RICE UNIVERSITY

Deriving Executable Models of Biochemical Network Dynamics from Qualitative Data

by

Derek Ruths

A THESIS SUBMITTED
IN PARTIAL FULFILLMENT OF THE
REQUIREMENTS FOR THE DEGREE

Doctor of Philosophy

APPROVED, THESIS COMMITTEE:

Luay Nakhleh, Assistant Professor, Chair
Computer Science

Oleg Igoshin, Assistant Professor
Bioengineering

Lydia Kavraki, Professor
Computer Science

Prahlad Ram, Assistant Professor
Systems Biology Department,
MD Anderson Cancer Center

Moshe Vardi, Professor
Computer Science

HOUSTON, TEXAS
APRIL 2009

Abstract

Deriving Executable Models of Biochemical Network Dynamics from Qualitative Data and Network Connectivity

by

Derek Ruths

Progress in advancing our understanding of biological systems is limited by their sheer complexity, the cost of laboratory materials and equipment, and limitations of current laboratory technology. Computational and mathematical modeling provide ways to address these obstacles through hypothesis generation and testing without experimentation—allowing researchers to analyze system structure and dynamics in silico and, then, design lab experiments that yield desired information about phenomena of interest. These models, however, are only as accurate and complete as the data used to build them. Currently, most models are constructed from quantitative experimental data. However, since accurate quantitative measurements are hard to obtain and difficult to adapt from literature and online databases, new sources of data for building models need to be explored. In my work, I have designed methods for building and executing computational models of cellular network dynamics based on qualitative experimental data, which are more abundant, easier to obtain, and reliably reproducible. Such executable models allow for in silico perturbation, simulation, and exploration of biological systems. In this thesis, I present two gen-

eral strategies for building and executing tokenized models of biochemical networks using only qualitative data. Both methods have been successfully used to model and predict the dynamics of signaling networks in normal and cancer cell lines, rivaling the accuracy of existing methods trained on quantitative data.

I have implemented these methods in the software tools PathwayOracle and Monarch, making the new techniques I present here accessible to experimental biologists and other domain experts in cellular biology.

Acknowledgments

I gratefully acknowledge the instrumental role of my advisor, Luay Nakhleh, in guiding, shaping, and contributing to all the work discussed in this thesis. I also would like to recognize the contributions of Dr. Prahlad Ram, his post-doc Dr. Melissa Muller, and his graduate student Jen-Te Tseng. All experimental procedures performed as part of this thesis were designed and conducted by them. Dr. Ram was also responsible for the curation of the EGFR network.

Derek Ruths and Luay Nakhleh were supported in part by a Seed Grant awarded to Dr. Nakhleh from the Gulf Coast Center for Computational Cancer Research, funded by John and Ann Doerr Fund for Computational Biomedicine. Derek Ruths was also supported in party by Grant Number R01CA125109 from the National Cancer Institute. Jen-Te Tseng was supported in part by a training fellowship from the Pharmacoinformatics Training Program of the Keck Center of the Gulf Coast Consortia (NIH Grant 5 T90 DK070109-03), and Melissa Muller and Prahlad T. Ram was supported in part by a Department of Defense grant BC044268 to P.T.R.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Advances in understanding and engineering cellular biology can translate into new, more effective treatments for diseases as well as innovative, biologically-inspired approaches to clean energy generation, waste disposal, and increased food production. However, progress in these areas is limited by the relatively slow pace of experimental work. Due to the sheer complexity of living systems, the cost of laboratory materials and equipment, and the limitations of current laboratory technology, bringing a laboratory experiment to completion can take weeks, even months. Furthermore, biologists are additionally limited in terms of what parts of the cellular system they can accurately measure. As a result, biological researchers increasingly depend on computational and mathematical models of biochemical processes that can predict both structural and dynamic properties of a modeled system much more quickly.

The daunting complexity of the cell largely derives from the fact that a cell's behavior emerges out of a remarkably tangled web of biochemical interactions, referred to, in its entirety, as the *cellular network*. Scientists divide this network into three functionally distinct components (see Figure 1.1): signaling, transcription, and metabolism. The *signaling network* senses the outside world and carries this information from the cell membrane through the cytoplasm and into the nucleus where it feeds into the *transcriptional network*. The transcriptional network regulates gene

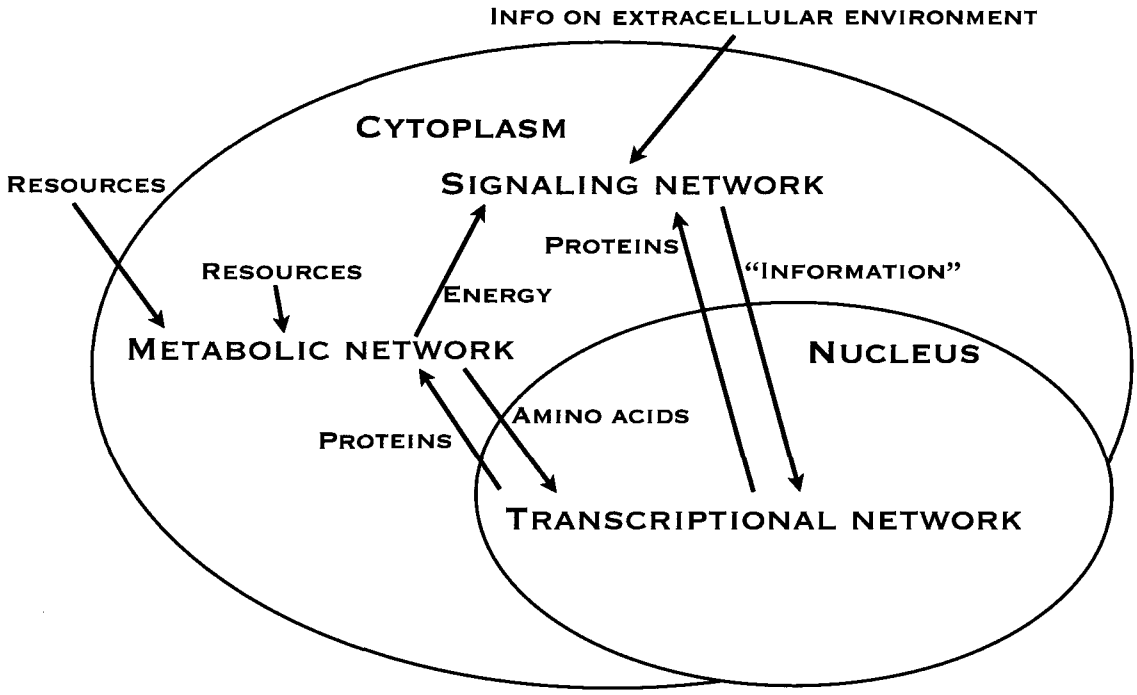


Fig. 1.1: The three major biochemical cellular networks. Signaling networks are responsible for sensing the external environment and delivering this information to the transcriptional network. This network, in turn, determines what genes are expressed. Differential gene expression changes the composition and, thus, the behavior of both the signaling and metabolic networks. The metabolic network is responsible for managing the cells resources such as energy and waste products.

expression levels which determine what messenger RNA and proteins are synthesized at any given point in time. These gene products eventually move either back into the signaling network, modifying its overall behavior, or into the *metabolic network* which is responsible for managing cellular resources such as energy (adenosine triphosphate or *ATP*) levels, amino-acids recycled from degradation of unneeded proteins, and other molecules essential to the functioning of the cell. The tight interplay among these three networks determines the behavior of a cell. At any given point in time, a cell in the human body may express any number of over 30,000 different proteins

and mRNAs encoded in the human genome. Different cell types at different stages in their life cycles express different combinations of these proteins. Thus, from the outset, mapping the vast and densely intertwined cellular network, a necessary step in unraveling the determinants of cellular behavior, seems an immensely challenging task.

Certainly, the best way to gain an understanding of these systems is by studying them directly in the laboratory, which is how biologists have traditionally approached the problem of charting cellular networks. Unfortunately, the current state of laboratory technologies, while an area of active and productive research, does not provide biologists with the tools they need to efficiently and effectively interrogate the many hundreds of thousands of biochemical reactions comprising the cellular network.

Traditional experimental techniques are intensely manual. Western blots, still a favored method for making accurate measurements of biological quantities require the involvement of laboratory technicians and researchers at many steps in the process, making each experiment expensive both in terms of laboratory resources and human effort. Such techniques, as they exist today, do not scale up to handle the many thousands of simultaneous measurements biologists would like to make.

High-throughput techniques have high error rates. The recent development of high-throughput techniques such as the gene expression arrays, reverse phase protein arrays, and mass spectra phosphoproteomics have made it possible for researchers

to make hundreds of measurements in a single experiment [RSSR09]. However, such techniques exhibit high error rates as experimental results may be strongly influenced by a range of factors including experimental equipment and the binding selectivity of biological probes [FBH+03, KR05, SV06].

Limited biological probes. Even when researchers resolve the high-error rates of high-throughput techniques, they will still need to overcome the overall lack of probes for measuring quantities of different biological compounds. In general, for a biologist to measure the concentration of a specific protein, she must have a probe that will detect it (usually by binding to it and, thereby, giving off a signal, such as by becoming fluorescent). Currently, developing these probes is a manual process. As few probes currently exist, developing them for all 30,000 proteins in the human genome is likely to be an expensive and long endeavor.

Limited ability to manipulate experimental conditions. Another issue complicating the process of studying cellular systems experimentally is the challenge of inducing specific cellular states. For example, if a biomedical researcher wants to study how a cancer cell responds to the elimination of a specific protein, he has a handful of options, each of which has significant side-effects that may dramatically change the way a cancer cell will response: small interfering RNA (siRNA) will suppress expression of the gene that codes for the protein, however siRNA is highly toxic to the cell and furthermore will not degrade a protein that has already been produced

by the cell; the gene can be completely deleted from the cancer cell's genome which will eliminate expression of the gene, but may have unintended effects on the overall layout of the genome, proximity of other genes and, therefore, their expression levels as well; finally, the biologist may use a drug (if one exists) that targets the specific protein, but must remain aware than many drugs have *off-target* effects that lead it to bind with other proteins in the cell. Thus, even properly setting up an experiment in the laboratory can be an arduous process that, ultimately, may require the biologist to accept various variables that are beyond his control.

These challenges that researchers face at many points in the process of designing, conducting, and interpreting the results of an experiment have provided much motivation for the development of modeling techniques that allow the biologist to "conduct" experiments *in silico.* The intention of such methods is not to entirely replace laboratory work, but rather to allow the experimentalist to use the computer to evaluate numerous possible experiments, refine their understanding of the biological system, and, ultimately, craft laboratory experiments that will yield the information that is most valuable to the researcher. This process of using computational techniques to evaluate, design, and improve experimental work is generally called *hypothesis generation* and *testing.*

The process of conducting experiments motivated by the predictions of computational or mathematical models also provides additional information that can be used to improve the correctness of the models. By identifying parts of a model that

agree or disagree with experimental results, it is possible to refine the model. Thus, the overarching process of modeling and experimentation is an iterative process that improves both predictive models as well as knowledge of biological system details.

Insofar as computational and mathematical techniques are applied to cellular networks, they generally fall into one of two categories: structural or dynamic models. Structural models pertain solely to the topology of the biochemical networks—capturing the patterns of connectivity that biochemical reactions create among a set of proteins and genes. Dynamic models characterize how the system as a whole behaves over time: given a starting state of the system (a specific set of protein concentrations or gene expression levels), such a model characterizes subsequent states that the system enters as time progresses.

In this thesis, we have focused on developing new computational techniques for abstract modeling of certain aspects of the dynamics of cellular networks. Thus far, we have restricted our attention to signaling networks which have been implicated in numerous diseases and are, therefore, heavily studied within the biomedical community. However, the methods we have developed can be extended to model metabolic and gene regulatory networks. This is an important topic for future work.

Our contribution is not only new computational models of network dynamics, but also a more general conceptual approach to the kind of data that can be used to build such models: our work departs from the traditional quantitative data-based training that characterizes current network dynamic modeling approaches in favor of using

qualitative data. This distinction has significant implications for the quality of and speed with which computational models of cellular network dynamics can be built.

For a computational or mathematical model of signaling network dynamics to make accurate predictions, it must be trained using known properties of the underlying biochemical network. This requires experimental data. Often this data is supplied by perturbation experiments which measure the dynamic response of the signaling network to different environmental and internal conditions. The measurements typically are read as changes in protein concentration, cell population size, or phenotypic outcomes. Numerous methods, ranging from ordinary differential equations to Bayesian networks, use these quantitative experimental measurements directly to infer model parameter values (e.g., [KBP+08, LW08, NWN+08, DGM+06]).

Parameter values (and, therefore the models they are contained in) can only be as accurate as the experimental results from which they were derived. When sufficient training data is available to determine accurate parameter values, existing quantitative modeling methods such as ordinary differential equations, can provide extremely accurate predictions of network dynamics. Often, however, such datasets are hard to obtain. Noisy data can be a significant source of modeling error since experimental results may be strongly influenced by a range of factors including limitations in laboratory technology (e.g., microarrays) and antibodies for measuring protein concentrations [FBH+03, KR05, SV06].

The availability of qualitative data makes them an appealing resource for deriving

predictive computational models. The biomedical literature is replete with qualitative observations of "dominant interactions", "increasing activity", and "antagonistic effects"—all of which characterize the results of perturbation experiments without depending on exact measurement values obtained. Additionally, public databases such as KEGG [KAG+08] and Science's STKE (`http://stke.sciencemag.org/cm`) report interactions as activating and inhibiting interactions (i.e., as $x \rightarrow y$ and $x \dashv y$), which are inherently qualitative. For example, while the kinase p-AKT may not always exhibit a 2.32 fold increase under a specific experimental condition, the fact that it increases may be highly reproducible.

The central question posed by this thesis is: can qualitative data alone be used to build predictive models of biochemical networks? The contributions of this thesis include two strategies for building computational models of signaling network dynamics using only qualitative data and the network topology. These two strategies serve different purposes. The first method is a stochastic execution strategy that uses only connectivity to predict the overall dynamics of a signaling network [RMT+08]. We apply this execution strategy to a Petri net, constructing a modeling method called the *signaling Petri net simulator*. While in this work we use the stochastic execution strategy with a Petri net, it may be applied to any other executable modeling technique such as state charts and boolean networks; we identify these extensions as a direction for future work. The performance of our method indicates that many of the major behavioral characteristics of a network can be derived from network

connectivity alone. This method will be discussed in Chapter 3.

The second method is a deterministic method that trains a parameterized model of a signaling network using only qualitative data [RN]. Using this method we determined the degree to which qualitative experimental data, in addition to network connectivity, allowed the construction of predictive models. At the core of this method, we use a non-linear optimization to derive parameter values from qualitative data. In order to evaluate the utility of such an approach, we have applied it to a state equation-based model of signaling network dynamics. However, like the stochastic execution method discussed above, the general optimization-based approach to parameterization may be applied to other executable modeling techniques as well; this is a topic for future work. The performance and accuracy of this method demonstrate that optimization-based parameter value search using qualitative data as constraints can yield accurate models of signaling networks. This method will be discussed in Chapter 4.

Overall, our results in this thesis show that, under the cell-lines and experimental conditions we considered when evaluating the performance of our methods, even though qualitative data is less precise than quantitative data, it can be used to generate computational models with accurate predictions of network dynamics.

In order to make our methods available for use by experimentalists, both have been implemented and deployed as software tools, which will be discussed in Chapter 5. PathwayOracle is a stand-alone tool that implements the signaling Petri net (for

dynamic analysis of signaling networks) and also includes features for analyzing the connectivity of signaling networks [RNR08]. Monarch is a web-based software tool that provides an implementation of the deterministic, optimization-based approach to model construction and execution [RN].

We begin the rest of this thesis in Chapter 2 with a discussion of the biological and computational background and prior work that creates context for the work we present here.

Chapter 2

Background

In this chapter, we introduce biological concepts and modeling techniques and methods relevant to the two methods at the core of this thesis.

Biological background. Because our methods have been designed to model signaling networks, we discuss the biochemical principles of cellular signaling. We validated our methods against actual experimental data which were generated using perturbation experimental techniques. We briefly discuss this experimental methodology as well.

Computational and mathematical modeling methods. Our methods should be understood within the context of existing approaches used to model biochemical network dynamics. These include mathematical approaches such as ordinary differential equations as well as a range of computational methods. In particular, we discuss tokenized systems, such as Petri nets, central to the two methods we present in this thesis, and their prior applications in biochemical network modeling.

2.1 Signaling Networks

Signaling networks are complex, interdependent cascades of signals that process extracellular stimuli, received at the plasma membrane of a cell, and funnel them to the nucleus, where they enter the gene regulatory system. These signaling networks underlie how cells communicate with one another, and how they make decisions about their phenotypic changes, such as division, differentiation, and death. Further, malfunction of these networks may alter phenotypic changes that cells are supposed to undergo under normal conditions, and potentially lead to devastating consequences on the organism. For example, altered cellular signaling networks can give rise to the oncogenic properties of cancer cells [Hun00, HW00], increase a person's susceptibility to heart disease [FCAB05], and have been shown to be responsible for many other devastating diseases such as congenital abnormalities, metabolic disorders and immunological abnormalities [Hun00, BMRT96].

The sensory information that propagates through signaling networks originates at *receptor* molecules which are, with rare exception, trans-membrane proteins: proteins lodged in the cell membrane in such a way that one portion of the protein dangles outside the cell and one portion sticks inside the cell. The extracellular portion of the receptor is designed to bind a specific molecule (called the ligand): a hormone, nutrient, toxin, or any myriad other type of molecules of interest to the cell. When the receptor binds its ligand, it undergoes a change in shape, called a conformational change, which alters the orientation and shape of the intracellular portion. As this

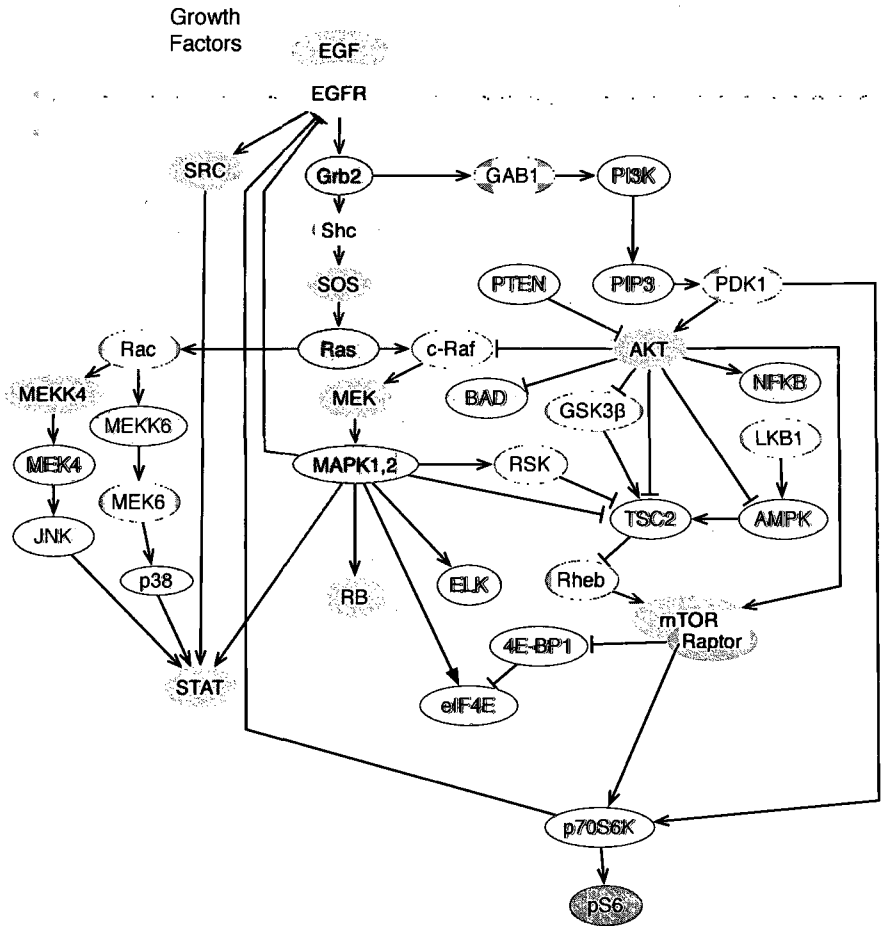


Fig. 2.1: A MAPK1,2 and AKT network downstream from EGFR, which we assembled from various sources, and used for the case study analysis in this work [RMT+08]. An edge $u \rightarrow v$ ending with an arrow indicates an activating reaction, while an edge $u \dashv v$ indicates an inhibiting reaction.

intracellular portion undergoes its change, it reveals a sequence of amino-acids, called a *domain*, that mediates a biochemical reaction with another protein in the cell. This begins a cascade of biochemical reactions which, much like the ligand-receptor interaction, consists of proteins changing the shape of other proteins. The "signal", the information that the ligand was found, propagates through these conformational changes in proteins.

Because of the complexity of signaling biochemistry, biologists often favor a conceptual abstraction in which a protein can be in one of two configurations: active, in which case it can interact with other proteins, and inactive, in which case it is relatively inert and cannot interact with other proteins. Proteins in the active state are carriers of the signal. An active protein, X, can either pass this signal along to protein Y through an *activating* interaction, or suppress this signal through an *inhibiting* interaction: $X \rightarrow Y$ and $X \dashv Y$, respectively.

Both kinds of these interactions can be observed in Figure 2.1. AKT, for example, inhibits c-Raf, TSC2, AMPK, and GSK3b; it activates mTOR. According to this model, when AKT becomes active, it will deactivate molecules of c-Raf, TSC2, AMPK, and GSK3b, while activating mTOR. This will have the effect of diminishing the signal passing through the former molecules, while increasing the strength of any signal passing through mTOR.

While we will adhere to this abstraction of signaling throughout this thesis, there are two aspects of the underlying biochemistry which are relevant to the task of modeling signaling networks: protein activity-levels and the enzymatic mechanisms of signaling.

Protein concentration. The dynamics of the propagation of signal through a signaling network greatly depends on the strength of the signal at any point in time. On a biochemical level, the strength of a signal in a given protein corresponds to the concentration of proteins of that type in an active state, hereafter referred to as

the *activity-level* of a protein. Thus, the methods we devise for modeling signaling network dynamics will necessarily account for not only the different types of proteins involved in the signaling network, but also for an abstraction of the activity-levels each of these proteins have over time.

Signaling through enzymatic reactions. Another important characteristic of signaling networks, which distinguishes them from metabolic networks and, less so, from transcriptional networks is that most signaling interactions occur through enzymatic reactions: when protein A modifies the active state of protein B, neither protein A's existence nor its activity-level changes. Thus, A can pass signal to or suppress signal in another protein *without losing the signal itself.* This is an entirely different paradigm from many metabolic processes in which chains of biochemical reactions build or break down molecules. Even transcription differs somewhat: mRNA molecules generally have to bind and remain bound to another mRNA or a strand of DNA in order to induce its effect. Thus, while in metabolic and transcriptional networks molecules are *consumed* as the network state evolves, in signaling networks proteins and their activities are generally conserved as they interact with other signaling proteins.

2.1.1 The EGFR Network

While our overarching goal is the development of computational methods that can be used with any signaling network, we needed a signaling network against which to

test our methods, study their performance, and evaluate their accuracy. We selected a network of pathways downstream of the epidermal growth factor receptor (EGFR).

The network we study and refer to throughout this thesis is shown in Figure 2.1. This network was chosen because the EGFR receptor and its downstream signaling network play a very important role in development, differentiation, and oncogenic transformation. Two very important signaling molecules within the cell are the Mitogen-activated protein kinase (MAPK) and protein kinase B (AKT), both of which can be activated by EGFR, and contains several potential regulatory pathways between them. We constructed a model network of EGF regulation of MAPK and AKT which includes several feedback and feed-forward loops all of which were constructed based on experimental findings from different laboratories around the world [KCCR04, MCEB+05, MLLA05, KM05, AHL+06, LSX+07, IOZ+06, ORS+06]. For both methods we propose in this thesis, we analyzed, both experimentally and computationally, the change in activity-level of several proteins in response to targeted perturbations (discussed in Section 2.2).

2.2 Perturbation Experiments

While it is often possible to measure and detect individual signaling proteins, much of the complexity of cellular signaling lies in the interactions among these proteins. One of the major methods used to detect new interactions and proteins is through perturbation experiments. Because numerous perturbation studies already exist in

the literature and because many biological and biomedical labs are set up to conduct perturbation experiments, we focused on perturbation experiments as the source for all experimental results used to train and test our biological models. Here we briefly describe the general structure of a perturbation experiment as well as the kind of data that it generates.

Within the context of signaling networks, the objective of a perturbation experiment is to determine the effect that a given signaling protein, X, has on the activity-levels of other proteins, say A, B, and C, present in a specific kind of cell (e.g., a breast cancer cell). In order to obtain this information, first a culture of the specific breast cancer cell-line is grown. The population of cells is divided into two different groups: the control and the perturbed groups. The perturbed group is exposed to a drug or other pharamcological or genetic agent that blocks protein X from the cell. This step is referred to as the perturbation. Since the activity-level of target protein, X, is usually reduced, this is often called either *knockout* or *knockdown.*

After protein X is perturbed, both cell populations are stimulated with a ligand that initiates a signaling cascade. The groups are left for some pre-determined amount of time and then are collected and lysed. Concentrations of active forms and total concentrations of proteins A, B, and C are measured in the control group and in the perturbed group. The activity-level of protein P is, $\frac{\text{Active protein concentration}}{\text{Total protein concentration}}$. Measurements of protein A in the two groups yield activity-level measurements A_c and A_p, activity-level in the control and the perturbed groups, respectively (similarly

for B_c and B_p, C_c and C_p).

Given these activity-level measurements, the biologist has now gained insights into the effect that protein X has on the activity-level of proteins A, B, and C under the specific conditions of the experiment. If $A_p > A_c$, then blocking X increased the activity-level of A, implying that X has an inhibitory effect on A, either directly or indirectly through other molecules. On the other hand, if $A_p < A_c$, then blocking X decreased the activity-level of A, implying that X has an activating effect on A.

Gaining insight into the effect of a given protein on the activity-level of other proteins in the network is a crucial step in determining the overall connectivity of the network. Of course, the observation that A's activity-level decreases when X is perturbed does not indicate that X activates A through a direct interaction. However, it does provide evidence that there is an activating path leading from X to A in the signaling network being studied.

It is worth noting that there are many variations that can be made to a perturbation experiment, some of which include:

- *Combinational perturbations* involve the perturbation of more than one protein (e.g., [NWN+08]). This type of experiment is gaining interest in biomedical labs in which there is interest in understanding the effect of multiple drugs on a given type of diseased cell.

- *Time series data collection* requires that control and perturbed populations be collected and lysed at multiple time points, yielding information about the

activity-levels of proteins at multiple time points (e.g., [HF96]). Note that this type of experiment can be much more time and resource demanding as individual control and perturbation groups must be set up for each time collection point. However, because multiple time points are represented, such experimental data can be more informative than single time-point experiments.

- *Varying the delivery and exposure time of the perturbation* can be used to control or study the strength of the perturbation effect [NWN+08].

2.3 Modeling Methods

Methods for modeling biochemical network dynamics fall into two classes: mathematical and computational models [FH07]. Mathematical models are based on the concept of the transfer function which relates biochemical quantities to one another (i.e., $x = 2 \cdot y$ indicates that the value of x is twice the value of y). Where scale and theory permit, the properties of the system can be analytically derived by analyzing this transfer function. As these systems expand in size and complexity beyond the realm of analytics, they can be simulated, which renders an estimate of how the system behaves over a specific period of time given a specific starting condition.

Computational models, in contrast, are rooted in the idea of the state machine: the biochemical system is modeled as one that moves from one global state to another. The system state captures the properties of individual proteins, RNA, and other molecules (i.e., activity-levels, concentration gradients, localization in the cell, etc.).

$$\frac{dx}{dt} = 2x + 3y$$
$$\frac{dy}{dt} = \frac{1}{2}xy$$

Fig. 2.2: A very simple system of ordinary differential equations in which the system being modeled has two quantities: x and y. The rate of change of each is a function of the quantities themselves.

The biochemical network's dynamics consists of a sequence of global states that are connected by some transition function that contains the rules governing how the internal properties of any given state may change between adjacent states.

