Pathway Models Decomposition and Composition Techniques for Parameter Estimation

Koh Yeow Nam, Geoffrey

A Thesis submitted for the degree of
Doctor of Philosophy

NUS Graduate School for Integrative Sciences and Engineering
National University of Singapore

2008

Abstract

A biological pathway is a complex system of interactions between the biochemical molecules in a cell. Due to these interactions, the state of the cell and the molecular concentration of its elements evolve dynamically with time to perform various functions. The ability to model and analyze the *dynamics* of such interactions is crucial to the understanding of the mechanisms underlying cellular functions and behavior.

Modeling the dynamics of biological pathways is usually done through the following steps: (1) inferring or identifying the participating molecules and their reactions, (2) forming the rate equations that determine the rate of the reactions, and (3) estimating the unknown rate parameters by fitting the model with experimental data. This approach is often hindered by lack of information about the parameters as well as the sparseness and noisiness of experimental data.

This thesis presents a principled approach to pathway modeling, with a focus on the parameter estimation problem for large pathway models. We decompose a large pathway model into smaller, possibly overlapping components. This approach greatly reduces the dimension of the parameter search space. In the initial approach, parameter estimation is carried out in a non-deterministic order on the individual smaller components. We show that applying this method on the Akt-MAPK signaling pathway achieves a significant improvement in computational time required, as compared to conventional techniques where the entire pathway is processed as a single component.

For large pathway models, their sub-modules may be elucidated and constructed independently. These sub-modules may overlap one another in their participating molecules and rate parameters. Similarly, pathway decomposition for the purpose of parameter estimation may also result in the components sharing pathway elements. Thus, the estimates for these parameters may conflict with one another across the different components. We address this issue by associating a *factor graph* - a probabilistic graphical model - with the pathway model. Then, after representing the parameter values as probability distributions, we use *belief propagation* - a message passing algorithm - to propagate constraints across the various components and reconcile their parameter estimates.

In addition to pathway composition and reconciliation, our approach can also be applied to integrate new experimental data in a systematic manner; without the need to do parameter estimation from scratch when additional experimental data becomes available.

We validate the feasibility of our approach by applying it on a synthetic model based on the Akt-MAPK signaling pathway. This pathway has 36 molecule species and 42 unknown parameters. We decompose the pathway into smaller components and build their corresponding factor graphs. Upon composition, the local dependencies are propagated globally using the *composite factor graph* to obtain the *maximum a posteriori* distributions of the parameters. We also apply our method to incorporate 5 new sets of experimental data in an incremental fashion, and demonstrate that the quality of our solutions is not compromised despite the fact that we do not re-estimate the parameters from scratch.

Acknowledgments

Firstly I would like to express my sincerest gratitude to my supervisors Professor P.S. Thiagarajan and A/Professor David Hsu. Their constant guidance over the past four years were instrumental in developing the ideas presented in this thesis. I am especially thankful to them for encouraging me to look further when extending my candidature for another semester. In doing so, the quality of the results in this thesis has improved considerably. They have also gone the extra mile of providing me with the necessary funding and support during that period so that I can concentrate on writing my thesis. In addition, they have also taught me several other aspects on the art of doing research.

I would also like to thank Dr Lisa Tucker-Kellogg, whom I have worked closely with over the past two years. Her conversations with me have always provided me with new ideas and inspirations, and she has been extremely supportive as a mentor. Special thanks to my collaborators A/Professor Marie-Véronique Clément and Teong Huey Fern from the Department of Biochemistry, NUS. They have been very patient in guiding me during my short stint with them, performing wet-lab experiments.

Lastly, I would like to thank my friends in NUS - Junxiang, Chu Ming, Calvin, Donny, Tsung-Han, Liubing - for breaking the monotony of school life with coffee-break interruptions and occasional solitiare showdown challenges.

I wish to dedicate this thesis to my family and my girlfriend Cindy, for their understanding and confidence in me over the past few years. I would not have reached where I am without their love and encouragement.

Contents

1 Introduction **1**

1.1 Context and Motivation 2

1.1.1 Pathway Structure 2

1.1.2 Rate Parameters 3

1.1.3 The Compositionality Challenge 4

1.2 Our Approach 5

1.2.1 Parameter Estimation Techniques 5

1.2.2 The Decomposition Technique 6

1.2.3 The Composition Technique 7

1.2.4 Data Integration 8

1.3 Overview 8

2 Background and Related Work **9**

2.1 Bio-Pathways 11

2.1.1 Gene Regulatory Networks 11

2.1.2 Metabolic Pathways 12

2.1.3 Signaling Pathways 14

2.2 Pathway Modeling 16

2.2.1 System of Ordinary Differential Equations 16

2.2.2 Boolean Networks 22

2.2.3 Petri Nets 23

2.2.4 Hybrid Functional Petri Nets 24

2.2.5 Hybrid Automata 27

2.3 Estimating the Pathway Parameters 29

2.3.1 Deterministic Algorithms 31

2.3.2 Stochastic Algorithms 32

3 Pathway Model Decomposition **35**
3.1 Components . . . 35
3.2 Decompositional Approach to Parameter Estimation . . . 39
3.2.1 Decomposition . . . 40
3.2.2 Grouping of the Components . . . 40
3.3 Case Study . . . 45
3.3.1 The Akt Pathway . . . 45
3.3.2 The HFPN Model of the Akt-MAPK Pathway . . . 48
3.3.3 Results . . . 52
3.4 Discussion . . . 55

