Hence in this case we end up targeting cell type '3' within the tumor even though it has the lowest relative ratio among all cell types.

Table 3.12: Payoff table for the population types '1', '2' and '3'. The three phenotypes in the game are autonomous growth (AG), invasive (INV) and glycolytic (GLY). The base payoff is 1 and the cost of moving to another location is c. The fitness cost of acidity is n, whereas k is the cost of having a less efficient metabolism (glycolytic). (Reprinted from [36], by permission of the Royal Society)

Population type $\rightarrow$ $\downarrow$	1	2	3
1	1/2	1	1/2-n
2	1-c	1-c/2	1-c
3	1/2+n-k	1-k	1/2-k

3.4.2.3 Example 3

Consider the case where there are three cell types in the tumor. Type '1' is metastatic in nature and the other two types are non-metastatic. Also assume that we have appropriate drugs which are able to target each of the subpopulation types. In this scenario we would not bother about the relative ratio of the cell types and the payoff matrix for the cell types. We would simply target cancer cell type '1' as our primary goal here would be to reduce the spread of cancer within the patients body. Hence in this example we do not bother about any other factor other than the fact that we have available treatment for the only metastatic subpopulation within the tumor.

3.4.2.4 Example 4

In this example, the mix of cancer cell types and its payoff matrix has been reproduced from [36]. Consider a scenario where subpopulation type '1' is an autonomous growth, subpopulation type '2' is invasive in nature and type '3' is glycolytic in nature. Suppose there are no drugs which can directly target the subpopulation type '2' but there are drugs which can directly kill off type '1' and type '3' subpopulation types. Here the relative ratio is given as $[1/3,\ 1/3,\ 1/3]$ and the payoff matrix is shown in table 3.12.

Here, our goal is to reduce the spread of the invasive (i.e. metastatic) cell type. We will try and see which of the other subpopulations should be targeted in order to reduce the overall growth

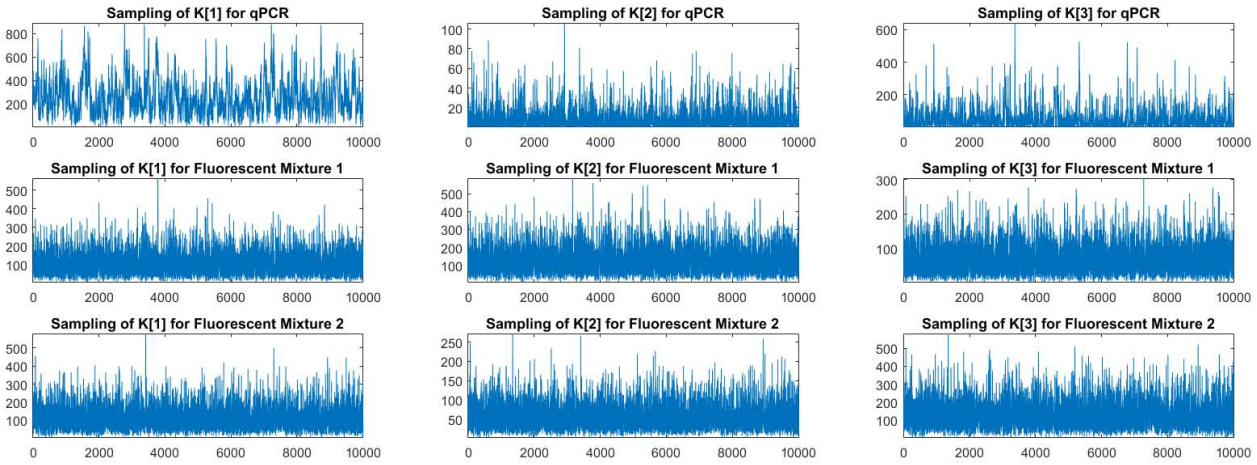

Figure 3.8: The sampling obtained for K in model I.

of the subpopulation type '2'. Suppose we target type '1'. Then in such a situation the type '2' and type '3' subpopulation no longer need to share resources with type '1' subpopulation. Hence, in order to calculate the RE we just need the payoffs due to interactions amongst type '2' and type '3' subpopulations. Now both type '2' and type '3' are half of the active tumor population. Hence fitness of subpopulation type '2' is $1-3c/4$ and fitness of subpopulation type '3' is $3/4-k$ using equation 3.25. So average fitness, using equation 3.26, is $7/8-3c/8-k/2$. Thus, using the RE in equation 3.27, the frequency growth rates of type '2' is $1/16-3c/16+k/4$ when we target type '1' subpopulation. Following the same steps, we calculate that when type '3' is targeted during treatment the frequency growth rates of type '2' is $1/16-3c/16$. Thus, we can clearly see that targeting type '3' subpopulation or the glycolytic cell type, we would be better off in terms of controlling the frequency growth rate of the type '2' or the metastatic subpopulation. This is similar in principle to what is mentioned in [36] and to what is shown in [59]. Hence, in this example we target the type '3' subpopulation.

3.5 Discussion

Before discussing the results, we need to be sure that the sampling done by the Metropolis-Hastings method was done efficiently. Thus, we need to look at the autocorrelation of the concentration parameters K and also a plot of the sampled values to see that the process achieved good

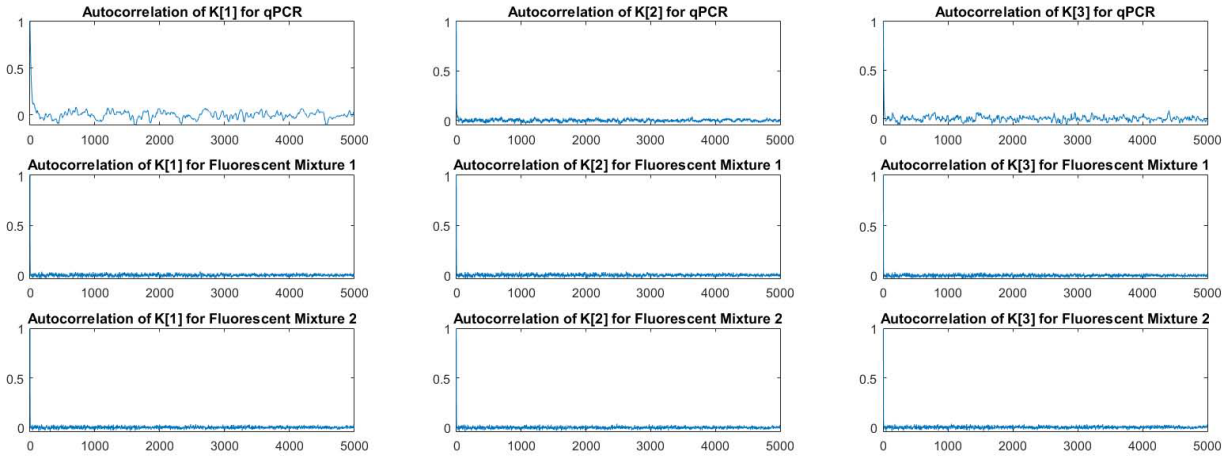

Figure 3.9: The autocorrelation for K in model I.

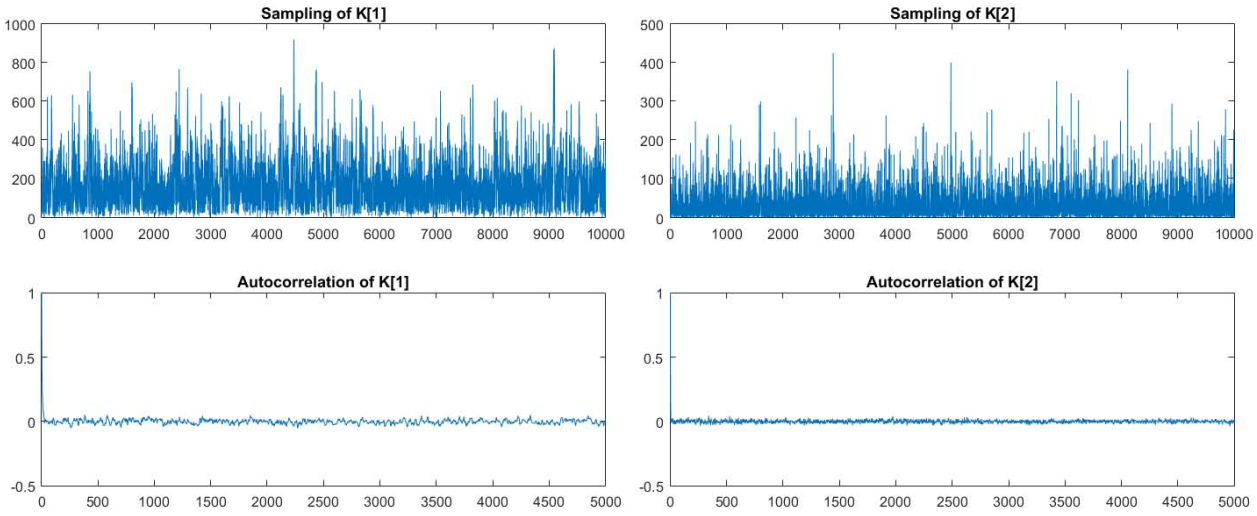

Figure 3.10: The sampled values and autocorrelation for K in model II.

mixing. Figure 3.8 shows the sampled values of K and figure 3.9 shows the autocorrelation for the K values in model I for the various datasets. For model II we can see both the sampling of K and autocorrelation of K in figure 3.10.

We can see from the figures that the samples are well mixed and are very close to being randomly sampled. We get similar results for the other estimated hidden variables during the mcmc process in both the models. This means that the samples are truly representative of the posterior distribution for the hidden variables.

In this section we have introduced two new Bayesian hierarchical models. These methods do better in comparison to the method presented in [26] due to the fact that the inactive state of an observable/attribute in real datasets need not be 0. Also, there can be a non-zero variance associated with the inactive state of an observable. Since in our formulation, the standard deviation is a multiple of the mean, a zero mean implies zero variance, which is a drawback of the method in [26]. The introduction of a non-zero hidden variable θ allows us to model variance and a non-zero mean associated with the inactive state of the observables. This helps to improve the estimation a lot in cases where the mean inactive state is actually not close to zero. In case of the qPCR data we can see that in an inactive state the average is close to 0.37 as obtained from [3] but in the fluorescent dataset the average value for the inactive state is 0.003 which is very close to zero. This is the reason that the model I outperforms the model in [26] (without θ) in case of the fluorescent data by only a small amount but the difference in performance is much more pronounced for the qPCR dataset. Another reason why our model outperforms the model in [26] for qPCR data is that the data in that case is much noisier which is seen from the fact that the estimate of c comes out close to 0.45. This contributes to a larger variance being associated with the inactivity levels of the observables in the qPCR data.

In case of model II we also get an extremely good estimation accuracy. In fact, the estimated values are equal to the actual relative ratio when calculated to four decimal places. This result is even better than what we obtained in Model I with a fixed θ. This could be because of the fact that model II had the freedom to automatically obtain the value of θ which allowed it to compute a better

estimate for θ (the estimate came out to be 0.2676). The reason we have to use two subpopulations for the qPCR data has to do with the number of underlying equations we get from the expression profiles and the number of unknowns we need to estimate, as explained in the intuition for model II. As we were estimating a new variable in model II (i.e. θ) we had to have one less unknown element in α so as to allow for sufficient independent equations for the unknowns.

Finally, once the process of estimation is done we need to make use of EGT so as to intelligently choose a target subpopulation. The existing literature on the use of EGT for cancer allows us to do just that. We can clearly see by looking at example 1 and example 2, how the nature of interaction between subpopulations leads us to choose different targets each time, even though the relative ratio of the subpopulations is the same. In any scenario where we have a payoff matrix similar to example 1 we would always choose the subpopulation with the highest relative ratio as the target, assuming that none of them is metastatic. If we have a single metastatic subpopulation which can be treated directly then we always target the metastatic subpopulation regardless of the relative ratio values or the payoff table as demonstrated by example 3. If we cannot directly target a metastatic subpopulation then using the principles of EGT we can target it indirectly as shown in example 4. There can be a large number of different possible scenarios. A general solution using EGT is beyond the scope of this work, but the idea here is to show how EGT could go along with the method of relative ratio estimation of subpopulations in order to customize cancer treatment.