The transition function takes an input state and yields one or more states that are legal transitions for the system. The existence of this function provides an implicit description of the state space of a biochemical network. Clearly, as the modeled biochemical network grows in complexity, the state space will quickly expand beyond the realm of exhaustive analysis or explicit modeling. However, even given such large state spaces, it is possible to use techniques derived from formal verification and model checking to identify properties of this state space and, in doing so, identify properties of the system dynamics. Where explicit state space analysis is not possible, the transition function can be executed: applied over and over in order to yield a sequence of states. Because the transition function is formalized as a sequence of steps that generate a new state from an input state, computational methods have been called *executable methods.*

2.3.1 Mathematical Models: Ordinary Differential Equations

The dominant mathematical technique used for modeling biochemical network dynamics is systems of ordinary differential equations (ODE). As shown in Figure 2.2, an ODE expresses the rate of change of one quantity (e.g., a protein's concentration) as a function of other quantities in the system (e.g., reaction rates, the concentration of other proteins). A system of ODEs, therefore, can express the rates of change of a set of protein concentrations, gene expression levels, or metabolite concentration. Such a system can be used in two different ways.

Simulation. Because a system of ODEs provides the rate of change of quantities over time, it can be used to project the behavior of a biochemical network forward from a specified starting point. ODEs have been successfully used to simulate the dynamics of many different biochemical systems including signaling (e.g., [NI02, APL05]), transcriptional (e.g., [CHC99]), metabolic (e.g., [Gor99]), and even multi-cellular systems (e.g., [THHD07]). In general, however, simulation is complicated by the fact that ODEs are, by their nature, continuous. Simulation requires that discrete time steps be identified and the rate of changes, which should be continuous, be broken into discrete functions evaluated at each time step. Particularly as the model increases in size (e.g., hundreds of proteins and interactions), the time discretization scheme selected can dramatically influence the results obtained from the simulation, making this a potential source of significant error in simulation.

Property analysis. Because an ODE is a closed-form expression of a biochemical network's dynamics, various analytical techniques can be applied to characterize the way that the network will behave under different conditions. A frequently studied property is the steady state solution of a system of ODEs: what is the behavior of the system we would expect to see after a "very" long period of time? Mathematical analysis can identify how the starting point of the system affects the long-term behavior of the system [GSBH07]. The power of these analytical techniques, however, decreases as the size and complexity of the ODEs increase. For large-scale systems of tens to hundreds of proteins, it is difficult, often practically impossible, to derive the properties of the system using mathematical analysis alone.

Both of these uses for ODE-based models depend on having values for parameters used in the model: reaction rates, diffusion constants, and stoichiometric coefficients frequently occur as values in individual ODEs. Obtaining biologically correct values for these parameters is crucial in building an accurate model of the underlying biochemical network. Furthermore, ODE-based models are highly sensitive to these parameter values: small variations in parameter values can dramatically change the overall behavior of the network. As a result, biologists must take great care when obtaining parameter values, which are typically derived from experiments.

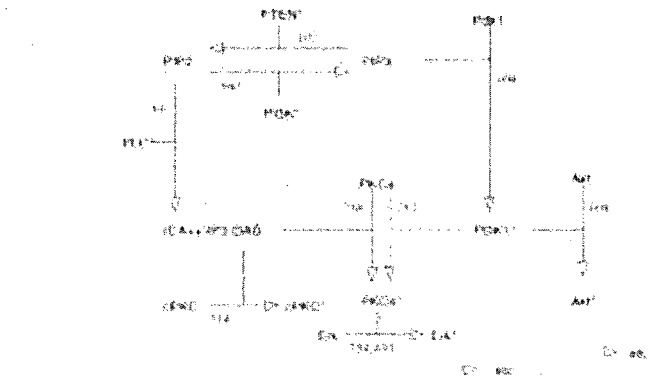
2.3.2 Piece-wise ODEs

As our focus in this thesis is to derive network dynamics from qualitative data and network connectivity, the class of qualitative piece-wise ODEs are noteworthy. In

recent years, piece-wise ODEs have been used to model dynamic features of genetic regulatory networks [DGH+04,dJGBH04]. The ODEs themselves are piece-wise linear functions - continuous, but not smooth. Insights into the model can be gained without having parameter values since all the points where two linear regions meet create a critical point where the behavior of the system may change. Identifying these different critical points allows the state space to be broken into qualitative states according to the critical points that bound them. Simulation of a piece-wise ODE, then, becomes the process of determining what sequence of qualitative states are visited as time proceeds. Furthermore, steady states of these systems can be studied without using parameter values by determining what qualitative states the system tends towards [DGH+04].

2.3.3 Executable Methods

Executable models share in common a discretized model of time: the system of interest is assumed to move from one state to another state between time points. As with ODE-based models, executable models can be used both for execution (the computational analog of simulation) and for analysis of the structure of the state-space and, therefore, analysis of properties of the modeled system. Different executable methods have different simulation and analytical capabilities. Here we discuss several of the most common executable modeling method classes. Each class is defined by the nature of the transition function used in the model.



```
rl [64].PIJKI.meta.PIP2.to.PIPJ]:
  PIP2 [?PIJKI:PIJKI - act] => PIPJ [?PIJKI:PIJKI - act] .

rl [107.Pten.pp.PIP].to.PIP2]:
  PIPJ [Pten - act] => PIP2 [Pten - act] .

rl [84.PLC.act.PIP2.Ca]:
  Ca++ [CM| PIP2 cm:Soup [?PLC:PLC - act][cyto:Soup]]
  =>
  [CM|cm:Soup [Ca++ DAG IP3] [?PLC:PLC - act][cyto:Soup]] .

rl [?58.DAG.act.nPKC]:
  [CM | DAG cm:Soup [cyto:Soup ?nPKC:nPKC]]
   =>
  [CM | DAG cm:Soup [cyto:Soup [?nPKC:nPKC - act]]] .
```

Fig. 2.3: An example of rewriting logic, taken from [EKL+02]. Note that the network above has been partially rewritten using a set of rules indicating how different objects are transformed or altered when the rules are applied.

Rewriting Logic

In rewriting logic specification systems (see Figure 2.3), the state of the system is a set of objects. The model consists of rules that indicate what combinations of input objects produce other objects. Rules can either specify specific inputs and outputs or classes of inputs and outputs (e.g., signaling proteins, chemical compounds, ligands, proteins with motif X, etc.). Given a current state, the next state is derived by applying one rule to the current state—effectively consuming some set of objects and producing some new set that is added to the current state.

Given a rewriting logic model, it is possible to either simulate the behavior of the

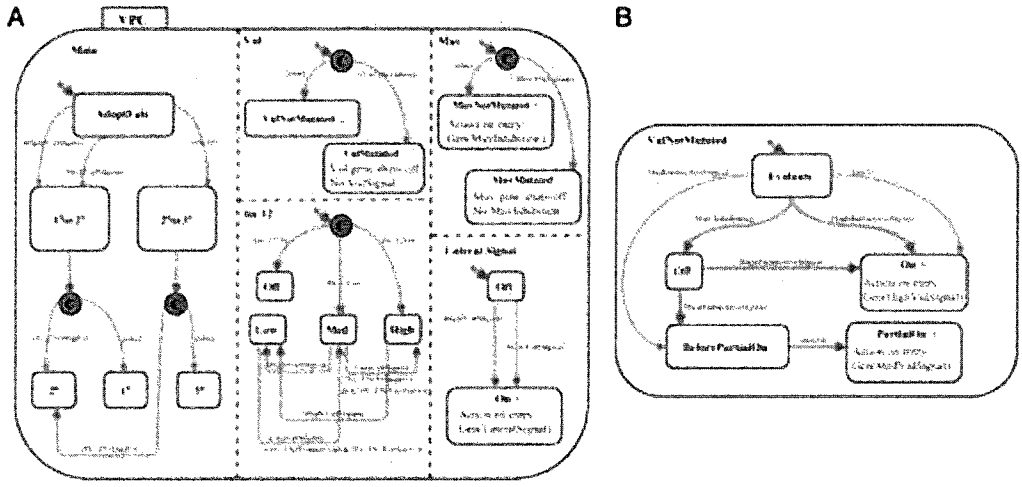


Fig. 2.4: An example of a statechart, taken from [FPHS05]. Individual boxes indicate states (or parts of a state) and arrows indicate legal transitions between states.

system by applying the rewriting rules iteratively or test, using methods from formal logic, whether certain properties hold for the model. PathwayLogic is a framework, built on top of the Maude software system, that has been used to model biochemical networks ranging in scale from signaling networks through neural networks [EKL+02, ITMB07].

Statecharts

A statechart (also called a *state diagram*) is a directed graph that explicitly models the state space of a system. As shown in Figure 2.4, states are represented as vertices and legal state transitions are the directed edges connecting vertices. In order to reduce the size of the state space, the model state may be broken into its orthogonal components: subsets of the state that are independent on one another.

Statecharts have been applied to modeling various biochemical systems such as

the aspects of C. elegans development [FPHS05]. For relatively small systems, statecharts provide useful analytical capabilities. However, the broader application of statecharts to biochemical networks is limited by the enormous size of the state space for large biochemical networks (which are now common). Because statecharts must explicitly represent the set of states (or orthogonal components), as the state space increases in size, so does the size of the representation. Biochemical networks do not exhibit significant independence among state variables: there are numerous indirect relationships among state variables. As a result, even decomposition of the network into orthogonal components does not significantly reduce statechart model sizes.

Boolean Networks

Fundamentally, a boolean network is a set of boolean variables whose values are determined by the values of other variables in the set. The term *network* is derived from the fact that these relationships between variables can be efficiently expressed as a directed graph in which vertices represent boolean variables and the directed edges indicate the relationships between the variables, as shown in Figure 2.5.

The first boolean networks were used to model genetic regulatory networks [Kau69]. Since then, boolean networks have been further extended for modeling gene networks and, more recently, signaling networks [LAA06]. In general, boolean networks have had more success in modeling transcriptional network dynamics than those of signaling networks. This is likely due to the fact that signaling networks have demonstrated greater sensitivity to concentrations of individual proteins: genes have been quite ef-

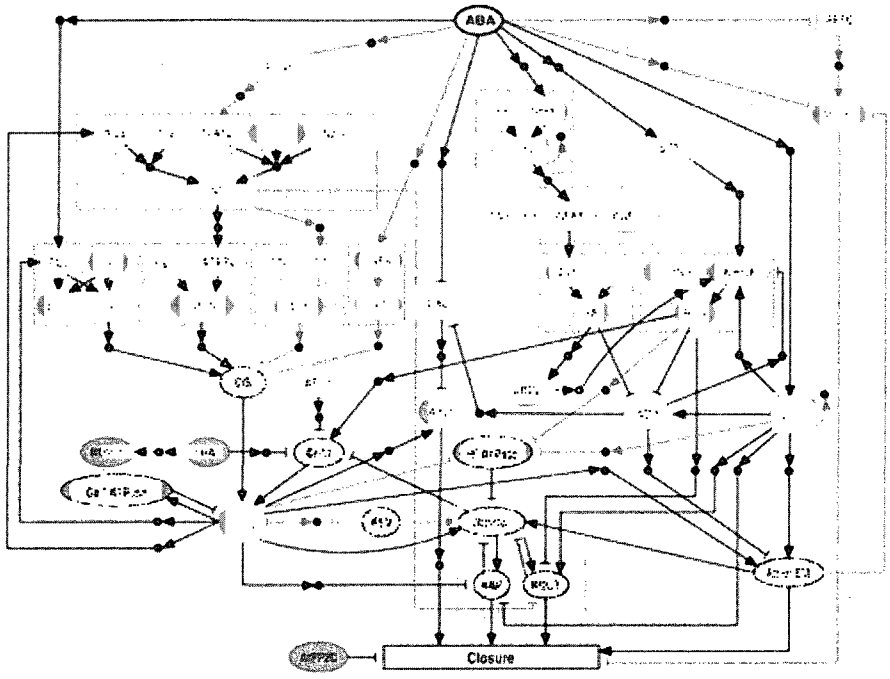


Fig. 2.5: An example of a boolean network, taken from [LAA06]. Individual nodes represent variables that can take on a boolean value. Edges indicate positive and inverting effects among variables. This boolean network models a signaling network in the stomata cell of a plant.

fectively modeled as either being on or off at any point in time [RMT+08].

One notable success in modeling signaling network dynamics is the work in [LAA06] in which the effects of protein knockdowns on signal transduction speed in plant stomata cells were predicted using a boolean model. This work will be discussed in greater detail in Chapter 3.

Petri nets

As mentioned above, one of the challenges in using boolean networks to model signaling networks is modeling signaling protein activity-levels, which are fundamentally continuous values. It is important to note that boolean networks are fully capable of

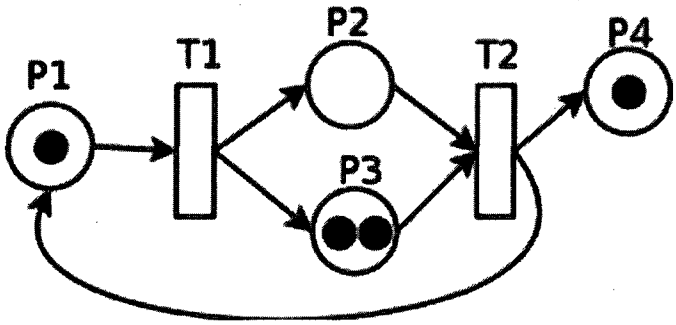


Fig. 2.6: A simple Petri net. Circles are places which can contain arbitrary quantities of resources called *tokens*. Rectangles are transitions that depict events that reallocate tokens from input places to output places.

modeling ranges of concentration values. Modeling this introduces a degree of complexity that may make purely boolean representations of signaling networks difficult to work with from an analytical point-of-view. The transformation from representing signals as on/off to a range of values (i.e., between 0 and n) requires replacing the single boolean node with $log_2 n$ nodes. The combined values of these nodes will represent the activity-level of the protein. Such a transformation will also require changes to the connectivity between these nodes and nodes representing the activity-level of other proteins. While this transformation is certainly possible, we are not aware of any efforts in this direction, likely because it faces several challenges: the size of the networks will be larger and the connectivity of networks will be more complicated; there is no one-to-one correlation between a protein and a node and an interaction and its edge; every transformation will require a fixed maximum activity-level which may be difficult to determine a priori.

Conceptually, Petri nets can be seen as a heavily extended version of boolean networks. In Petri nets, boolean variables are replaced by *places* which can hold an

arbitrary number of discrete units called *tokens*. As shown in Figure 2.6, structurally, a Petri net is a directed bipartite graph in which a node is either a place (that holds tokens) or a *transition* which represents an event that reallocated tokens. When a transition *fires*, it moves tokens from its inputs (which are places) to its outputs (which are also places). Thus, tokens circulate through the system according to the firing of these transitions. Execution of a Petri net involves firing a sequence of transitions which simulates the effect of the corresponding sequence of events happening in the underlying system.

Petri nets have been used extensively in the engineering community for the purpose of modeling resource allocation and communication systems [DA05]. In addition to their utility for visualizing and simulating processes, a great deal of work has been done in the area of studying the mathematical and formal properties of Petri nets. Three properties, in particular, have been studied extensively:

Reachability is the problem of deciding whether there is a sequence of transitions (called a *firing sequence*) that when fired in order will transition the system from one specific state to another specific state. Establishing reachability between two states indicates the potential for the system undergo a specific set of changes. In biology, reachability can be used to determine whether it is possible for a system to enter a specific state when starting from a known initial condition (e.g., [SHK06]).

Liveness of a transition, t, asks for guarantees about when t will be able to fire.

A transition can only fire when tokens exist in all its input places. Thus, a transition may (1) never be able to fire (L_0), (2) be fired at least once (L_1), (3) be able to fire some finite number of times (L_2), (4) be able to fire an infinite number of times under a specific condition (L_3), or (5) be able to fire an infinite number of times under any condition (L_4). Though liveness has been mentioned within the context of biological system modeling, we know of no instances where this property has been actually used to study a biochemical network [Cha07].

Boundedness is the problem of deciding whether there is a positive number that is an upper bound on the number of tokens a place ever have. When boundedness can be proven, it is relevant to biochemical systems insofar as it can establish the maximum activity-level, gene-expression level, or metabolite concentration that a given molecule can achieve (or it can establish that a molecule may have no bound on its concentration) [Cha07, SBSW07].

Many different forms of Petri nets exist and, depending on the formalization, may be easier or harder to prove the properties above for.

There have been significant efforts in modeling biochemical networks as Petri nets. These efforts concern either studying the properties of the biochemical system using theoretical properties of the Petri net (e.g., [LSG+06, PRA05, SHK06]) or executing the Petri net to obtain an estimate of the biochemical network dynamics (e.g., [SBSW07, MTA+03]).

As our work in this thesis focuses on the question of estimating biochemical net-

work dynamics, this latter research direction is of direct relevance to our work here. The most comprehensive work in this area has been done using hybrid functional Petri nets (HFPNs) [Cha07, DFM+04, MTA+03]. A hybrid functional Petri net is a Petri net in which edges have weights and transitions have probabilistic firing functions. The edge weights indicate the number of tokens which will be moved along it when the associated transition fires (i.e., the number of tokens that will be removed from the input places or the number of tokens that will be put into the output places). The probabilistic firing functions determine when the associated transition fires and commonly are written as the likelihood of the transition firing again after not firing for a period τ. For example, if a transition should fire in a normal distribution centered around 1 second of delay (with a standard deviation of σ^2), then the likelihood that the transition will fire after a delay of τ is $\frac{1}{\sigma\sqrt{2\pi}} \exp(-\frac{(\tau-1)^2}{2\sigma^2})$. Thus, unlike in the basic firing model in which the firing sequence is deterministic, the firing time of transitions in HFPNs is stochastic in an attempt to model the stochastic nature of the biochemical reactions being modeled.

HFPNs have been successfully used to model numerous biological systems ranging from cellular signaling (e.g., [DFM+04]) to large multicellular systems (e.g., [SBSW07]). Because of the complexity of the model and the numerous parameters required in order to build such a model, HFPNs function somewhat as a discrete analog to the ordinary differential equation. Thus, while the many modeling parameters allow for extremely accurate models, they also present serious issues for the rapid construction

of models, particularly when the underlying systems are large or largely uncharacterized (few parameters and mechanisms are known).

2.4 Our Work

In this thesis, we introduce two new methods for modeling the dynamics of signaling networks. Both are computational methods: both involve a discrete model of time and emphasize a specific execution strategy that evolves the state of the system from one time step to another. In Chapter 3 we present a new Petri net-based model and execution strategy; in Chapter 4 we present a tokenized model in which the state of the system evolves according to explicitly written state equations.

Chapter 3

From Connectivity to Dynamics: Stochastic Execution

Chapters 1 and 2 presented evidence and arguments supporting the need for computational tools in biological and biomedical research into cellular biological networks. While many computational and mathematical tools and methods already exist, we argued for the need for methods that could use qualitative data: data that could be quickly and reliably derived from experiments, data that is abundantly available in many online databases and in the literature. In this chapter, we present the first of two methods we have developed to achieve this goal [RMT+08].

A challenge posed by many existing methods is that they rely on having kinetic parameters for all biochemical reactions comprising the network: parameters whose values must be derived from quantitative experimental data, which is recognized to be a significant obstacle to their utility in experimental work [Bai01, SHK06]. Here we present a novel signaling network simulator that can provide many of the insights of existing dynamic methods without needing any kinetic parameters. Our approach makes use of recent discoveries that network structure alone can determine many aspects of a network's dynamics [LAA06, AC03, KB05, Bai01]. When compared against experimental results, our method correctly predicted 90% of the cases considered. In

those where it did not agree, our approach provided valuable insights into discrepancies between known network structure and experimental observations.

Our is not the first attempt to approach the network dynamics modeling problem using only network connectivity. Several recent efforts in this direction have produced encouraging results. An approach using a boolean network simulation method, based on work in the area of gene regulatory networks, successfully used only signaling network connectivity information to predict the speed of signal transduction through a stomata signaling network [LAA06]. The use of piecewise linear systems of ODEs, as discussed in Chapter 2, have also had success in analyzing some of the dynamics of gene regulatory and signaling networks without using exact kinetic parameters (e.g. [GK73, dJGBH04, MOMR08]). The challenge to extending the method in [LAA06] to model individual protein responses to signal transduction is the boolean model used to discretize the signal as it propagates. As discussed in Chapter 2, a boolean network can simulate any discrete network. Two problems arise: such boolean networks will be large and no one-to-one correlation between the boolean network's structure and the underlying biochemical network will exist under such a transformation. It is worth noting that we are not aware of any investigations into this question which, certainly, is a potential direction for future work. If the one-to-one correlation is preserved, then such two-state models of signal transduction simplify the underlying biochemistry to the point where it is difficult to model changes in protein concentration more precisely than present or absent. Modeling such gradients of concentration changes and the

effects of those changes may be important to predicting individual protein responses, motivating our effort to devise more fine-grained ways to model and simulate the dynamics of signaling networks. The challenges to using linear-piecewise ODEs to model a signaling network center around the issue of identifying all the ODEs required to model the underlying network as well as scalability issues involved in simulating large systems of ODEs.

In [RMT+08], we extend the synchronized Petri net model and firing policy such that the resulting framework models cellular signaling processes. We call this extension the signaling Petri net (SPN). By coupling this with a novel strategy for Petri net execution and sampling, we obtain a method capable of characterizing some dynamics of signaling networks while using only connectivity information about these networks.

To validate our method, we studied the EGFR network discussed in Chapter 2 and shown in Figure 2.1 in two breast cancer cell lines. We analyzed, both experimentally and computationally, the change in activity-level of several proteins in response to targeted manipulation of TSC2 and mTOR-Raptor. In these experiments, the predictions from our method agreed with experimental results in over 90% of the cases, and in those where they did not agree, our method correctly identified discrepancies that could be traced back to incompleteness in the network connectivity model.

3.1 Methods and Models

Our approach combines elements of the boolean network simulator in [LAA06] with a synchronized Petri net model [DA05]. In [LAA06], Li *et al.* present a non-parametric approach that accurately predicts the speed of signal propagation through a network. However, as their method assumes a binary model of activation—every protein is either active (*true*) or inactive (*false*)—modeling a range of activity-levels requires a significant transformation in the meaning of nodes and the overall connectivity of the network. Petri nets, in contrast, preserve the meaning of edges as interaction and nodes as proteins. However, while able to model concentrations using tokens, existing approaches require parameters describing the kinetic characteristics of the network, which are typically difficult to obtain.

Our method models signal flow as the pattern of token accumulation and dissipation within places (proteins) over time in the Petri net. Transitions in the network represent directed protein interactions; each transition models the effect of a source protein on a target protein. Through transition firings, the source can influence the token-count[1] of the target, modeling the way that signals propagate through protein interactions in cellular signaling networks.

In order to overcome the issue of modeling reaction rates in the network, signaling dynamics are simulated by executing the signaling Petri net (SPN) for a set number of steps (called a *run*) multiple times, each time beginning at the same initial marking.

[1]By *token-count*, we refer to the number of tokens assigned to a protein place at a specific time point.

For each run, the individual signaling rates are simulated via generation of random orders of transition firings (interaction occurrences). When the results of a large enough number of runs are averaged together, we find that the series of token-counts correlate with experimentally measured changes in the activity-levels of individual proteins in the underlying signaling network. In essence, the tokenized activity-levels computed by our method should be taken as abstract quantities whose changes over time correlate to changes that occur in the amounts of active proteins present in the cell. It is worth noting that some of the most widely used experimental techniques for protein quantification—western blots and microarrays—also yield results that are treated as qualitative statements, but not exact measurements, of protein activity-levels within the cell. Thus in some respects, the predictions returned by our SPN-based simulator can be interpreted like the results of a western blot or microarray experiment[2] looking at changes relative to "control".

The key insight behind our approach is the assumption that, while all network parameters determine the actual signal propagation to some extent, the network connectivity is the most significant single determinant. While this is clearly a gross simplification, several researchers have observed that the connectivity of a biological network dictates, to a great extent, the network's dynamics [AC03, LAA06, KB05, KPST04]. Some have conjectured that biological network connectivities have evolved to have a stabilizing effect on the overall network behavior, making the network more

[2]Though it should be emphasized that experimental results provide measurements whereas computational simulations provide only predictions.

resilient to local fluctuations in other network parameters such as reaction rates and protein binding affinities [AC03,KB05]. Here we present the *signaling Petri net* (SPN) model and the signaling Petri net-based simulator whose designs collectively utilize this assumption and couple it with a Petri net tokenization scheme that quantifies the changes in protein activity-levels that occur as signals propagate through the network. In the following sections, we describe the synchronized Petri net, how we extended it to create the signaling Petri net, and a novel strategy for executing the signaling Petri net to simulate signaling network dynamics.

3.1.1 Petri Nets

A Petri net is a graph that consists of two types of nodes, *places* and *transitions* [DA05]. Edges in the graph, called *arcs*, are directed and connect places to transitions or transitions to places. Thus, the Petri net is a bipartite graph. Formally, a Petri net is a 4-tuple $Q = \langle P, T, I, O \rangle$ where

- $P = \{p_1, p_2, ..., p_m\}$ is the set of places,
- $T = \{t_1, t_2, ..., t_n\}$ is the set of transitions,
- $I = \{i_1, i_2, ..., i_k\}$ is the set of input arcs where for all $(u, v) \in I$, $u \in P$ and $v \in T$, and
- $O = \{o_1, o_2, ..., o_l\}$ is the set of output arcs where for all $(u, v) \in O$, $u \in T$ and $v \in P$.

In order to simulate a dynamic process, a number of tokens is assigned to each place in order to indicate the presence of some quantitative property. This assignment of tokens to places encodes the state of the system and is called a marking, denoted $\mathbf{m}$, which is a vector of size $|P|$, where $\mathbf{m}[i]$ indicates the number of tokens at place i. A *marked Petri net*, $R = \langle Q, \mathbf{m}_0 \rangle$, is a Petri net with a marking $\mathbf{m}_0$, called the initial marking. For the remainder of this chapter, the term *Petri net* (PN) refers to a marked Petri net.

Changes in the state of the system are simulated by *executing* the Petri net—evaluating the effect of transitions on the marking of the network. These changes in marking are induced by sequential *firing* of one or more transitions. When a transition fires, it removes a token from each place connected to it by input arcs and adds a token to each place connected to it by output arcs[3]. A transition can only fire when it is *enabled*, meaning that each of its input places has at least one token in the current marking. If a transition t, when fired on a marking $\mathbf{m}_1$, produces marking $\mathbf{m}_2$, then we write $\mathbf{m}_1 | t \rangle \mathbf{m}_2$.

This notation can be extended to represent the effect of firing a series of transitions. A *firing sequence*, $\sigma = (t_1, t_2, ..., t_j)$ is a sequence of transitions. The sequence's cumulative effect on the system's state is denoted $\mathbf{m}_0 | \sigma \rangle \mathbf{m}_f$ where $\mathbf{m}_0$ is the initial marking and $\mathbf{m}_f$ is the marking produced by the firing of the sequence of transitions in the order specified in σ. In this chapter , we write $\mathbf{m}_g^\sigma$ to indicate the marking

[3]The number of tokens removed from inputs and added to outputs can be specified by weighting the input arcs. However, as our extension does not use this weighting property, we do not consider this very common PN formulation here.

produced from $\mathbf{m}_0$ by the first g transitions in σ. Therefore, in the above example, $\mathbf{m}_0^\sigma = \mathbf{m}_0$ and $\mathbf{m}_{|\sigma|}^\sigma = \mathbf{m}_f$.

For a more complete introduction to types of Petri nets and their properties, we refer the reader to [DA05].

Synchronized Petri Nets

Synchronized Petri nets model systems in which the firing of a transition is triggered by a specific event that occurs in the environment. The marked Petri net is extended to include a set of these events and a mapping function that assigns an event to each transition. When transition t's assigned event occurs, transition t is fired. Formally, a *synchronized Petri net* is a 3-tuple $\langle R, E, Sync \rangle$, where:

- $R = \rangle P, T, I, O \langle$ is a marked Petri net,
- $E = \{E_1, E_2, ..., E_s\}$ is a set of events, and
- Sync : $T \rightarrow E \cup \{\mathbf{e}\}$ maps each transition in the Petri net to an event. Event $\mathbf{e}$ is the *always occurring event.* Any transition associated with $\mathbf{e}$ is always immediately fired upon becoming enabled.