4 Composition and Reconciliation **56**
4.1 Pathway Composition . . . 58
4.2 Factor Graphs . . . 58
4.2.1 Factor Graph representation of Biological Pathways . . . 59
4.3 The Compatibility Function . . . 62
4.3.1 Local Scores and Global Scores . . . 62
4.4 Parameter Beliefs . . . 64
4.5 Belief Propagation . . . 64
4.5.1 Belief Propagation on Factor Graphs . . . 66
4.5.2 Loopy Belief Propagation . . . 67
4.5.3 Max-Product and Sum-Product . . . 68
4.6 Composition . . . 69
4.7 Data Integration . . . 69
4.8 A Small Case Study: Branching Signaling Pathway . . . 72
4.8.1 Results and Discussion . . . 73

5 An Integrated Case Study **77**
5.1 Pathway Model . . . 77
5.1.1 Minimal Components . . . 79
5.1.2 Factor Graph Representation . . . 80
5.2 Extensions to the Algorithm . . . 82
5.2.1 Representing Traces . . . 82
5.2.2 Directed Sampling of the Parameter Space . . . 83
5.3 Data Integration . . . 84
5.4 Results and Discussion . . . 86

6 Conclusion **89**
6.1 Our Contributions 89
6.2 Future Work 90
6.3 Composition and Updating Pathway Models 92

A Supplement Data for Chapter 3 **94**
A.1 Knockdown Experiments 95
A.2 Treatment Experiments 99
A.3 Time Series Data 105
A.4 Validation Data Set 109

B Supplement Data for Chapter 5 **112**

List of Figures

2.1 Central Dogma of Molecular Biology 10
2.2 Gene Network for S. Cerevisiae Mating Response 12
2.3 Glycolysis Pathway 13
2.4 Signaling Pathways in Baker's Yeast 15
2.5 Michaelis Menten Kinetics 21
2.6 Example of a Boolean network 22
2.7 Basic elements of a Petri net 24
2.8 Example of a Petri net model 25
2.9 Additional elements for an HFPN model 25
2.10 HFPN representation of a biochemical reaction 26
2.11 Hybrid automata model of an excitable cell 28

3.1 Component 37
3.2 Minimal Component 40
3.3 Component Graph 43
3.4 Grouping and Ordering of Components 44
3.5 Converting a cyclic component graph to an acyclic graph 46
3.6 Steady-state western blot images of the observed phenomena 48
3.7 Schema of the Akt-MAPK signaling pathways 49
3.8 HFPN model of the Akt and MAPK pathways 50
3.9 Comparison of simulation results 53
3.10 MEK and ERK profiles 53
3.11 Bad profile with DPI treatment 54

4.1 Overlapping Components 57
4.2 Simple factor graph example 59
4.3 Simple pathway conversion to factor graph 60

4.4 Connecting the factor graphs 61
4.5 Belief propagation example 65
4.6 Sum-product and Max-product 68
4.7 Pathway composition 70
4.8 Data integration 71
4.9 Branching signaling pathway 72
4.10 Pathway component and factor graph 74

5.1 The Akt-MAPK Pathway 78
5.2 Component Graph 80
5.3 Factor Graph of the individual Components 81
5.4 Discretizing a molecule trace 83
5.5 Average Performance of the Algorithms 87

List of Tables

3.1 Rate reactions and their parameters 51

4.1 Results for simulation 75

5.1 Rate equations 79
5.2 Initial Conditions 85
5.3 Belief Propagation in Pathway Composition and Data Integration . . . 86
5.4 Comparison of evaluation runs 87

A.1 Control experiment 95
A.2 Treatment with 20 nM Akt siRNA 96
A.3 Treatment with 50 nM Akt siRNA 96
A.4 Treatment with 100 nM Akt siRNA 97
A.5 Treatment with 20 nM PDK1 siRNA 97
A.6 Treatment with 50 nM PDK1 siRNA 98
A.7 Treatment with 100 nM PDK1 siRNA 98
A.8 Control experiment 99
A.9 Treatment with PP2A siRNA 100
A.10 No Serum 100
A.11 Treatment with PP2A siRNA and no Serum 101
A.12 Treatment with DPI 101
A.13 Treatment with DPI and PP2A siRNA 102
A.14 Treatment with LY297002 102
A.15 Treatment with LY297002 and PP2A siRNA 103
A.16 Treatment with DPI and PP2A siRNA 103
A.17 Treatment with DPI, LY294002 and PP2A siRNA 104
A.18 Control experiment 105
A.19 Time series data with no Serum 106

A.20 Time series data with LY294002 treatment 106
A.21 Time series data with LY294002 treatment and no Serum 107
A.22 Time series data with DPI treatment 107
A.23 Time series data with DPI treatment and no Serum 108
A.24 Control experiment 109
A.25 Validation time series data with no Serum 110
A.26 Validation time series data with LY294002 treatment 110
A.27 Validation time series data with LY294002 treatment and no Serum . . 111

B.1 Result for BP 112
B.2 Result for SRES 113
B.3 Result for GA 113

Chapter 1

Introduction

Conventional approaches in molecular biology involve the identification of individual elements and understanding their primary interactions. However, it is increasingly recognized that detailed studies of the constituents alone cannot satisfactorily explain the many biological phenomena that arise in the cell (Nurse, 2003). Systems biology brings us a step further by focusing on understanding how all these elements function together to bring about *systemic* properties.