3.6 Conclusion

In this section we looked at the problem of estimating and finally identifying the target subpopulations within a tumor when the nature of subpopulations is known. We applied our model successfully to different types of datasets. One thing we should note here is that the exact same model worked well for both qPCR and fluorescent dataset because of the fact that the conditional dependencies proposed in the model hold true for both the datasets. For example, the assumption that the observed values were derived by the ratio of two normally distributed variables was valid for both the datasets, due to the way in which they were normalized. Even if the same conditional dependencies do not hold for the different datasets captured using various methods, the relevant

hierarchical model could easily be worked out by using the principles presented here.

4. CANCER HETEROGENEITY: AN ARTIFICIAL NEURAL NETWORK APPROACH

4.1 Introduction

Cancer is an umbrella term used to refer to a whole array of diseases caused by mutations, chromosomal rearrangements, etc. with the property that the affected cell divide and grow uncontrollably. The disruption of the normal functioning within a cell gives it a proliferative advantage over its normal neighbors [35]. To successfully target cancer or manage the progression of cancer, we need to thoroughly understand the underlying mechanisms in cell division, cell growth and cell death. The interactions between various proteins involved in cellular processes can be modelled in various ways such as Boolean modeling [16, 3] and differential equations [6, 7]. Making use of such models it becomes possible to reasonably predict the behavior of the protein signaling under different conditions (such as under the presence of mutation, under the influence of protein kinase inhibitors, etc.). However, in the case of cancer we might not always have knowledge of the exact nature of the mutations and how the protein signaling pathways are affected. Hence, the problem takes a different form where we might have the observed data for the activity of some key proteins and on the basis of that we try and identify the type of treatment we are supposed to give a patient. In this section our focus is to customize treatment for each patient on basis of the observational data we have at hand.

There is already some existing literature tackling the problem of treatment customization for cancer. In [16] the authors propose a method to identify the best drug combinations under the condition that a single protein has been mutated in the signaling pathway of the cells within the tumor. In another work [3] this was extended to include the scenario where there might be multiple mutations within the signaling pathway of the mutated cell. In both of these works, the tumor was assumed to be homogeneous in nature i.e. the nature of the mutations in all the tumor cells was the same. Another work [26] described a method applicable when the tumor is heterogeneous in nature i.e. the nature of the mutations within the different tumor subpopulations is different. This work

devised a method to identify the proportion of various subpopulations assuming that the nature of the various subpopulations is known. This work was refined further in [4] (also shown in section 3) by improving the accuracy of the method and making use of evolutionary game theory to identify the target subpopulation in order to better customize treatment. However, these methods cannot be extended to situations where the nature of the mutations in the subpopulations is not known a priori and the tumor is heterogeneous in nature. For example, it will be extremely tedious to extend the rules presented in [3] to heterogeneous tumors even if the required conditions are satisfied. Also, the works in [16, 3] only deal with tumors where the mutation referred to as stuck-at-faults cause a gene/protein to be constitutively active or to be rendered inactive. It does not apply to scenarios where a mutation referred to as bridging faults in [16, 3] could introduce new signaling mechanisms in the signaling pathways. We need to be able to address these mutations too. The range of the possible mutations and the spectrum of the possible combinations of these mutated subpopulations in a heterogeneous tumor make the problem of analyzing and estimating the proportion of various subpopulations of tumor as described in [26, 4] extremely difficult, especially when the nature of the subpopulations within the tumor is unknown.

Motivated by this, our primary goal is to develop a model to be able to identify the best possible drug combination for a patient while minimizing the possible side effects. This model should be able to address the various types of mutations which behave either as stuck-at-faults or as bridging faults along the signaling pathway. Our model should also be able to deal with a heterogeneous tumor regardless of the number of subpopulations or the interactions occurring between these various subpopulations. To do so, we resort to creating two feedforward neural network models trained using backpropagation to learn the hidden structure of the observed data. This model will help us in identifying the best possible drug combination given the observation data collected from the tumors.

In the next subsection we describe a method to generate synthetic data so as to capture the various uncertainties associated with the actual observed data. We also briefly discuss the real-world data collected from actual cancer cell lines. Then we move on to describing the neural network

models and present the results that were achieved using these models both on the synthetic data and on the observed data. Then we present a discussion of our work and a concluding subsection.

4.2 Cancer Datasets

In this subsection we describe the process underlying the generation of the synthetic data which has been used for the purpose of studying the usefulness of a neural-network based model in the context of treatment customization for cancer patients. We also give a brief overview regarding the real-world dataset collected from cancer tissues.

4.2.1 Synthetic Data Generation

Our objective is to make the synthetic data as realistic as possible. In the synthetic data generated for this work, the input consists of the activity level of genes transcribed by transcription factors and reporter genes under the influence of different drug combinations and the output consists of binary vectors indicating the presence or absence of drugs in the drug combination used for treatment. Each data point corresponds to a single patient. Thus, the input is basically the expression level of a fixed set of proteins under different conditions and the output is the best possible drug combination for the particular scenario. The goal of customized therapy is twofold. First, we want to reduce the activity of the aggressive subpopulations within the tumor and second, we want to reduce the side effects of the drug on the patient by using the minimal number of drugs. The data points generated take into account the heterogeneous nature of cancer. Keeping this in mind we describe the process of data generation in more detail.

As the tumors are mostly heterogeneous in nature one of the key characteristics we decide for each of the data samples is the number of subpopulations that are present in the tumor. The total number of subpopulations is obtained by adding one to a random number generated from a Poisson distribution centered at 3. The reason for adding 1 to the randomly generated Poisson variable is to eliminate the scenario where we have 0 cancerous subpopulations within the tumor. Thus, the average number of tumor subpopulations we have over the complete dataset is 4.

Another variable which has to be decided upon, is the possible mutation locations in the tumor

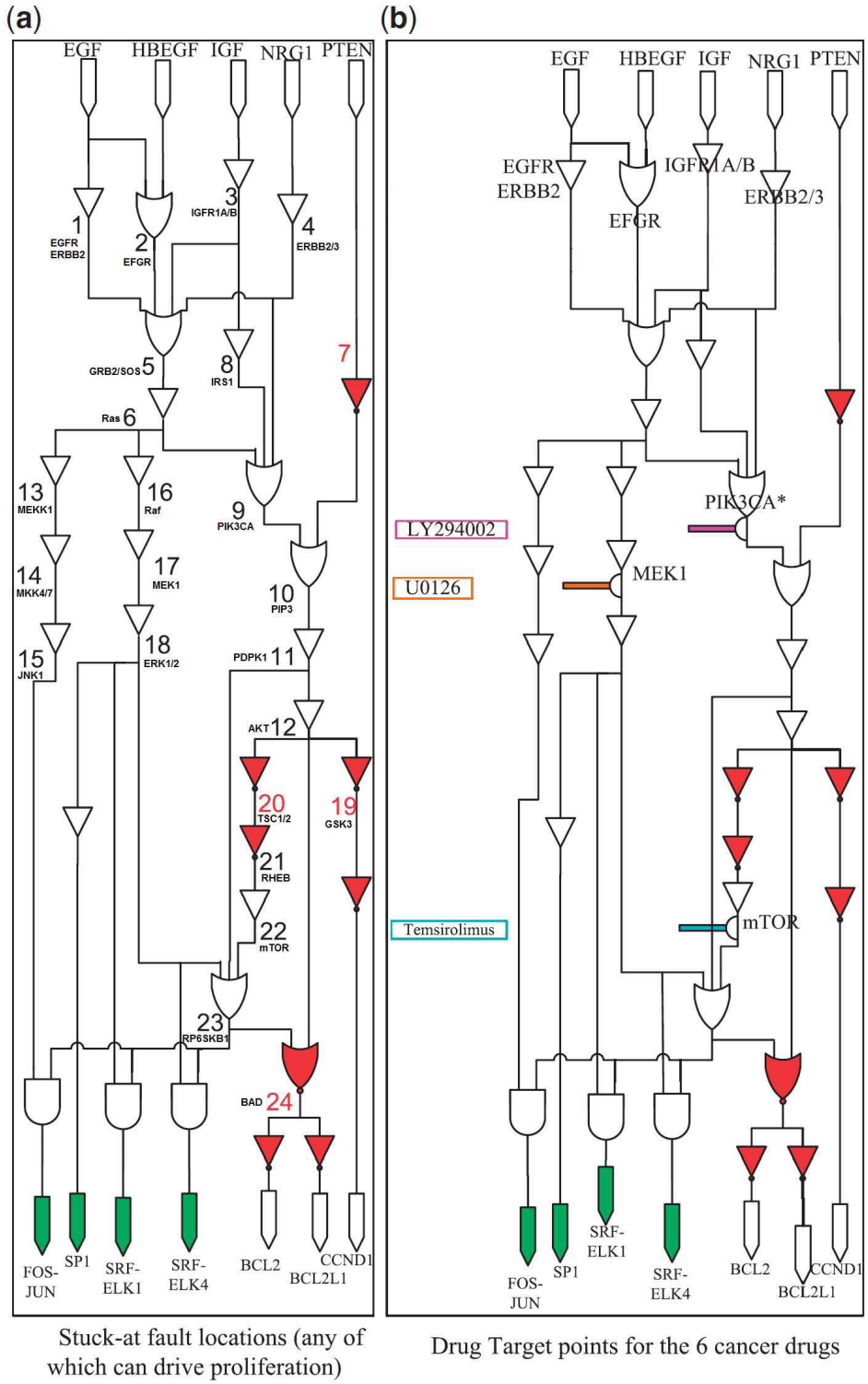

Figure 4.1: (a) The Boolean model of the growth factor signaling pathway (proposed in [16]) along with the proteins which are involved in the signaling (b) shows the protein kinase inhibitors and their target proteins along the signaling pathway. (Derived from [3])

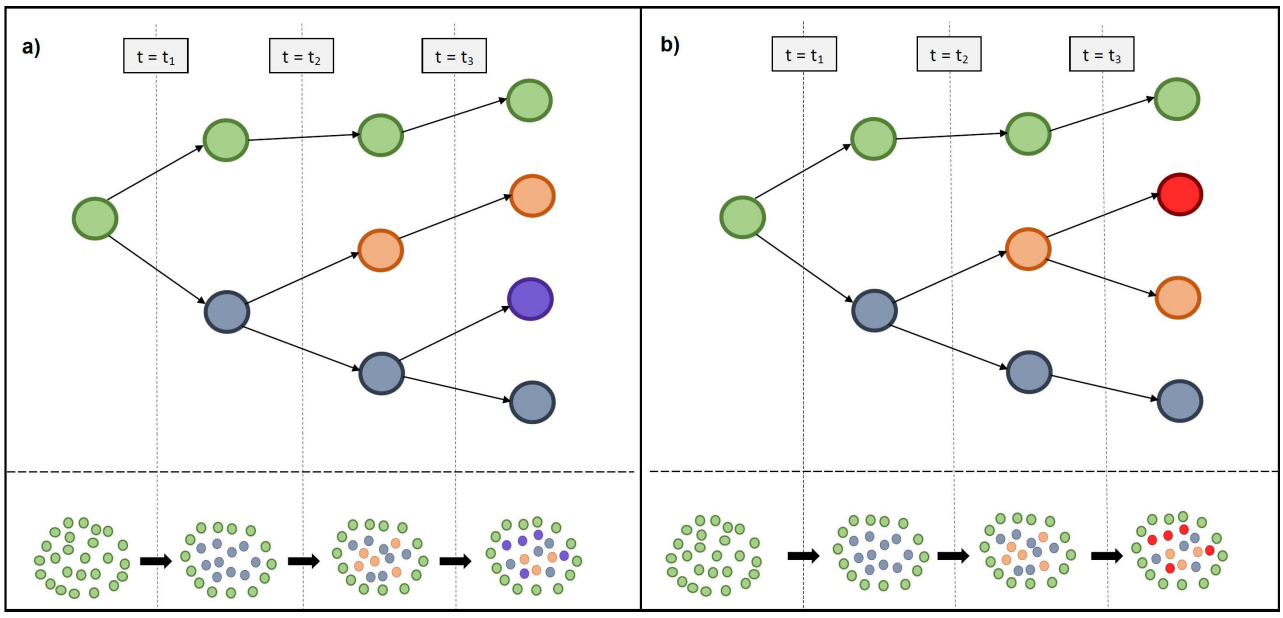

Figure 4.2: The evolution of cancer is shown here with branching occurring at 3 time steps and thus a total of three mutated subpopulations after the final time step.

cells. On the basis of the existing knowledge, we have selected some of the key genes associated with cancer such as IGFR [63], Ras [64, 65], IRS1 [66, 67], AKT [68], MEK1 [69] and mTOR [70, 38, 71] as the possible mutation locations along the growth factor (GF) signaling pathway (shown in Fig. 4.1). Thus, some of these genes would be mutated in the various subpopulations within a tumor. Another thing to note here is that, as we have at least one mutation in each subpopulation, the total number of different subpopulations possible is $2^6 - 1 = 63$. Thus, the total number of subpopulations cannot exceed 63. But, there is always a chance, however miniscule, that we obtain a number greater than 63 in the previous step while deciding the total number of subpopulations. In such a situation we simply discard the value and randomly pick an integer from the set $\{1, 2, 3, , 63\}$ with an equal probability.