When executing a synchronized Petri net, transition an enabled t is fired only when its associated event Sync(t) occurs. The order in which events are generated depends upon the environment which generates them. Just as in the marked Petri net, when a transition fires, it removes one token from each place connected by input arcs and gives one token to each place connected by output arcs.

PROCEDURE SIMULATE(S, B, r)

1. Set the initial marking of S
2. For $i = 1$ to r
   (a) Generate a random sequence with length $B \cdot |P|$ of the events in E
   (b) Simulate the network by executing the transitions associated with the events in the generated sequence
   (c) Record the number of tokens at each node, for each time block
3. For each node, compute the number of tokens at each time unit t, averaged over r

Fig. 3.1: A high-level outline of the procedure for simulating a signaling network. The input to the procedure is a signaling Petri net, S, the number of time units to simulate the network for, B, and the number of runs for which to repeat the simulation, r. The random generation of event ordering is employed to simulate the stochasticity in reaction rates and the differing times of signal arrivals.

As will be discussed in the next sections, we extend the synchronized Petri net paradigm to model the dynamics of a signaling network. To our knowledge, ours is the first use of the synchronized Petri net to model biochemical systems. In principle, it is well suited to signaling networks since places represent proteins, tokens represent concentrations, and transitions represent directed protein interactions. A model of signaling event occurrence can be used to generate events and fire transitions, providing a way of simulating the signaling network's behavior. These and other design details will be discussed in the next section.

3.1.2 The Signaling Petri Net-based Simulator

A high-level sketch of our simulator is given is Figure 3.1. Details and rationale for specific design decisions will be discussed in subsequent sections.

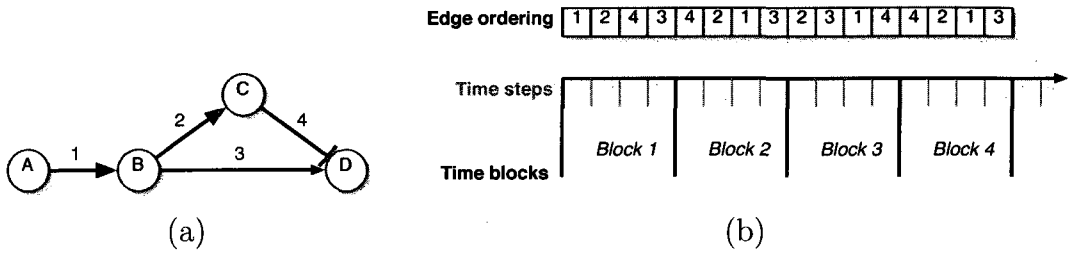


Fig. 3.2: (a) By changing the speed of signaling edge 3, the value of D at the end of a single simulation step can be reversed. If edge 3 is slower than the cascade $B \rightarrow C \dashv D$, then D will be active. If edge 3 is faster than the cascade, then D will be inactive. (b) An example of how the simulator might evaluate the individual edges during a run. In each time block, every edge is evaluated once. Each edge evaluation corresponds to one time step. Note that the order of the edge evaluation is shuffled during each time block in order to sample the space of possible relative signaling rates.

During the simulation, the input signaling Petri net is executed multiple times on a firing sequence constructed by the signaling event generator. The signaling event generator imposes an ordering on transition firing such that it creates a two-time scale simulation. The smaller time scale is discretized as the firing of a single transition. This unit is referred to as the *firing* time scale. Firing steps are nested within a larger time scale, called time *blocks*, in which each transition is fired exactly once. Thus, there are $|T|$ firings per block. Since the simulation is run for the specified number of time blocks, B, there are $B \cdot |T|$ firing steps in the simulation.

The time structure for an example simulation is illustrated in Figure 3.2. This dual-time approach is necessitated by the rate parameter sampling strategy we employ. Since the rate parameters are not known, our method executes many simulation runs (Step 2 in Figure 3.1) in order to sample the space of possible rate parameters. The markings returned by these runs are then averaged (Step 3 in Figure 3.1). The

only requirement placed on the different rate parameter values is that all events occur within the same larger time frame—the time block. Therefore, within every time block each edge is evaluated once, yet the order of evaluation is not necessarily the same across blocks.

This idea of evaluating random event orderings within a two-time scale system has appeared before in the domain of transcriptional networks [CAS05]. In that study, Chaves *et al.* employed a two-time scale formulation of network updates similar in concept to the one we describe here. In their work, they assumed a boolean model of regulation and characterized the effect of different relative rates of transcription within the same network on the final steady state reached. In contrast, our method is designed to operate on tokenized models of signaling networks with the ultimate intent of predicting the activity-level changes of proteins in the underlying signaling network over time.

In the next sections, we discuss in greater detail the core design decisions underlying our method: the signaling Petri net, transition firing, signaling network event generator, constructing the initial marking for the model, and sampling signaling rates. We then discuss how our strategy can be used to predict the outcome of perturbation experiments.

The Signaling Petri Net

The goal of our method is to predict the signal flow through a cell-specific network under specific experimental conditions. As a result, the signaling Petri net model must

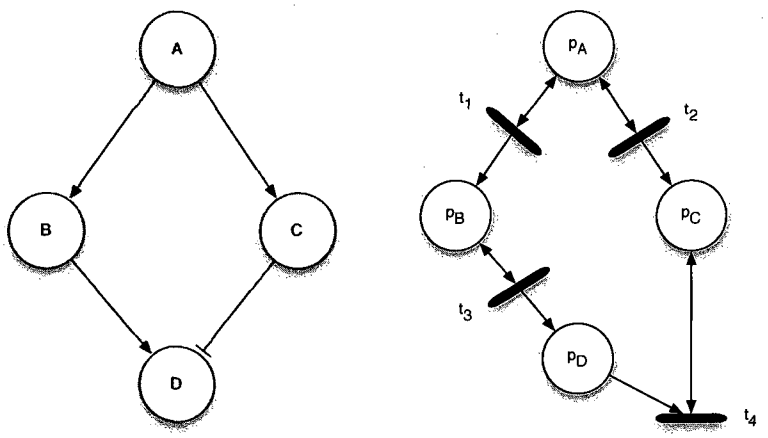


Fig. 3.3: An example signaling network (a) and its corresponding signaling Petri net (b). Each signaling protein in the network, A, B, C, and D, are designated as places p_A, p_B, p_C, and p_D. Each signaling interaction becomes a transition along with its input and output arcs. Note that the connectivity for an activating edge differs from that of an inhibitory edge.

characterize the connectivity of the signaling network, the connectivity-level network properties that are unique to the cell type and experimental conditions under which the network is being studied, and the signaling processes of activation and inhibition.

The signaling Petri net is a synchronized Petri net with:

1. a specific way of modeling activating and inhibiting interactions using places, transitions, and arcs,

2. a one-to-one correspondence between events and transitions such that every transition is associated with a unique event,

3. modified rules regarding how many tokens are moved in response to a transition firing, and

4. a signaling network event generator.

Places correspond to the activated forms of signaling proteins. The number of tokens assigned to place p in marking $\mathbf{m}_s$, $m_s(p)$, abstractly represents the amount of active protein p present in that network state. Signaling interactions are modeled using transitions and their connected input and output arcs. Each transition, t, is associated with a unique signaling event, e, such that when e occurs, transition t fires. Figure 3.3 shows the equivalent signaling Petri net for a signaling network. Note that some of the edges in the Figure 3.3 are bi-directional. These arcs are called *read arcs* and indicate that the place at the endpoint is both an input and an output. When its associated transition fires, a read arc removes and then returns the same number of tokens to the place, having the overall effect of "reading" tokens from the place.

Formally, a signaling Petri net is the 3-tuple $S = \langle R, E, \text{Sync} \rangle$, where:

- $R = \langle P, T, I, O \rangle$ is a marked Petri net,
- E is a set of signaling events such that $|E| = |T|$ and there is no *always occurring event*, and
- Sync : $T \rightarrow E$ is a one-to-one mapping which assigns each transition a unique signaling event.

The initial marking of a signaling Petri net, $\mathbf{m}_0$, represents the state of rest from which the network is starting and being simulated. Proteins whose concentrations are known to be high are given a large number of tokens, and those whose concentrations are known to be low are assigned few or zero tokens. Attention to the initial

marking is central to modeling cell-specific networks. In many cell lines, specific proteins are known to contain mutations that render them perpetually active or inactive [NCF+06]. Furthermore, experimental studies frequently involve the targeted manipulation of various proteins within the network. Both of these phenomena induce state changes in certain proteins at various time points that must be modeled. The way in which these are modeled will be discussed when the simulator design is explained.

Transition Firing

When a signaling interaction $A \rightarrow B$ (A *activates* B) or $A \dashv B$ (A *inhibits* B) occurs, it has the effect of changing the state of the system by modifying the activity-level of A and/or B. Thus, in the SPN used to model this network, the associated transition, t, will fire at time τ and produce marking $\mathbf{m}_{\tau+1}$ from $\mathbf{m}_{\tau}$. The way in which $\mathbf{m}_{\tau+1}$ is computed from $\mathbf{m}_{\tau}$ depends on the set of input and output arcs attached to the transition t as well as the number of tokens moved by the transition.

The combination of input and output arcs connected to a transition is determined exclusively by the type of interaction and the transition firing model. However, different topologies, combinations of input and output arcs, are needed to model the different biochemical processes that mediate protein-protein interactions in a signaling network. Here we examine four of the most common biochemical processes, identify the corresponding topological structures, and ultimately devise a modeling policy best suited for non-parametric simulation of signal flow.

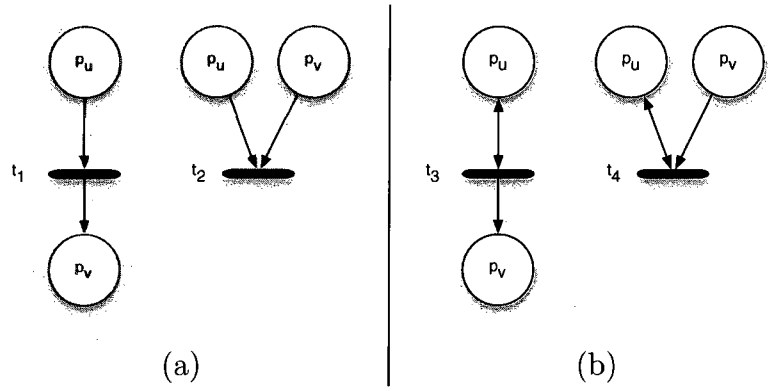


Fig. 3.4: The topological structures for differing signaling processes. (a) The token consumption structures for complexing and recruitment. Transition t_1 encodes activation of v by the binding or consumption of u. Transition t_2 encodes deactivation of v by the binding or consumption of u. In both cases, the number of tokens of p_u decreases immediately after transitions t_1 and t_2 fire. (b) The token conserving structures for PTM and GTP/ATP binding. Transition t_3 encodes enzymatic activation of v by u. Transition t_4 encodes enzymatic inhibition of v by u. In both cases, the number of tokens of p_u remains unchanged immediately after transitions t_3 and t_4 fire.

In *post-translational modification* (PTM), a protein mediates the addition or removal of a phospho group at a specific phosphorylation site on another protein. In *GTP/ATP* binding, a protein triggers the exchange of GDP (ADP) from GTP (ATP) on another protein. In a *recruitment* process, a protein mediates the relocalization of another protein to a different part of the cell. Finally, in a *complexing* process, a protein binds to another protein to create a complex, which can then participate in other reactions. In the first two processes, the mediating protein usually acts as an enzyme that participates in the reaction but is not consumed by the reaction. In the latter two processes, the participating protein often becomes unavailable to other reactions, transiently while the protein recruitment is taking place and for longer durations when complexing occurs. To model these two cases, we identified the two

different token-passing policies implemented by the different topological structures depicted in Figure 3.4:

- *Token consumption.* In this policy, $u \rightarrow v$ consumes tokens in u in order to generate new tokens for v. In order to model this, p_u is connected to transition t_1 through an arc and p_v is connected to t_1 through an output arc. When t_1 fires, some number of tokens in p_u are moved into p_v. Similarly, $u \dashv v$ consumes tokens in u in order to consume tokens in v. This is modeled by connecting p_u to t_2 with an input arc and p_v to t_2 with an input arc. When t_2 fires, some number of tokens are removed from both p_u and p_v. This policy models a recruitment or complexing event in which u binds to another molecule, thereby creating a molecule of type v. *A molecule of type u has been consumed in order to generate or deactivate a molecule of type v.*

- *Token conservation.* In this policy, $u \rightarrow v$ generates new tokens for v while conserving those in u. In order to model this, p_u is connected to transition t_3 through a read arc. Node p_v is connected to t_3 through an output arc. When t_3 fires, some number of tokens in p_u is read (but not removed) and copied into p_v. Similarly, $u \dashv v$ consumes tokens in v while conserving those in u. This is modeled by connecting p_u to t_4 with a read arc (described earlier in this section) and p_v to t_4 with an input arc. When t_4 fires, some number of tokens in p_u are read and removed from p_v. Enzymes will often behave in this way: inducing a change in a molecule (v) without themselves undergoing any change. *A molecule*

of type u has induced a change in a different molecule of type v without itself changing state.

Ideally, for each interaction in the network, the associated transition could be embedded in the topology corresponding to the interaction's underlying biochemical mechanism. However, connectivity-level knowledge of the network does not provide this information for each interaction. In the absence of these details, we use one token-passing policy for all interactions in the network. We implemented and tested both the consuming and conserving policies and found that token conservation provides significantly more accurate results for signaling networks when compared to experimentally-derived data. This is not surprising, as post-translational modification and GTP/ATP binding events are responsible for many state changes in signaling networks [IB89, Hun00, JLI00, Bra95]. It is worth noting that our approach does not restrict the network structure to token conserving topologies. Thus, it is possible to use the token consumption topologies where such processes are known to occur. However, as our focus in this chapter is on designing a purely non-parametric simulation method, we consider the use of information regarding the biological mechanism of signaling as a potential way to further improve the accuracy of our method's predictions and identify this as a direction for future work.

The transition topologies, as described above, do not designate how the number of tokens added to or removed from p_v is determined. However, we know that in biochemical signaling networks concentration has an effect on the strength of a signaling

event [GBP06, EI04b, EI04a]. Specifically, the higher u's concentration, the stronger its effect on v—the more tokens that p_u has, the more tokens of p_v should be affected (generated or consumed).

However, because of the stochastic nature of the underlying biochemistry, it would be inaccurate to assume that *all* active u molecules will always participate in an interaction with v. In order to accomodate this observation, when transition t fires, we randomly select the number of p_u's tokens to be involved in each signaling event[4]. Note that, according to our choice of topology, p_u can always be identified as the node connected to the transition by a read arc. In this chapter, we assume a uniform distribution for selecting the number of tokens involved in a given signaling event, but acknowledge that other distributions may be more appropriate under certain circumstances and identify this as a topic deserving further consideration.

Let $m_s(x)$ denote the number of tokens in node x at time s. For an interaction (u, v), under the token conservation policy detailed above, u's token-count remains unchanged after the firing of t, whereas v's token-count is updated based on the following formula:

$$m_s(v) = \begin{cases} m_{s-1}(v) + \text{random}(0, m_{s-1}(u)) & \text{if} \quad u \rightarrow v \\ \max\{0, m_{s-1}(v) - \text{random}(0, m_{s-1}(u))\} & \text{if} \quad u \dashv v \end{cases}, \tag{3.1}$$

where $\text{random}(p, q)$ is a random integer m drawn from a uniform distribution over the range $[p, q]$.

If we employ the policy of token passing with consumption, then after $m_s(v)$ has

[4]By *signaling event*, we refer to a single evaluation of an interaction in the model network

been computed based on Formula (3.1), $m_s(u)$ is updated as:

$$m_s(u) = m_{s-1}(u) - \min\{m_{s-1}(u), |m_s(v) - m_{s-1}(v)|\}. \tag{3.2}$$

Signaling Network Event Generator

The SPN topology and transition token-number selection policy alone do not specify the speed with which individual signaling interactions occur. However, such rates must be accounted for when simulating a signaling network. ODEs characteristically model such details as reaction rate constants; parameterized Petri nets specify these in a variety of ways including transition firing rates and firing probabilities [HR04, Cha07]. In synchronized Petri nets, the environment controls the generation of events. Thus, the signaling network event generator is responsible for controling the timing and ordering of signaling events. However, as our objective is a non-parametric simulation method, our approach must either estimate these parameters or operate without explicit knowledge of them.

While there has been some work in the area of predicting reaction rates, all results of which we are aware require knowledge about the mechanism of signaling (e.g., [BFGH06]). As a result, without enriching the SPN model, it is doubtful that rate parameters can be accurately estimated.

For this reason, the signaling network event generator operates without explicit knowledge of the rate parameters. To compensate for this "missing" knowledge, we make use of an observation of signaling networks discussed earlier: a network's connectivity determines its dynamics. Several studies have found that the connectivity

of biochemical networks desensitizes them to small fluctuations in the kinetic biochemical parameters [AC03, KB05, KPST04]. This robustness to parameter values can be understood both within the context of evolution and survival of the organism in an uncertain environment. Evolution is a stochastic process that tweaks signaling network parameters across generations—thus robustness to exact parameter values is a highly desirable property as it ensures that an offspring remains viable despite fluctuations in the exact tuning of its cellular machinery. If this property holds, then small fluctuations in the rate parameters should have a marginal effect on the overall propagation of signal through the network. Parameter value robustness also improves the fitness of an organism in a variable environment: temperature changes, shifts in humidity, and other environmental factors can also induce small changes in the kinetic parameters of a biochemical network. We can consider these small effects to be noise obscuring the underlying dynamics of the network connectivity. By taking many samples of the network dynamics under a variety of reaction rate assignments and then averaging these dynamics, we simultaneously reduce the noise introduced by any one rate assignment and strengthen the underlying dynamic characteristics of the network's connectivity.

However, since reaction rate constants can vary by several orders of magnitude—from 10^{-10} to 10^{3}, the task of correctly selecting parameters *close* to the true parameters is non-trivial. In fact, without having some estimate of the actual rate parameters, it is unclear as to how to measure closeness at all. Clearly, these are

PROCEDURE GENERATESIGNALINGEVENTS(E, n)

1. $k = |E|$
2. σ an an empty array of size $(k \times n)$
3. $i = 1$
4. for $b = 1$ to n
   (a) $E' = E$
   (b) while $E' \neq \emptyset$
      i. e = a random event from E'
      ii. $\sigma[i] = e$
      iii. $E' = E' - \{e\}$
      iv. $i = i + 1$
5. Return σ

Fig. 3.5: The algorithm that implements the signaling network event generator. This routine generates the time block/firing structure. Given a set of events, E, and the number of blocks for which the SPN will be executed, n, GENERATESIGNALINGEVENTS generates n blocks of events, each consisting of $|E|$ events ordered randomly. In each block, every event in E occurs exactly once.

among the issues that make parameter estimation so difficult for ODE and parameterized Petri net approaches. Since our comparisons will be relative and not absolute, we take a relative approach to modeling rate parameters. The space of possible rate values is *the space of possible signaling event orderings.*

This idea is illustrated in Figure 3.2(a). Protein A affects the activity of protein D through two separate pathways. Assuming that A is active to begin with, the relative speed of these two pathways determines the final activity of D. If the pathway through C is faster than the pathway $B \rightarrow D$, then D will be active. However, if the pathway speeds are reversed, then D will remain inactive. The overall outcome of this

network can be represented without any use of numeric reaction rates by representing the reaction rates as an ordering over all the edges in the network. We can extend this idea to the SPN by observing that there exists a unique event for each signaling edge in the signaling network.

This sampling strategy is the motivation for the dual-time framework depicted in Figure 3.2(b) and implemented by the signaling network event generator shown in Figure 3.5. *Time blocks* are the larger time intervals during which every signaling event occurs exactly once. Since every transition in the SPN is associated with a unique event, each transition will fire exactly once in each time block. *Transition firings* are the smaller time units that impose a strict sequential order on the occurrence of signaling events. While this strict sequentiality of firing models relative reaction rates, it also discretizes the effect of signaling events. Though this is consistent with the definition of transition firing in discrete time Petri nets (only one transition is evaluated at a given point in time) [DA05], in biological signaling networks there is no such serial evaluation constraint. However, our validation with experimental data suggests that this discretization approximation does not affect the overall validity of the simulation results.

Defining the Initial State

As mentioned previously, the initial state of the SPN is the initial marking, $\mathbf{m}_0$. As the SPN provides no explicit information on how this marking should be built, we propose three ways to construct the initial state: zero, basal, or experimentally

derived. In a zero initial state, the simulator initializes all proteins to have zero tokens. The basal initial state is a random distribution of activation levels intended to model the cell when no impluses due directly to external stimuli are propagating through the signaling network. Though a basal network is considered at rest, in general it will not have a zero marking since signal flows are known to occur even in unstimulated signaling networks through autocrine and paracrine secretions by the cells. The experimentally derived initial state is based on knowledge about the activity levels of various proteins just prior to the addition of the external stimuli.

When accurate experimental data is available such as results from microarrays or western blots, the experimentally derived initial state may be the most accurate. A challenge in using experimental data, however, is determining how best to assign numbers of tokens based on the experimentally observed activity levels.

In the absence of reliable experimental data, the basal initial state seems more accurate than the zero initial state. However, it presents the challenge of properly selecting the basal activity-levels to assign to each protein in the model network. In [LAA06], a basal initial state was constructed by activating a small number of randomly selected proteins in the signaling network. However, the work in [LAA06] was done using a boolean model. Translating this approach into a tokenized model creates the additional complexity of determining how many tokens each basally active protein should receive. The correct values are likely to depend on the specific signaling network and experimental conditions.

We performed preliminary tests to compare the effect of using different basal versus zero markings on the outcome of the simulator. We found that the basal and zero states produced indistinguishable predictions so long as less than 30% of the proteins were activated and a small number of tokens (< 5) were used when constructing the basal marking. This is not as surprising as it may seem at first. Inhibitory edges will quickly consume a small number of tokens scattered throughout the network, effectively returning much of the network to the zero state before a stimulation event can propagate through.

Furthermore, while validating our method, we also compared the predictions produced by SPNs based on a zero initial state and experimentally derived initial state. These, too, did not produce noticeably different final results for similar reasons as discussed above. Details of these comparisons will be discussed further in the Results and Discussion sections.

However, since all three initial state construction strategies yield qualitatively identical predictions, using zero initial states has the advantage of invoking the fewest unnecessary assumptions about the network (as in the case of the basal initial state) and requiring the least experimental data (as in the case of the experimentally derived state). Nonetheless, in our implementation of the tool, we allow for using any one of these three initial state construction strategies.

Modeling Cell-Specific Signaling Networks

Whereas consensus signaling networks typically represent the connectivity in normal cells and are typically collected from observations in different experiments, many experiments are conducted on abnormal cells in which oncogenic mutations, gene knockous, and pharmacological inhibitors have altered the behavior of various signaling nodes in the network. In an SPN, these alterations to the signaling network can be modeled by adding/removing transitions (and associated input/output arcs) and explicitly setting the token count for various proteins in the initial state.

The two network alterations which are commonly induced by oncogenic mutations, gene knockouts, or pharmacological inhibitors are constitutively high or low protein activity-levels, meaning that a protein is either unable to be inhibited or unable to be activated. The simulator allows for proteins to be specified as either fixed *High* or *Low*. Here we explain how these are modeled by changes to the SPN.

If protein u is fixed high, then this protein cannot be inhibited. Thus, all transitions that remove tokens from p_u are removed from the SPN. The fact that u is high, however, also suggests that it maintains a higher activity level in general. Therefore, in the initial state, $m_0(p_u) = H$, where H is a non-zero number of tokens. Since all inhibiting transitions have been removed from the SPN, throughout any execution, place p_u will always have at least H tokens.

In experiments, we have observed that the choice of the value of H does not change the relative outcome of the simulations. While H will affect the actual number of

tokens present in a given place as well as the number of time blocks required to observe certain activity-level changes, the relative changes in activity-level (number of tokens) among different proteins (places) do not change. As a result, one is free to select any reasonable value of H (for our experiments, we used $H = 10$) as long as this H is held constant across all simulations whose results will be compared.

If protein u is fixed low, then this protein cannot be activated. Thus, all transitions that add tokens to p_u are removed from the SPN. The fact that u is low, however, also suggests that it maintains a constantly low activity level in general. Therefore, in the initial state, $m_0(p_u) = L$, where L is a small number of tokens (in our simulations we use $L = 0$). Since p_u is only inhibited, we observed that all constitutively low proteins quickly had their marking reduced to zero.

Unlike the value of H, extra caution must be taken when selecting values for representing L. A value of L that is too large can destabilize the early propagation of signal through the network. In our experiments, we obtained best results for values of L very close to or equal to zero ($L \leq 2$). Beyond this, the final results obtained depended on other values in the network, the strength of the signal, and the duration of the simulation.

Simulating a Signaling Network

Figure 3.6 provides a more detailed version of the simulation algorithm outlined in Figure 3.1. Steps 1 and 2 of the SIMULATE procedure construct the modified initial marking and network topology to incorporate perpetually high proteins, H,

PROCEDURE SIMULATE(S, H, L, B, r)

1. For each $p \in H$
   - $m_0(p) = 10$;
   - $I = I - \{(p,t) : t \in T \text{ and } (t,p) \notin O\}$
2. For each $p \in L$
   - $m_0(p) = 0$;
   - $I = I - \{(t,p) : t \in T\}$;
3. for $i = 1$ to r
   - σ^e = GenerateSignalingEvents(E, B);
   - Execute $\mathbf{m}_0^i|\sigma\rangle\mathbf{m}_{B|T|}^i$;
4. For each $p \in P$ and $0 \leq b \leq B$
   - $\overline{m}_b(p) = \frac{1}{r}\sum_{i=1}^{r} m_{b|T|}^i(p)$;
5. Return $(\overline{\mathbf{m}}_1, \overline{\mathbf{m}}_2, \ldots, \overline{\mathbf{m}}_B)$

Fig. 3.6: SIMULATE predicts the signal flow through the SPN S. The simulation is run for B time blocks; the results of r runs are averaged to produce the final result. Most of the work is done by the signaling Petri net execution procedure detailed in the preceding sections. This execution actually performs an individual run. This procedure takes the initial marking, $\mathbf{m}_0$ and applies the sequence of transitions triggered by the event sequence, σ^e. This ordering, generated by the algorithm in Figure 3.5, has the dual time structure in which each block of edges contains every event in E exactly once. Each firing evaluates the effect of one transition. The markings at the end of each time block are extracted in Step 5.

and perpetually low proteins, L. In this chapter, proteins that are assigned high activity-levels receive an initial token count of 10 in order to model a higher-than-average initial activity-level. As discussed earlier, using other values of H scale the activity-levels of all the proteins in the network, but will not qualitatively change their relative activities.

The loop in Step 3 runs r individual simulation runs. Each run receives a different

event ordering, σ^e, thereby implementing the interaction rate sampling strategy. The time block/step structure is contained within the ordering σ^e (see Figure 3.5(c)). As a result, the SPN execution step simulates the events by firing their associated transitions. Only markings at the end of time blocks are sampled.

After SIMULATE finishes collecting the time block markings from all the runs, Step 4 computes the average markings for each time block and Step 5 returns these averages.

Simulating a Perturbation Experiment

We tested the accuracy and speed of our method by simulating the effect of two different targeted manipulations to the EGFR network in Figure 2.1. We compared these predictions to experimental results produced by performing the actual manipulations on two separate cancer cell lines.

The perturbations we considered in this study altered the constitutive activity-level of various proteins in the network (as opposed to affecting specific signaling interactions). Therefore, we modeled the perturbations as changes in the high and low proteins—H^c and L^c for the control[5] network and H^p and L^p for the perturbed network.

A variant of the SIMULATE method was required to quantify how a perturbation changed the protein token-counts for each time block. Figure 3.7 shows the algorithm we used. In the procedure DIFFERENTIALSIMULATE, the input S provides

[5]By *control*, we refer to the normal cell line.