The systems view of molecular biology has given rise to a *network-centric* approach. Biological pathways - the molecule species and their interactions - are often represented as complex graph-like structures. In these structures, the nodes typically represent the molecule species and the edges represent chemical interactions. It not only provides a visually appealing way for researchers to record their knowledge about the pathways and their mechanisms, it also allows topological network analyses where system level properties such as redundancy in output signals and pathway robustness to multiple inputs can be studied (Papin and Palsson, 2004). There are several free online repositories that store pathway models, such as KEGG[1] (Ogata et al., 1999) and BioCarta[2].

Modeling of the biological pathways and their analyses are increasingly gaining the attention of researchers from fields such as mathematics, physics, computer science and engineering. The Davidson Laboratory at the California Institute of Technology focuses on building *cis*-regulatory networks that control endomesoderm specification in the sea urchin embryo (Bolouri and Davidson, 2002; Davidson et al., 2002). Some groups try to infer the entire gene network from microarray gene expression data using probabilistic graphical models such as Bayesian networks (Friedman, 2004; Kim et al.,

[1] http://www.genome.jp/kegg/
[2] http://www.biocarta.com

2003) and information-theoretic methods like the Relevance networks (Margolin et al., 2006). From the control theorists' and computer scientists' perspective, understanding the control mechanisms and qualitative analyses are of interest. They use models of computation such as Boolean networks (Raeymaekers, 2002; Shmulevich et al., 2002), Petri nets and their variants (Matsuno et al., 2006), and hybrid automata (Mishra and Policriti, 2004; Ye et al., 2006) to represent and simulate the pathways.

With the aid of breakthroughs in high data throughput technology such as DNA microarrays and laser scan cytometry (Pozarowski et al., 2006), the systems approach can potentially yield valuable insights into complex biological phenomena such as embryonic stem cell differentiation (Keller, 2005), inflammation, homeostasis and apoptosis (Jacobson and McCarthy, 2002). The understanding of the mechanisms behind these phenomena can yield important benefits such as the identification of effective drug targets. One such target that has been identified is the enzyme *transkelotase*, which is a key enzyme in the regulation of ribose synthesis in tumor cells (Cascante et al., 2002). Other applications include directed differentiation in stem cell therapy, where multipotent progenitor cells can be coaxed into specific lineages (cell types) for the treatment of cardiac diseases (Segers and Lee, 2008).

While systems biology is the overall framework for understanding cell functions through a holistic systems approach, the use of computational methods and algorithms to perform such studies is becoming prevalent, forming the subfield - *Computational Systems Biology* (Kitano, 2002). In this thesis, our focus is on the latter and we shall refer to it simply as *systems biology*.

1.1 Context and Motivation

In order to gain a systems-level understanding of biological functions, one requires mathematical models that can capture not only the structure of the pathways, but also the intra- and intercellular *dynamics*. These dynamics describe the spatial and temporal changes of molecular concentrations due to the various reactions in the cell.

The forms of the reactions depend on the resolution of the model and the level of detail required for analysis. Regardless of the formalisms used, dynamical modeling of biological pathways throws us several challenges which we need to address.

1.1.1 Pathway Structure

One of the challenges faced by researchers is the incomplete information on the pathway structure. The most common approach to constructing these pathways is to curate

them manually from various literature sources. For instance, a detailed pathway map of the epidermal growth factor receptor signaling pathway has been put together by an extensive literature search involving 242 papers accessible from PubMed[3]. This pathway contains 211 reactions and 322 molecular species (Oda et al., 2005). Although it is currently recorded as a static map, it can be a comprehensive source of reference for further dynamical studies.

Advances in high throughput technology have also been instrumental in helping to identify the key players and their interactions in biological pathways. For example, Kel et al. (2005) have utilized microarray data, together with sequence analysis to reveal the key transcription factors in signaling pathways such as the T-cell antigen receptor pathway. Given the identity of these transcription factors, further experiments can then be carried out to elucidate their roles and functions in the relevant pathways, using which researchers can study their impact on the dynamics of the system.

Some researchers have attempted to use genome-wide time-series microarray data to *infer* the structure of the underlying pathways. However, these efforts are limited by the low sensitivity of the algorithms. A review of the common inference algorithms has shown that on a nine-gene sub-network of the SOS pathway in *Escherichia coli* with 6 sets of time-series data, the highest sensitivity (i.e. percentage of interactions recovered) is obtained using the Bayesian network formalism. Even so, only 69% of the true edges were recovered (Bansal et al., 2007). Moreover, this technique of pathway inference is targeted towards the reconstruction of gene regulatory networks, and not other types of pathways. Also, it does not truly *reverse-engineer* the mechanistic reactions of the pathways. Rather, it attempts to deduce causal or correlational relationships between the pathway elements using experimental data.