Once we have settled on the number of subpopulations and the location of the mutations, the next step is to decide on the subpopulations to be introduced in the tumor. In order to decide on the subpopulations to be introduced, we make use of the branching model [72, 73] of cancer evolution. Thus, as shown in the examples presented in figure 4.2, initially we have normal cells

only represented by green colored cells. The first mutation is introduced at some time $t = t_1$ during the process of cell division. This new mutated subpopulation is represented by the blue colored cell, which forms the entirety of the tumor as shown in figure 4.2. Next, at time $t = t_2$ another mutation is introduced to the already mutated cell to give rise to a new subpopulation (represented by orange cells in both fig. 4.2(a) and 4.2(b)). Thus, in both these examples the blue cell has a single mutation but the orange cell has two mutations. Finally, at time $t = t_3$ one of the mutated subpopulations within the tumor further gives rise to a third subpopulation. In fig. 4.2(a) the blue subpopulation gives rise to the new subpopulation (shown in purple) with two mutations whereas in fig. 4.2(b) the orange subpopulation gives rise to the third subpopulation (shown in red) with three mutations. In fig. 4.2 there is a total of 3 time steps involved but if during the process of generating a data sample we decide on n tumor subpopulations, then we would have n such time steps.

We do not worry about the exact time when a new subpopulation is introduced because the total number of subpopulations is fixed beforehand. However, we do need to decide which subpopulation gives rise to the new subpopulation. We base this on the mutation rates of the subpopulations. Thus, the higher the mutation rate, the greater is the chance for the subpopulation to mutate. There is some evidence that a mutator phenotype is introduced at an early stage of the cancer [74, 75, 76] which helps increase the mutation rates and thus leads to tumor heterogeneity. Keeping this in mind we generate three kinds of datasets. In each of the datasets a different function controls the relative mutation rate (relative to normal subpopulation) given the total number of mutations N_m within a subpopulation. We describe the three functions in the equations 4.1-4.3.

$$f_1(N_m) = 1 + 2.5 \times ln(N_m + 1) \tag{4.1}$$

where ln represents the natural logarithm.

$$f_2(N_m) = 1 + \pi \times \sqrt{ln(N_m + 1)} \tag{4.2}$$

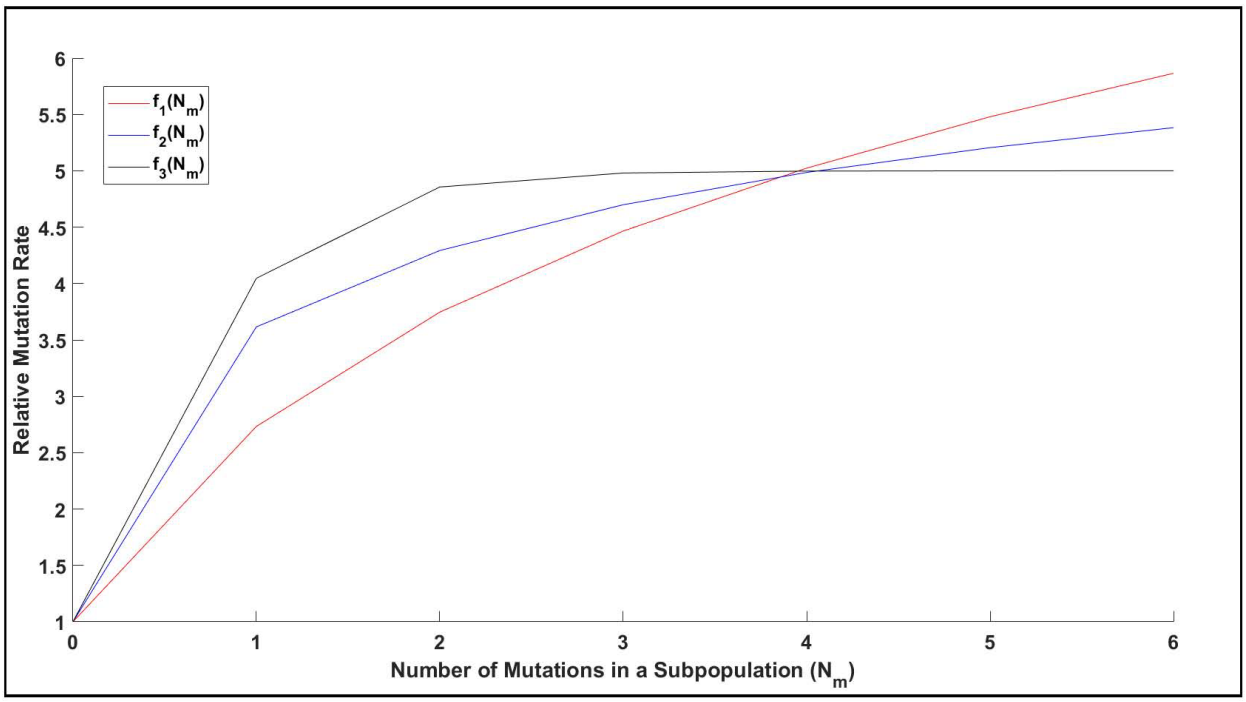

Figure 4.3: The plot of the relative mutation rate vs the total number of mutations within a subpopulation.

where ln represents the natural logarithm.

$$f_3(N_m) = 8 \times \mathcal{S}(2 \times N_m) - 3 \tag{4.3}$$

where $\mathcal{S}$ represents the sigmoid function.

As our data generation allows for a maximum of 6 mutations, (due to the 6 possible mutation locations in the signaling pathway) we show how each of these functions behaves in the plot in fig. 4.3, for N_m ranging from 0 to 6. We can see that the function f_1 provides for a more gradual rate of increase in the mutation rate. However, the functions f_2 and f_3 take into account the mutator phenotype which boosts the mutation rate as the first mutation is added and the first subpopulation is created. The difference between f_2 and f_3 is that, the function f_2 has a gradual rate of increase in the mutation rate after the initial boost where as f_3 quickly reaches a saturation point in the mutation rate after the initial boost. Each of these functions is used to assign a weight (proportional

to the mutation rate) to each of the existing subpopulations on the basis of the number of mutations in them. These weights allow us to randomly select the subpopulation which will branch off into a new subpopulation. Once the subpopulation is selected we allow for a new mutation to occur in the subpopulation, with the specific mutation choice being equally likely from the remaining mutation locations. If the new subpopulation has already been added to the population by the branching off of another subpopulation, we discard it and again use the mutation rate functions f_i to select the subpopulation which will branch off and repeat the process all over again till we get a new subpopulation.

During the implementation, instead of going over the process again and again we separately allocate weights to all the subpopulations which could originate from the existing subpopulations. This also helps to reduce the run time of the data generation algorithm. For example, consider the scenario at the end of all the time steps in fig. 4.2(a). Let us assume that the subpopulation represented by blue cells has a mutation in the (Ras) gene, the subpopulation represented by the violet cells has mutations at (Ras, mTOR) gene locations where as the subpopulation represented by orange cells has mutations at (Ras, AKT) gene locations. Now suppose that we had 4 time steps. Then, both the violet and orange subpopulations have equal chances of mutating and both these subpopulations could lead to the subpopulation with mutations at (Ras, mTOR, AKT) gene locations. Thus, in our implementation the subpopulation (Ras, mTOR, AKT) has twice the weight of being the new subpopulation as compared to the subpopulation (Ras, mTOR, IRS1) or the subpopulation (Ras, AKT, IRS1) both of which can originate from only one existing subpopulation. Note that the existing subpopulation (Ras) could again branch off to give rise to (Ras, mTOR) which already exists at the 4^{th} time step. In order to avoid this type of situation we force the weights of the existing subpopulations to be 0. Thus, this method allows us to directly select the next subpopulation from the existing subpopulations in a single try.

Once we have selected the subpopulations we decide on the proportions of the various subpopulations by deciding upon the relative ratio parameters controlling the impact of the various subpopulations on the tumor as a whole. The relative ratio parameters are obtained from a Dirich-

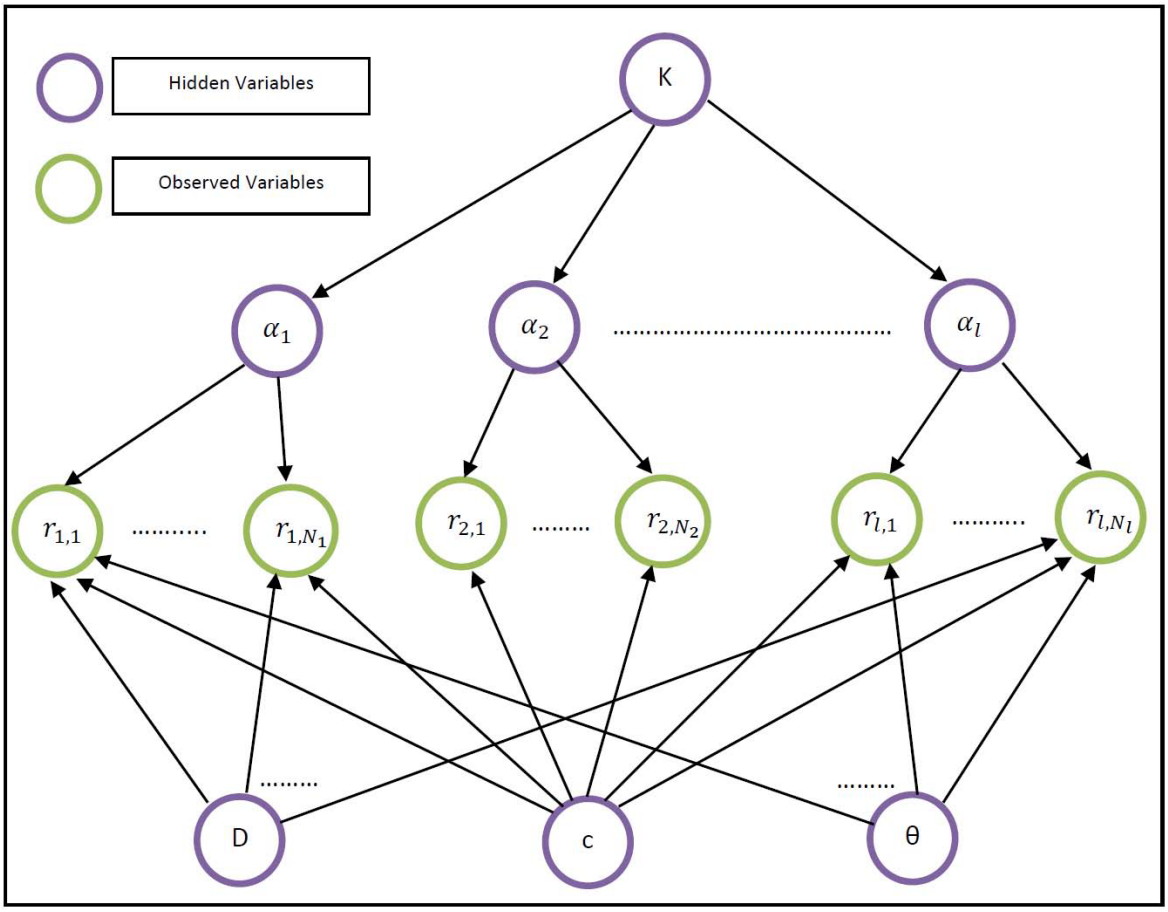

Figure 4.4: A hierarchical model portraying the generation of observations from an underlying tumor tissue (derived from the model presented in [4]).

let distribution with concentration parameter of dimension equal to that of the number of tumor subpopulations. The concentration parameter K is a set of randomly selected real values in the range $(10, 150)$. The reason we allow for random relative ratio parameters is because the growth rates of each of the subpopulations is normally different and also the time steps at which branching occurs has a wide range of variation. The reason we limit the range of concentration parameters to $(10, 150)$ is so that the Dirichlet distribution is neither too sharp which will reduce variability amongst the relative ratios of different genes, nor do we want it to be too much spread out and thus remove any similarity between the relative ratio values of different genes. The relative ratio of the various subpopulations is what controls the final observed output which we generate for our work. A hierarchical model [40, 41, 42] describing the generation of observed values was first described in [26] and refined further in [4] to obtain much better accuracy.