PROCEDURE DIFFERENTIALSIMULATE(S, H^c, L^c, H^p, L^p, B, r)

1. $S^c = S$, $S^p = S$;
2. For each $p \in H^c$
   (a) $m_0^c(p) = 10$ and $I^c = I^c - \{(p,t) : t \in T^c \text{ and } (t,p) \notin O^c\}$
3. For each $p \in L^c$
   (a) $m_0^c(p) = 0$ and $I^c = I^c - \{(t,p) : t \in T^c\}$;
4. For each $v \in H^p$
   (a) $m_0^p(v) = 10$ and $I^p = I^p - \{(v,t) : t \in T^p \text{ and } (t,v) \notin O^p\}$
5. For each $v \in L^p$
   (a) $m_0^p(v) = 0$ and $I^p = I^p - \{(t,v) : t \in T^p\}$;
6. for $i = 1$ to r
   (a) $\sigma^e = \text{GenerateSignalingEvents}(E,T)$;
   (b) Execute $\mathbf{m}_0^c|\sigma\rangle\mathbf{m}_{B|T|}^c$;
   (c) Execute $\mathbf{m}_0^p|\sigma\rangle\mathbf{m}_{B|T|}^p$;
   (d) For $j = 0$ to B
      i. $\mathbf{d}_j^i = \mathbf{m}_{j|T|}^p - \mathbf{m}_{j|T|}^c$
7. For each $p \in P$ and $0 \leq b \leq B$
   (a) $\Delta_b(p) = \frac{1}{r}\sum_{i=1}^{r} d_b^i(p)$;
8. Return $(\Delta_1, \Delta_2, \ldots, \Delta_B)$;

Fig. 3.7: The algorithm for predicting the effect on signal propagation of a targeted manipulation on signaling network with connectivity G. The 'c' and 'p' superscripts are used to denote parameters in the *control* and *perturbed* versions, respectively, of the SPN.

the consensus SPN. Inputs H^c and L^c specify the control high and low proteins, the inputs H^p and L^p specify the perturbed high and low proteins. After Steps 1—5 construct two separate SPNs for the control and perturbed conditions, the loop in Step 6 performs r independent simulations over the control and perturbed models. Step 6d computes the difference between the markings at the end of each time block

in the perturbed and control networks. The marking difference $\mathbf{d}_j^i = \mathbf{m}_j^p - \mathbf{m}_j^c$ yields the marking $\mathbf{d}_j^i$ where $d_j^i(v) = m_j^p(v) - m_j^c(v)$ for each $v \in P$. Following the loop, the marking differences are averaged to obtain the time series $(\Delta_1, \Delta_2, \ldots, \Delta_B)$ where $\Delta_b(v)$ is the average change in the token-count for protein v at the end of time block b.

For values of $|\Delta_b(v)| > 0$ for a given molecule v, we can conclude that the perturbation caused a change in the activity-level of v at the end of time block b only if the difference observed is statistically significant. We use a t-test to determine whether this change is statistically significant for protein v at the end of time block b. Computing the t-test for two distributions (control and perturbation) requires knowledge of the mean ($\mu_{c,b}$ and $\mu_{p,b}$) as well as the variance (σ_c^2 and σ_p^2) for both distributions. In order to obtain these parameters for the control network, a large number, X, of independent simulations is run. Simulation i provides a single series of markings, $(\overline{\mathbf{m}}_1^i, \overline{\mathbf{m}}_2^i, \ldots, \overline{\mathbf{m}}_B^i)$. The mean is then computed:

$$\mu_{c,b,v} = \frac{\sum_{i=1}^{X} \overline{m}_b^i(v)}{X}.$$

The variance is computed similarly:

$$\sigma_{c,b,v}^2 = \frac{\sum_{i=1}^{X} (\overline{m}_b^i(v) - \mu_{c,b,v})^2}{X - 1}.$$

The parameters $\mu_{p,b,v}$ and $\sigma_{p,b,v}^2$ for the perturbed network are computed as described above by substituting the perturbed network for the control network. Using these parameters, the t-value for molecule v at the end of time block b can be computed

from the formula

$$\text{t} - \text{value} = \frac{\mu_{c,b,v} - \mu_{p,b,v}}{\sqrt{\frac{\sigma^2_{c,b,v}}{X} + \frac{\sigma^2_{p,b,v}}{X}}}.$$

The statistical significance of the difference can then be obtained by comparing the t-value to the desired critical value.

Note that the DIFFERENTIALSIMULATE procedure and the associated significance test can predict the effect not only of perturbations, but also of any two different experimental (or cellular) conditions imposed on the same signaling network. As a result, in addition to perturbation experiments, our method can also be used to study the effects of other phenomena that induce changes in the propagation of signal through a signaling network.

3.2 Materials

3.2.1 Cell-specific Signaling Network Models

Figure 2.1 shows the signaling network we analyzed. We obtained the core connectivity from a published literature survey on the EGFR network [ICG05]. We added to this several other well-established interactions taken from literature [KCCR04, MCEB+05, MLLA05, KM05, AHL+06, LSX+07, IOZ+06, ORS+06]. The response of this network to various perturbations was measured and simulated in two separate breast cancer cell lines: MDA231 and BT549. The core signaling Petri net used, S^{EGFR}, is captured by the following signaling proteins and interactions:

- Places (the set P): v_{EGFR}, v_{SRC}, v_{Rac}, v_{MEKK4}, v_{MEK4}, v_{JNK}, v_{MEKK6}, v_{MEK6},

v_{STAT}, v_{Grb2}, v_{Shc}, v_{SOS}, v_{RB}, v_{ELK}, v_{BAD}, v_{NFKB}, v_{RAS}, v_{GAB1}, v_{PIP3}, v_{PI3K}, v_{PDK1}, v_{PTEN}, v_{c-Raf}, v_{AKT}, v_{LKB1}, v_{MEK}, $v_{GSK3\beta}$, v_{AMPK}, v_{TSC2}, $v_{MAPK1,2}$, v_{RSK}, v_{Rheb}, $v_{mTOR-Raptor}$, v_{4EBP1}, and v_{p70S6K}.

- Protein interaction network structures (the combination of arcs and transitions):

$v_{EGFR} \rightarrow v_{Grb2}$,

$v_{Grb2} \rightarrow v_{Shc}$, $v_{Shc} \rightarrow v_{SOS}$, $v_{SOS} \rightarrow v_{Ras}$, $v_{Grb2} \rightarrow v_{GAB1}$, $v_{GAB1} \rightarrow v_{PI3K}$, $v_{EGFR} \rightarrow v_{SRC}$, $v_{SRC} \rightarrow v_{STAT}$, $v_{PI3K} \rightarrow v_{PIP3}$, $v_{PIP3} \rightarrow v_{PDK1}$, $v_{Ras} \rightarrow v_{c-Raf}$, $v_{PDK1} \rightarrow v_{AKT}$, $v_{Ras} \rightarrow v_{Rac}$, $v_{Rac} \rightarrow v_{MEKK4}$, $v_{MEKK4} \rightarrow v_{MEK4}$, $v_{MEK4} \rightarrow v_{JNK}$, $v_{JNK} \rightarrow v_{STAT}$, $v_{Rac} \rightarrow v_{MEKK6}$, $v_{MEKK6} \rightarrow v_{MEK6}$, $v_{MEK6} \rightarrow v_{STAT}$, $v_{PDK1} \rightarrow P_{p70S6K}$, $v_{PTEN} \dashv v_{AKT}$, $v_{AKT} \dashv v_{c-Raf}$, $v_{AKT} \dashv v_{GSK3\beta}$, $v_{AKT} \dashv v_{TSC2}$, $v_{AKT} \dashv v_{AMPK}$, $v_{AKT} \dashv v_{BAD}$, $v_{AKT} \rightarrow v_{NFKB}$, $v_{AKT} \rightarrow v_{p70S6K}$, $v_{LKB1} \rightarrow v_{AMPK}$, $v_{MEK} \rightarrow v_{MAPK1,2}$, $v_{MAPK1,2} \rightarrow v_{RB}$, $v_{MAPK1,2} \rightarrow v_{ELK}$, $v_{MAPK1,2} \rightarrow v_{STAT}$, $v_{GSK3\beta} \rightarrow v_{TSC2}$, $v_{AMPK} \rightarrow v_{TSC2}$, $v_{MAPK1,2} \dashv v_{EGFR}$, $v_{MAPK1,2} \dashv v_{TSC2}$, $v_{MAPK1,2} \rightarrow v_{p70S6K}$, $v_{MAPK1,2} \rightarrow v_{RSK}$, $v_{RSK} \dashv v_{TSC2}$, $v_{TSC2} \dashv v_{Rheb}$, $v_{Rheb} \rightarrow v_{mTOR-Raptor}$, $v_{AKT} \rightarrow v_{mTOR-Raptor}$, $v_{mTOR-Raptor} \rightarrow v_{4EBP1}$, $v_{mTOR-Raptor} \rightarrow v_{p70S6K}$, $v_{p70S6K} \dashv v_{EGFR}$,

$v_{SRC} \dashv v_{SRC}$, $v_{Rac} \dashv v_{Rac}$, $v_{MEKK4} \dashv v_{MEKK4}$, $v_{MEK4} \dashv v_{MEK4}$, $v_{JNK} \dashv v_{JNK}$, $v_{MEKK6} \dashv v_{MEKK6}$, $v_{MEK6} \dashv v_{MEK6}$, $v_{STAT} \dashv v_{STAT}$, $v_{Grb2} \dashv v_{Grb2}$, $v_{Shc} \dashv v_{Shc}$, $v_{SOS} \dashv v_{SOS}$, $v_{Ras} \dashv v_{Ras}$, $v_{c-Raf} \dashv v_{c-Raf}$, $v_{MEK} \dashv v_{MEK}$, $v_{MAPK1,2} \dashv v_{MAPK1,2}$, $v_{RB} \dashv v_{RB}$, $v_{ELK} \dashv v_{ELK}$, $v_{RSK} \dashv v_{RSK}$, $v_{GAB1} \dashv v_{GAB1}$, $v_{PIP3} \dashv v_{PIP3}$, $v_{PI3K} \dashv v_{PI3K}$, $v_{PDK1} \dashv v_{PDK1}$, $v_{AKT} \dashv v_{AKT}$, $v_{BAD} \dashv v_{BAD}$,

$v_{NFKB} \dashv v_{NFKB}$, $v_{AMPK} \dashv v_{AMPK}$, $v_{mTOR-Raptor} \dashv v_{mTOR-Raptor}$, $v_{p70S6K} \dashv v_{p70S6K}$, $v_{pS6} \dashv v_{pS6}$, $v_{4EBP1} \dashv v_{4EBP1}$.

Notice that the last several edges are self-inhibitory loops (e.g., $v_{Ras} \dashv v_{Ras}$). These loops are used to model regulatory mechanisms that are not present in the model network.

For molecules that do not have specific inhibitory edges modeled in the network, we use the self-inhibitory loop to prevent exponential increase in the token counts and to model inhibitory mechanisms beyond the scope of the network. For example, consider the molecule *Ras* in the network shown in Figure 2.1. In the model, this protein is not inhibited. However, biologically we know that Ras has intrinsic GTPase function which inactivate itself [JBS99]. In order to model this, we introduce a self-inhibitory loop.

The differences between the two cell-specific networks are captured by following activity assignments to various proteins in the SPN. In the MDA231 cell line, $H^{MB} = \{v_{Ras}, v_{EGF}\}$ and $L^{MB} = \emptyset$. In the BT549 cell line, $H^{BT} = \{v_{EGF}\}$ and $L^{BT} = \{v_{PTEN}\}$.

Of the two perturbations we considered, one significantly knocked down the activity-level of TSC2 and the other knocked down mTOR-Raptor. While the core SPN still modeled these networks, separate *perturbed* activity-assignments were required for each cell line-perturbation pairing: $L^{MB-TSC2} = L^{MB} \cup \{v_{TSC2}\}$, $L^{MB-mTOR} = L^{MB} \cup \{v_{mTOR-Raptor}\}$, $L^{BT-TSC2} = L^{BT} \cup \{v_{TSC2}\}$ and $L^{BT-mTOR} = L^{BT} \cup \{v_{mTOR-Raptor}\}$.

3.2.2 Setup for Perturbation Experiments

The following experiments were conducted by Dr. Prahlad T. Ram's lab. We include description and results for readability and self-containment of the thesis manuscript.

Cell culture and stimulation. Human MDA-MB-231 (MDA231) and BT549 breast cancer cells were routinely maintained in RPMI supplemented with 10% FBS. For signaling experiments, logarithmically growing cells were serum-starved for 16 hours and then subjected to treatments by epidermal growth factor (EGF) (20 ng/mL) (Cell Signaling Technology, Beverly, MA) for 30 minutes. Controls were incubated for corresponding times with DMSO. To knock down TSC2, cells were treated with short interfering RNA (siRNA) (Dharmacon, Lafayette, CO) for 72 hours prior to EGF stimulation. Control cells were transfected with non-targeting (N/T) siRNA (Dharmacon, Lafayette, CO) prior to EGF treatment.

Antibodies. The following antibodies were used for immunoblotting: anti-phospho-p44/42 MAPK, anti-phospho-GSK3β(S21/S9); anti-phospho-AKT(ser473); anti-phospho-TSC2(T1462); anti-phospho-mTOR(S2448); anti-phospho-P70S6K(T389) (Cell Signaling Technology, Boston, MA); and anti-β-Actin (Sigma-Aldrich, St. Louis, MO).

SDS-PAGE and Immunoblotting. Cells were lysed by incubation on ice for 15 minutes in a sample lysis buffer (50 mM Hepes, 150 mM NaCl, 1mM EGTA, 10 mM Sodium Pyrophosphate, pH 7.4, 100 nM NaF, 1.5 mM MgCl2, 10% glycerol, 1% Triton X-100 plus protease inhibitors; aprotinin, bestatin, leupeptin, E-64, and

pepstatin A). Cell lysates were centrifuged at 15,000 g for 20 minutes at 4C. The supernatant was frozen and stored at -20C. Protein concentrations were determined using a protein-assay system (BCA, Bio-Rad, Hercules, CA), with BSA as a standard. For immunoblotting, proteins (25 μg) were separated by SDS-PAGE and transferred to Hybond-C membrane (GE Healthcare, Piscataway, NJ). Blots were blocked for 60 minutes and incubated with primary antibodies overnight, followed by goat anti-mouse IgG-HRP (1:30,000; Cell Signaling Technology, Boston, MA) or goat anti-rabbit IgG-HRP (1:10,000; Cell Signaling Technology, Boston, MA) for 1 hour. Secondary antibodies were detected by enhanced chemiluminescence (ECL) reagent (GE Healthcare, Piscataway, NJ). All experiments were repeated a minimum of three independent times.

3.2.3 Setup for Perturbation Simulations

To select the block duration parameter, B, we compared the experimentally derived fold change of AKT in the MDA231 cell line to the AKT fold changes predicted for $B = 10, 20, 50, 100$, and 1000. We found $B = 20$ to be the best fit and used this value for all simulations in this study.

We also experimented with input parameter r, the numbers of individual simulation runs averaged per simulation. We tried a range extending from $r = 100$ to $r = 1000$. We found that no observable changes occurred in trends for $r \geq 400$. Therefore, $r = 400$ was used for all simulations in this study.

We considered both the zero and experimentally derived initial states as the ini-

Table 3.1: The experimentally derived initial markings used in the simulations.

Molecule	Control	TSC2 Inhibited
mTOR-Raptor	0	1
TSC2	0	0
GSK3β	5	3
p70S6K	0	2
AKT	0	0
MAPK	2	6

MB231

Molecule	Control	TSC2 Inhibited
mTOR-Raptor	5	5
TSC2	6	0
GSK3β	3	6
p70S6K	0	0
AKT	7	7
MAPK	1	2

BT549

tial markings for the TSC inhibition simulations. The experimental states for both cell lines were derived from western blots produced from cells that were incubated in DMSO and serum-starved for 16 hours. Unsampled molecules were assigned a marking of zero. The number of tokens assigned to each sampled molecule was directly proportional to the darkness of the line on the western blot. This assignment was done by hand, though devising automated and standardized methods for the construction of experimentally derived initial states is an important direction for future work. Since most of the molecules in the network were not sampled, only mTOR-Raptor, TSC2, GSK3β, p70S6K, AKT, and MAPK were given non-zero markings. The initial markings used are shown in Table 3.1.

Since experimental results for the mTOR-Raptor inhibition were obtained from

literature, we did not have experimental results for construction of experimentally-derived initial states. Therefore, we used the zero initial states for the mTOR-Raptor inhibition simulations.

3.3 Results

In order to evaluate the accuracy of our simulation method, we tested its predictions of the effect of targeted manipulations on two cell-specific versions of the signaling network depicted in Figure 2.1. In each cell line, a TSC2-specific siRNA was applied and the concentration of several key proteins in the EGFR network were sampled 30 minutes after stimulation with EGF. This was repeated in the absence of the TSC2 siRNA in order to obtain the concentration in the control network. We also collected a corpus of literature detailing the response of signaling proteins activity-levels to the inhibition of mTOR-Raptor using Rapamyacin [SAS05, ORS+06]. Predictions were generated by our simulator for the TSC2 and mTOR-Raptor perturbations in both cell lines.

3.3.1 Simulation

To simulate a perturbation, we used two networks both based on the signaling network shown in Figure 2.1: the control network for the cell line and the perturbed network for the cell line. The control networks for the cell lines were different because it was important to model the cell-specific mutations. In the case of the BT549 cell

Cell Line	Western Blot Results	Simulation Results

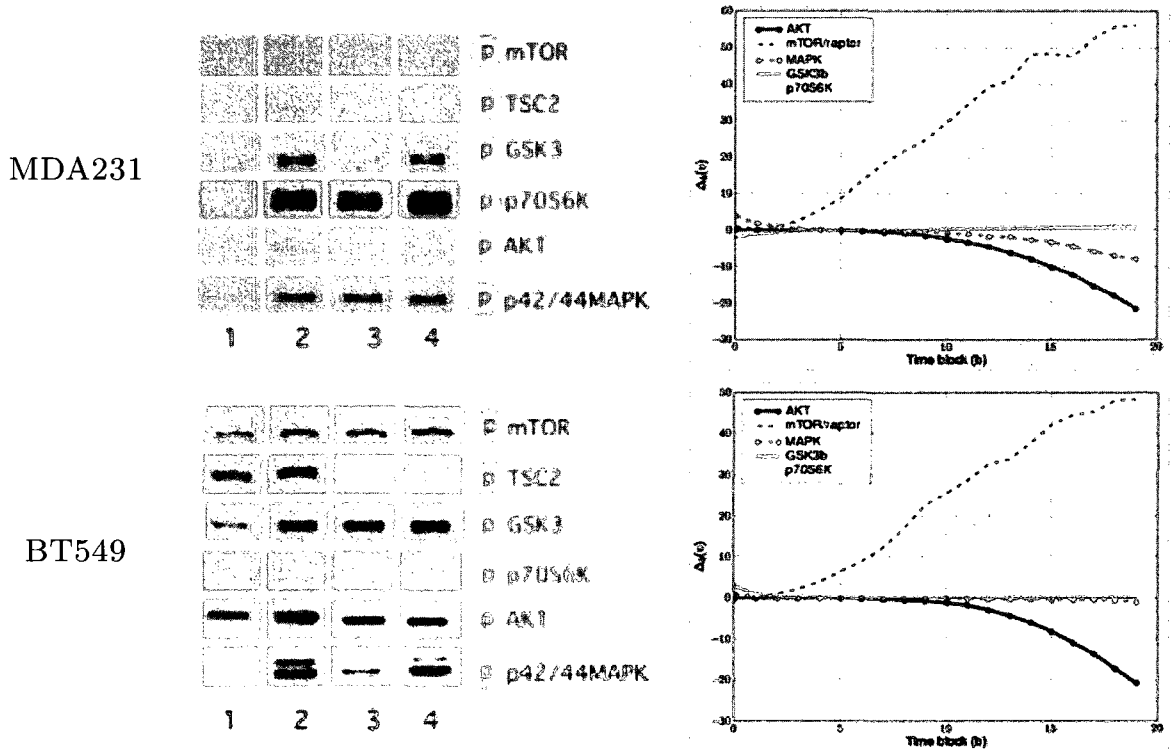


Fig. 3.8: The results of the TSC2 perturbation experiments and simulations. In the western blots, columns (or lanes) are as follows: (1) non-targeting (NT) control siRNA, (2) NT siRNA + EGF, (3) TSC2 siRNA, (4) TSC2 siRNA + EGF. The effect of the TSC2 siRNA on a given molecule can be assessed by comparing column 4 against column 2. For each molecule in the western blot, there is a corresponding simulation curve showing the predicted change in protein activity over time. For the purposes of this analysis, we compared the concentration change after 20 time steps (the left-most data points in the plots) for each molecule. Each simulation point corresponds to the average of 400 measurements that were computed using the procedure described in Figure 3.7. Experimentally-derived initial states were used in the simulations. The results of both the experiments and simulations are qualitatively summarized in Table 3.3.

line, there is a mutation that leads to the loss of PTEN, which makes AKT always active. In the MDA231 cell line, there is a mutation in Ras, which makes it always active. As shown in the formulation of the model, these are modeled using fixed activity assignments in the simulator.

The TSC2 (mTOR-Raptor) perturbed network for a cell line was created by taking

Table 3.2: The t-values for the molecules sampled in the microarray. The critical value for an alpha value of 0.05 with 50 samples is 2.0086. Note that the t-values for all molecules except for GSK3β are larger than this value, confirming that these changes are statistically significantly.

Molecule	t-value in MDA231	t-value in BT549
MAPK	16.35	18.93
p70S6K	14.22	5.83
TSC2	21.65	8.28
AKT	6.60	9.55
GSK3β	0.42	0.10
mTOR-Raptor	41.72	30.53

the control network and fixing the activity-level of TSC2 (mTOR-Raptor) to zero for the duration of the simulation, effectively simulating the pharmacological inhibition of the protein. For each cell-line/perturbation pair, we ran the simulator on the control and perturbed networks using the DIFFERENTIALSIMULATE procedure in Figure 3.7 which computed the change in token-counts induced by the perturbation for all proteins in the model. These change plots are shown in Figure 3.8 for TSC2 and in Figure 3.9 for mTOR-Raptor. Be ran the simulations using both experimentally-derived initial states as well as zero initial states. The initial state used did not change the overall trends observed in the simulations.

Using the t-test described in the Methods section, we also computed the statistical significance of the final time block ($b = 20$) for each molecule considered. For each molecule considered, 400 runs, 20 time blocks, and 50 samples were used. With the exception of GSK3β which did not show a significant response to the perturbation, the changes of all other proteins sampled were beyond the 0.05 significance level (see

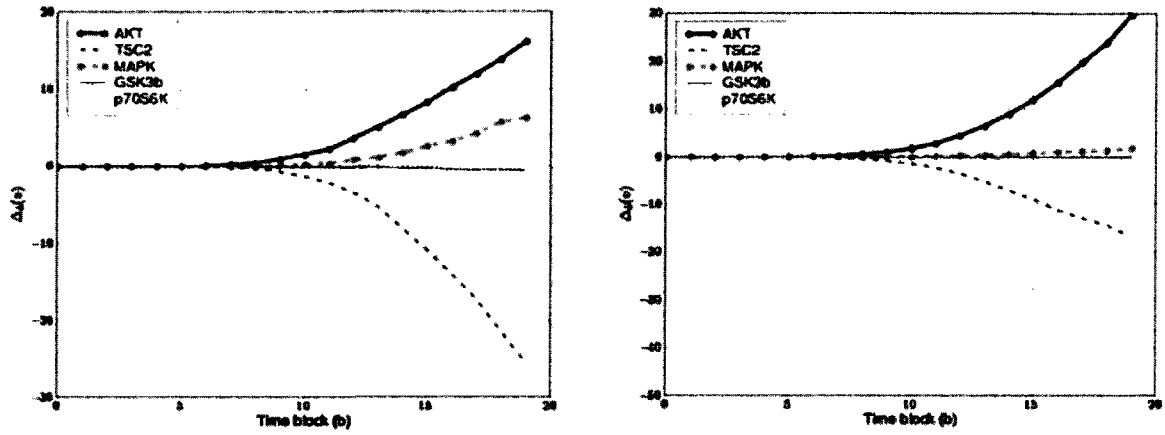
Table 3.2). The statistical insignificance of the change in GSK3β is not surprising since, as shown in Figure 2.1, GSK3β is solely activated by LKB, a molecule fixed high in both cell lines. Thus, we should not expect either perturbation to have a significant effect on the activity of GSK3β, which is what the t-value indicates.

Table 3.3: Summary of the effect of perturbation reported by experimental and simulated methods. The up arrow ($\uparrow$) indicates that the perturbation caused a rise in the level of the phosphorylated protein; the straight line ($-$) indicates no change; and the down arrow ($\downarrow$) indicates that a decrease occurred. Values in the *Experiment* column were estimated by comparing lanes 4 and 2 in Figure 3.8. We estimated the *Simulation* column by determining whether the top quartile of the distribution for the final time point was above, below, or at zero. In some cases it is difficult to judge for certain whether the total quantity of the phosphorylated protein changed or remained the same—both for the experimental and computational cases. In these situations, we indicated the uncertainty by listing the possible changes that the protein *could* have feasibly undergone.

	MDA231			BT549	
Protein	Experiment	Simulation	Protein	Experiment	Simulation
mTOR	$\uparrow$	$\uparrow$	mTOR	$\uparrow$ or $-$	$\uparrow$
TSC2	$\downarrow$	$\downarrow$	TSC2	$\downarrow$	$\downarrow$
GSK3β	$-$	$-$	GSK3β	$-$	$-$
p70S6K	$\uparrow$	$\uparrow$	p70S6K	$\downarrow$	$\uparrow$
AKT	$\downarrow$ or $-$	$\downarrow$	AKT	$\downarrow$	$\downarrow$
MAPK1,2	$-$	$-$	MAPK1,2	$-$	$-$

3.3.2 Experimental Results

After the TSC2 perturbation was applied to a cell line, the protein concentrations were collected using western blots. Details are given in the Section 3.2.2. The western blot results are shown in Figure 3.8.



(a) (b)

Fig. 3.9: The predicted response of the network to an mTOR-Raptor perturbation in the (a) MDA231 and (b) BT549 cell lines. Our method predicts that the amount of available AKT increases in response to the perturbation, which is in agreement with results published in the literature [SAS05, ORS+06]. Our method also predicts that the activity-level of p70S6K in the MDA231 cell line decreases in response to the perturbation, which has been observed experimentally [CRF05]. Each point corresponds to the average of 400 measurements that were computed using the procedure described in Figure 3.7.

3.4 Discussion

As can be seen in Table 3.3, our method correctly predicted the *relative* protein activity-level changes induced by the TSC2 perturbation in both cell lines, for most molecules sampled. Notice that *no change* (−) was reported for the predicted response of MAPK to the TSC2 perturbation despite the fact that a small change did occur in its marking during the simulation (see Figure 3.8) and the t-value for the change is significant (see Table 3.2). At first, interpreting this value as *no change* may seem misleading. However, one of the significant challenges in experimental perturbation experiments is separating true system responses from the background noise created by experimental variables that cannot be precisely controlled (among

them cell population sizes, variability in microarray antibody binding effectiveness, limited sensitivity of hardware and software used to quantify experimental results). As a result, a common practice is to only consider those substantial changes that are well beyond the background noise level. Our interpretation of the small predicted change in MAPK as *no change* reflects the fact that such small changes would not be detectable in microarray or western blot results. Thus, though such a small fluctuation might have occurred in the real data, it would not have been detected by the biologists and most likely would appear in the experimental data to have not changed.

Similar reasoning guided our decision to characterize the simulation (and experimental) results as either up ($\uparrow$), down ($\downarrow$), or no change ($-$) in general. Since the amount of protein registered in a microarray or western blot is not always a reliable indicator of the exact amount of protein (or protein form) being measured, biologists are often reluctant to report degrees of increases or decreases—preferring qualitative observations such as *up* or *down* which are less subject to influence by extraneous experimental conditions. It is true that our simulation method produces precisely quantified increases or decreases which can be taken to indicate degrees of change in response to perturbations. However, as experimental techniques cannot reliably measure degrees of increase or decrease, we judged the qualitative (up or down) characterization to be a more reliable way of validating our method. Certainly, our method provides additional information of degrees of change and we consider studying the accuracy of these degrees to be an important area for future work.