1.1.2 Rate Parameters

To enable the dynamical simulation of biological pathways, apart from knowing the pathway structure and the forms of the various reactions, one also requires the values of the parameters. These parameters can be rate constants for elementary reactions such as association and dissociation constants, or they may be specific to certain biochemical kinetic models such as the Michaelis constants and Hill coefficients (Klipp et al., 2005). Slight deviations in some of the parameter values may result in widely varying model responses, such as the dampening of oscillating signals and signal amplification. Hence, obtaining the correct values for the parameters plays a crucial role in the modeling and

[3]http://www.ncbi.nlm.nih.gov/PubMed/

analysis of pathways.

Very often, the values of these parameters are unknown, and have to be obtained through *in vitro* measurements, or by collectively fitting them to experimental data. Direct measurements of the parameters *in vitro* involve isolating the relevant molecule species, followed by measuring the rates of their reactions in controlled environments. This method requires huge amounts of effort, due to the large number of parameter values to be determined, especially in eukaryotic cells. Furthermore, some of the compounds are unstable, for instance the intermediary complexes of enzyme-catalyzed reactions, thus making measurements difficult. A major criticism of using *in vitro* parameters is that they do not accurately represent the actual physiological conditions in the cell. This discrepancy has been shown in the yeast glycolytic pathway (Teusink et al., 2000). Using parameters measured *in vitro*, only half of the simulated concentration profiles can match *in vivo* experimental data.

The use of parameter estimation algorithms to fit the model to *in vivo* experimental data must also address its own set of challenges. In particular, for signaling pathways, most of the experimental data are in the form of Western blot images. These data are noisy and qualitative in nature. Fitting the model to such data may result in inaccuracy of the estimated parameter values. For large pathway models, there is the issue of a high dimensional search space, which makes parameter estimation a computationally intensive task (Koh et al., 2006).

1.1.3 The Compositionality Challenge

A recurring theme in most research endeavors in systems biology is that more pathways are being studied together in order to understand their collective influences on one another. Construction of these pathway models should be carried out in an iterative and integrative manner. Recent analyses on cross-talk interactions between different pathways have also played a major role in this integrative approach, where different pathways are now being *composed* together to form larger ones. Hence, there should be means to extend the pathways whenever new elements or reactions are discovered (Gadkar et al., 2005). Structurally, composing pathways together is straight forward, involving only the adding of edges and nodes to the extended pathway model. However, ensuring the consistency of the pathway dynamics with the new elements and rate parameters is much more challenging.

Within a pathway model, the parameter estimates are seldom universally correct. This is because current data sets are never complete, and are usually replete with technical and biological noise. Instead, one can estimate the parameters for the pathway

model using currently available data, and update them when there is new data. This new data can originate from various research groups with different measuring techniques. Hence a principled method to iteratively refine existing model parameters is crucial in pathway modeling.

1.2 Our Approach

To facilitate large-scale modeling of dynamical pathway models and integration of data, we present a modular approach to pathway modeling, with an emphasis on parameter estimation.

Our approach can be broadly divided into four steps: (i) exploiting the pathway topology and locations of the available experimental data to break it down into smaller, possibly overlapping components, (ii) estimating the parameters of the components, (iii) systematically composing the smaller pathways together while ensuring that the parameter estimates are globally consistent with experimental data and finally, (iv) fine-tuning the estimates by integrating information from new experimental data as it becomes available.

The methods presented in this thesis can be extended and applied to other types of biological pathways, as long as their structures can be captured by appropriate graphical formalisms such as Petri nets, and their dynamics can be described by differential equations. However, we will focus primarily on *signaling pathways* due to their inherent modular structure (Klipp and Liebermeister, 2006), which makes them more amenable to decomposition.

1.2.1 Parameter Estimation Techniques

As mentioned previously, a major obstacle in biological pathway modeling is the determination of the large number of unknown rate parameters.

Parameter estimation involves minimizing a certain objective function while traversing the parameter search space. Several algorithms have been developed to perform this minimization, given experimental data. There are different ways of defining the correctness of the estimated parameters. Gonzalez et al. (2006) assess the strengths of *Simulated Annealing* - a common algorithm employed in such problems - by a direct comparison of the recovered parameters with their nominal values. In contrast, Mendes and Kell (1998) acknowledge that there will be more than one possible set of parameters. They use an objective function as the basis for comparison instead. This objective function is defined as the amount of deviation of the simulated concentration profiles

produced using the estimated parameters as compared to experimental data. In this thesis, our notion of suitable parameters follows the latter, i.e. the parameter combination that produces simulated concentration profiles that best matches the experimental data.

One is also interested in the end products of the estimation process, which are the values of the parameters. In a conventional approach, each parameter takes on a point value in the real number space. However, after obtaining the set of estimated parameters, what does one do when newer, and possibly more accurate, data is made available? Furthermore, due to the *multi-modal nature* of the problem there might be more than one possible values for a parameter. Moreover, the pathway model might not even be sensitive to some parameters, i.e. the system output remains relatively unchanged when those parameters vary over several orders of magnitude.

Hence, a point based representation of the parameters is somewhat unrealistic although it is needed for simulating the pathway model. In this thesis, we adopt a probabilistic representation where a parameter is viewed as a probability distribution over its range of values. Each distribution describes the current state of knowledge about the parameter. The higher the probability of a parameter value, the more likely that assignment will produce a model output that fits data well. Conversely, uncertainties in the parameter will translate into a *flatter* distribution, indicating that either the pathway is not sensitive to that parameter, or there is not enough information to constrain its values. A key advantage of this representation is that the parameter estimates can be updated by performing a *pointwise product* of their *a priori* distributions with the distributions obtained from fresh experimental data.