Figure 4.4 shows the hierarchical model derived from the work presented in [26, 4]. We have utilized this model to generate our data. In figure 4.4, we can see the concentration parameter K [26, 4] described earlier. α_i, the relative ratio parameters, are obtained from the Dirichlet distribution with concentration parameter K. Thus, α_i represents the relative impact [26, 4] of each subpopulation on the overall expression of the gene i in the tumor. In our data generation model, we obtain the observation for the output nodes shown in figure 4.1. That refers to one gene transcribed by each of the transcription factors FOS-JUN, SP1, SRF-ELK respectively and the reporter genes BCL2, CCND1. Hence, we have five output genes for which we are generating observation data and thus we have 5 α_i's. D is a set consisting of the vector $d_{i,j}$ [26, 4] which represents the gene expression profile for the j^{th} observation for the i^{th} gene across the various subpopulations. For example, in a tumor consisting of 3 subpopulations if a gene i is supposed to be active for subpopulation 1, inactive for subpopulation 2 and active for subpopulation 3 for the j^{th} observation then $d_{i,j} = [1\,0\,1]$. This gene expression profile $d_{i,j}$ can be computed by using the growth factor signaling diagram shown in figure 4.1. In our case we assume that the input proteins (EGF, HBEGF, IGF, NRG1, PTEN) have the values given in the tuple $(0\,0\,0\,0\,1)$ i.e. all the inputs are inactive except for PTEN which is active.

Even in an inactive state the expression of the gene need not be exactly 0. It can be defined to have an average value of θ [4] which is another hidden variable of the hierarchical model. Thus, we get a modified expression profile $b_{i,j} = [1\,\theta\,1]$. The average expression value [26, 4] for gene i in the j^{th} observation will be the dot product of α_i and $b_{i,j}$ i.e. $\alpha_i b_{i,j}^T$. However, there is always a variance associated with such observations [39, 77] which can be modeled by the final hidden parameter c in the hierarchical model. c is the coefficient of variance associated with the observations and thus the actual j^{th} observed value for the i^{th} gene would be a sample $o_{i,j}$ drawn from the Normal distribution with a mean $\alpha_i b_{i,j}^T$ and a standard deviation $c \times \alpha_i b_{i,j}^T$. We used $c = 0.17$ [39] and $\theta = 0.2676$ [4] on the basis of our prior work. To obtain the final observed values $r_{i,j}$ we normalize $o_{i,j}$ against the expression value $t_{i,j}$ of the gene in the control experiment. In the control experiment, all the genes across all the subpopulations are forced to be active. Thus, $t_{i,j}$ can be obtained by sampling from the Normal distribution with a mean 1 and standard deviation c. Hence, $r_{i,j}$ is the ratio of two Normally distributed random variables [26, 4, 39].

Each gene is observed 4 times under the effect of the 5 different drug combinations which are $\{$No drugs, U0126, U0126 + LY294002, Temsirolimus, Temsirolimus + U0126 + LY294002$\}$. The 3 drugs and their specific target locations are shown in figure 4.1. When a drug is applied to the tissue sample it renders the target protein/gene location inactive. Thus, using this knowledge along with the information of the mutation locations and the input values we can easily compute the expression profile $d_{i,j}$ for all the genes. As we are observing 5 genes under the effect of 5 drug combinations 4 times each, we have a total of $5 \times 5 \times 4 = 100$ observations for a tumor tissue sample. Thus, we generate 20000 vectors of dimension 100 each to form the training dataset and separately generate 2000 vectors to form the testing dataset. We carried out this procedure thrice and each time used a separate function defining the relative mutation rate.

Note that each observation vector in each of the datasets represents a sample obtained from a single patient. In the dataset we also defined a target set for each of the observations. The drug combination is decided upon by first identifying the most aggressive subpopulation in the mix. The subpopulation with the maximum number of mutations is marked as the most aggressive. How-

ever, if there are ties for the maximum number of mutations within the population then we choose the subpopulation with the largest concentration parameter from among the ones with the most number of mutations. The reason for this approach is to reduce the growth rate of the tumor while minimizing the possible side effects from using too many drugs. Thus, our goal is to target the most aggressive subpopulation instead of targeting all the subpopulations at the same time. The drug combination which gives the best possible output but makes use of the least number of drugs is marked using a binary vector. As we have 3 drugs we use a binary vector of size 3 to indicate the presence or absence of the different drugs. For example, suppose a tissue sample has two subpopulations where one subpopulation has a mutation in the Ras gene and the other subpopulation has mutations in the Ras and the IRS1 gene. In this case the second subpopulation with two mutations is considered to be the most aggressive. In a normal cell with no mutation and the input as given earlier we expect all the outputs to be inactive. Hence, our goal is to drive all the outputs to inactivity. We can see from figure 4.1 that applying the drug combination U0126 + LY294002 + Temsirolimus we can force all the outputs to 0 whereas any drug combination consisting of a single drug cannot achieve the same outcome. However, the two drug combination U0126 + LY294002 can also achieve the same result. We prefer the two drug combination U0126 + LY294002 rather than the three drug one in order to have reduced side effects. This drug combination corresponds to the binary vector $(1\,1\,0)$.

By following the procedure described above, we generated a total of 3 datasets. We name the datasets SD1, SD2 and SD3 where 'SD' represents synthetic data. The datasets SD1, SD2 and SD3 were generated by using the relative mutation rate functions f_1, f_2 and f_3 (defined in eqs. 4.1-4.3) respectively.

4.2.2 Real World Data Collection

The real-world dataset uses samples from three different human cancer cell lines. The cell lines were collected from patients with colon carcinoma (HCT116), metastatic melanoma (A2058) and colorectal adenocarcinoma (SW480). The cell lines are marked with specific fluorophores and thus they emit specific wavelengths of light under excitation which can be used in identifying

the cells [56]. In this case the HCT116 cell lines were marked with the red wavelength emitting fluorophores and the A2058 cells were marked with the green wavelength emitting fluorophores. The general procedures for preparing the samples are described in [56].

The observed values were collected using a simple experimental set up. During the start of the experiment all the samples consisted of a mixture of the three cell lines in one of two different relative ratios. Each of the samples were observed either under the influence of exactly one drug such as Lapatinib and Temsirolimus or without the application of any drugs over a period of 48 time steps. Hence, the factors differentiating the observed samples were the initial proportion of the cell lines the sample started with, the time step in which the data were collected and the application of a specific drug or the absence of any drug. There were a total of 18 replicates of the samples observed under similar conditions (i.e. same starting proportion, same time step and similar drug application). For each sample the average green/red emission intensity for all the cells was observed and calculated. We combined the green/red emission values from all the 18 replicates to form a single vector of dimension 36. As we have a total of three drug combinations (i.e. only Lapatinib or only Temsirolimus or no drugs), a total of 48 time steps and two different starting proportion of mixtures, we have a total of $48 \times 3 \times 2 = 288$ data vectors. The proportion of the cell lines vary due to the effect of the different drugs over a period of time which make them very useful for testing the efficacy of our feedforward neural network model. We also randomize the ordering of these vectors in order to remove any dependence that might exist between consecutive data vectors. As the number of vectors is only 288 and the dimension of each vector is large we reduce the dimension of each of the vectors in order to avoid problems related to overfitting of the model. We achieve this by combining the green/red emission intensity from sets of three sample replicates. As there is a total of 18 replicates observed under similar conditions, we get 6 sets of 3 replicates each. We then take the average of the green/red emission values collected from each of the sets and use these values to get our modified vector of dimension $36/3 = 12$.

We then create a target binary vector of dimension 3 for each of these 288 vectors. In order to do this, we first compute the relative proportion of the three cell-lines (i.e. HCT116, A2058 and

SW480) averaging over the 18 replicates. This is computed by looking at the average emission level of green/red wavelength obtained from all the individual cell lines (i.e. HCT116, A2058 and SW480) under the application of similar drugs and during the same time steps. The average green/red emission values for the individual cell lines and that of the mixtures are used to compute the relative ratio of the cell lines by solving a simple set of simultaneous equations. For example, assume that the proportion of three cell lines HCT116, A2058 and SW480 is $p_1 : p_2 : p_3$. Also suppose the average green/red emission levels for the individual cell lines HCT116, A2058 and SW480 are (g_1, r_1), (g_2, r_2) and (g_3, r_3) respectively. Also suppose that the average green/red emission level for all the replicates of the mixture is (g, r). Then we solve the three simultaneous equations 4.4-4.6 in order to obtain the relative ratios $p_1 : p_2 : p_3$ of the cell lines in the mixture.

$$p_1 + p_2 + p_3 = 1 \tag{4.4}$$

$$p_1 g_1 + p_2 g_2 + p_3 g_3 = g \tag{4.5}$$

$$p_1 r_1 + p_2 r_2 + p_3 r_3 = r \tag{4.6}$$

We assume that the cell line which has the largest proportion in the samples corresponding to the vector will be targeted for treatment and we assume a set of three fictitious drugs D1, D2 and D3 which target the cell lines HCT116, A2058 and SW480 respectively. Thus, if A2058 forms the largest subpopulation within the samples then we should use D2 for treatment and hence the target vector is $(0\,1\,0)$. Similarly, if SW480 forms the largest subpopulation then the target vector is $(0\,0\,1)$. Hence, this gives us a total of 288 data vectors and their respective target vectors.

Finally, we randomly choose 240 data vectors to form our training set and the remaining 48 data vectors forms our testing set. We name this dataset FD (abbreviation for fluorescent dataset).

4.3 Neural Network Model

We train two feedforward neural network models on the cancer observation data. The goal is to be able to predict the best possible course of treatment from a set of drugs on the basis of the observation data. The advantage of using a neural network model is that it can directly learn the hidden rules and the structure inherent to the data, without any human intervention. Now, we look into the details of both the models, which we have trained in this work to learn the underlying association between the observation data and the best drug combination for treatment. Both the models are minor variations of each other and provide an advantage on different aspects of the drug prediction as will be elaborated upon later.

4.3.1 Model I Description

As mentioned earlier, we have used a feedforward neural network model. This model has been implemented by using the neural network toolbox in Matlab. The training data is broken up into three blocks with the initial 80% of the data being used for training, the next 10% of the data being used as the first validation set and the final 10% of the data being used as the second validation set. The dataset has been randomized beforehand so as to prevent the neural network from training on the data in a specific region of the feature vector space rather than generalizing over the complete vector space.

As a preprocessing step we perform feature scaling of the data so as to transform the features to the range $[-1, 1]$. This generally helps in faster convergence [78] of the model to its minimum. It is trained using the scaled conjugate gradient backpropagation [79]. The details of this backpropagation method are explained in [79]. This method is generally good for large datasets and the synthetically generated dataset described in the previous subsection falls in that category. It also works well for relatively smaller datasets as we shall see when we apply our model to the fluorescent data obtained from cancer cell cultures. Mean squared error mse, shown in equation 4.7 is used to evaluate and backpropagate the errors during the training process, where N_s is the total number of samples in the dataset.