Table 3.4: The number of paths connecting several pairs of compounds in the EGFR model used in our simulations. The multiple paths connecting pairs of proteins highlight the complex interactions present within the network that give rise to its overall dynamic behavior.

Source Protein	Destination Protein	Number of Paths
EGFR	TSC2	7
AKT	mTOR-Raptor	6
MEK	EGFR	4
AKT	p70S6K	8

Our method also correctly predicted the activity-level change of AKT in response to mTOR-Raptor inhibition as reported by a number of studies [SAS05, ORS+06]. Further, our method predicted that, when mTOR-Raptor is inhibited, the level of p70S6K in the MDA231 cell line decreased, which also had been observed experimentally [CRF05].

The only incorrect prediction made by our method was the activity-level change of p70S6K in the BT549 cell line. However, BT549 cells contain a retinoblastoma tumor suppressor (RB) mutation [NCF+06] which could alter p70S6K phosphorylation [MVG+02]. It is a strength of our simulator that the discrepancy between our method's predictions and the experimental results identified a section of the model in which additional connectivity has been found which might account for the difference observed.

The predictions made by our simulator would be exceedingly difficult to derive by visual or manual inspection. Table 3.4 shows the number of paths between several pairs of compounds within the network. Where there is more than one path connect-

ing two molecules, feed forward and feed backward loops are present. Attempting to determine, by hand, how these different loops will interact with one another is, by itself, a difficult endeavor even when not considering the additional task of deriving the rest of the network dynamics simultaneously. For the larger networks that are now becoming available, computational analysis becomes even more crucial to obtaining insights into the dynamic behavior of the network.

Despite the complexity of the network dynamics, it was straight-forward to find and integrate the connectivity information used to build it. Most of the information sources [KCCR04, MCEB+05, MLLA05, KM05, AHL+06, LSX+07, IOZ+06, ORS+06] established the *existence* of various pathways and provided few or no biochemical or kinetic details. As a result, the literature we used would have provided little assistance is building a parameterized Petri Net or ODE model. Due to the proliferation of curated signaling network repositories and searchable literature archives, connectivity information is relatively abundant, which makes the ad hoc assembly of networks a relatively straight-forward endeavor. This further underscores the advantage of using our method over ODEs or parameterized Petri Nets to quickly model and characterize some of the dynamics of a signaling network.

For simulations that will be compared to experimental results, the time parameter must be selected carefully. The time parameter, B, indicates how many time blocks our method will simulate. The time block is an abstract unit of time. Therefore, before comparing experimental results and predictions, it is necessary to determine

how many seconds, minutes, or hours correspond to a time block. This can be done by comparing a prediction of the simulator with the experimentally measured activity-level of one or two proteins at several time points in order to determine what time blocks correspond to the different sampled time points. In the present study, we calibrated our time blocks only once for two cell lines and six experimental conditions (two cell lines, with/without TSC2, with/without mTOR-Raptor). To select the time parameter we used the experimentally measured activity changes in two proteins at two time points. In contrast to other predictive dynamic analysis tools which require multiple time points and multiple protein samples in order to calibrate simulation and model parameters, our method has relatively low time and resource investment.

Besides the time parameter, the other component of our simulations which involved experimentally-obtained knowledge was the initial states. The experimentally-derived initial states require that some experimental data be available providing information on the initial concentrations of individual signaling proteins in the network prior to stimulation. However, in the network that we considered here, the overall behavior of the network and of individual signaling proteins was resilient to changes in the initial states used. Zero and experimentally-derived both produced the same overall change predictions. Thus, while experimentally-derived initial states may be important for the simulation of some networks, it may well be the case that many networks (such as the one we considered in this chapter) can be simulated without this knowledge—further reducing the experimental work that must be done prior to

simulation.

The fact that our simulator produced accurate predictions for a variety of experimental conditions using the one core network model and set of simulation parameters also distinguishes our method from other predictive approaches. The only aspect of the model that was modified during the simulations were activity-levels reflecting the immediate effects of either the underlying tumor mutations (Ras and PTEN) or the perturbations (mTOR-Raptor and TSC2 targeted manipulation). In contrast, the accuracy of ODEs and parameterized Petri nets predictions are known to be sensitive to small changes to the model. For comparative studies such as the one conducted in this chapter, an ODE or parameterized Petri net model might need to be re-constructed with different parameters for each experimental condition of interest. As a result, while it is possible to obtain our simulation results using these models, it remains beyond the capabilities of any existing ODE or parameterized Petri net system to provide insights into the effects of experimental conditions on the dynamic behavior of a signaling network with so little initial time and resource investment.

Though our method's predictions will not be as accurate as the results returned by a correctly parameterized ODE, biologists using our method can derive information about a network's dynamic behavior without having to conduct extensive experimentation and computationally expensive parameter estimation. This novel capability offers scientists the exciting prospect of being able to test hypotheses regarding signal propagation in silico. As a result, by using our method researchers can evaluate a

wide array of network responses in order to determine the most promising experiments before even entering the laboratory.

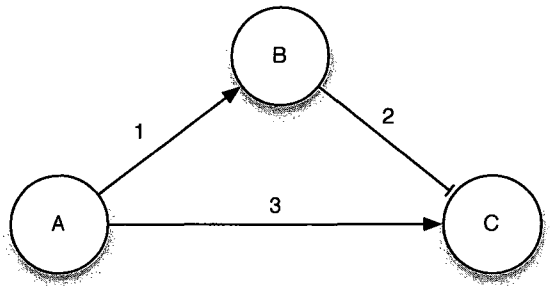
Chapter 4

From Connectivity and Qualitative Data to Dynamics: Deterministic Execution

In Chapter 3 we presented a method that used only network connectivity to predict the dynamics of a signaling network. Despite the absence of any parameters, this method accurately predicted the behavior of the signaling network in 90% of conditions considered. This success was largely due to the fact that network structure appears to be a major determinant of the overall network dynamics. Nonetheless, parameters can be important. Consider the feedforward motif shown in Figure 4.1.

In this example, A has both activating and inhibiting effects on C. What is the overall effect of A on C? Does A have the overall effect of activating or inhibiting C? Answering this question is central to accurately predicting the dynamics of the network. Given such a network, the signaling Petri net approach discussed in Chapter 3 would sample the different relative rates and arrive at some sort of average between the two—in this case, the overall effect would be 'no change' (one path activates, one path inhibits). However, more than likely one of these paths is stronger nudging the overall effect to be either activating or inhibiting. In order to model this situation, parameters are needed to weight the effect of the different interactions.

These weights must come from experimental data. Often this data is supplied by



(a)

Edge Ordering	Effect of A on C
1,2,3	$\rightarrow$
2,1,3	$\rightarrow$
2,3,1	$\rightarrow$
1,3,2	$\dashv$
3,1,2	$\dashv$
3,2,1	$\dashv$

(b)

Fig. 4.1: Example signaling network where the parameters' weights do matter. Table (b) shows the overall effects that different evaluation orderings would create.

perturbation experiments which measure the dynamic response of the signaling network to different environmental and internal conditions. The measurements typically are read as changes in protein concentration, cell population size, or phenotypic outcomes. Numerous methods, ranging from ordinary differential equations to Bayesian networks, use these quantitative experimental measurements directly to infer model parameter values (e.g., [KBP+08, LW08, NWN+08, DGM+06]).

However, as has been discussed previously, using such quantitative data introduces a variety of challenges that require the experimentalist to expend significant effort. We have proposed that qualitative data can be used for building predictive models and, in this chapter, present a method that uses qualitative data to derive parameterized models of signaling networks [RN]. Our method uses a simplified discrete-time model of signal propagation in which each protein has a degradation rate parameter and each interaction has a weight parameter that abstractly models both strength and speed.

The values for these parameters are determined by solving a non-linear optimization problem in which the model equations themselves and the qualitative results from perturbation experiments are used to define the space of valid parameter values. The final values selected maximize the number of qualitative results that the parameterized model can reproduce through simulation.

We use our new method to build a predictive model of a network of signaling pathways downstream of EGFR (see Figure 2.1) in the MCF-7 cell-line. A number of perturbation experiment results for this cell-line were reported in [NWN+08]. To determine model parameter values, we supply our method with the qualitative results of three independent perturbation experiments from [NWN+08]. On the remainder of the dataset, the trained model correctly predicts the effect of a perturbation on a protein's activity-level 85.7% (60 out of 70 predictions) of the time. This predictive accuracy is particularly favorable when compared with the method in [NWN+08] which required training on the numerical results of 20 perturbation experiments from the same dataset to achieve the same recall/prediction accuracy.

After validating our method, we investigate the derived interaction weight parameters. The paths with the strongest weights correspond to interactions with known significance in the MCF-7 cell-line. This analysis both validates our method and demonstrates how parameterized models produced by our method can be used to gain cell-specific insights.

To our knowledge, ours is the first method to use purely qualitative assertions to

perform an entirely automated search for biochemical network model parameter values. However, the idea of moving away from requiring quantitative data has appeared in a number of other methods. The use of piece-wise linear differential equations to model genetic interactions introduced a way of simulating qualitative state changes using differential systems of equations [DGH+04]. Flux-balance analysis uses only connectivity and stoichiometric constants to infer the steady-states of metabolic networks and other systems for which mass conservation laws exists [PPW+03, PP04]. In the signaling domain, the signaling Petri net employs only connectivity (and no parameters) to predict signaling network dynamics [RMT+08].

Mendes *et al.* [MK98] provide a good overview of the use of non-linear optimization for inferring network structure and parameters. Recently, in [LW08], a multi-objective optimization scheme was proposed to infer an S-system structure for experimental time-series data. The work presented in [KBP+08] uses non-linear optimization algorithms to fit linear models of regulatory biochemical networks to time-series data. In our work, we assume the network connectivity is known, and devise a method for parameterizing it so as to achieve a maximally predictive model [RN].

4.1 Method

4.1.1 A simplified model of signaling network dynamics

Dynamic models of biochemical systems fall into two classes: continuous-time and discrete-time. Continuous-time schemes typically model the behavior of the system

as a first-order differential equation

$$\frac{dY}{dt} = f(Y(t))$$

where $Y(t)$ is a vector containing the values of the state variables at time t. The trajectory that the system state vector follows at time t is determined by some function of the current state, $f(Y(t))$.

Discrete-time models, in contrast, explicitly break time into a series of steps in which the behavior of the system is expressed as the inductive formula

$$Y_{t+1} = f(Y_t) \tag{4.1}$$

where $f(x)$ is the transition function that evaluates to the next state visited after x. Often such discrete-time models are linear in the system state variables, in which case the state transition formula can be rewritten

$$Y_{t+1} = AY_t$$

where A is the transition matrix. In models of metabolic networks, A corresponds to the stoichiometric matrix. This correlation does not extend to signaling systems, however, since the underlying biochemical reactions are rarely explicitly modeled. Regardless of the interpretation of A, a given state variable y^i_{t+1} is determined by

$$y^i_{t+1} = \sum_{1 \le j \le |Y|} a_{i,j} y^j_t$$

where $a_{i,j}$ is the element of A at row i, column j. Thus, the system's next state depends entirely upon the current state and the elements of A. These $a_{i,j}$ are the parameters of the system. Once the values of these have been determined and a starting condition, Y_0 has been specified, the model is complete.

Though continuous-time models seem to express the biochemical processes more accurately (the underlying system is continuous in nature), discrete-time models enjoy a number of practical advantages over continuous-models that can make them the better suited for certain types of problems: (1) though the underlying biochemical systems may be continuous, time-series data is inherently discrete, representing one or more time points at which the state of the system was observed; (2) the inductive structure of Equation 4.1 makes it easy to derive the state space of the system; and (3) Equation 4.1 allows the explicit derivation of the finite sequence of states visited given a starting state and a number of time steps. The third property is of particular interest to us here as we use the finiteness property of this sequence to efficiently find parameter values for a model that satisfy certain qualitative properties.

In order to take advantage of this finite state sequence property, we build a discrete-time model of a signaling network with the form:

$$y_{t+1}^i = \max(\delta_i y_t^i + \sum_{j \in A_i} w_{j,i} y_t^j - \sum_{j \in H_i} w_{j,i} y_t^j, 0). \tag{4.2}$$

State variable i corresponds to the activity-level of a signaling protein, δ_i is the degradation rate of that protein, A_i are other proteins in the system that activate i, and H_i are other proteins in the system that inhibit i. Since A_i and H_i specify the proteins that interact directly with i, the A_i's and H_i's for all i's in the system constitute the *connectivity* of the system—the directed interactions that connect the proteins in the system together. The parameter $w_{j,i}$ denotes the strength of the effect that j has on i through the interaction that connects them. Models similar to this

have been used to capture transcriptional dynamics (e.g., [CMW07, KC08]).

Therefore, in our model of a signaling network each individual protein has a separate equation of the form in Equation 4.2; collectively they are called the state equations, S. For a signaling network, the state equations are determined by the set of proteins in the system, $\mathbb{P} = \{1, ..., N\}$, and the activating and inhibiting interacting protein sets for each protein $i \in \mathbb{P}$, A_i and H_i, respectively. When the parameters δ_i and $w_{i,j}$ are specified and a starting point is selected, the resulting system can be simulated by iteratively evaluating the state equations for increasing values of time, t.

4.1.2 Qualitative data from perturbation experiments

To determine values for δ_i and $w_{i,j}$, we require qualitative data from perturbation experiments. A perturbation experiment activates or inhibits the function of one or more proteins (called *targets*) through the use of various mechanisms such as drugs, gene knockouts, or siRNA. These perturbations have varying effects on the response of other proteins and cell phenotypes to signaling events. For a given signaling protein, the perturbation's effect is measured by comparing the activity-level of that protein in an unperturbed cell to the activity-level of the same protein under the perturbed condition. Ordinarily the cell is stimulated prior to measuring the activity-levels in order to determine how the perturbed protein(s) influence the signal that reaches other proteins.

Given the unperturbed and perturbed activity-levels for proteins X, Y, and Z

(X_u and X_p, Y_u and Y_p, Z_u and Z_p, respectively), we can make qualitative assertions about the effect of the perturbation on the activity-level of each protein: $X_u < X_p$ if X increased in response to the perturbation, $Y_u > Y_p$ if Y decreased, and $Z_u = Z_p$ if Z exhibited no change.

It is possible to make many other kinds of qualitative assertions about the experimental results. For example, the biologist may observe that the perturbed concentration of Z is greater than that of Y: $Z_p > Y_p$; or that the unperturbed value of Z appears to be two times that of X: $Z_u = 2X_u$. In fact, any observations taking such forms can be used to constrain the parameter values of the model. However, using such constraints must be done with great care since comparison across protein types and conditions may not be meaningful due to differing concentrations and measurement accuracy for various protein types.

For the remainder of this chapter, we consider (without loss of generality) the three fundamental assertions: $Y_u < Y_p$, $Y_u > Y_p$, and $Y_u = Y_p$ as the types of qualitative data that constrain the training process.

4.1.3 Training a model using qualitative data

Given the connectivity for a signaling network of interest—the sets A_i and H_i for all proteins i in the system—we designed a training method that takes a set of qualitative data from perturbation experiments and infers values for the parameters δ_i and $w_{i,j}$ that make the resulting model reproduce the maximum number of qualitative behaviors specified possible (when the appropriate perturbed conditions are

simulated).

Our method works by converting the model and the qualitative data into a series of constraints for a non-linear optimization problem. The optimization algorithm is directed to find values for all δ_i and $w_{i,j}$ such that the model's behavior satisfies as many qualitative data constraints as possible.

The specific optimization problem is constructed as follows. Given a simulation time period, T, we use the discrete-time model in Equation 4.2 to explicitly write the activity-level of each protein at each time step in terms of (1) activity-levels from prior time steps and (2) the parameters whose values we seek. Therefore, we construct one constraint having exactly the form of Equation 4.2 for each protein in the model for each condition being used for training. The number of conditions equals one plus the number of perturbations being used for training (the "extra" condition is the un-perturbed condition).

Once these model constraints have been constructed, all the qualitative data constraints can be written in terms of the protein activity-level values (y_t^i) for the different conditions.

Modeling perturbation experiments

Formally, a perturbation experiment can be characterized as the set of inhibited proteins, $P \subseteq \mathbb{P}$. The perturbed signaling network is structurally the same as the unperturbed network except where the perturbation has its effect. As a result, the state equations of the perturbed network, S^P, are largely the same as those in the

unperturbed network, S^0:

$$S^P[i] := \begin{cases} S^0[i] & \text{if } i \notin P \\ y^i_{t+1} = 0 & \text{if } i \in P \end{cases}$$

where $S^X[i]$ is the state equation for protein i under condition X.

Given the state equations for a perturbation experiment, S^P, and the unperturbed signaling network, S^0, we can compute the qualitative change in protein i due to the perturbation by simulating both networks from some initial state Y_0. The predicted qualitative change in protein i is:

$$\hat{q}_i = \begin{cases} < & \text{if } \Delta_i < -\epsilon \\ > & \text{if } \Delta_i > \epsilon \\ = & \text{if } -\epsilon \leq \Delta_i \leq \epsilon \end{cases}$$

where $\Delta_i = \sum_{0 \leq t \leq T}(S^0[i,t] - S^P[i,t])$ is the difference in the activity-level of protein i over the time of the simulation ($S^X[i,t]$ denotes the value of the state equation for protein i at time t under condition X). The ϵ parameter is incorporated into the definition in order to desensitize the measure to extremely small, probably insignificant, changes (e.g., $\Delta_i = 10^{-12}$ most likely does not indicate a change of any significance).

Training a model using qualitative data from perturbation experiments

To train a model, S^0, a set of qualitative changes are provided, $Q = \{(p_1, q_1), (p_2, q_2), ..., (p_R, q_R)\}$ where $q_r \in \{<, >, =\}$ indicates the way that the activity-level of protein p_r changed in response to perturbation S^P with respect to the unperturbed system S^0.

The objective of the training procedure is to select an initial condition, Y_0, degradation rates, δ_i, and interaction weights, $w_{i,j}$, such that when the original and perturbed systems are simulated (S^0 and S^P, respectively), $\hat{q}_p = q$ for $(p, q) \in Q'$ such that $Q' \subseteq Q$ and $|Q'|$ is as large as possible.

As with most training procedures, ours is a search for parameter values that cause the model to which they belong to behave in a certain way. We formalize the parameter search as a non-linear optimization problem in which the parameters are free variables constrained by the

1. state equations in S^0 and S^P,

2. the qualitative behavioral assertions, Q, and

3. a set of logical constraints: $0 \leq \delta_i \leq 1$ (the activity-level of a protein can never fall below zero), and $w_{i,j} \geq 0$ (the effect of a protein can not be negative).

It is worth noting that, because this non-linearity takes such a regular form, we suspect that there may be more optimal search strategies than a general non-linear optimization algorithm. We identify this as a topic for future work.

In order to build the non-linear optimization problem, a simulation time, T, must be specified. Optionally, a set of weights for individual constraints can be specified $\Omega = \{\omega_1, ..., \omega_{|Q|}\}$. Conceptually, these weights can be used to make the optimizer favor satisfying certain constraints over others. If Ω is not specified, all constraints are assumed to be equally important (i.e. $\omega_r = 1$ for all $1 \leq r \leq |Q|$). The problem is then constructed as follows:

- **Free variables**
  - $S^0[i,t]$ - the activity-levels for each protein, $1 \leq i \leq N$, for each time step, $0 \leq t \leq T$, in the original network
  - $S^P[i,t]$ - the activity-levels for each protein, $1 \leq i \leq N$, for each time step, $0 \leq t \leq T$, in the perturbed network
  - $0 \leq \delta_i \leq 1$ - the degradation rate of each protein
  - $w_{i,j} \geq 0$ for all interactions - the interaction weight of each edge in the network
  - $X[r] \in \{0,1\}$ for all qualitative data constraints $1 \leq r \leq R$.
- **Constraints**
  - State equations for the unperturbed and perturbed network: S^0 and S^P
  - Qualitative changes due to perturbations as characterized in Q:
    - The following rule is produced for all rules (p_r,'<'): (proteins that increased in response to the perturbation)

$$X[r]\left(\sum_{t=0}^{T} S^0[p_r,t] - \sum_{t=0}^{T} S^P[p_r,t]\right) < X[r](-\epsilon_r)$$

    - The following rule is produced for all rules (p_r,'>'): (proteins that decreased in response to the perturbation)

$$X[r]\left(\sum_{t=0}^{T} S^0[p_r,t] - \sum_{t=0}^{T} S^P[p_r,t]\right) > X[r]\epsilon_r$$

* The following rule is produced for all rules (p_r,'='): (proteins that did not change in response to the perturbation)

$$X[r]\left|\sum_{t=0}^{T} S^0[p_r,t] - \sum_{t=0}^{T} S^P[p_r,t]\right| \leq X[r]\epsilon_r$$

- **Objective function:** maximize $\sum_{r=1}^{R} \omega_r X[r]$

The choice to use ϵ_r rather than a strict inequality was based on the need to ensure that the optimization algorithm did not satisfy the condition using a trivial difference (e.g., 10^{-20}) and the desire to incorporate support for changing the difference thresholds that signaled a qualitative change (recall the use of a similar ϵ parameter earlier in the definition of Δ_i).

When all constraint weights are equal (e.g., $\omega_r = 1$ for all r), then the objective function forces the optimization algorithm to find parameter values that satisfy the maximum number of qualitative constraints. Giving the optimization algorithm the flexibility to ignore specific constraints is important since certain network structures might make satisfying some qualitative constraints impossible. In these cases, rather than failing outright, the optimization algorithm simply satisfies all other qualitative constraints.

The constraint weights, Ω, are used to bias the optimizer towards satisfying certain perturbation constraints over others. This is useful when some experimental results have higher confidence than others. In such cases, the more highly supported experimental result constraints can be given larger weights in order to cause the optimizer to favor satisfying them over other results in which the researcher has less

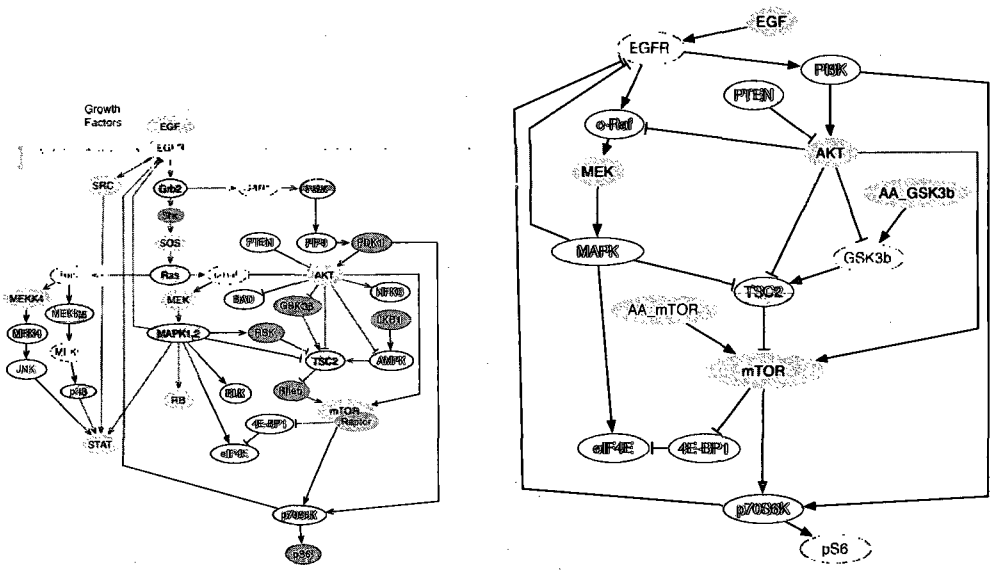
confidence.

4.2 Results

4.2.1 Testing the predictive power of our method on MCF-7 cells

In order to test the ability of our approach to predict dynamic properties of a signaling network, we evaluated its performance on a series of perturbation experiments conducted on the MCF-7 cell-line and published in [NWN+08]. In these experiments, a series of proteins were targeted: EGFR (ZD1839), mTOR (rapamycin), MEK (PD0325901), PKC-δ (rottlerin), PI3-kinase (LY294002), and IGF1R (A12 anti-IGF1R inhibitory antibody). In total, 21 different perturbation experiments were conducted. In each, one or two of these molecules were inhibited, after which EGF stimulation was applied. Phospho-levels for several proteins were measured at the end of each experiment: p-AKT-S473, p-ERK-T202/Y204, p-MEK-S217/S221, p-eIF4E-S209, p-c-RAF-S289/S296/S301, p-P70S6K-S371, and pS6-S235/S236. The effect of these perturbations on two phenotypic processes, cell cycle arrest and apoptosis, were also measured.

For our analysis, we considered a subset of molecules in the EGFR network presented in Figure 2.1 and shown again in Figure 4.2(a). Based on this subset, we considered all protein targets except IGF1R and PKC-δ—both of which are not recognized members of EGFR signaling [CFR99, SVL+92]. This provided a set of 10 perturbation experiments (out of the 21 in [NWN+08]). We included phospho-levels



(a) (b)

Fig. 4.2: (a) A detailed diagram of the EGFR signaling network [RMT+08]. (b) The EGFR signaling network largely restricted to the proteins inhibited or measured in the experiments reported in [NWN+08].

for all proteins measured. Since our current methods are focused on signaling processes, we did not consider the two phenotypic processes since these are the result of a combination of signaling, transcriptional, and metabolic processes.

The complete network in Figure 4.2(a) was reduced in order to minimize the number of proteins and interactions in the model for which measurement information was not available. The motivation for this is to limit the number of parameters whose values are unconstrained by observations, which otherwise makes the parameter space much larger. Clearly, however, it is desirable to support such unmeasured proteins in a predictive model. We identify the problem of extending our methods to handle such unconstrained signaling members as a direction for future work.

The connectivity for the network induced by the measured molecules is shown in Figure 4.2(b). This reduced form of the EGFR network was obtained by keeping only proteins that either (1) were targets, (2) were measured, or (3) were required to maintain connectivity among targets and measured proteins in a non-trivial way. GSK3b was retained in order to ensure that TSC2 had at least one activating input. The molecules AA_mTOR and AA_GSK3b were added in order to model significant sources of activity that reside outside of the EGFR network (GSK3b activity is largely determined by environmental factors and mTOR is activated by Rheb which maintains a high basal activity-level).

To test the predictive ability of our method, we performed a cross-validation procedure in which the model parameters were trained using qualitative data from three experiments. The resulting model was then used to simulate the remaining seven perturbation conditions. The predicted activity-levels from these simulations were interpreted as qualitative observations (e.g., the perturbation caused an increase/decrease/no-change in p-AKT). These predicted observations were then compared to the true qualitative changes in the data. The correctness of the trained model was taken to be the percent of predictions that agreed with the qualitative experimental results.

Each different triplet of perturbation experiments yielded a different parameterized model (the full set of training triplets and their predictive accuracy is provided in the Supporting Information). Most triplet training sets yielded models with $> 70\%$

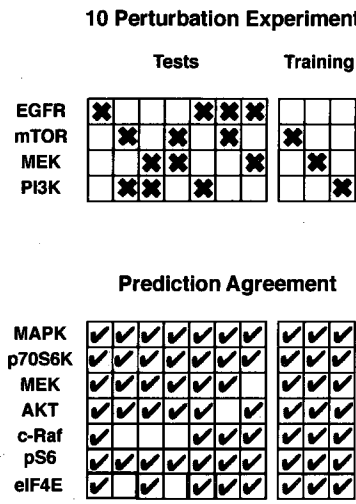


Fig. 4.3: The agreement of one of the best trained model's predictions with perturbation experiments reported in [NWN+08]. Columns are the individual experiments, rows correspond to molecules. The columns set apart to the far right constitute the three experiments used to train the model. In the perturbation experiments matrix, a bold "**x**" indicates inhibited molecules. In the prediction agreement matrix, a "✓" square indicates that our method's prediction for that molecule in that condition agreed with the experimental measurement. Our method correctly predicted 85.7% (60 out of 70) of the test experiment measurements.

accuracy. We observed that good training sets corresponded to those whose perturbation targets were well-distributed throughout the network. The best trained models obtained had 85.7% (60 out of 70) predictive accuracy, one of which was selected for further analysis and is shown in Figure 4.3. As a point of comparison, the predictive model reported in [NWN+08] was trained and tested on this same dataset. Though they trained their method on 20 of the 21 experiments, their method's ability to recall the correct qualitative change for a given molecule in a specific perturbation experiment was 85.7% (60 out of 70). Thus, despite using much less and only qualitative data, our method was able to predict the behavior of individual molecules with a

comparable degree of accuracy.