1.2.2 The Decomposition Technique

For most estimation algorithms, a common requirement is the ability of the pathway model to be *simulated* using current parameter guesses. The model output can then be compared with experimental data to assess the correctness of the guesses. As it turns out, sub-networks can be identified such that, given appropriate input signals, they can be simulated independently. This feature allows parameter estimation to be carried out on a component-level for the relevant parameters. The presence of experimental data (especially time series data) can also *decouple* certain pathway reactions. For instance, the absence of an active enzyme can serve to dissociate upstream with downstream reactions.

Thus, our approach for decomposition involves exploiting the pathway topology, as well as locations of experimental data to identify the components of the pathway which

can be simulated independently. These components may not be mutually exclusive, i.e. they may share pathway elements, and some components may be embedded in others.

Decoupling of pathway models has also been explored, although for different purposes by Bourqui et al. (2007) and Holme et al. (2003). Most works on pathway decomposition are concerned with the structural analyses on pathway topology. The only other work we know of on decomposition, aimed specifically at the parameter estimation problem, was done by Voit and Almeida (2004). Their approach requires the pathways to be modeled using the *S-System* formalism (Voit, 2000) and it requires large amounts of accurate experimental data.

1.2.3 The Composition Technique

Our decompositional approach is able to drastically reduce the high dimensional search space, enabling parameter estimation to be performed on large pathway models. However, due to the presence of shared parameters and the multi-modal nature of the problem, estimating the shared parameters in the context of different components will yield different values. In addition, there is a partial order imposed on the components during parameter estimation. Upon estimating the parameters of a component, they will be fixed for the subsequent estimation of other components. This may result in a sub-optimal global solution, especially when information from other components may suggest better estimates for the fixed parameters.

The same issue arises when different pathways are put together to form a larger pathway. This is because previously estimated parameter values do not represent the "true" kinetic parameters. Rather, they represent partial information that is obtained from experimental data on the separate pathways. Hence when the model is put together, some of the values have to be adjusted. The most direct approach when composing different pathways together is to use all the available experimental data, and re-performing the entire parameter estimation process on the updated model. Again, the problems encountered when the pathway size gets larger makes this method highly prohibitive.

In this thesis, we propose the use of a probabilistic approach for pathway composition. We represent the parameters as probability distributions and use a probabilistic graphical model - factor graphs (Kschischang et al., 2001) - to capture the dependencies between the parameters. The factor graphs are derived from the individual smaller components. Upon composition of the pathways, their factor graphs will also be combined together to form a larger composite factor graph.

Crucially, we use the inference technique of *belief propagation* on the factor graph to

reconcile and update the probability distributions of the parameters. Belief propagation is a message passing algorithm that can propagate local constraints throughout the pathway. Upon convergence, the *posterior* distributions of the parameters will have taken into account information from all the local constraints.

1.2.4 Data Integration

Our approach for pathway composition can also be applied to integrate new experimental data. Using the factor graph formalism, we propose to capture the dependencies suggested by the new set of experimental data as a separate *factor graph instance*. Thereafter, data integration can be achieved by composing all the factor graph instances together, followed by belief propagation. The advantage of this approach is that data need not all be available at the same time. In addition, the factor graph formalism can serve as an alternative data structure to store the experimental data, such that parameter estimation can be performed incrementally without having to re-simulate the entire pathway model.

1.3 Overview

This thesis is organized as follows. Chapter 2 gives the background information on pathway modeling, including biological and computational aspects. We also consider the common models used to capture the dynamics of biological pathways. Further, we review some standard parameter estimation techniques. In Chapter 3, we formulate the concept of components and describe our decompositional approach to pathway modeling and parameter estimation, using the Akt-MAPK signaling pathway as a case study. In Chapter 4, we present the factor graph representation of biological pathways. We describe in detail the inference technique of belief propagation, and show how we can reconcile pathway estimates using it. Our approach for data integration will also be explained here. In Chapter 5, we demonstrate the use of our modeling methodology on a large synthetic pathway model based on the Akt-MAPK signaling pathway. We compare the reliability and efficiency of our methodology to some of the conventional parameter estimation algorithms, as well as its performance when experimental data sets are introduced incrementally. Finally, we conclude by discussing our approach and possible future lines of research.

Chapter 2

Background and Related Work

In this chapter, we present the background knowledge on biological pathways and review some of the models that are used to capture pathway dynamics. We also describe some of the algorithmic techniques that are commonly used to tackle the parameter estimation problem.

The basic elements of the cell that perform cellular functions are the *biomolecules* such as proteins, lipids, hormones and nucleic acids. With current technological advances, there are now numerous experimental techniques that researchers can use to observe and measure the activities of the various biomolecules in the cell. Proteins and certain metabolites can be tagged with fluorescent proteins to monitor their relative concentration levels and locations in the cell (Wilson and Walker, 2005). Techniques such as laser scan cytometry (Pozarowski et al., 2006) and flow cytometry (Shapiro, 2003) allow some forms of real-time monitoring of enzymatic activities in processes such as cell-cycle and cell-cell interactions. Gene expression levels for almost the entire genome can be measured using high throughput experimental techniques such as those involving DNA microarray technology. Most of the experimental techniques are developed to aid the identification of biomolecules and their interactions. This is an important first step in achieving a systems-level understanding of biological functions.