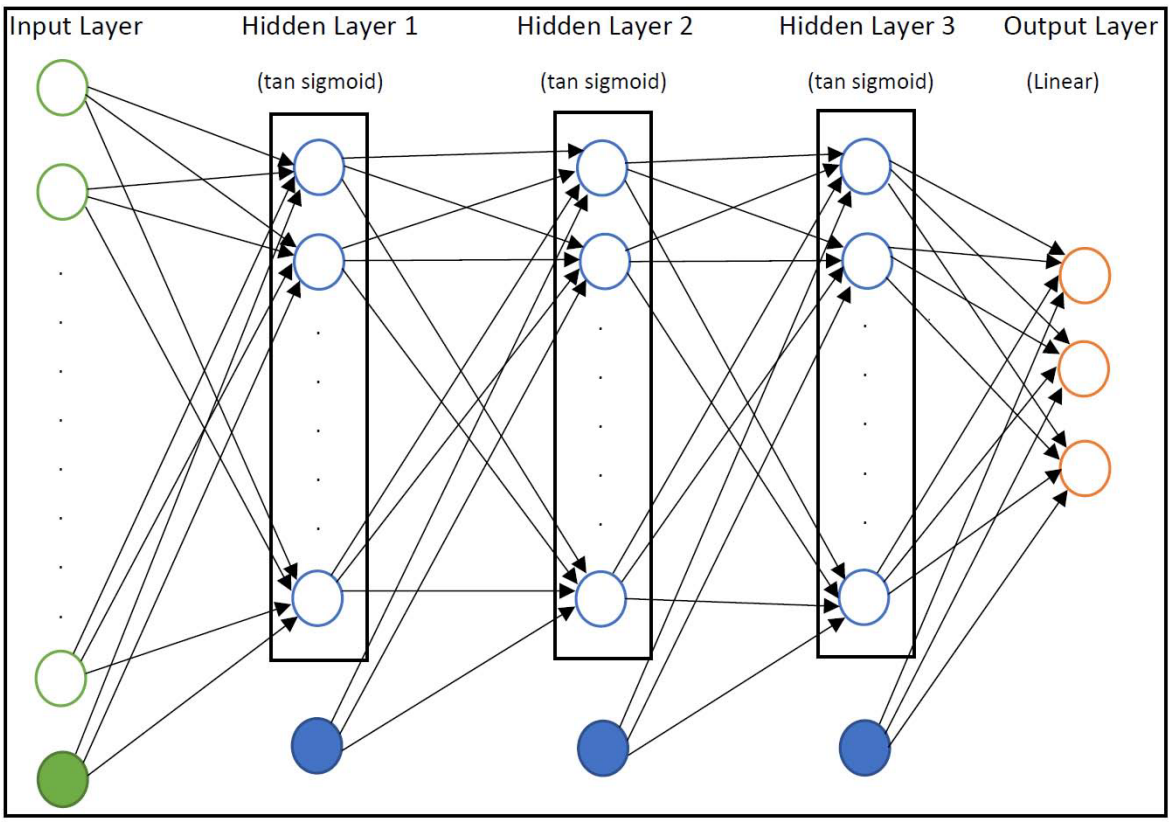

Figure 4.5: Architecture of the feedforward neural network-based model I.

$$mse = \frac{1}{N_s}\Sigma_{i=1}^{N_s}(y_i - t_i)^T(y_i - t_i) \tag{4.7}$$

where y_i is the output vector and t_i is the target value vector.

In our neural network model, we have used three hidden layers of the same size. The neural network model diagram is shown in figure 4.5. Each node in the hidden layer uses a tan sigmoid (tansig) activation function described in equation 4.8.

$$tansig(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}} \tag{4.8}$$

The nodes in the output layer use a linear activation function. The reason for using a linear activation function in the output layer was to improve the performance as using a sigmoid activation function in the output layer caused the performance to drop. This is due to the fact that it is very

difficult to update the weights by backpropagating errors over the sigmoid activation function once the node reaches the saturation region. Thus, the back propagated error is very small and the change in the weight becomes negligible in each epoch of training. However, we do still use tan sigmoid activation in the hidden layer because we do need the nonlinear nature of such activation functions for the model to successfully learn the associations between the input and the target output values. As we already have three hidden layers with non-linear activation functions we use a linear activation function for the final layer. We also use a bias in all the layers (shown in solid colors in figure 4.5) so as to account for any bias associated with the data or the output of the previous layer. This is especially useful in the output layer considering the fact that the output of the tan sigmoid activation function lies in the range $(-1, 1)$ whereas we expect the output to be approximately in the range $(0, 1)$ since our target vector has binary values.

In order to obtain the final parameters, we calculate the mse of the first validation set in each epoch and stop the training of the model when the mse of the first validation set does not improve for 25 consecutive epochs. Then we roll back the model by 25 epochs and set the model parameters to the values the model had when the mse for the first validation set was at the lowest. An example is shown in figure 4.6 where we trace the mse obtained during the training of the model on dataset SD3. This helps us to prevent the model from overfitting on the training data.

Once the model has been trained we need to find the threshold in order to help us decide whether a drug is to be administered or not. In other words, the threshold helps convert the real valued output into binary values 0 or 1. Since this is a classification task and the datasets are unbalanced in the sense that approximately 0.75 of the target values in each of the three simulated datasets are 1's and exactly $2/3^{rd}$ of the fluorescent dataset target values are 0's we need to make sure that the classification is balanced with respect to both the classes i.e. 0 and 1. Hence, we make use of the Matthews correlation coefficient MCC [80, 81], popularly used in bioinformatics [81, 82] for classification tasks, while choosing the threshold. We vary the value of the threshold from 0 to 1 in steps of 0.01 and choose the value which gives us the largest value for the MCC on the second validation set. The formula used to compute the MCC is given in eq. 4.9.

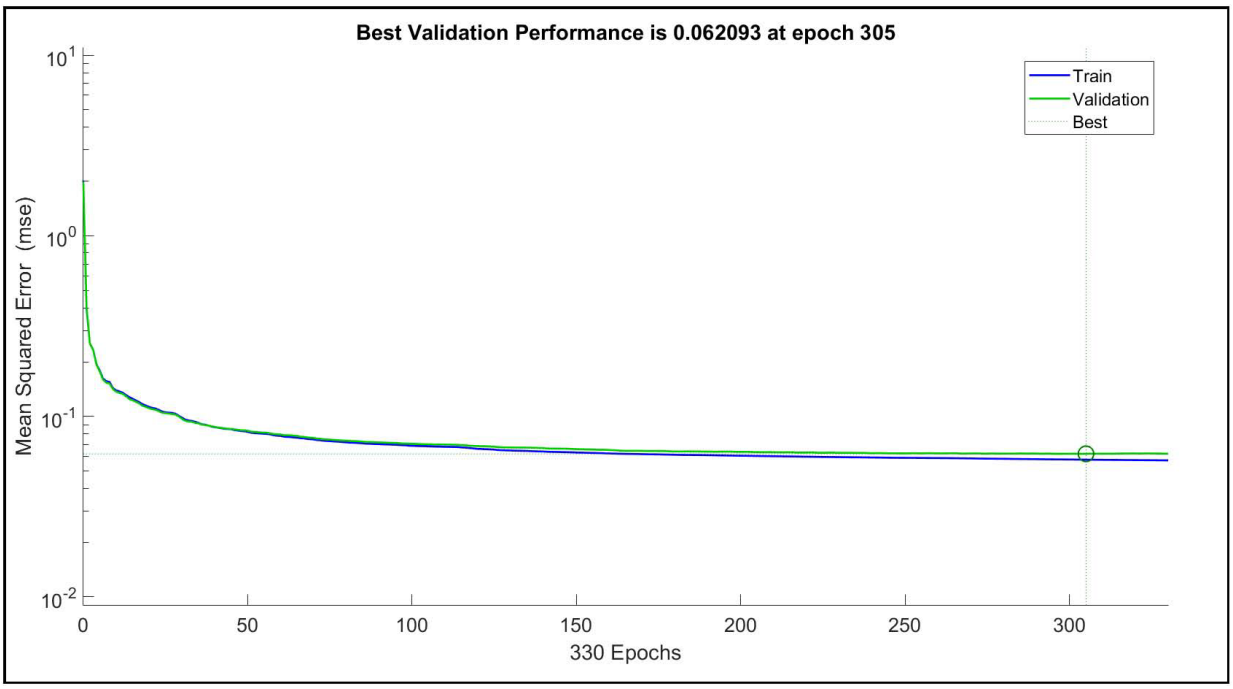

Figure 4.6: Mean squared error (mse) plot obtained during Model I training using the SD3 dataset showing the minimum on the validation set at epoch 305.

$$MCC = \frac{TP \times TN - FP \times FN}{\sqrt{(TP + FP)(TP + FN)(TN + FP)(TN + FN)}} \tag{4.9}$$

Note that TP refers to the number of the true positives, TN refers to the number of true negatives, FP refers to the number of false positives and FN refers to the number of false negatives during classification. Figure 4.7 shows the MCC plots and receiver operating characteristic (ROC) curves for each of the output nodes which were obtained while training model I using dataset SD3. The final selected threshold value for each of the output nodes is marked in red in the corresponding position on the MCC curve and on the ROC curve. We chose separate threshold values for each of the output nodes, while training Model I, on all the datasets in a similar manner (i.e. by using the MCC values for assessment).

We also set a limit, on the total number of epochs for which the model can train, at 1000. The minimum allowed performance gradient is set at 10^{-6} and thus when the value drops below the threshold the training is stopped. These settings allow us to prevent the model from training

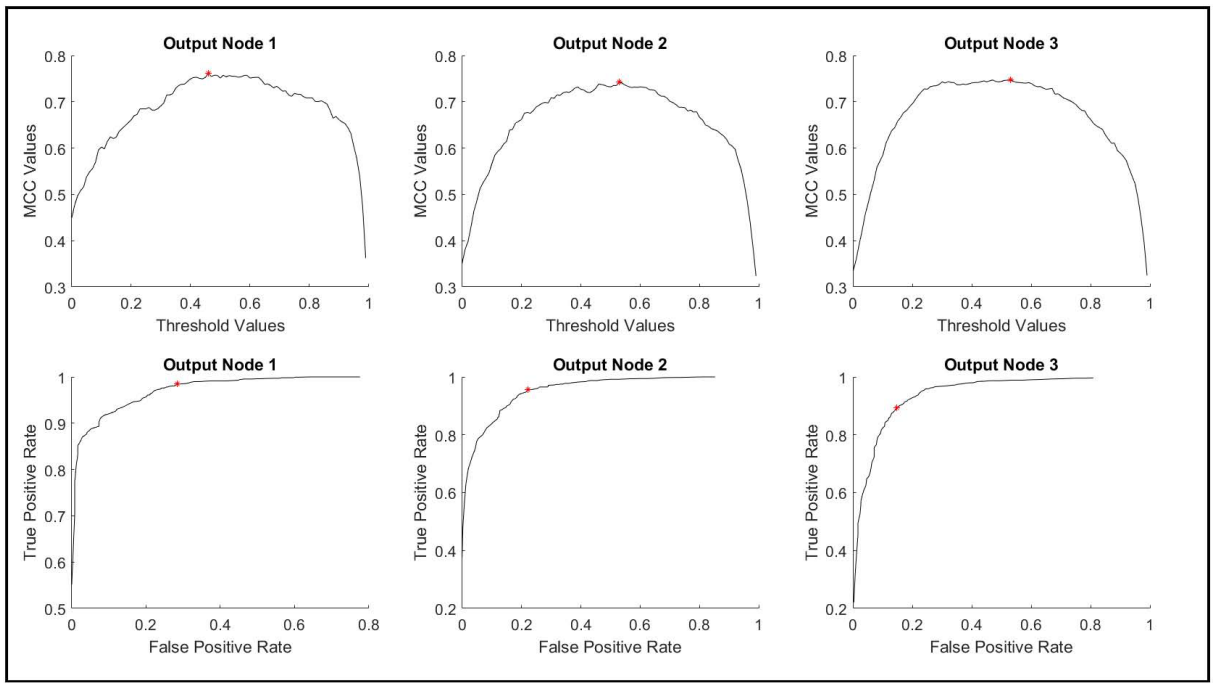

Figure 4.7: MCC curves and ROC plots obtained by varying the threshold values for a trained Model I using SD3 datset.

indefinitely.