The errors in our method's predictions may be due to cell-specific signaling properties, some of which are suggested in [NWN+08]. The most significant source of error stems from the misprediction of c-Raf under three different perturbations. c-Raf is activated by Ras and by various isoforms of PKC, none of which is PKCδ [CFR99, SVL+92]. Nonetheless, [NWN+08] detects a significant interaction between PKCδ and c-Raf suggesting that, in the MCF-7 cell-line, this isoform may have some interaction with c-Raf. The absence of such a signaling mechanism in our model could well account for the prediction errors of c-Raf.

The incorrect prediction of eIF4E under the MEK/mTOR perturbation may be related to the complicated mechanisms actually governing eIF4E. Experimental results report eIF4E *increasing* in response to this perturbation. Regardless of parameter values, the connectivity of our model cannot explain this since MAPK and mTOR are the only activators of eIF4E activity. This suggests that the increase in eIF4E activity in response to this perturbation is either the result of an entirely different mechanism or experimental error.

Both our method and the method in [NWN+08] failed to correctly predict the response of AKT to the EGFR/mTOR perturbation. Under the perturbation, AKT is reported to have shown no change (0.0 fold increase). While it is certainly possible for AKT to have not changed, it is also possible that the change (up or down) was sufficiently small as to not register as a change during analysis: note that in [NWN+08],

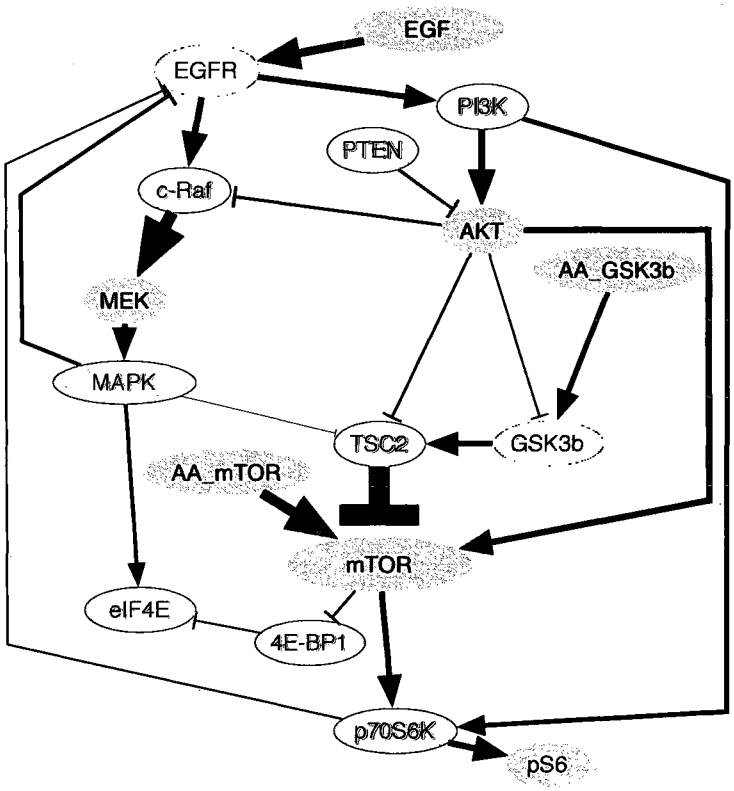
the AKT blots are quite dark and cover much of the channel, factors that make discerning small fold changes more difficult. It is also possible that AKT signaling occurs differently in the MCF-7 cell-line due to a known mutation in PIK3CA (the catalytic subunit of PI3K) which causes MCF-7 cells to have higher basal levels of AKT phosphorylation than normal cells [SVL+92].

Like AKT, the incorrect prediction of MEK activity under the EGFR/MEK perturbation may be the result of the existence of some mechanism not present in our model. Typically, MEK is activated through the pathway EGFR $\rightsquigarrow$ c-Raf $\rightsquigarrow$ MEK. However, MEK is observed to increase while c-Raf activity drops, which cannot be explained by interactions in the model. Thus, other cell-specific signaling pathways may dominate MEK's activity under this perturbation. Close inspection of the western blots for c-Raf in [NWN+08] also raise the possibility that the reported changes are simply artifacts of the western blots themselves.

4.2.2 Interpretation of Interaction Weights

In addition to predictive capabilities, our method produces a model whose parameters have been derived from experimental data. There are several aspects of the interaction weights (shown in Figure 4.4) inferred for the EGFR network in the MCF-7 cell-line that offer insights into cell-specific signaling. The four heaviest pathways in the network are:

- EGFR $\rightarrow$ c-Raf $\rightarrow$ MEK $\rightarrow$ MAPK,



Interaction	Weight
4E-BP1 ⊣ eIF4E	0.158
AA_GSK3b → GSK3b	0.462
AA_mTOR → mTOR	0.939
AKT ⊣ c-Raf	0.271
AKT ⊣ GSK3b	0.162
AKT → mTOR	0.569
AKT ⊣ TSC2	0.214
c-Raf → MEK	1.130
EGF → EGFR	0.762
EGFR → c-Raf	0.554
EGFR → PI3K	0.491
GSK3b → TSC2	0.481
mTOR ⊣ 4E-BP1	0.199
mTOR → p70S6K	0.522
MAPK → eIF4E	0.321
MAPK ⊣ EGFR	0.299
MAPK ⊣ TSC2	0.055
MEK → MAPK	0.476
p70S6K ⊣ EGFR	0.157
p70S6K → pS6	0.546
PI3K → p70S6K	0.389
PI3K → AKT	0.616
PTEN ⊣ AKT	0.271
TSC2 ⊣ mTOR	2.127

Fig. 4.4: The EGFR signaling network model with relative interaction weights depicted by the width of arrows.

- EGFR → PI3K → AKT,
- EGFR → PI3K → p70S6K → pS6, and
- AA_GSK3b → GSK3b → TSC2 ⊣ mTOR.

Notice that the first three constitute the three ways in which EGF signal enters the network through the receptor. The interaction weights suggest a relative ordering in the strength of these different signaling paths (listed by signaling endpoint): pS6 < AKT < MAPK.

Cell-specific behavior of AKT. Our model suggests that the EGFR $\rightsquigarrow$ AKT pathway is much less significant than the c-Raf pathway. This is a surprising result when the general significance of the PI3K pathway is considered. Our method appears to have identified a cell-specific attribute, since MCF-7 has a PI3K mutation that induces the constitutive overexpression of AKT [SCY+08]. Additional evidence in support of this hypothesis is that, in our model, AKT was given a degradation rate slower than the network average degradation rate (0.53 compared to the network-wide average degradation rate of 0.47, see Supporting Information) which will cause AKT to maintain its activity-level for longer than other members of the network.

Also notice that the relative strengths of $EGFR \rightsquigarrow MAPK$ and $EGFR \rightsquigarrow AKT \rightarrow mTOR$ suggests a relative ordering of the negative feedback loops that regulate $EGFR$. Because the $MAPK \dashv EGFR$ interaction receives stronger signal than the $p70S6K \dashv EGFR$ interaction, it is likely the case that in the MCF-7 cell-line, $MAPK$ is the stronger negative regulator of EGFR. This coincides with the results in [NWN+08] in which they found significant evidence of negative regulation of EGFR by MAPK, but no indication for that of p70S6K.

Tumor cell use of GSK3b. GSK3b participates in regulating a number of important cellular processes including cell cycle and energy metabolism [Mar08]. A mounting body of experimental evidence also suggests that it may be a mechanism by which cancer cells satisfy their significant energy demands. The strong activation of GSK3b

Table 4.1: The contribution that correct connectivity and trained parameters make to overall model accuracy for the EGFR network.

Connectivity	Parameters	Accuracy
Correct	Trained	85.7% (60/70)
Correct	Random	59.3% (approx. 40/70)
Random	Trained	21.2% (approx. 15/70)
Random	Random	0.4% (approx. 3/70)

(and the very strong inhibition of mTOR) in our model may be an indication that MCF-7, a breast cancer cell-line, belongs to the class of tumor cells that up-regulates certain cell processes partially through increased GSK3b activity.

The presence of these pathways in our model as strong chains of interactions both provides additional evidence for the predictive capabilities of our method and demonstrates how the parameters of the models can be used to gain insights into the system being studied.

4.2.3 The Importance of Connectivity and Parameters

Since in Chapter 3 we discussed a method that used only connectivity to predict the activity-levels of signaling proteins, an important question to answer is how much the presence of well-trained parameters contributed to the accuracy of our method. In order to understand the contribution of parameters and connectivity in this regard, we evaluated the accuracy achieved by a model (1) with the correct connectivity, but random parameter values, (2) random connectivity with trained parameter values, and (3) random connectivity and random parameter values. Correct connectivity

PROCEDURE RANDOMIZE(V,E)

1. $D[v] = \text{degree}(v, E)$ for all $v \in V$
2. $E' = \emptyset$
3. For each $e \in E$
   - $u, v, t = e$
   - Choose $x \in V$ s.t. $D[x] > 0$
   - $D[x] = D[x] - 1$
   - Choose $y \in V$ s.t. $D[y] > 0$
   - $D[y] = D[y] - 1$
   - $E' = E' \cup \{(x, y, t)\}$
4. Return $G' = (V, E')$

Fig. 4.5: The algorithm used to randomize the connectivity of a network $G = (V, E)$.

corresponded to the connectivity in Figure 4.2; random connectivity corresponds to a network with all the nodes and edges in the correct network, connected in a randomized pattern (with node degree preserved). The algorithm used to randomize a network's connectivity is shown in Figure 4.5. Trained parameters refers to using the optimal training dataset to select good parameter values; random parameters refers to using parameter values selected within a range of 0 to 1 for retention parameters and 0 to 15 for interaction weights (note that various ranges for parameter values were tested with no change in the overall results we report next). For each scenario considered, 1000 networks were constructed and their accuracy tested against the 7 remaining datasets. Table 4.1 shows the outcome of the results.

The results of these experiments indicate that connectivity is, by far, the most significant contributor to the accuracy of the model's predictions. Even when random

parameters are used, predictions are correct nearly 60% of the time. Having trained parameters, however, does have an impact on accuracy: evidenced by the fact that trained parameters increase accuracy by another 25%.

What these results also show is that training parameters is not always succeptible to the issue of overfitting. While there is always concern that a sufficiently complicated system can always be parameterized to produce certain behavior, for the EGFR network considered here, the degree of connective complexity could only be fit to 21% (approximately 15 out of 20 data points) of the experimental data through training of parameter values.

4.2.4 Selecting Good Training Sets

Not surprisingly, obtaining the right training dataset is central to building a good predictive model. In the previous sections in this chapter we used the training dataset that yielded the most accurate model. However, researchers using this and other methods often do not have the benefit of knowing, a priori, what the best training dataset is for their model. In these cases, the question is: what training dataset will give the model with the most accurate predictions? While we identify this as a direction for future work, in the course of this study we have observed that path coverage appears to distinguish particularly good training datasets and comment on this here.

Recall that models are being trained on the results of perturbation experiments. In each experiment one or more molecules are perturbed and then some (pre-determined)

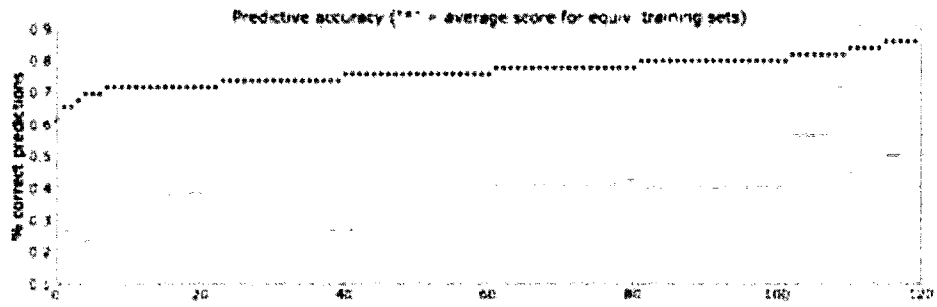


Fig. 4.6: The x-axis is the set of all possible three-experiment training datasets derived from [NWN+08], ordered according to how good each training dataset is (determined in terms of how accurate the resulting model's predictions were). The solid line shows the % of accurate predictions made by the model trained on the dataset. Note that many datasets yield equivalent degrees of accuracy. The dotted line indicates the coverage of the training dataset as computed using Equation 4.3. The position of the dot indicates the average coverage for all datasets that had the same resulting accuracy. Note the trend of better datasets having better coverage.

set of molecules, called the observables, are measured. Our aim in selecting a training dataset is to build a model that predicts the behavior of those same observables under a range of different perturbation conditions.

In order to understand the properties of a good training set, we analyzed all 120 of the three-experiment training sets that could be derived from the data in [NWN+08]. We considered the quality of the training set to be the accuracy of the predictions of the resulting model. The quality of these datasets are shown in Figure 4.6.

In careful analysis of these different training datasets, we discovered a correlation between training dataset quality and how well the perturbations in the training data covered the paths among observables in the network. Formally, let Π be the set of complete paths connecting all observable and stimulated proteins in the signaling network and let $\mathbb{P} = \{P_1, P_2, ..., P_n\}$ be the set of perturbations in the training dataset of interest ($P_i \subseteq V$ is a subset of the molecules in the network being studied that

were perturbed in experiment i). We define the coverage measures as

$$C(\mathbb{P}) = \frac{|\{\pi \in \Pi : \exists P \in \mathbb{P}, P \subseteq \pi\}|}{|\Pi|} \tag{4.3}$$

$C(\mathbb{P})$ is the fraction of all paths connecting observable and stimulated proteins along which a perturbation of the training dataset lies. Figure 4.6 shows how this measure correlates to the quality of the training set. While the curve is not monotonic, there is a clear association between better training datasets and higher coverage scores, as computed by the equation above.

While this is only a preliminary result, intuitively, we expect that datasets that perturb more paths in the network may produce better models. By perturbing more paths that influence observables, the contribution of more parts of the network are represented in the training dataset. The problem in using this measure is that it weights all paths equally. Were all paths equally important to the behavior of the network, this would be an appropriate assumption. However, the contribution of different paths are not all equal (otherwise we would not need to know parameters at all). Therefore, an improving this measure, it will be important to take into consideration the relative contributions that different paths make to the overall behavior. It is unclear as to whether such information can be determined a priori. However, this is certainly an area that deserves additional investigation.

4.3 Discussion

The performance of our method on the MCF-7 cell-line perturbation experiments underscores the two important focal points of our work: simple models of signaling network dynamics are sufficient to capture much of the behavioral complexity of biochemical systems and qualitative data can be effectively used to derive meaningful parameters for computational models. Using the results from only three independent perturbation experiments, our training method identified model parameters that predicted the response of signaling proteins to a variety of other perturbations with over 85.7% (60 out of 70) accuracy.

In our analysis we have shown that experimental results can be used to construct predictive models even when confidence in the exact values obtained is not high. Furthermore, our method can leverage experimental data that is readily available from a wide array of other sources including online databases and literature. In such sources, it is difficult to know whether the numerical values reported can be applied directly to one's system. Nevertheless, the qualitative trends in the data are still a rich source of information. Using our method, these trends can now be used to train predictive models of signaling networks. Even scientific intuitions and untested hypotheses can be easily incorporated into the training process. Having the ability to evaluate such unverified (or unverifiable) ideas can be an important step in the larger process of gaining scientific insight into a system's characteristics. Our methods provide scientists working on cellular signaling this ability.

Chapter 5

Tools

In Chapters 3 and 4 we have discussed two methods for building and executing models of signaling network dynamics. In order to make these methods available to biologists, we have developed software tools that make it possible to easily construct, load, and execute models through a user interface.

This chapter is broken into two sections. In Section 5.1 we discuss the PathwayOracle software tool, implementing the signaling Petri net simulator. In Section 5.2, we discuss the Monarch web application, implementing the deterministic simulator.

5.1 PathwayOracle

In this section, we present the software tool *PathwayOracle* [RNR08]. The core functionality of this system revolves around the signaling Petri net simulator, described in Chapter 3. In order to deliver more comprehensive capabilities, its scope has been broadened such that it provides an integrated environment for connectivity-based structural and dynamic analysis of signaling networks, supporting

- visualization of signaling network connectivity;
- two versions of the signaling Petri net simulator where

  - the first allows prediction of signal flow through a given network for a specific experimental condition, and
  - the second predicts the difference in signal flow through a given network induced by two different experimental conditions;

- enumeration of the paths connecting arbitrary pairs of nodes in the network; and
- visualization of experimental concentration data on the signaling network display.

Directions for future versions include expanded capabilities in all three areas of analysis—dynamic, structural, and experimental—with a focus on providing effective ways of integrating results from each together. *PathwayOracle* has been designed in a modular fashion in order to facilitate extension of existing capabilities and the addition of new features.

In the following subsections, we explain the architecture and core concepts underlying PathwayOracle and then examine the individual features, how they can be used, and how they compare to existing tools.

5.1.1 Implementation

PathwayOracle is written in Python [pyt]. The user experience is oriented around visualization of and interaction with three main types of data: the signaling network, markings, and paths. At any given time, one signaling network is open, which is the

basis for all analyses. Any simulation or concentration data is loaded and inspected as markings. Currently all static analyses revolve around paths, which are the third data type. In the following subsections, these individual data types and the user interfaces to them are discussed in more detail.

The Signaling Network Model

While the implementation of our methods use the signaling Petri net model discussed in an earlier section of this paper, we provide a simpler and more convenient representation of the network to the user which omits the internal topology of the transitions and allows the user to specify interactions simply as either activating or inhibiting. Thus, for the remainder of this paper we use the following definition of the signaling network which is consistent with the experience the user will have when working with *PathwayOracle.* The signaling network connectivity is a directed graph $G = (V, E)$ where

- V is the set of nodes, which are signaling proteins and complexes (hereafter referred to collectively as *signaling nodes*) and
- E is the set of edges, which are signaling interactions. Each edge is of one of two types: $u \rightarrow v$ for activation and $u \dashv v$ for inhibition.

Within *PathwayOracle*, each signaling node has a name, unique within the network. A signaling edge has no properties besides its type and is only defined by its *source* and *target.*

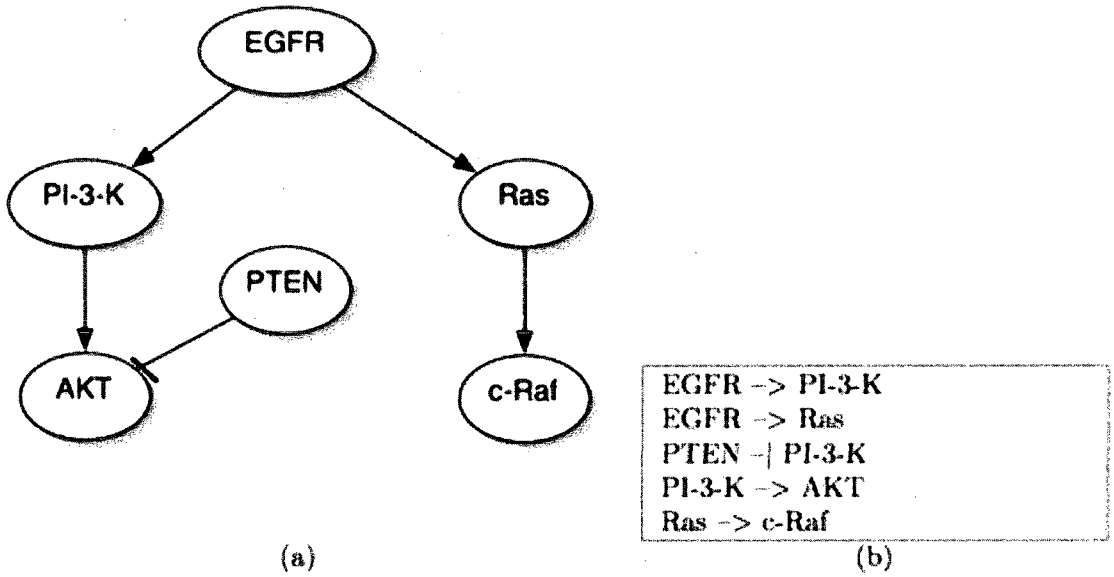


Fig. 5.1: An example of a Network in the Connectivity Format. (a) A graphical representation of a signaling network's connectivity. (b) The signaling network in (a) written in the *Network Connectivity Format.*

In order to facilitate the rapid construction of such signaling network models, we devised a file format called the *Connectivity Format.* It is capable of expressing both general networks as well as paths. When representing a network in the format, as shown in the example in Figure 5.1(b), one signaling interaction is written on a line with the format

$$u \text{ -> } v \qquad \text{or} \qquad u \text{ -| } v$$

where u is the name of the source signaling node and v is the name of the target signaling node. Each node is taken to represent the active form of the protein it is named for. Thus, from the example above, the interaction PI-3-K$\rightarrow$AKT means that the active form of PI-3-K increases the activity-level of AKT whereas the interaction PTEN$\dashv$AKT means that the active form of PTEN decreases the activity-level

of AKT. While these types of unparameterized relationships can be represented in SBML, SBML was designed for encoding much more information than just connectivity [HFS+03]. As a result, we deemed it appropriate to design a more concise format for our purposes. However, in a future release, *PathwayOracle* will support loading and saving in the SBML format.

At a given point in time, only one signaling network can be open in *PathwayOracle*. The main window displays a graphical representation of the network. The layout of the network can be modified by dragging nodes or by *shift*-clicking on edges to create, remove, or move waypoints. These layouts can be saved with the network and loaded again.

Signaling Network Markings

In signaling networks, signal flow is measured and quantified as the fluctuation of concentrations of various forms of signaling proteins over time. In *PathwayOracle*, we model concentrations using the concept of a network *marking*, which was adapted from Petri nets in which it was first used [PRA05].

Markings. In *PathwayOracle*, a marking, μ is an assignment of real values to the nodes of a signaling network such that every signaling node receives a value. Earlier, the concept of a marking was introduced as the assignment of tokens to protein places in the signaling Petri net. In a signaling Petri net, tokens are discrete. In *PathwayOracle*, a marking is an average of the markings from many independent

simulation runs, which gives rise to the real, rather than integral values, assigned by the marking.

As discussed earlier, the value of the marking of a signaling node, $\mu(v)$, can be interpreted as an estimate of the concentration or change in concentration of the active form of the signaling protein v (we call the amount of the active form of the signaling protein its *activity-level*). The two different versions of the simulator generate markings with these different meanings. The first simulator predicts the signal flow due to an experimental condition and generates markings whose values are taken to represent the actual activity-level of signaling protein present over the assumed basal levels. The second version of the simulator predicts the difference in signaling due to changing experimental conditions. The values assigned by markings produced by this simulator correspond to the *change* in the activity-level of the protein induced by the change in experimental condition. This will be discussed further in the Results and Discussion section.

Marking Series. In order to model signal *flow*, a single marking is not enough since it only provides a single snapshot of concentrations throughout the network. A *marking series* is an sequence of markings, $(\mu_1, \mu_2, ..., \mu_T)$ in which the marking μ_t is a snapshot of the concentration distribution at time step t. Thus, it is possible to see how the activity-level of protein v changed by plotting the values $\mu_1(v), \mu_2(v), ..., \mu_T(v)$. *PathwayOracle* provides the ability to do this.

mTOR/raptor,	AKT,	EGFR,	RSK
0.3,	0.2,	0.1,	1.1
...	...	...	...
2.1,	0.001,	0.1,	1.5

(a)

,	mTOR/raptor,	AKT,	EGFR,	RSK
DMSO,	0.3,	0.2,	0.1,	1.1
...	...	...	...	...
EGF_30min,	2.1,	0.001,	0.1,	1.5

(b)

Fig. 5.2: Examples of marking series and group file formats. (a) An example marking series dataset in the *comma-separated value* file format. The first row specifies the signaling proteins whose concentrations were measured. Each row thereafter specifies the concentration for a given time step: row i specifies the concentrations for each signaling protein at time step $i - 1$. (b) An example marking group dataset in the *comma-separated value* file format. The first row specifies the signaling proteins whose concentrations were measured. The first column specifies the names for each marking in the group dataset. The numbers in each row specify the concentration measured for each signaling protein in that marking.

PathwayOracle supports loading a marking series dataset from *comma-separated value* (*.csv*) files. As shown in Figure 5.2(a), the file has a header row which specifies, for each column, the name of the molecule whose concentration values will appear in that column. Each subsequent row contains the value assignments for a marking: the second row contains the marking for time step 1, the third row contains the marking for time step 2, and so on.

Marking Groups. In many experiments, the activity-level of various proteins are sampled at different time points and under different experimental conditions. Since the *marking series* is not able to represent changes due to different experimental conditions, we introduced the more general concept of a *marking group* in which each marking can correspond to an arbitrary activity-level distribution. Each marking is given a descriptive label that can be used to identify the conditions under which the activity-level was sampled.

Like the marking series, a marking group is loaded from a *.csv* file. However, unlike

the marking series in which each row corresponds to a time step, in the marking group, each row corresponds to an independent marking (experimental condition). As shown in Figure 5.2(b), the first row is a header row specifying the molecule names for each column, the first column specifies the names for the individual markings (experimental conditions).

The Marking Manager

PathwayOracle includes a specific user-interface, the *Marking Manager*, designed to manage the three different types of markings. The Marking Manager provides a central interface within which it is possible to view all markings loaded and inspect them in ways that are relevant to their type (marking, marking series, or marking group). The specific ways in which markings can be inspected will be discussed further in the *Results* section.

5.1.2 Signaling Paths

The current structural analysis capabilities available in *PathwayOracle* allow inspection of signaling paths within the network. A signaling path p is a sequence of nodes, $(v_1, v_2, ..., v_k)$ where $v_i \in V \ \forall 1 \leq i \leq k$, and $(v_i, v_i + 1) \in E \ \forall 1 \leq i < k$. In this case, we say that node v_1 is the *source* of path p, and node v_k is the *target* of p. Given a path, a variety of statistics may be of interest to the user. Additionally, it may be useful to view the path within the larger network. *PathwayOracle* provides these capabilites which will be discussed in the Results and Discussion section.

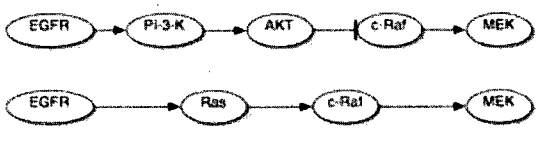


Fig. 5.3: An example of a Path in the Connectivity Format. (a) A graphical representation of two signaling paths. (b) The signaling paths in (a) represented in the *Connectivity Format.* Each line corresponds to a single signaling path.

Sets of paths can be saved to a file and loaded back into a session. Like networks, paths are also stored in the Connectivity Format. When representing a set of paths, as shown in Figure 5.3, the full node names and the edge types are written so that all path information is directly available within the file itself. One line contains one path.

5.1.3 Results

PathwayOracle provides a variety of tools for analyzing the structural and dynamic properties of a signaling network based on its connectivity. While its main differentiating feature is the ability to predict signal flow through a network using only the connectivity of the signaling network, *PathwayOracle* also provides the ability to visualize the network, analyze its connectivity, and inspect concentration-based experimental data.