The Central Dogma of Molecular Biology

The *Central Dogma of Molecular Biology* is the framework for understanding the transfer of *information* from the gene sequences to the proteins that carry out the various cellular functions. Although it has exceptions, the central dogma is regarded as one of the key guiding principles of molecular biology (Thieffry and Sarkar, 1998).

The central dogma states that DNA molecules encode all the genetic information in

the form of nucleotide sequences. Genetic information is transmitted from the parent to the daughter cells by *replication* to form another DNA molecule during the synthesis phase of the cell-cycle. Within the cell, it utilizes this information by forming messenger ribonucleic acid (mRNA) molecules using the DNA sequence of the gene as a template - a process known as *transcription*. The quantity of mRNA molecules being synthesized is known as the expression level of the gene. For *eukaryotic* cells, the mRNA molecules are exported from the nucleus into the cytoplasm. The sequences in the mRNA molecules are then further read as triplet codons by protein complexes known as ribosomes. Complexes of initiating and elongation factors bring *aminoacylated transfer RNAs* (tRNA) into the ribosome-mRNA complex, and link the amino acids according to the mRNA sequence to form the protein polypeptide chains. This process is known as *translation*. The newly synthesized proteins will then drive all other cellular functions via various chemical reactions.

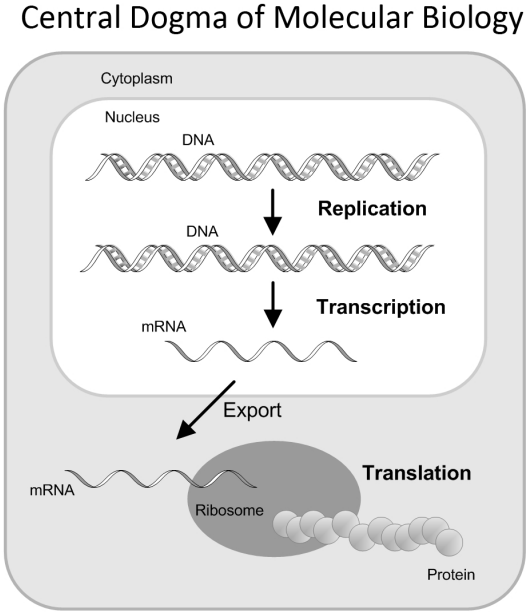


Figure 2.1: The central dogma of molecular biology for eukaryotic cells.

From the central dogma, we can see that there are several ways in which cells can regulate their functions. In response to various stimuli, proteins such as transcription factors can catalyze or inhibit transcription, affecting the gene expression, and subsequently their corresponding protein concentration levels. Upon translation, modifications to the proteins such as phosphorylation and acylation can also have significant

impacts on their activity levels. By varying the protein concentration levels and their activity, the cells can behave differently and perform different functions. Similarly, any disruptions to these mechanisms due to mutations can also lead to the onset of various diseases, such as cancer and Alzheimer's disease.

2.1 Bio-Pathways

The main operations in the cell are the various chemical reactions between its molecules. At the gene level, these reactions may include direct interactions such as the binding of transcription factors to their binding sites on the DNA molecules. Between proteins it may be complex formation or post-translational modifications such as phosphorylation, where the protein conformation or state is modified due to phosphate ions being added to its side chains.

A natural way to record and understand these mechanisms is through a network-centric approach where the nodes of the network represent the molecules and the edges represent interactions. The graphs depict the series of chemical reactions in the cell, and are known as *bio-pathways*.

There are several ways of deriving the graphs that capture the molecular interactions in the cell. These graphs can be broadly classified into one of the following types - *gene regulatory networks*, *metabolic pathways* and *signaling pathways*. Each of these pathway types captures different aspects of cellular functions.

2.1.1 Gene Regulatory Networks

Gene regulatory networks describe the interactions between the genes in the cell. The nodes in these networks represent the genes, and sometimes include mRNA and proteins. However, unlike metabolic and signaling pathways, the edges do not model actual chemical reactions. Instead, they represent *causal influences* between the genes of the cell, since genes do not directly interact with one another. Some of them encode for transcription factors - proteins that catalyze the expression activity of their target genes. Figure 2.2 shows a subset of a gene regulatory network for the *S. Cerevisiae* mating response.

High-throughput experimental techniques such as DNA microarray can show the relative expression levels of several thousand genes in a single experiment. With this huge amount of data, methods such as hierarchical and functional clustering can be used to identify and categorize the genes to discover regulatory relationships between them. The basic concept underlying these types of analyses is "guilt by association",