4.3.2 Model II Description

This model is a variation on the model described in the previous subsection. The training dataset is split up into two blocks. The first block consisting of 80% of the data samples is used for training the model and the remaining 20% of the data is used for validation purposes. Similar to model I we use the same preprocessing and perform feature scaling. We make use of the scaled conjugate gradient backpropagation method [79] to train our model.

In order to implement the model II we make use of the architecture shown in figure 4.8. Similar to model I there are 3 hidden layers which use the tan sigmoid activation function as described in eq. 4.8 and consist of the same number of nodes. The output layer uses a softmax activation. Softmax activation calculates the output vector as shown in the equations 4.10 and 4.11.

$$z_j = \Sigma_{i=0}^{N_h} w_{ij} o_i^{(3)} \tag{4.10}$$

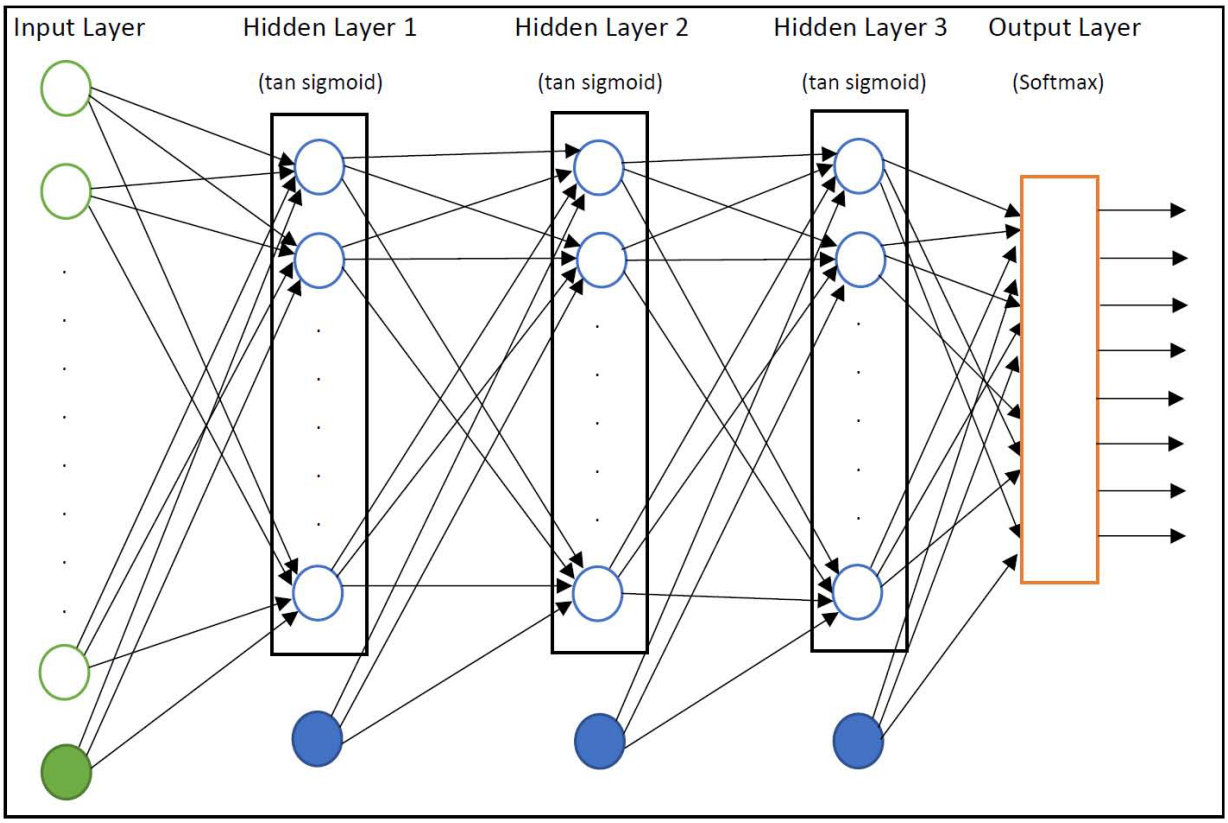

Figure 4.8: Architecture of the feedforward neural network-based model II.

such that $o_i^{(3)}$ is the output of the i^{th} node in the 3^{rd} hidden layer.

$$y_j = \frac{e^{z_j}}{\sum_{k=1}^{K} e^{z_k}} \tag{4.11}$$

such that y_j is the output of the j^{th} node in the output layer.

Note that in eq. 4.10, $i = 0$ refers to the bias node which implies that $o_0^{(3)}$ is equal to 1 and N_h is the total number of nodes in the hidden layer. In eq. 4.11 K is the total number of nodes in the output layer.

As we are using a softmax output we modify the target vectors so that the model would be compatible with the target vector. In this section we are dealing with the issue of identifying the correct drug combination. Therefore, the modified target binary vector will be 1 for exactly one output node, corresponding to the correct drug combination, and 0 for the rest of the output nodes.

Table 4.1: Modification of target binary vectors so as to be compatible with SoftMax activation at the output layer.

Original target vector	**Modified target vector**
0 0 0	1 0 0 0 0 0 0 0
1 0 0	0 1 0 0 0 0 0 0
0 1 0	0 0 1 0 0 0 0 0
1 1 0	0 0 0 1 0 0 0 0
0 0 1	0 0 0 0 1 0 0 0
1 0 1	0 0 0 0 0 1 0 0
0 1 1	0 0 0 0 0 0 1 0
1 1 1	0 0 0 0 0 0 0 1

Note that all the data samples have an associated target binary vector of dimension 3. These 3 binary values in the target vector correspond to the 3 drugs which are being considered in the datasets, as explained earlier. Thus, the total number of different drug combinations possible is $2^3 = 8$. Hence, in order to modify the target vectors, we simply map each of the 8 possible target binary vectors to vectors of length 8 as shown in table 4.1.

In this model we make use of the cross-entropy error function CEE for the purpose of evaluation and backpropagation of error during the process of model training. It is defined as the dot product of the logarithm of the model output vector and the target vector averaged over all the data samples (eq. 4.12).

$$CEE = -\frac{1}{N_s}\Sigma_{i=1}^{N_s} ln(y_i^T) \times t_i \tag{4.12}$$

such that y_i, t_i are the model output and target vectors respectively for the i^{th} sample and N_s is the total number of samples in the dataset.

Similar to model I, we also compute the CEE over the validation set in each epoch of training and stop the training of the model when the CEE of the validation set does not improve for 25 consecutive epochs. Then we roll back the model by 25 epochs and assign the model parameters that the model had when the CEE for the validation set was at the lowest, as the final model parameters. We get a plot similar to figure 4.6 but with the performance being measured by cross-

entropy error. Again, as in model I we set the limit on the total number of epochs for which the model can train at 1000. The minimum allowed performance gradient is set at 10^{-6} and thus when the value drops below the threshold the training is stopped. These settings allow the models to achieve the best possible weight values in an acceptable amount of time.

4.3.3 Implementation and Results

We train both model I and model II on each of the four datasets (i.e. SD1, SD2, SD3 and FD) and get a total of $4 \times 2 = 8$ models. In the simulated datasets, the training data had a total of 20000 samples each with a feature dimension of 100. For these 3 simulated datasets we fix the size of the hidden layers at 25 nodes (excluding the bias node) for both model I and model II. Increasing or decreasing the size of the hidden layer causes the performance to drop. Thus, this was found to be the optimum value to use given the dataset of this size. In the case of the fluorescent dataset FD we had a total of 240 samples of dimension 12 in the training dataset. We definitely could not use a model of the same size as used for the simulated datasets. The reason is that the number of independent model weights we would have to learn from the training data would far exceed the total number of data samples. Thus, when training model I and model II, for the FD dataset, we found that the optimum size of the hidden layers was 3 nodes (excluding the bias). Thus, we had a total of $13 \times 3 + 4 \times 3 + 4 \times 3 + 4 \times N_{out} = 63 + 4N_{out} < 100$ weights to train, where N_{out} is the total number of output nodes which varies depending on whether we are using model I or model II.

Once we trained model I and model II for all the datasets we tested the models using the corresponding test dataset. The test dataset as described earlier had a total of 2000 samples for the simulated dataset and the fluorescent dataset had a total of 48 samples for testing.

Now we look at the results we obtained by applying both the models to the four datasets (three simulated and the fluorescent dataset). We use 3 evaluation metrics during the testing of the model. First, the drug combination accuracy (DCA) for both the models is calculated. In this evaluation criteria we compare the predicted drug combination with the target drug combination. Second, the drug prediction accuracy (DPA) is evaluated for both the models. In this evaluation criteria we

Table 4.2: Confusion matrix for the example test set.

Total drug usage values = 6		**Predicted drug usage values**	
		0	**1**
Actual Drug usage values	**0**	$TN = 2$	$FP = 0$
	1	$FN = 1$	$TP = 3$

Table 4.3: Testing Results for the four datasets using two types of models.

Datasets ↓	**Model I**			**Model II**		
	DCA	**DPA**	**MCC**	**DCA**	**DPA**	**MCC**
SD1	81.50%	91.47%	0.7749	83.40%	91.47%	0.7763
SD2	80.90%	91.80%	0.7833	82.65%	91.68%	0.7787
SD3	81.80%	91.90%	0.7864	82.50%	91.72%	0.7812
FD	85.42%	91.67%	0.8148	87.50%	91.67%	0.8125

compare the predicted usage of the drugs individually to the targets. Third, we compute the MCC value across all the output nodes combined for all the testing samples. This helps us in making sure that the model is not biased in terms of the detecting whether a drug is to be used or not, for treatment. Suppose we consider the example where we have two predicted drug combinations $[1\,1\,0]$, $[0\,1\,0]$ and two corresponding target drug combinations $[1\,1\,0]$, $[0\,1\,1]$. In the first evaluation metric we will get an accuracy of 50% as one predicted drug combination matches with the target combination. In using the second evaluation metric the accuracy would come out to be 83.33% as five out of the six drug usage predictions match with the target values. For the third evaluation metric we need to calculate the TP, TN, FP and FN values. These can be calculated from the confusion matrix shown in table 4.2. Using eq. 4.9 we obtain $MCC = 0.7071$.

We evaluated both the models on all the 4 datasets using these 3 evaluation metrics. The results are presented in table 4.3.

All these models were trained using scaled conjugate gradient backpropagation as mentioned earlier and all these models stopped training when the performance of the model did not improve for the validation for 25 epochs. For all the datasets while training model I we also decided on the

threshold values using the MCC plots similar to those shown in figure 4.7.

4.4 Discussion

We can see that both of these models provide comparable results in terms of prediction accuracy DCA, DPA and also the MCC values. However, if we observe closely we would see that model II outperforms model I in terms of DCA by 1 to 2% for all the 4 datasets. This is understandable in light of the fact that in model II each output node represents a single drug combination whereas in model I each output node corresponds to a single drug. During the training of the neural networks, model II learns the associations between observation vector and the specific drug combinations directly whereas model I learns the relationship between observation vector and the corresponding drug usage values. Thus, model II outperforms model I when we compute DCA.

Due to the nature of the target vectors used to train model I we can see that the model I always outperforms model II when we use the evaluation metric DPA albeit by a small amount. The reason we do not see a large difference in performance when comparing DPA for both the models is that there are two competing factors here. Although learning the specific relationship between observation vector and the drug usage values does provide model I an advantage in the evaluation of DPA but when model II learns the association between observation vector and the drug combination correctly it gets three drug usage values correctly and thus improves its DPA. Thus, model I cannot outperform model II by a significant amount on DPA evaluation metric as in the case with the DCA evaluation metric. Hence, we can see that the DPA values are the same for both the models in the datasets SD1 and FD and the model I has a greater DPA value than model II for datasets SD2 and SD3.