With the exception of the signaling Petri net simulator, PathwayOracle's features can be found in various combinations in other tools. Figure 5.4 provides a matrix of the features and capabilities of several tools most commonly-used for signaling

Analysis Type	Features	CellDesigner	CellIllustrator	CellNetAnalyzer	COPASI	Cytoscape	Matlab SB Toolkit	PathwayOracle
	Open Source	-	-	✓	✓	✓	-	✓
	Visual Network Editor	✓	✓	✓	-	✓	-	✓
Experimental Data Analysis	Microarray Visualization	-	-	-	-	✓	-	✓
	Microarray Analysis	-	-	-	-	✓	-	-
	Structural Statistics	-	-	✓	-	✓	-	-
	Path Finding	-	-	✓	-	✓	-	-
Structural Analysis	+/- Path Finding	-	-	✓	-	-	-	✓
	Flux Analysis	-	-	✓	✓	-	-	-
	Boolean Analysis	-	-	✓	-	-	-	-
	ODE Simulation	✓	-	-	✓	-	✓	-
Dynamic Analysis	Hybrid PN Simulation	-	✓	-	-	-	-	-
	Signaling PN Simulation	-	-	-	-	-	-	✓

Fig. 5.4: A comparison of features supported by tools commonly used for signaling network analysis. The table shows the features and analytical capabilities supported by different tools commonly used for the analysis of signaling networks. Tools included in the comparison are: CellDesigner [FMKT03], CellIllustrator [NDMM03], CellNetAnalyze [KSRG07], COPASI [HSG+06], Cytoscape [SMO+03], the System Biology Toolkit for Matlab [SJ06], and PathwayOracle.

network analysis. While other tools support a variety of simulation techniques, PathwayOracle, alone, provides non-parameterized simulation capabilities. It is worth noting that the commercial software package CellIllustrator [FMKT03] provides Petri net-based simulation capabilities. The difference between CellIllustrator and PathwayOracle Petri net approaches is the extensive set of kinetic parameters required by CellIllustrator in order to simulate a biological system. In this regard, hybrid functional Petri nets, the underlying technology used by CellIllustrator, are not significantly different from ODEs.

Another important distinguishing characteristic of PathwayOracle is the combination of features that it supports. Biological network analysis is a multi-faceted process that may involve structural, dynamic, and data analysis in parallel. Whereas other tools tend to focus on one or two of these general areas of analysis, we considered it important for PathwayOracle to incorporate all three in order to provide the

researcher a single environment in which all their analysis could be done. In future releases we plan to increase PathwayOracle's support for all three of these directions of investigation: structural, dynamic, and data analysis.

In the remainder of this section, we discuss the features currently available in PathwayOracle.

Network Visualization

As in many other computational analysis tools for signaling networks (e.g., [SMO+03, FMKT03]), an interactive graphical representation of the signaling network connectivity is at the center of the *PathwayOracle* interface. The main window provides a visualization of the signaling network connectivity. This visualization interface allows the user to edit the layout of the network by clicking on and dragging nodes and by *shift*-clicking on edges to create, remove, or move waypoints. Waypoints are points that lie on an edge. Holding down *shift* will display all edge waypoints. Existing waypoints can be dragged to change the path that an edge follows. Right-clicking on a waypoint will remove it. Left-clicking on a straight segment of the edge will create a new waypoint.

The network visualization also provides a view onto which path and experimental data analysis may be mapped. As will be discussed in subsequent sections, selected paths may be highlighted in this view and markings from experiments can set the colorings of individual nodes.

Network Signal Flow Simulation

The main feature differentiating *PathwayOracle* from other tools, such as CellDesigner [FMKT03] and COPASI [HSG+06], is its ability to simulate signal flow using an unparameterized signaling network model. Simulations can be performed in two different ways. In the first (*Single Simulation*), the simulator predicts the signal flow through the network for a specific experimental condition. In the second (*Differential Simulation*), the simulator predicts the difference in signal flow due to two different experimental conditions on the same network. These simulation methods themselves are described in [RMT+08]. Here we focus on how simulations are configured, run, and analyzed.

Whereas the consensus networks typically represent the connectivity in normal cells, many experiments are conducted on abnormal cells in which oncogenic mutations, gene knockous, and pharmacological inhibitors have altered the behavior of various signaling nodes in the network. In *PathwayOracle* users can model these cell- and experiment-specific conditions by specifying each signaling node as either *High*, *Low*, or *Free*. The *High* state models any condition under which a protein's activity-level is held high for the duration of the experiment. This may be due to external stimulation or a known mutation in the protein that makes it constitutively active, for example. Similarly, a *Low* state models any phenomenon that forces a protein to have a persistently suppressed activity-level. This may be due to mutations that render the protein inactive, gene knockouts, or pharmacological inhibitors that force

the activity-level of the protein low. In general, most signaling nodes will be *Free*, which means that their activity-level is unconstrained throughout the simulation. Only those nodes designated as *High* or *Low* will have their activity-level fixed for the duration of the simulation.

In order for a protein to be held high during the simulation, it is necessary to indicate the initial activity-level that the protein will be elevated to. This is done by specifying the number of tokens that the protein will receive. Since a protein with a *High* state cannot be inhibited (even if inhibitory edges target it in the actual network), the protein's activity level will never fall below this initial value. The initial value for a High protein is indicated by placing it in parentheses next to the protein's name, as shown in Figure 5.5.

Two other parameters that must be specified for a simulation are:

- the number of simulation runs to perform and
- the number of time blocks

The number of runs sets the number of independent simulations whose time block markings are averaged together to yield the overall simulation markings. In general, using more runs is a tradeoff between reliability of the results and simulation speed. In practice, the number of runs needed is dependent on the signaling network model and should be selected by observing the reproducability of the simulation results. An appropriate number of iterations will be large enough so that for a given experimental condition, the results are very similar across multiple simulations.

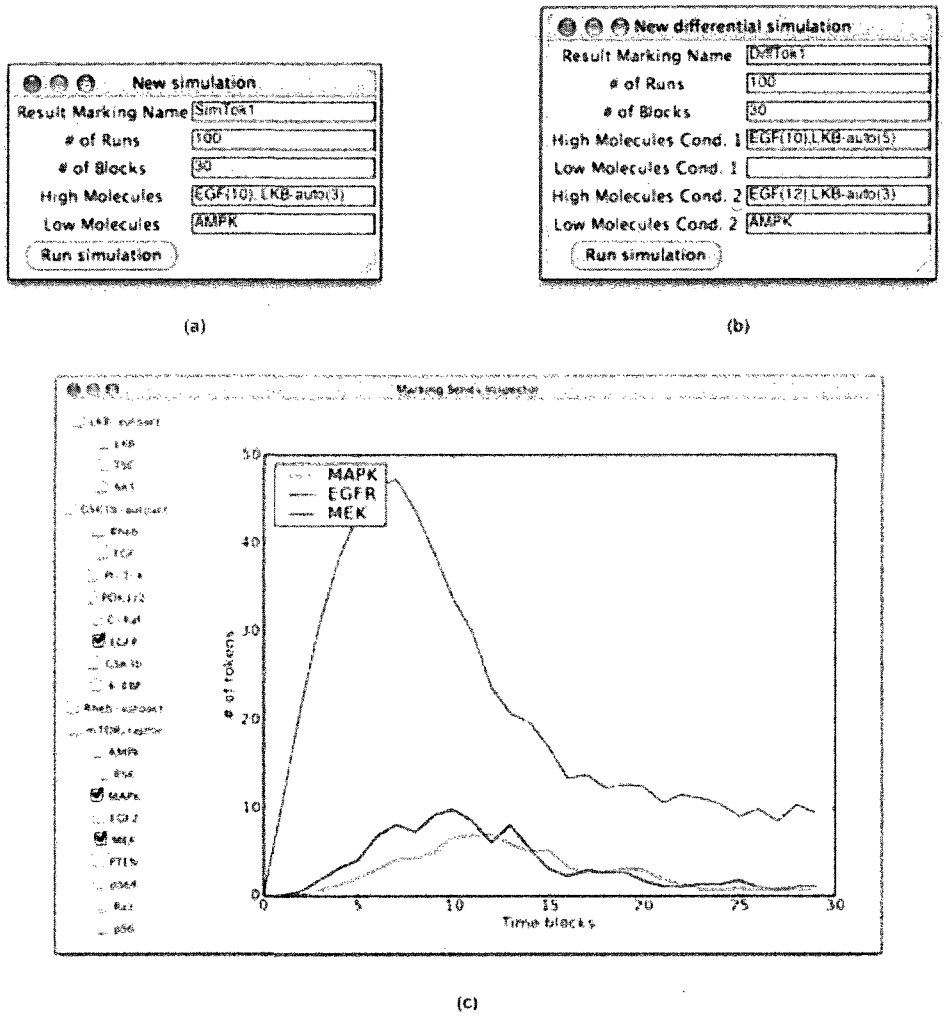


Fig. 5.5: The tokenized simulator user interface. (a) The setup window for the tokenized simulator. The simulation is being configured to have two High nodes, EGF and LKB-auto. EGF will be initialized with a token-count of 10, LKB-auto with a token-count of 3. The token-count of AMPK will be zero for the duration of the simulation. (b) The setup window for the differential simulator. Two different scenarios are being compared through simulation: different token assignments are being tried with EGF and LKB-auto, with and without AMPK being fixed low. (c) The plot window for the marking series generated by a simulation. Observe that the signaling nodes whose activity-levels are plotted correspond to those selected in the checklist directly to the left of the plot.

The time block, as discussed earlier, is a fundamental unit of time in the simulator. The appropriate number of time blocks for which to simulate will vary depending on the size of the signaling network and the scale of the network behavior of interest.

Generally it should be selected by running simulations for a variety of time block values and determining which yields the most biologically reasonable activity-level changes for a known protein. While this is a manual process in the current version of *PathwayOracle*, we are investigating automated methods for estimating the number of time blocks by training against experimental time series data.

In *PathwayOracle*, the setup window for the *Single Simulation* (see Figure 5.5(a)) prompts the user for a single experimental condition. The setup window for the *Differential Simulation* (see Figure 5.5(b)) prompts the user for two experimental conditions. Both simulators produce a marking series. The tokenized simulation marking series corresponds to the activity-level time series predicted for the specified experimental condition. The differential simulation marking series corresponds to the change in activity-levels over time produced by switching from experimental condition 2 to experimental condition 1.

The marking series produced by a simulation can be accessed through the Marking Manager. Choosing to *inspect* a marking series will present the user with a blank plot. By selecting signaling nodes, the plot is populated by the marking series values for individual nodes over time, as shown in Figure 5.5(c).

While this plot generation capability exists in many other dynamic simulation tools, the simplicity of the model used for simulation and the speed with which a simulation runs set *PathwayOracle* apart from other tools which require specification of the numerical values of kinetic parameters for each reaction in the network of interest

(e.g., [FMKT03, HSG+06]). *PathwayOracle*, because of its novel approach, does not have such requirements. It is worth noting, however, where *PathwayOracle* provides approximations of signal flow, an ODE generates the actual concentration changes using extremely detailed and accurate models of the underlying biochemistry. The simulators in *PathwayOracle* provide an attractive, time- and resource-saving alternative this more exhaustively parameterized techniques. In particular, *PathwayOracle's* features will benefit researchers interested in quickly assessing characteristics of signal flow in their network.

For some networks, biologists will have partial knowledge of kinetic parameters or of other biological details which the signaling Petri net model does not, at present, consider. By integrating this knowledge into the simulator, it may be possible to improve the simulator's predictions. We identify this as a direction for future investigation. As the signaling Petri net simulator is extended, these new capabilities will be incorporated in future releases of *PathwayOracle*.

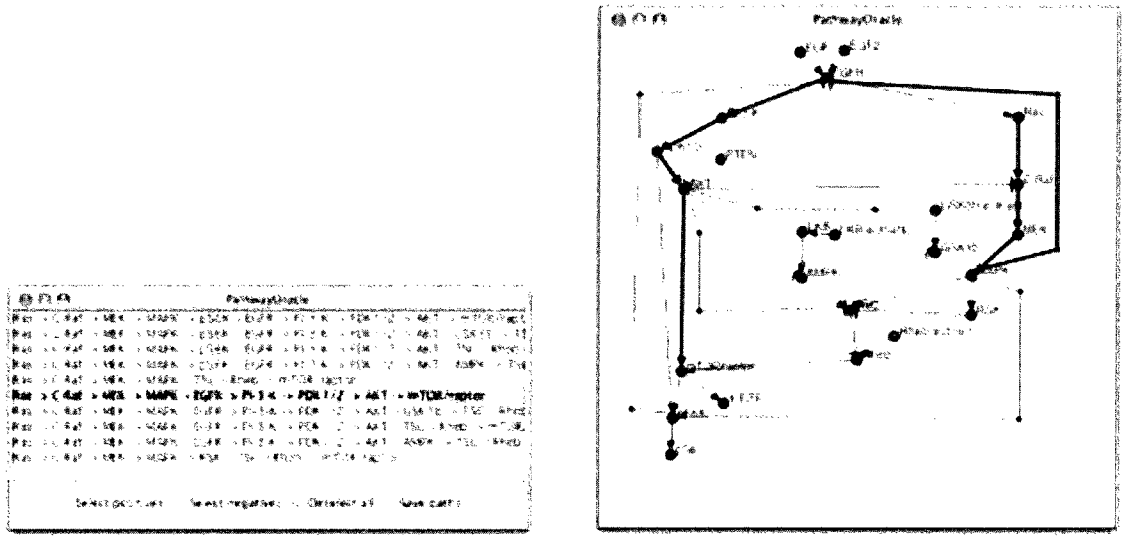
Signaling Path Analysis

The use of the simulators and plotting tools allows the user to observe trends in the activity-level of individual signaling nodes over time. Since the activity-level of a node is determined by the activity-level of other nodes in the network, the activity-level time series of a node may be explained by changes in the activity-level history of nodes upstream of it. In order to investigate such indirect interactions, it is useful to enumerate all the paths leading from a specific protein to the protein of interest.

PathwayOracle provides this capability. Additionally, it provides various statistics on the set of paths linking two signaling nodes as well as a classification of the effect of each path as either *coherent* or *incoherent* (e.g. [Alo07]).

A coherent path is a directed series of interactions that leads from x to y such that an increase in the activity-level of x causes an increase in the activity of y and a decrease in the activity-level of x causes a decrease in the activity-level of y. An incoherent path is a directed series of interactions leading from x to y such that an increase in the activity-level of x causes a *d*ecrease in the activity-level of y and a decrease in the activity-level of x causes a *i*ncrease in the activity-level of y. It is possible to classify a path p as either coherent or incoherent by counting the number of inhibitory edges along p. *A path with an even number of inhibitory edges is coherent; a path with an odd number of inhibitory edges is incoherent* [KSRLS06].

This logic is assumed in *PathwayOracle*. All simple paths (paths without loops) connecting two specified signaling nodes are enumerated by an exhaustive depth-first search. These paths then are classified as either coherent or incoherent, and presented to the user for further inspection in a window similar to the one shown in Figure 5.6(a). When a path is selected in the results window, it is highlighted in the main window, allowing the user to evaluate it within the context of the complete network (see Figure 5.6(b)).



(a) (b)

Fig. 5.6: The path interrogation user interface. (a) The result window enumerating the set of all paths between *Ras* and *mTOR/raptor*. (b) The main network view showing the selected path highlighted.

Experimental Data Analysis

A model of the connectivity of a signaling network makes it possible to identify components of the model that are inconsistent with experimental data or visa versa. *PathwayOracle* enables this kind of analysis by allowing users to load experimental concentration data and visualize it both as a heatmap (see Figure 5.7(a)) or superimposed on the network view (see Figure 5.7(b)). Several other software tools provide similar capabilities (e.g., [SMO+03]). In *PathwayOracle*, experimental concentration data is loaded as a marking group in which a single marking corresponds to a condition for which concentrations were sampled. Figure 5.7(a) shows a marking group with 24 conditions (rows). The concentration of seven signaling proteins were sampled for each condition. This is the heatmap view for the marking group. When a

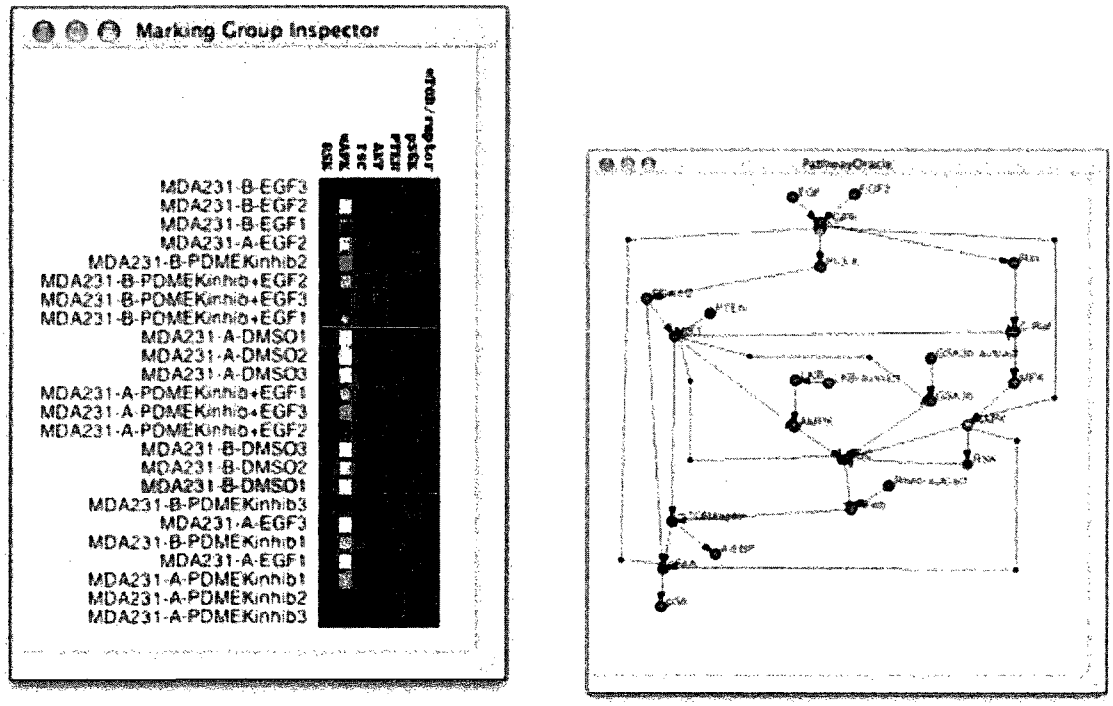


Fig. 5.7: The marking group user interface. (a) The heat map visualization of a marking group. The selected marking, MDA231-B-DMSO1, is highlighted in blue. (b) The color distribution for the selected marking in the group is applied to the network view in the main window. Note that signaling nodes for which values were not given are not assigned a color on the valid red to green spectrum.

specific marking in the group is selected, the colors for that marking are applied to the network view. This is particularly useful when assessing whether the experimental data is consistent with the interactions in the model. In Figure 5.7, the MDA231-B-DMSO2 marking has been superimposed on the network. We can see that RSK has a relatively low concentration despite the high concentration of MAPK. Given that, in the model, RSK is activated by MAPK, this combination of activity-levels seems unlikely to occur. Such an inconsistency suggests that there may be other signaling interactions contributing to the overall activity-level of RSK. Such an insight can help

a researcher quickly identify areas where the model or experimental results need to be re-evaluated or improved.

5.2 Monarch

In this section we discuss the Monarch software system [RN]. The purpose of the Monarch software system is to provide an implementation of the deterministic method described in Chapter 4. Because of the computational tools needed to solve the optimization problem used by the trainer, deploying Monarch as a stand-alone application presents issues both from an installation and a resource standpoint. Thus, Monarch is a web application (located at `http://www.monarchscience.com`) which uses a dedicated server to run training and simulation processes.

This web interface is the major reason that the deterministic simulator has not been incorporated into the PathwayOracle software package. In the remainder of this section, we will discuss the architecture of the back-end system as well as the features supported by the front-end.

5.2.1 Back-end System Architecture

Conceptually, the Monarch system consists of three parts: the training algorithm, the simulation algorithm, and the front-end. With the exception of one 3rd party tool (BONMIN), the entire system is implemented in Python. The training and simulation algorithms constitute the back-end. The front-end allows user interaction with these

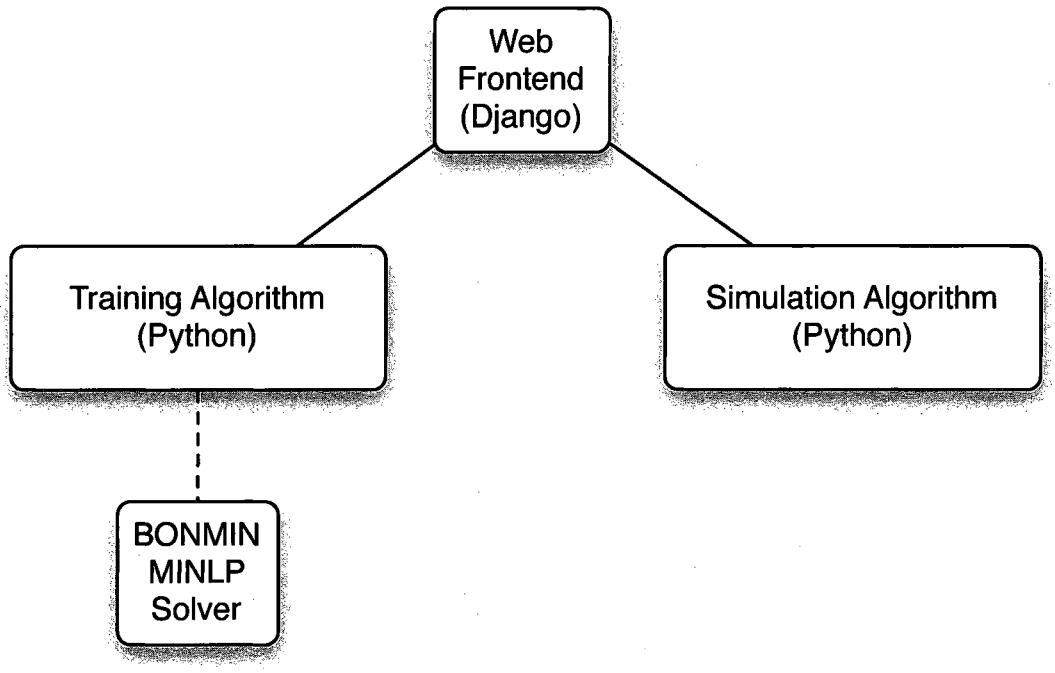


Fig. 5.8: The different components comprising the architecture of the Monarch system. The dashed line connecting the Training Algorithm and the Bonmin MINLP Solver indicates that the Bonmin solver is run on a separate machine dedicated to running MINLP solving jobs.

algorithms.

Training Algorithm

The training algorithm takes a model of a signaling network's connectivity, the number of timesteps for which to parameterize the model, and a set of qualitative constraints on the dynamics of the network. It produces a parameterized model of the signaling network. The parameterized model is the input connectivity with a degradation rate specified for each node and a weight parameter specified for each edge.

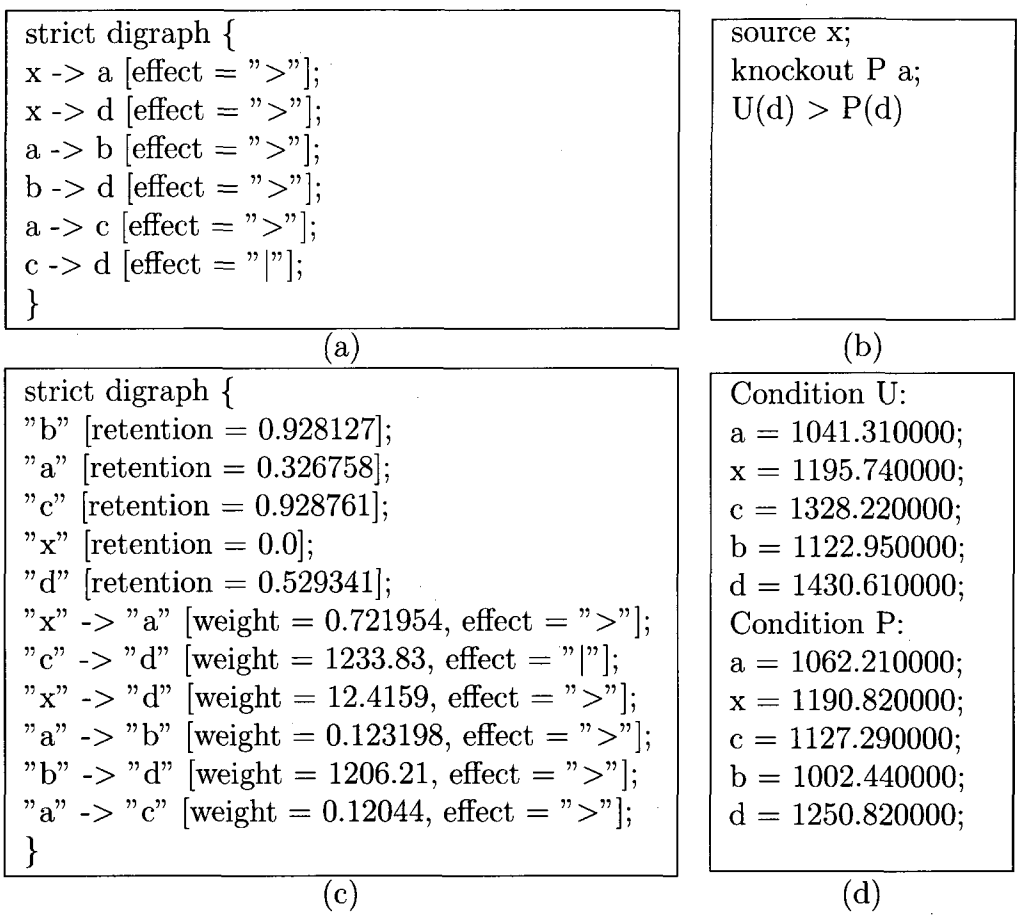
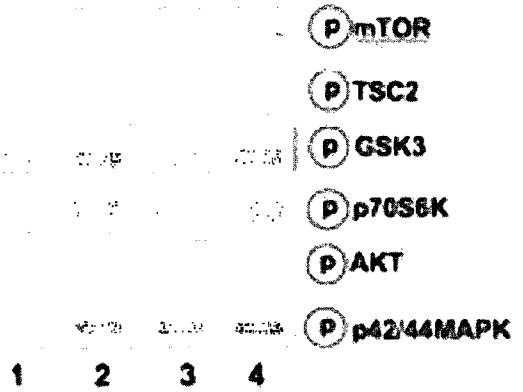


Fig. 5.9: Examples of the input accepted and output produced by the Monarch system: (a) the signaling network's connectivity, (b) the qualitative constraints, (c) the parameterized model, and (d) the initial conditions for each condition included in the training process.

Representing connectivity. As shown in Figure 5.9(a) and (c), the DOT format is used to represent both the model of connectivity as well as the parameterized model [GN99]. The DOT format specifies the properties of one node or one edge on each line. When parameterized, each node has a retention parameter (= 1 − degredation) and each edge has a weight property.



(a)

source EGF;
knockout P TSC2;
U(mTOR) < P(mTOR);
U(GSK3b) = P(GSK3b);
U(p70S6K) < P(p70S6K);
U(AKT) > P(AKT);
U(MAPK) = P(MAPK);

(b)

Fig. 5.10: Example of how qualitative constraints are derived from a western-blot. In this example, the results from the western-blot (a) were used to derive the qualitative constraints (b). The experimental data in (a) was generated by conducting two different experiments (shown in columns 2 and 4). In lane 2, the cell-line was exposed to EGF, which induced the propagation of signal through the EGFR receptor. In lane 4, first a TSC2 inhibitor was applied to the cell-line, followed by EGF. The individual activity-levels of proteins were measured under both conditions. The EGF stimulation was captured by the qualitative constraint "source EGF;" which indicates that the signal will originate from the node labeled "EGF". The TSC2 knockout in the second condition is captured by the the constraint "knockout P TSC2;" indicates that the node labeled "TSC2" has an activity-level of zero under the condition "P". The remaining constraints are derived by comparing lanes 2 and 4. For example, comparing mTOR in lane 2 and 4 reveals that mTOR had less activity in the control condition than in the perturbed condition. Thus: $U(mTOR) < P(mTOR)$.

Representing qualitative constraints. Qualitative constraints, as discussed in Chapter 4, assert the relationship between the activity-level of a protein observed in one condition versus the activity-level of a protein observed in a second condition. Thus, qualitative constraints take the form $C_1(X) \cdot C_2(X)$ where $C_1(X)$ is the activity-level of protein X under condition C_1, $C_2(X)$ is the activity-level of protein X under

condition C_2, and $\cdot \in \{<, >, =\}$ is the relation between the two (less-than, greater-than, or equal, respectively). These constraints can be derived from experiments as shown in Figure 5.10: Figure 5.10(a) shows a set of western-blots and (b) shows the qualitative constraints that are derived from it.