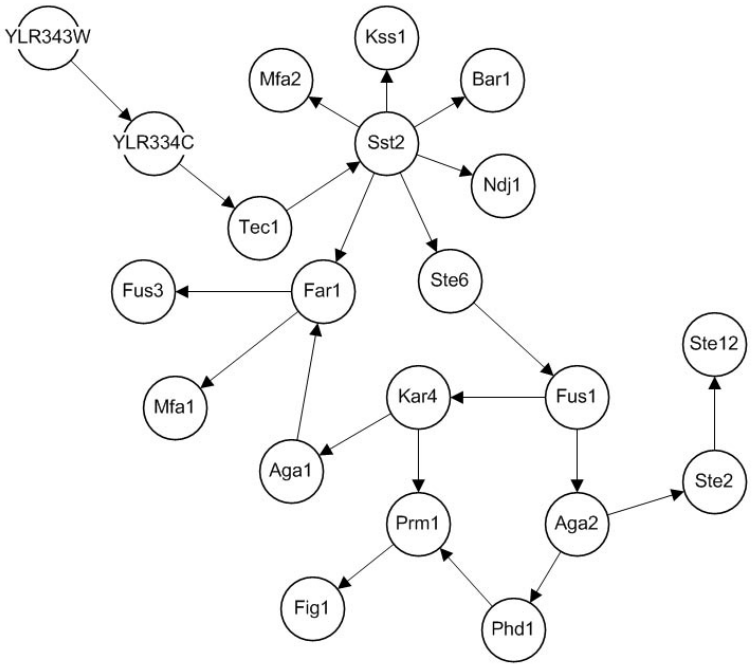


Figure 2.2: A sub-network of the gene regulatory network for the *S. Cerevisiae* mating response. Adapted from Friedman (2004)

where genes that show similar expression profiles are assumed to share a common regulatory rule. In the recent years, the use of probabilistic graphical models such as Bayesian networks (Friedman et al., 2000) and information-theoretic network inference algorithms such as ARACNe (Margolin et al., 2006) are gaining popularity.

2.1.2 Metabolic Pathways

Metabolism is the set of chemical processes that can be classified into anabolic and catabolic reactions. Anabolic reactions are involved in the synthesis of biomolecules. Catabolic reactions, on the other hand, break them down into smaller units, and release energy in the process. These reactions are usually involved in the maintenance and house keeping operations in the cell, allowing it to survive and reproduce. It is also through metabolism that the cell produces energy to enable all other chemical reactions. The networks of these reactions are collectively known as *metabolic pathways*.

One such pathway is the Glycolytic metabolic pathway (Figure 2.3). In this pathway, glucose is broken down into pyruvate via a multi-stage process. From these reactions, energy is released in the form of *adenosine triphosphate* (ATP), which is an essential molecule for driving other reactions. Pyruvate, which is an important bio-

chemical compound, can then be further converted into carbohydrates, fatty acids, the amino acid alanine as well as ethanol, where they serve as inputs to other processes.

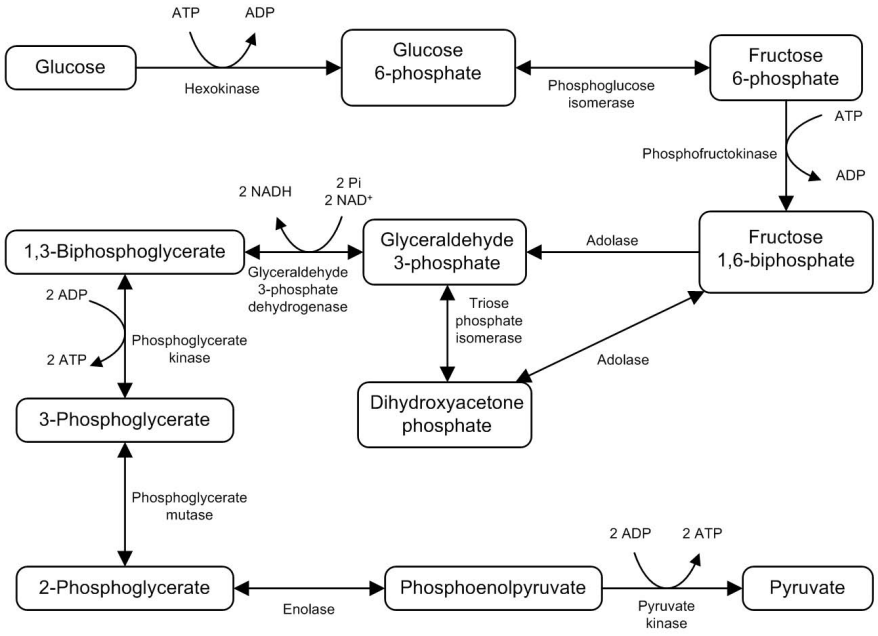


Figure 2.3: The glycolytic metabolic pathway.

The reactions in a metabolic pathway represent the steps in which compounds are being degraded or synthesized from the substrates to the end products. Achieving an equilibrium, or steady-state in the *mass flux*, or the flow of substance, of these pathways is vital in maintaining homeostasis and allowing the proper functioning of the cell.

There are several analytical techniques that operate on this steady-state assumption. Methods such as flux balance analysis (FBA) and metabolic control analysis (MBA) can identify critical as well as rate-limiting elements and reactions in the metabolic pathways. They have been applied to shortlist potential drug targets in the Mycolic Acid metabolic pathway for the treatment of tuberculosis (Raman et al., 2005). Other forms of analyses such as the study of elementary flux modes, stoichiometric and structural analyses (Klamt and Stelling, 2003; Schilling et al., 1999) have also enabled researchers to artificially engineer pathways in order to achieve high-efficiency production of desired biochemicals (Lee et al., 2005).

The analysis of metabolic pathways is one of the earliest works in systems biology. There is now a huge repertoire of databases and software tools developed to aid the study of such pathways (Krieger et al., 2004).