In the MCC evaluation metric, model I outperforms model II on an average by a small amount. The reason for this is that, we specifically select the binary thresholds for model I on the basis of MCC calculation. Thus, it does help model I to improve its MCC during the evaluation of the model. However, it does not provide a major advantage in terms of MCC evaluation due to a competing factor which plays a role. We explain the competing factor by the use of an example. Consider that we have these 4 target vectors in the dataset $[1\,1\,1]$, $[0\,1\,1]$, $[1\,0\,1]$, $[1\,0\,0]$. If we

consider each of the output nodes, 3 out of 4 times the output is 1 and only once it is 0. In such a scenario it is quite easy for the model I to be biased towards the output 1 by the very nature of the dataset. However, if we look at model II each of the target vectors gives rise to 4 different drug combinations which are represented by a different output node. The architecture of model II also forces exactly one output node to have a maximum value, with a very high degree of probability, and this node is considered to have a value 1 and the remaining output nodes have a value 0. Thus, the architecture of model II allows the target data to be better spread out across 8 classes in this example. This reduces the chances for the model to develop bias towards certain types of output.

Another interesting thing to note here is that DPA values are all greater than 90% whereas DCA values lie within the range $80 - 90\%$. This is easily understood as for a drug combination to be correct all three output nodes have to be correct whereas for a single drug prediction to be correct just one output node needs to be correct. For example, consider the target vector $[1\,0\,1]$ and suppose we get the predicted vector $[1\,0\,0]$. This adds 2 correct predictions to the DPA score whereas it does not contribute to the DCA score. Thus, mathematically speaking DCA$\leq$DPA always holds true when we are testing a particular model on a given dataset.

Thus, we see that each of the models do hold some advantage over the other. Given the nature of the models, we can see that model II would be extremely useful in scenarios where multiple drugs work in tandem in order to treat the patient. In such cases model II would better select the correct drug combination. In scenarios where drug combination itself is not important but rather we need more accuracy regarding the usage of drugs, model I would be a better option to use.

4.5 Conclusion

In this section we generated a simulated dataset to closely resemble real-world data obtained from the cancer cell lines on the basis of existing knowledge. Then we trained two neural network based models on the simulated datasets and also a fluorescent dataset. We saw the advantages of using each type of model over the other. This section shows how neural network models are able to realize the hidden associations between the observations and the target outputs even though there is a lot of variations associated with such biological data. Another important step which needs

to be taken is collecting more real-world data so that models of the types considered here can be exploited to their fullest potential.

5. SUMMARY AND CONCLUSIONS

5.1 Summary

In this dissertation, we have considered various problems with the common goal to find a way to customize the drug treatment for cancer patients. The approach used to solve each of these problems exploits the monotonic properties of certain Boolean networks, the conditional dependencies of the hierarchical models, the principles of evolutionary game theory and the learning potential of the neural network models. In order to customize the treatment, we try to get the best possible output in terms of containing the spread and growth of the tumor while minimizing the side effects of the treatment at the same time.

First, in section 2, we looked at the scenario when the tumor is homogeneous in nature and made use of Boolean networks satisfying monotonic properties [3] to predict the best possible drug combination for the patients. Here our prediction is based upon the observations obtained from the tissue sample under the influence of a subset of the possible drug combinations. However, the signaling pathway has to satisfy the monotonic properties and the mutations have to correspond to stuck-at-faults for this method to be applicable. The strategy of using a small subset of possible drug combinations to predict the best possible drug combination is inherent to all the methods presented in this work.

Second, in section 3, we solved the case when the tumor is heterogeneous in nature and we have prior knowledge regarding the behavior of the subpopulations within the tumor. We applied a Bayesian hierarchical models along with evolutionary game theory [4] in order to predict the target subpopulation and consequently provide a method for the customization of cancer treatment in such a scenario. The Bayesian hierarchical model allows us to accurately estimate the relative ratios of the various subpopulations. Then, evolutionary game theory allows us to identify the subpopulation which should be targeted first so as to achieve the best possible results in terms of controlling the growth and spread of the tumor. This particular approach can deal with mutations on

signaling pathways which need not satisfy the monotonicity requirement as dictated by the method presented in section 2. This can also deal with the mutations which correspond to a bridging fault assuming that the nature of the new signaling is known apriori. However, in this case we do make the assumption that the nature of the all the constituent subpopulations of the tumor are known to us either via the knowledge of mutations along the pathway or via some phenotypic behavior of the tumor cells.

Finally, in section 4, we considered the scenario where the tumor is heterogeneous in nature where the number of the subpopulations and/or nature of the subpopulations is unknown. We made use of neural network based modeling [82] to identify the best possible drug combination. This method is much more powerful as, in this case, we need not have any prior knowledge of the subpopulations within the tumor and we need not make any assumptions as to the nature of the signaling pathway or the type of mutations present. However, this method does make use of data from lots of cancer patients whereas the methods presented in sections 2 and 3 only makes use of the observation data from the single patient for whom we are trying to predict the best possible drug combination and signaling pathway knowledge.

5.2 Further Study

There are various aspects of the problem which needs to be considered in the future so as to bring this work closer to the realm of clinical applications. In this work we have just used the total number of drugs as an indicator for the toxicity of a drug combination. We can further improve upon this by making use of a detailed analysis of the toxicity of various drug combinations at different concentrations so as to better customize the treatment for patients. Another way to make our methods more useful is to improve the run time of our presented methods while maintaining the accuracy levels of our model. Some work has been done in this regard in [57].

An important aspect of this research, which will be taken up later, is finding innovative ways for feature selection in order to obtain features which are more discriminative in terms of deciding the output. The use of relevant features will help us in achieving better estimation of the best possible drug combination. This can be directly applied to all the methods presented in sections 2,

3 and 4. We should also combine various categories of features (e.g. body mass index, hormone levels, etc.) along with protein/gene expression, emission values to get more informative features and get a better predictive model.

We made use of EGT along with the hierarchical model to demonstrate how powerful a tool it can be when used alongside Bayesian networks. However, in order to use EGT we need to be have the payoff matrix for various cell lines. One possible direction of research for the future could be to try and estimate this payoff matrix by making use of the time series data or perhaps combining the estimation of the payoff matrix along with the other hidden variables in the hierarchical model to get a more comprehensive solution to the problem. The improvement in accuracy which has been obtained in this work, in identifying the relative impact of the various subpopulations, makes Bayesian hierarchical models more significant in the context of customizing cancer treatment and hence an approach worth pursuing for further research.

Another possible direction of future research would be to make use of the time-series data and the response of the patients to the various drug combination therapies over a period of time. Such a model would not only learn which drug combination works best but also make use of the knowledge regarding the drug combinations to which the patients did not respond well, during treatment, and further improve the prediction of the best possible drug combination. Another thing which such a model would take into account is the time duration involved for a patient to respond to the various types of treatment. This direction of future research would definitely bring us a step closer to achieving the goal of treatment customization for cancer patients.

REFERENCES

[1] Rebecca SY Wong, "Apoptosis in cancer: from pathogenesis to treatment," *Journal of Experimental & Clinical Cancer Research*, vol. 30, 26 September 2011.

[2] Bruce Alberts, Alexander Johnson, Julian Lewis, Martin Raff, Keith Roberts and Peter Walter, *Molecular Biology of the Cell*. Garland Science, 2002.

[3] Bibhu P. Mishra, Aniruddha Datta and Vijayanagaram Venkatraj, "Boolean networks with monotonic properties and their utilization in the efficient selection of kinase-inhibitor combination therapies for the treatment of cancer," Under Review.

[4] Bibhu P Mishra, Aniruddha Datta, Chao Sima, Jianping Hua, Rosana Lopes and Michael Bitner, "Understanding cancer tissue heterogeneity using bayesian hierarchical models along with evolutionary game theory," Under Review.

[5] Bibhu P Mishra, Aniruddha Datta, Chao Sima, Jianping Hua, Rosana Lopes and Michael Bittner, "Cancer heterogeneity: An artificial neural network approach," Under review.

[6] James M. Bower and Hamid Bolouri, *Computational Modeling of Genetic and Biochemical networks*. MIT press, 2001.

[7] Neema Jamshidi and Bernhard Ø. Palsson, "Mass action stoichiometric simulation models: incorporating kinetics and regulation into stoichiometric models," *Biophysical Journal*, vol. 98, January 2010.

[8] Ting-Chao Chou, "The mass-action law based algorithm for cost-effective approach for cancer drug discovery and development," *American Journal for Cancer Research*, 2011.

[9] Paul Sjoberg, Per Lotstedt and Johan Elf, "Fokker-planck approximation of the master equation in molecular biology," *Computing and Visualization in Science*, vol. 12, pp. 37–50, December 2015.

[10] Ritwik K. Layek, Aniruddha Datta and Edward R. Dougherty, "From biological pathways to regulatory network," *Molecular Biosystems*, vol. 7, pp. 843–851, March 2011.

[11] Stuart A. Kauffman, "The origins of order : self organization and selection in evolution," *Oxford university press*, 1993.

[12] Sui Huang and Donald E. Ingber, "Shape-dependent control of cell growth, differentiation, and apoptosis: Switching between attractors in cell regulatory networks," *Experimental Cell Research*, vol. 261, pp. 91–103, November 25 2000.

[13] Herman F. Fumiã and Marcelo L. Martins, "Boolean network model for cancer pathways: Predicting carcinogenesis and targeted therapy outcomes," *PLOS One*, 26 July 2013.

[14] Sriganesh Srihari, Venkatesh Raman, Hon Wai Leong, and Mark A. Ragan, "Evolution and controllability of cancer networks: A boolean perspective," *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, vol. 11, no. 1, 2014.

[15] Roland Somogyi and Carol Ann Sniegoski, "Modeling the complexity of genetic networks: Understanding multigenic and pleiotropic regulation," *Complexity*, vol. 1, pp. 45–63, July/August 1996.

[16] Ritwik Layek, Aniruddha Datta, Michael Bittner and Edward R. Dougherty, "Cancer therapy design based on pathway logic," *Bioinformatics*, vol. 27, no. 4, pp. 548–555, 2011.

[17] Miron Abramovici, Melvin A. Breuer and Arthur D. Friedman, *Digital Systems Testing and Testable Design*. Wiley-IEEE press, 1994.

[18] Ambili Remesh, "Toxicities of anticancer drugs and its management," *International Journal of Basic & Clinical Pharmacology*, vol. 1, July-August 2012.

[19] Ian H. Plenderleith, "Treating the treatment: Toxicity of cancer chemotherapy," *Canadian Family Physician*, vol. 36, October 1990.

[20] Sebolt-Leopold JS, "Advances in the development of cancer therapeutics directed against the ras-mitogen-activated protein kinase pathway," *Clin. Cancer Res.*, vol. 14, p. 36516, June 2008.

[21] Mauricio Burotto, Victoria L. Chiou, Jung-Min Lee and Elise C. Kohn, "The mapk pathway across different malignancies: A new perspective," *Cancer*, vol. 120, p. 34463456, 15 November 2014.

[22] Robert A. Weinberg, *The Biology of Cancer*. Garland Science, 2006.

[23] Ingo Wegener, *The Complexity of Boolean Functions*. John Wiley & Sons Ltd, 1987.

[24] Yves Crama and Peter L. Hammer, *Boolean Functions - Theory, Algorithms, and Applications*. Cambridge University Press, 2011.

[25] Boris Kovalerchuk, Evangelos Triantaphyllou, James F. Ruiz, Vetle I. Torvik and Evgeni Vityaev, "The reliability issue of computer-aided breast cancer diagnosis," *Journal of Computers and Biomedical Research*, vol. 33, pp. 296–313, August 2000.

[26] Anwoy Kumar Mohanty, Aniruddha Datta and Vijayanagaram Venkatraj, "A model for cancer tissue heterogeneity," *IEEE Trans Biomed Eng.*, vol. 61, pp. 966–74, March 2014.

[27] M. Bébien, S. Salinas, C. Becamel, V. Richard, L. Linares, R. A. Hipskind, "Immediate-early gene induction by the stresses anisomycin and arsenite in human osteosarcoma cells involves mapk cascade signaling to elk-1, creb and srf," *Oncogene*, vol. 22, pp. 1836–1847, March 2003.