The training algorithm uses the input network connectivity and the qualitative constraints to construct a mixed-integer non-linear programming problem using the formulation discussed in Chapter 4. This problem is then given to the BONMIN solver, a program specifically designed to solve such problems. The results returned by BONMIN specify the (1) protein retention and interaction weight values for the model, (2) the initial condition of the network, and (3) the set of constraints satisfied. These details are then presented back to the user through the front-end.

Simulation Algorithm

The simulation algorithm accepts a parameterized model of the signaling network's connectivity, an initial condition, and a specific number of time steps for which to run the simulation. The simulator itself is implemented entirely in Python. The simulator computes the activity-level of each protein over the course of the simulation and returns these values to the user through the front-end.

5.2.2 Web Front-end

The front-end is responsible for delivering dynamically generated web pages to the user's browser and receiving jobs submitted by the user through these pages. To

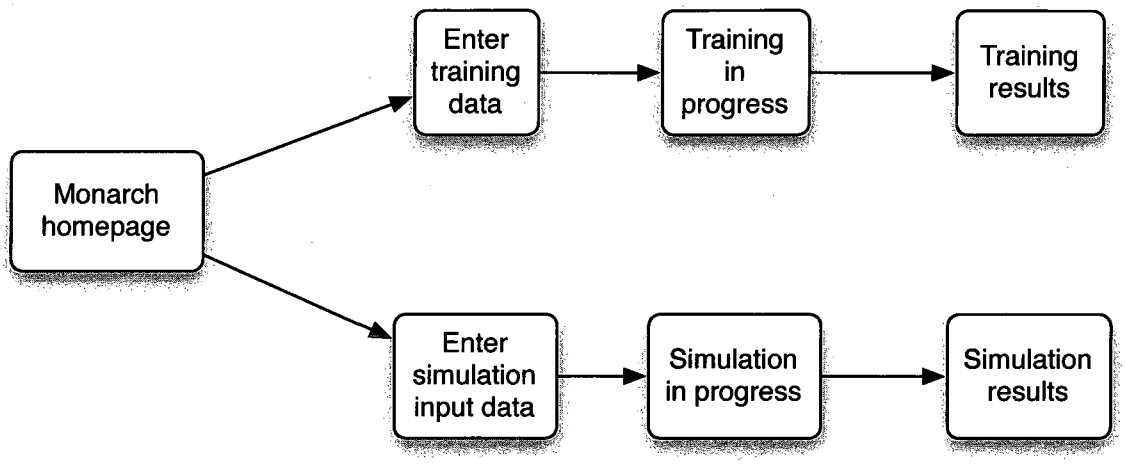


Fig. 5.11: The workflow of the Monarch front-end: (a) the process of training a network model and (b) the process of simulating the model.

achieve this purpose, it was implemented as a Django application, which enables the rapid development of dynamic content webpages.

Training a network model. As shown in Figure 5.11(a), the front-end presents the user with a page on which to specify inputs. After submitting the training request (by clicking the "Train" button, the user sees a "Training in progress" page. When training is complete, this page is replaced with a page showing the results of the training: a parameterized model and the initial conditions for each condition used to train the model. The user is also shown any constraints that are not satisfied by the trained model (indicating that not all constraints could be fit to the connectivity specified).

Simulating a network model. Figure 5.11(b) shows the workflow for simulation of a network model. The user is presented with a page requesting inputs: (1) a

parameterized model of network connectivity, (2) the initial conditions to use as a starting point for the simulation, and (3) the number of time steps for which to run the simulation. After submitting the simulation request (by clicking the "Simulate" button, the user sees a "Simulation in progress" page. When the simulation finishes, this page is replaced with a page showing the results of the simulation: the sum-total of the activity-level of each protein during the course of the simulation.

Chapter 6

Conclusions and Future Work

Unraveling and understanding biochemical networks is a central challenge facing cellular biological and biomedical research with a multitude of implications for bio-engineering, clean energy production, and better treatments for devastating diseases such as cancer. Limitations of current laboratory technology make it difficult to perform large-scale analysis and studies on the biological systems of interest, motivating the need for good computational or mathematical models that can be investigated through other means.

To date, modeling has involved the use of quantitative data to estimate the parameter values for models. While such models can be very accurate, they require significant investments to be made in obtaining the experimental data necessary to build them. Often this experimental can be difficult or impossible to obtain.

In this thesis, we have proposed that qualitative experimental data can be used to build predictive computational models of biochemical network dynamics. While qualitative data has been used extensively for the purpose of reconstructing network connectivity or studying trends in cellular behavior, little work has been done on the question of how qualitative data might be used to actually predict the behavior of biochemical networks. To this end, in this thesis we have presented two novel methods that do exactly this for signaling network dynamics.

The **signaling Petri net** is a method that uses signaling network connectivity as the underlying model. A stochastic simulation method evaluates how signal propagates through this network connectivity over time. Under the networks and experiments we have used thus far, our method shows the ability to predict the correct behavior of the network over 90% of the time. Because the model used has no parameters or weights, it is possible to quickly build models for use with signaling Petri nets from existing maps of signaling networks, from online databases such as KEGG, and from the collected experience of researchers themselves.

The **Monarch system** is a method that also uses signaling network connectivity as the underlying model. Unlike the signaling Petri net, this model also has parameters whose values are trained using qualitative experimental data—data that is much easier to obtain and more readily available from literature and online data repositories. Once parameter values are trained, the model can be used to predict the dynamics of the signaling network under a variety of conditions. Under a broad array of experiment considered here, our method correctly predicts the affect of a network perturbation 85% of the time (60 out of 70 predictions). This degree of accuracy is competitive with existing ODE-based models that must be trained on quantitative data. In general, ODE-based methods require nearly 10 times the amount of training data in order to achieve comparable degrees of accuracy.

We have implemented both of these methods and made them available as the software tools PathwayOracle and Monarch. The goal of this investment in imple-

mentation is to make our methods accessible to experimental and theoretical biologists who are in need of such modeling capabilities.

PathwayOracle is an integrated software environment in which biologists may conduct structural and dynamic analysis of signaling networks of interest. This tools is distinguished from other tools in the field of systems biology by its ability to predict the signal flow through a network using a simplified, connectivity-based model of the signaling network. Simulations are fast and, based on a published study, predictors of signal propagation. This novel simulation capability, combined with support for structural analysis of connectivity between pairs of proteins and for analysis of certain kinds of experimental data make *PathwayOracle* a powerful asset in the experimentalist's endeavor to gain a more complete understanding of the cellular signaling network.

Monarch is a web-based tool that focuses on delivering the ability to quickly build and simulate models of signaling network dynamics from connectivity and qualitative data.

6.1 Future Directions

We consider the success of our methods and tools in predicting the dynamics of signaling networks to be compelling evidence in favor of using qualitative data to build models of biochemical networks. As our work thus far has considered only signaling networks, we identify several directions for future work:

Modeling transcription and metabolism Transcriptional networks are responsible for determining how gene expression levels change over time and respond to various cellular events. Metabolic networks are responsible for managing and regulating the resources available to the cell (e.g., energy and amino acids). As important contributors to overall cellular behavior, it is important to identify ways of building models of these networks using qualitative data.

Integrated models of cellular networks Ultimately, cellular behavior emerges from the interactions among the different biochemical networks present in the cell. Qualitative data is often the only data we have about cell-level events, making the development of methods for building combined models of transcription, signaling, and metabolism from qualitative data important.

Multi-cellular models Though disease is often investigated on the cellular or sub-cellular levels, most diseases involve the failure of cellular systems: whether organ systems or interacting groups of cells (e.g., diabetes results from the failure of pancreas, muscle, fat, and liver cells). Thus, building multi-cellular models of cellular dynamics is an important long-term goal.

6.1.1 PathwayOracle

Our goal is to develop *PathwayOracle* into an integrated and expansive suite of tools that allow the biologist to extract as much information as possible from models of signaling network connectivity and experimental data relating to those models.

We consider future directions for *PathwayOracle* to fall into several categories: network construction, network augmentation, experimental and computational analysis integration, and architecture.

One of the benefits of working with connectivity models of signaling networks is the abundance of databases and other online resources that publish connectivity-level data. Future versions of *PathwayOracle* will have support for querying such databases for connectivity components and, ultimately, for automated connectivity construction based on a set of signaling nodes specified by the user.

Analysis of network connectivity and topology is increasingly relevant to biological research. We intend to expand PathwayOracle's structural analysis features to include the ability to search for and identify motifs in the signaling networks.

Network connectivity can also be inferred from experimental data, which provides another direction for research and development. By using experimental results to identify inconsistencies between experimental results and the current network model, it may be possible for *PathwayOracle* to augment the network with new connectivity based on hints supplied by experimental results. At present only experimental concentration data is supported. However, as experiments produce more information beyond concentration profiles of signaling nodes, we plan to expand the experimental data that *PathwayOracle* can load, visualize, and use as part of network analyses.

Experimental results can also provide computational analysis methods information that can improve their final predictions or decompositions. Taking advantage of

the additional, potentially obfuscated, information present in experimental results to improve the results returned by computational tools is a major goal for future versions of *PathwayOracle.*

A longer term direction for *PathwayOracle* is the integration of transcriptional and metabolic network analysis. In the biological systems of interest, the behavior of any one of these networks is dependent on the characteristics of the other two. As a result, developing a complete understanding of signaling, transcriptional regulation, or metabolism depends in part on integrating knowledge from the others.

Finally, an ongoing priority in the design of *PathwayOracle* is its role as an open platform for the development and deployment of new analytical capabilities by other groups. Currently *PathwayOracle* employes a modular architecture that facilitates easy integration of new functionality. However, in future releases we plan to expose a plugin interface which will make it easier to developers and researchers to develop and deploy tools within PathwayOracle.

6.2 Monarch

The current version of Monarch provides users the ability to train and simulate models of signaling network dynamics through a web interface. Future work on this tool will add or improve on a number of capabilities:

Extend qualitative constraints Currently the language used to specify qualitative constraints only allows three relations between molecule activity-levels in

different conditions. We recognize the need for a more sophisticated language allowing the specification of many other types of qualitative relationships such as scaled relations (e.g., $U(X) < 2 * P1(X)$).

Integration into PathwayOracle At present, Monarch is a web-based application. However, by wrapping all the web-based functions in a simple XML-based API, it will be possible to embed all the front-end capabilities in the stand-alone client PathwayOracle.

Qualitative constraints from experimental data Biologists frequently keep experimental results in spreadsheets. Allowing biologists to directly upload these experimental results as input qualitative constraints for the training algorithm eliminates the time-consuming process of having the biologist convert all their experimental observations into individual qualitative constraints.

Abbreviations

EGF	epidermal growth factor
EGFR	epidermal growth factor receptor 2
Ras	v-Ha-ras Harvey rat sarcoma viral oncogene homolog
c-Raf	v-raf-1 murine leukemia viral oncogene homolog 1
MEK	MAPK1,2 kinase
MAPK1,2	mitogen-activated protein kinase 1,2
RSK	ribosomal protein S6 kinase, 90kDa, polypeptide 1
PI3K	phosphoinositide-3-kinase
PTEN	phosphatase and tensin homolog
PDK1	pyruvate dehydrogenase kinase, isozyme 1
AKT	v-akt murine thymoma viral oncogene homolog 1
LKB1	serine/threonine kinase 11
GSK3β	glycogen synthase kinase 3 β
AMPK	protein kinase, AMP-activated, beta 1 non-catalytic subunit
TSC2	tuberous sclerosis 2
Rheb	Ras homolog enriched in brain
mTOR	Mammalian Target of Rapamycin
p70S6K	ribosomal protein S6 kinase, 70kDa, polypeptide 1
4EBP1	eukaryotic translation initiation factor 4E binding protein 1
ODE	ordinary differential equation
siRNA	small interfering RNA
RB	Retinoblastoma tumor suppressor protein

siRNA	small interfering RNA
PN	Petri Net
SPN	Signaling Petri Net

Bibliography

[AC03] Aldana M and Cluzel P. A natural class of robust networks. Proceedings of the National Academy of Sciences 100(15):8710–8714 (2003)

[AHL+06] Avruch J, Hara K, Lin Y, Liu M, and Long X. Insulin and amino-acid regulation of mTOR signaling and kinase activity through the Rheb GTPase. Oncogene 25(48):6361–6372 (2006)

[Alo07] Alon U. An introduction to systems biology: Design principles of biological circuits. Mathematical and Computational Biology Series (2007)

[APL05] Araujo R, Petricoin E, and Liotta L. A mathematical model of combination therapy using the EGFR signaling network. BioSystems (2005)

[Bai01] Bailey J. Complex biology with no parameters. Nature Biotechnology 19:503–504 (2001)

[BFGH06] Blinov M, Faeder J, Goldstein B, and Hlavacek W. A network model of early events in epidermal growth factor receptor signaling that accounts for combinatorial complexity. BioSystems 83:136–151 (2006)

[BMRT96] Belloni E, Muenke M, Roessler E, and Traverso G. Identification of sonic hedgehog as a candidate gene responsible for holoprosencephaly. Nat Genet 14:353–356 (1996)

[Bra95] Bray D. Protein molecules as computational elements in living cells. Nature 376:307–312 (1995)

[CAS05] Chaves M, Albert R, and Sontag E. Robustness and fragility of boolean models for genetic regulatory networks. Journal of Theoretical Biology 235:431–449 (2005)

[CFR99] Corbit K, Foster D, and Rosner M. Protein Kinase C delta mediates neurogenic but not mitogenic activation of mitogen-activated protein kinase in neuronal cells. Molecular and Cellular Biology 19(6):4209–4218 (1999)

[Cha07] Chaouiya C. Petri net modelling of biological networks. Briefings in Bioinformatics 8(4):210–219 (2007)

[CHC99] Chen T, He H, and Church G. Modeling gene expression with differential equations. In Pacific Symposium on Biocomputing, 29–40 (1999)

[CMW07] Ciliberti S, Martin O, and Wagner A. Robustness can evolve gradually in complex regulatory gene networks with varying topology. PLoS Comput Biol 3(2):e15 (2007)

[CRF05] Chen Y, Rodrik V, and Foster D. Alternative phospholipase D/mTOR survival signal in human breast cancer cells. Oncogene 24:672–679 (2005)

[DA05] David R and Alla H. Discrete, Continuous, and Hybrid Petri Nets. Springer (2005)

[DFM+04] Doi A, Fujita S, Matsuno H, Nagasaki M, and Miyano S. Constructing biological pathway models with hybrid functional Petri nets. In Silico Biology 4(3):271–291 (2004)

[DGH+04] Dejong H, Gouze J, Hernandez C, Page M, Sari T, and Geiselmann J. Qualitative simulation of genetic regulatory networks using piecewise-linear models. Bulletin of Mathematical Biology 66(2):301–340 (2004)

[DGM+06] Dojer N, Gambin A, Mizera A, Wilczynski B, and Tiuryn J. Applying dynamic bayesian networks to perturbed gene expression data. BMC Bioinformatics 7(1):249 (2006)

[dJGBH04] de Jong H, Geiselmann J, Batt G, and Hernandez C. Qualitative simulation of the initiation of sporulation in Bacillus subtilis. Bulletin of Mathematical Biology 66:261–299 (2004)

[EI04a] Eungdamrong N and Iyengar R. Computational approaches for modeling regulatory cellular networks. Trends in Cell Biology 14(12):661–669 (2004)

[EI04b] Eungdamrong N and Iyengar R. Modeling cell signaling networks. Biol Cell 96(5):355–362 (2004)

[EKL+02] Eker S, Knapp M, Laderoute K, Lincoln P, and Talcott C. Pathway logic: Executable models of biological networks. In Electronic Notes in Theoretical Computer Science, vol. 71 (2002)

[FBH+03] Fang Y, Brass A, Hoyle DC, Hayes A, Bashein A, Oliver SG, Waddington D, and Rattray M. A model-based analysis of microarray experimental error and normalisation. Nucleic Acids Research 31(16):e96 (2003)

[FCAB05] Feldman D, Carnes C, Abraham WT, and Bristow MR. Mechanisms of disease: beta-adrenergic receptors—alterations in signal transduction and pharmacogenomics in heart failure. Nature Clinical Practice Cardiovascular Medicine 2:475–483 (2005)

[FH07] Fisher J and Henzinger TA. Executable cell biology. Nat Biotechnol 25(11):1239–1249 (2007)

[FMKT03] Funahashi A, Morohashi M, Kitano H, and Tanimura N. CellDesigner: a process diagram editor for gene-regulatory and biochemical networks. Biosilico 1:159–162 (2003)

[FPHS05] Fisher J, Piterman N, Hubbard E, and Stern M. Computational insights into Caenorhabditis elegans vulval development. Proceedings of the National Academy of Sciences 102(6):1951–1956 (2005)

[GBP06] Gianchandani E, Brautigan D, and Papin J. Systems analyses characterize integrated functions of biochemical networks. Trends in Biochemical Sciences 31(5):284–291 (2006)

[GK73] Glass L and Kauffman S. The logical analysis of continuous, non-linear biochemical control networks. J Theor Biol 39:103–129 (1973)

[GN99] Gansner ER and North SC. An open graph visualization system and its applications to software engineering. Software Practice and Experience 00(S1):1–5 (1999)

[Gor99] Goryanin I. Mathematical simulation and analysis of cellular metabolism and regulation. Bioinformatics (1999)

[GSBH07] Grimbs S, Selbig J, Bulik S, and Holzhütter H. The stability and robustness of metabolic states: identifying stabilizing sites in metabolic networks. Mol Syst Biol 3:146 (2007)

[HF96] Huang C and Ferrell J. Ultrasensitivity in the mitogen-activated protein kinase cascade. Proceedings of the National Academy of Sciences 93:10078–10083 (1996)

[HFS+03] Hucka M, Finney A, Sauro H, Bolouri H, and Doyle J. The systems biology markup language (SBML): a medium for representation and exchange of biochemical network models. Bioinformatics 19(4):524–531 (2003)

[HR04] Hardy S and Robillard P. Modeling and simulation of molecular biology systems using Petri nets: modeling goals of various approaches. J Bioinform Comput Biol 2(4):619–637 (2004)

[HSG+06] Hoops S, Sahle S, Gauges R, Lee C, and Pahle J. COPASI - a COmplex PAthway SImulator. Bioinformatics 22:3067–3074 (2006)

[Hun00] Hunter T. Signaling—2000 and beyond. Cell 100(1):113–127 (2000)

[HW00] Hanahan D and Weinberg R. The hallmarks of cancer. Cell 100(1):57–70 (2000)

[IB89] Iyengar R and Birnbaumer L. G Proteins. Academic Press (1989)

[ICG05] Inoki K, Corradetti MN, and Guan KL. Dysregulation of the TSC-mTOR pathway in human disease. Nat Genet 37(1):19–24 (2005)

[IOZ+06] Inoki K, Ouyang H, Zhu T, Lindvall C, and Wang Y. TSC2 integrates Wnt and energy signals via a coordinated phosphorylation by AMPK and GSK3 to regulate cell growth. Cell 126(5):955–968 (2006)

[ITMB07] Iyengar M, Talcott C, Mozzachiodi R, and Baxter D. Executable symbolic modeling of neural processes. In NETTAB (2007)

[JBS99] Joneson T and Bar-Sagi D. Suppression of Ras-induced apoptosis by the Rac GTPase. Molecular and Cellular Biology 19(9):5892–5901 (1999)

[JLI00] Jordan J, Landau E, and Iyengar R. Signaling networks: The origins of cellular multitasking. Cell 103(2):193–200 (2000)

[KAG+08] Kanehisa M, Araki M, Goto S, Hattori M, Hirakawa M, Itoh M, Katayama T, Kawashima S, Okuda S, Tokimatsu T, and Yamanishi Y. KEGG for linking genomes to life and the environment. Nucleic Acids Research 36:D480–D484 (2008)

[Kau69] Kauffman SA. Metabolic stability and epigenesis in randomly constructed genetic nets. Journal of Theoretical Biology 22:437–467 (1969)

[KB05] Klemm K and Bornholdt S. Topology of biological networks and reliability of information processing. Proceedings of the National Academy of Sciences 102:18414–18419 (2005)

[KBP+08] Kim J, Bates DG, Postlethwaite I, Heslop-Harrison P, and Cho KH. Linear time-varying models can reveal non-linear interactions of biomolecular regulatory networks using multiple time-series data. Bioinformatics 24(10):1286–1292 (2008)

[KC08] Kwon YK and Cho KH. Quantitative analysis of robustness and fragility in biological networks based on feedback dynamics. Bioinformatics 24(7):987–994 (2008)

[KCCR04] Karbowniczek M, Cash T, Cheung M, and Robertson G. Regulation of B-Raf kinase activity by tuberin and Rheb is mTOR-independent. Journal of Biological Chemistry 279(29):29930–29937 (2004)

[KM05] Kwiatkowski D and Manning B. Tuberous sclerosis: a GAP at the crossroads of multiple signaling pathways. Human Molecular Genetics 14(Review Issue 2):R251–R258 (2005)

[KPST04] Kauffman S, Peterson C, Samuelsson B, and Troein C. Genetic networks with canalyzing boolean rules are always stable. Proceedings of the National Academy of Sciences 101(49):17102–17107 (2004)

[KR05] Kriel DP and Russel RR. There is no silver bullet - a guide to low-level data transforms and normalization methods for microarray data. Briefings in Bioinformatics 6(1):86–97 (2005)

[KSRG07] Klamt S, Saez-Rodriguez J, and Gilles E. Structural and functional analysis of cellular networks with CellNetAnalyzer. BMC Syst Biol 1:2 (2007)

[KSRLS06] Klamt S, Saez-Rodriguez J, Lindquist J, and Simeoni L. A methodology for the structural and functional analysis of signaling and regulatory networks. BMC Bioinformatics 6:56 (2006)

[LAA06] Li S, Assmann SM, and Albert R. Predicting essential components of signal transduction networks: a dynamic model of guard cell abscisic acid signaling. Plos Biol 4(10):e312–e328 (2006)

[LSG+06] Li C, Suzuki S, Ge Q, Nakata M, and Matsuno H. Structural modeling and analysis of signaling pathways based on Petri nets. Journal of Bioinformatics and Computational Biology 4(5):1119–1140 (2006)

[LSX+07] Liang J, Shao S, Xu Z, Hennessy B, and Ding Z. The energy sensing LKB1-AMPK pathway regulated p27(kip1) phosphorylation mediating the decision to enter autophagy or apoptosis. Nature Cell Biology 9(2):218–224 (2007)

[LW08] Liu PK and Wang FS. Inference of biochemical network models in S-system using multiobjective optimization approach. Bioinformatics 24(8):1085–1092 (2008)

[Mar08] Martinez A. Preclinical efficacy on GSK-3 inhibitors: Towards a future generation of powerful drugs. Med Res Rev 28(5):773–796 (2008)

[MCEB+05] Ma L, Chen Z, Erdjument-Bromage H, Tempst P, and Pandolfi PP. Phosphorylation and functional inactivation of TSC2 by Erk implications for tuberous sclerosis and cancer pathogenesis. Cell 121(2):179–193 (2005)

[MK98] Mendes P and Kell DB. Non-linear optimization of biochemical pathways: applications to metabolic engineering and parameter estimation. Bioinformatics 14(10):869–883 (1998)

[MLLA05] Manning B, Logsdon M, Lipovsky A, and Abbott D. Feedback inhibition of Akt signaling limits the growth of tumors lacking Tsc2. Genes & Development 19(15):1773–1778 (2005)

[MOMR08] Muller M, Obeyesekere M, Mills GB, and Ram PT. Network topology determines dynamics of the mammalian MAPK1,2 signaling network: bifan motif regulation of C-Raf and B-Raf isoforms by FGFR and MC1R.

The Journal of the Federation of American Societies for Experimental Biology 22:1393–1403 (2008)

[MTA+03] Matsuno H, Tanaka Y, Aoshima H, Doi A, and Matsui M. Biopathways representation and simulation on hybrid functional Petri net. In Silico Biology 3(3):389–404 (2003)

[MVG+02] Makris C, Voisin L, Giasson E, Tudan C, and Kaplan D. The Rb-family protein p107 inhibits translation by a PDK1-dependent mechanism. Oncogene 21(51):7891–7896 (2002)

[NCF+06] Neve R, Chin K, Fridlyand J, Yeh J, and Baehner F. A collection of breast cancer cell lines for the study of functionally distinct cancer subtypes. Cancer Cell 10(6):515–527 (2006)

[NDMM03] Nagasaki M, Doi A, Matsuno H, and Miyano S. Genomic object net: I. a platform for modelling and simulating biopathways. Appl Bioinformatics 2(3):181–184 (2003)

[NI02] Neves S and Iyengar R. Modeling of signaling networks. Bioessays (2002)

[NWN+08] Nelander S, Wang W, Nilsson B, She QB, Pratilas C, Rosen N, Gennemark P, and Sander C. Models from experiments: combinatorial drug perturbations of cancer cells. Mol Syst Biol 4:11 (2008)

[ORS+06] O'Reilly K, Rojo F, She Q, Solit D, and Mills G. mTOR inhibition induces upstream receptor tyrosine kinase signaling and activates Akt. Cancer Research 66(3):1500–1508 (2006)

[PP04] Papin J and Palsson B. The JAK-STAT signaling network in the human B-Cell: An extreme signaling pathway analysis. Biophysical Journal 87:37–46 (2004)

[PPW+03] Papin J, Price N, Wiback S, Fell D, and Palsson B. Metabolic pathways in the post-genome era. Trends in Biochemical Sciences 28(5):250–258 (2003)

[PRA05] Peleg M, Rubin D, and Altman R. Using Petri net tools to study properties and dynamics of biological systems. Journal of the American Medical Informatics Association 12(2):181–199 (2005)

[pyt] Official website for the Python programming language. URL `http://www.python.org`

[RMT+08] Ruths D, Muller M, Tseng JT, Nakhleh L, and Ram PT. The signaling Petri net-based simulator: A non-parametric strategy for characterizing the dynamics of cell-specific signaling networks. PLoS Comput Biol 4(2):e1000005 (2008)

[RN] Ruths D and Nakhleh L. Deriving predictive models of signaling network dynamics from qualitative experimental data. Mol Syst Biol Submitted

[RNR08] Ruths D, Nakhleh L, and Ram P. Rapidly exploring structural and dynamic properties of signaling networks using PathwayOracle. BMC Syst Biol 2:76 (2008)

[RSSR09] Ritz A, Shakhnarovich G, Salomon AR, and Raphael BJ. Discovery of phosphorylation motif mixtures in phosphoproteomics data. Bioinformatics 25(1):14–21 (2009)

[SAS05] Sarbassov D, Ali S, and Sabatini D. Growing roles for the mTOR pathway. Current Opinion in Cell Biology 17(6):596–603 (2005)

[SBSW07] Steggles L, Banks R, Shaw O, and Wipat A. Qualitatively modelling and analysing genetic regulatory networks: a Petri net approach. Bioinformatics 23(3):336–343 (2007)

[SCY+08] She QB, Chandarlapaty S, Ye Q, Lobo J, Haskell KM, Leander KR, DeFeo-Jones D, Huber HE, and Rosen N. Breast tumor cells with PI3K mutation or HER2 amplification are selectively addicted to AKT signaling. PLoS ONE 3(8):e3065 (2008)

[SHK06] Sackmann A, Heiner M, and Koch I. Application of Petri net based analysis techniques to signal transduction pathways. BMC Bioinformatics 7:482–498 (2006)

[SJ06] Schmidt H and Jirstrand M. Systems Biology Toolbox for MATLAB: a computational platform for research in systems biology. Bioinformatics 22(4):514–515 (2006)

[SMO+03] Shannon P, Markiel A, Ozier O, Baliga N, and Wang J. Cytoscape: A software environment for integrated models of biomolecular interaction networks. Genome Res 13(11):2498–2504 (2003)

[SV06] Steinhoff C and Vingron M. Normalization and quantification of differential expression in gene expression microarrays. Briefings in Bioinformatics 7(2):166–177 (2006)

[SVL+92] Sozeri O, Vollmer K, Liyanage M, Firth D, Kour G, Mark GE, and Stabel S. Activation of the c-Raf protein kinase by protein kinase C phosphorylation. Oncogene 7(11):2259–2262 (1992)

[THHD07] To T, Henson M, Herzog E, and Doyle F. A molecular model for intercellular synchronization in the mammalian circadian clock. Biophysical Journal 92:3792–3803 (2007)