[28] P. Dhawan, A. Bell, A. Kumar, C. Golden, K. D. Mehta, "Critical role of p42/44(mapk) activation in anisomycin and hepatocyte growth factor-induced ldl receptor expression: activation of raf-1/mek- 1/p42/44(mapk) cascade alone is sufficient to induce ldl receptor expression," *J. Lipid Res.*, vol. 40, pp. 1911–1919, October 1999.

[29] K. J. Livak, and T. D. Schmittgen, "Analysis of relative gene expression data using real-time quantitative pcr and the $2^{-\Delta\Delta C_t}$ method," *Methods*, vol. 25, no. 4, pp. 402–408, 2001.

[30] R. W. Clarkson, C. A. Shang, L. K. Levitt, T. Howard, and M. J. Waters, "Ternary complex factors elk-1 and sap-1a mediate growth hormone induced transcription of egr-1 (early growth response factor-1) in 3t3-f442a preadipocytes," *Mol. Endocrinol.*, vol. 13, no. 4, pp. 619–631, 1999.

[31] D. Levens, "How the c-myc promoter works and why it sometimes does not," *J. Natl. Cancer I. Monographs*, vol. 39, pp. 41–43, 2008.

[32] D. Rozek, and G. P. Pfeifer, "In vivo protein-dna interactions at the c jun promoter: preformed complexes mediate the uv response," *Mol. Cell. Biol.*, vol. 13, no. 9, pp. 5490–5499, 1993.

[33] H. G. Xu, R. Jin, W. Ren, L. Zou, Y. Wang, and G. P. Zhou, "Transcription factors sp1 and sp3 regulate basal transcription of the human irf-3 gene," *Biochimie.*, vol. 94, pp. 1390–7, June 2012.

[34] Sriram Sridharan, Ritwik Layek, Aniruddha Datta and Jijayanagaram Venkatraj, "Boolean modeling and fault diagnosis in oxidative stress response," *BMC Genomics*, vol. 13(Suppl 6):S4, 2012.

[35] Peter C. Nowell, "The clonal evolution of tumor cell populations," *Science*, vol. 194, pp. 23–8, 1 October 1976.

[36] David Basanta and Alexander R. A. Anderson, "Exploiting ecological principles to better understand cancer progression and treatment," *Interface Focus*, vol. 3, 6 Aug 2013.

[37] Garcia-Cao I, Song MS, Hobbs RM, Laurent G, Giorgi C, de Boer VC, Anastasiou D, Ito K, Sasaki AT, Rameh L, Carracedo A, Vander Heiden MG, Cantley LC, Pinton P, Haigis MC and Pandolfi PP., "Systemic elevation of pten induces a tumor suppressive metabolic state," *Cell*, vol. 149, pp. 49–62, 30 March 2012.

[38] JB Easton and P J Houghton, "mtor and cancer therapy," *Oncogene*, vol. 25, p. 64366446, 16 October 2006.

[39] Y. Chen, E. R. Dougherty, and M. L. Bittner, "Ratio-based decisions and the quantitative analysis of cdna microarray images," *J Biomed Opt.*, vol. 2, pp. 364–74, October 1997.

[40] A. Gelman, J. B. Carlin, H. S. Stern and D. B. Rubin, *Bayesian Data Analysis*. Chapman and Hall/CRC, 2004.

[41] A. Gelman, and J. Hill, *Data Analysis Using Regression and Multilevel/Hierarchical Models*. Cambridge University Press, 2007.

[42] P. D. Hoff, *A First Course in Bayesian Statistical Methods*. Springer Texts in Statistics, 2009.

[43] Alan E. Gelfand and Adrian F. M. Smith, "Sampling-based approaches to calculating marginal densities," *Journal of the American Statistical Association*, vol. 85, pp. 398–409, June 1990.

[44] George Casella and Edward I. George, "Explaining the gibbs sampler," *The American Statistician*, vol. 46, no. 3, pp. 167–174, 1992.

[45] Radford M. Neal, "Slice sampling," *Annals of Statistics*, vol. 31, no. 3, pp. 705–767, 2003.

[46] N. Metropolis, Arianna W. Rosenbluth, Marshall N. Rosenbluth, and Augusta H. Teller, "Equation of state calculations by fast computing machines," *J. Chemical Physics*, vol. 21, p. 10871092, 1953.

[47] Peter D. Hill, "Kernel estimation of a distribution function," *Communications in Statistics - Theory and Methods*, vol. 14, no. 3, pp. 605–620, 1985.

[48] M. C. Jones, "Simple boundary correction for kernel density estimation," *Statistics and Computing*, vol. 3, no. 3, pp. 135–146, 1993.

[49] R. W. Clarkson, C. A. Shang, L. K. Levitt, T. Howard, and M. J. Waters, "Ternary complex factors elk-1 and sap-1a mediate growth hormone induced transcription of egr-1 (early growth response factor-1) in 3t3-f442a preadipocytes," *Mol. Endocrinol.*, vol. 13, no. 4, pp. 619–631, 1999.

[50] H. G. Xu, R. Jin, W. Ren, L. Zou, Y. Wang, and G. P. Zhou, "Transcription factors sp1 and sp3 regulate basal transcription of the human irf-3 gene," *Biochimie*, vol. 94, pp. 1390–7, June 2012.

[51] M. V. Mityaev, E. P. Kopantzev, A. A. Buzdin, T. V. Vinogradova, and E. D. Sverdlov, "Functional significance of a putative sp1 transcription factor binding site in the survivin gene promoter," *Biochemistry*, vol. 73, no. 11, pp. 1183–1191, 2008.

[52] F. Verrecchia, J. Rossert, A. Mauviel, "Blocking sp1 transcription factor broadly inhibits extracellular matrix gene expression in vitro and in vivo: implications for the treatment of tissue fibrosis," *J. Invest. Dermatol.*, vol. 116, pp. 755–763, May 2001.

[53] S. L. Samson, and N. C. Wong, "Role of sp1 in insulin regulation of gene expression," *J. Mol. Endocrinol.*, vol. 29, no. 3, pp. 265–79, 2002.

[54] G. Pag'es and J. Pouysst'egur, "Transcriptional regulation of the vascular endothelial growth factor gene-a concert of activating factors," *Cardiovasc. Res.*, vol. 65, no. 3, pp. 564–573, 2005.

[55] A. K. Mohanty, A. Datta, and V. Venkatraj, "A conjugate exponential model for cancer tissue heterogeneity," *IEEE Journal of Biomedical and Health Informatics*, p. 699709, March 2016.

[56] C. Sima, J. Hua, R. Lopes, A. Datta, and M. L. Bitner, "Detecting cell growth and drug response in heterogeneous populations: A dynamic imaging approach," *IEEE 16th International Conference on Bioinformatics and Bioengineering (BIBE)*, p. 121128, 2016.

[57] Thomas N. Seyfried, and Leanne C. Huysentruyt, "On the origin of cancer metastasis," *Crit Rev Oncog.*, vol. 18, no. 1-2, p. 4373, 2013.

[58] David Basanta, Robert A. Gatenby, and Alexander R. A. Anderson, "Exploiting evolution to treat drug resistance: combination therapy and the double bind," *Mol Pharm.*, vol. 9, pp. 914–21, 2 April 2012.

[59] David Basanta, "Cell-cell interactions and evolution using evolutionary game theory,"

[60] Jorge M. Pacheco, Francisco C. Santos and David Dingli, "The ecology of cancer from an evolutionary game theory perspective," *Interface Focus*, vol. 4, 6 Aug 2014.

[61] Karl Sigmund, "Introduction to evolutionary game theory," *Proceedings of Symposia in Applied Mathematics*, vol. 69, 2011.

[62] Jacek Miekisz, "Evolutionary game theory and population dynamics, in: Capasso v., lachowicz m. (eds) multiscale problems in the life sciences," *Lecture Notes in Mathematics, Springer*, vol. 1940, 2008.

[63] Stergios J. Moschos and Christos S. Mantzoros, "The role of the igf system in cancer: From basic to clinical studies and clinical applications," *Oncology*, vol. 63, no. 4, pp. 317–32, 2002.

[64] Ian A. Prior, Paul D. Lewis, and Carla Mattos, "A comprehensive survey of ras mutations in cancer," *Cancer Res.*, 15 November 2012.

[65] Alberto Fernández-Medarde and Eugenio Santos, "Ras in cancer and developmental diseases," *Genes Cancer*, vol. 2, pp. 344–58, March 2011.

[66] Katerina Mardilovich, Shannon L Pankratz and Leslie M Shaw, "Expression and function of the insulin receptor substrate proteins in cancer," *Cell Commun Signal.*, vol. 7, 17 June 2009.

[67] A Wu, J Chen and R Baserga, "Nuclear insulin receptor substrate-1 activates promoters of cell cycle progression genes," *Oncogene*, vol. 27, pp. 397–403, 10 Jan 2008.

[68] Michael A. Davies, "Regulation, role, and targeting of akt in cancer," *J Clin Oncol.*, vol. 29, pp. 4715–7, 10 December 2011.

[69] Christopher J. Caunt, Matthew J. Sale, Paul D. Smith and Simon J. Cook, "Mek1 and mek2 inhibitors and cancer therapy: the long and winding road," *Nat Rev Cancer*, vol. 15, pp. 577–92, October 2015.

[70] Brian C. Grabiner, Valentina Nardi, Kivanc Birsoy, Richard Possemato, Kuang Shen, Sumi Sinha, Alexander Jordan, Andrew H. Beck, and David M. Sabatini, "A diverse array of cancer-associated mtor mutations are hyperactivating and can predict rapamycin sensitivity," *Cancer Discov.*, vol. 4, pp. 554–63, May 2014.

[71] Helena Pópulo, José Manuel Lopes and Paula Soares, "The mtor signalling pathway in human cancer," *Int J Mol Sci.*, vol. 13, no. 2, pp. 1886–918, 2012.

[72] Durrett R., "Branching process models of cancer in:branching process models of cancer," *Mathematical Biosciences Institute Lecture Series, Springer*, vol. 1.1, 7 December 2014.

[73] Philipp M. Altrock, Lin L. Liu and Franziska Michor, "The mathematics of cancer: integrating quantitative models," *Nat Rev Cancer*, vol. 15, pp. 730–45, December 2015.

[74] Lawrence A. Loeb, Jason H. Bielas, and Robert A. Beckman, "Cancers exhibit a mutator phenotype: Clinical implications," *Cancer Res.*, vol. 68, pp. 3551–7, 15 May 2008.

[75] Lawrence A. Loeb, Keith R. Loeb, and Jon P. Anderson, "Multiple mutations and cancer," *Proceedings of the National Academy of Sciences of the United States of America*, vol. 100, p. 776781, 4 February 2003.

[76] Alexander Davis, Ruli Gao, and Nicholas Navin, "Tumor evolution: Linear, branching, neutral or punctuated," *BBA - Reviews on Cancer*, vol. 1867, pp. 151–161, April 2017.

[77] Y. Chen, V. Kamat, E. R. Dougherty, M. L. Bittner, P. S. Meltzer, and J. M. Trent, "Ratio statistics of gene expression levels and applications to microarray data analysis," *Bioinformatics*, vol. 18, no. 9, pp. 1207–1215, 2002.

[78] Joel Grus, *Data Science from Scratch*. O'Reilly Media, April 2015.

[79] Martin Fodslette Moller, "A scaled conjugate gradient algorithm for fast supervised learning," *Neural Networks*, vol. 6, no. 4, pp. 525–533, 1993.

[80] D. M. W. Powers, "Evaluation: From precision, recall and f-measure to roc, informedness, markedness & correlation," *Journal of Machine Learning Technologies*, vol. 2, no. 1, pp. 37–63, 2011.

[81] Mauno Vihinen, "How to evaluate performance of prediction methods? measures and their interpretation in variation effect analysis," *BMC Genomics*, vol. 13(Suppl 4): S2, 2012.

[82] Sabri Boughorbel, Fethi Jarray, Mohammed El-Anbari, "Optimal classifier for imbalanced data using matthews correlation coefficient metric," *PLoS One*, vol. 12, 2 June 2017.