2.1.3 Signaling Pathways

Signaling pathways depict the chemical reactions that occur in response to external and internal stimuli. These chemical reactions serve to transmit information through the cell via changes in the concentration levels of the signaling molecules.

The stimuli often present themselves as molecules and ions, collectively termed as *ligands*. The ligands typically latch themselves onto membrane-bound protein receptors on the cell surface. These receptors are usually trans-membrane, with one part being in the cytosol. The binding of the ligands will cause the receptors to undergo conformational changes, which will either "activate" the proteins in the cytoplasm or act as scaffolds to bring together the internal proteins and their respective activators.

Within the cell, signals are transmitted via a secondary messenger system, which consists of a series of protein modifications - notably *phosphorylation*. In this modification, a phosphate group is covalently bonded to the side chain of a protein by an enzyme, thereby activating it. Conversely, the removal of the phosphate ions, or *dephosphorylation*, de-activates the protein. The enzymes that catalyze such reactions are the kinases and phosphatases respectively. Some of the proteins being activated are kinases themselves. Upon activation, they will phosphorylate downstream molecules. This cascading effect is a common theme in signaling pathways. Research suggests that these cascades serve several purposes. They enable the cell to perform noise filtering, signal amplification, and to produce a bistable switch-like behavior when the input exceeds a certain threshold concentration (Chen and Thorner, 2005). At the end of the cascade, the active proteins will subsequently translocate to the nucleus where they will regulate the transcriptional machinery, thus eliciting the appropriate cellular responses. The various reactions between the proteins and their interactions that transmit the inter- and intra-cellular messages constitute a signaling pathway. An example of signaling pathways in baker's yeast is shown in Figure 2.4.

Unlike metabolic pathways which provide mainly mass transfer, signaling pathways enable the transfer of information. It is the transient behavior in signaling pathways that determines their functions. Hence even though the reaction networks constituting metabolic and signaling pathways are similar, steady-state analysis techniques are seldom relevant on signaling pathways. Instead, dynamical studies based on computer simulations of the signaling pathways are more prevalent. Techniques such as sensitivity analysis on the pathway parameters have been performed for various applications including predicting targets for stem cell fate control (Mahdavi et al., 2007) and determining the roles of the proteins IKK and IκBα in controlling oscillations in the NFκB signaling pathway (Ihekwaba et al., 2004). Simulation studies also enable researchers to

uncover *emergent behaviors* - e.g. ultrasensitivity and bistability - which are otherwise impossible to reveal simply by studying the static pathway maps (Bhalla and Iyengar, 1999).

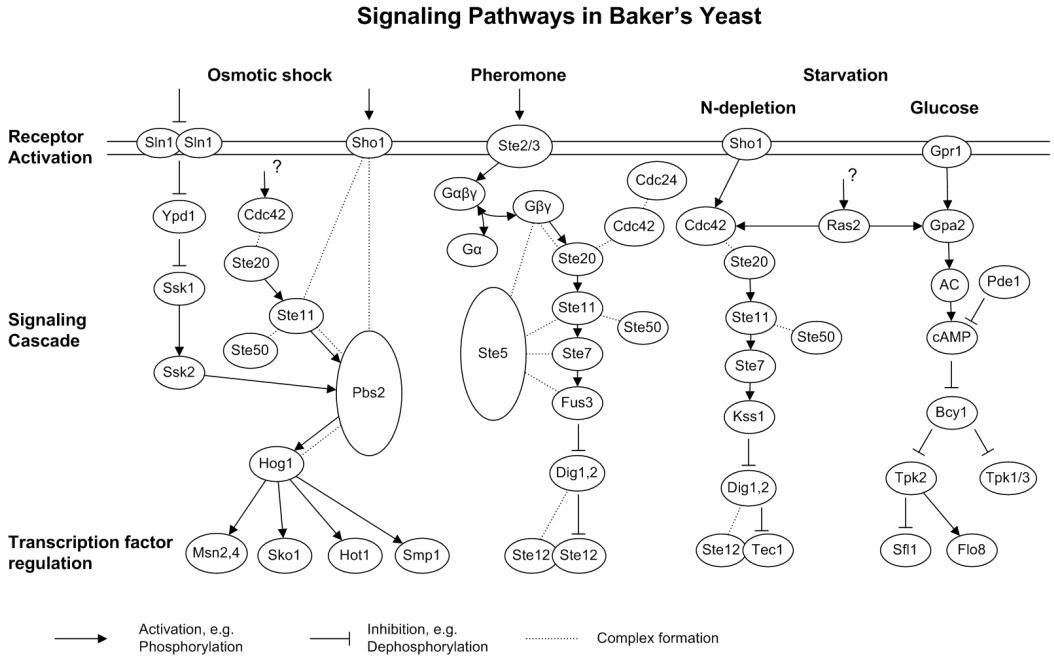


Figure 2.4: An example of the various signaling pathways in the baker's yeast. The pathways shown are triggered by the various conditions - osmotic shock, pheromone from cells of opposite mating type, and growth pathway stimulated by starvation conditions (Klipp and Liebermeister, 2006).

Signaling pathways are complex biological mechanisms that use elaborate mechanisms such as feedback loops, cycles and amplification cascades in order to elicit proper cellular responses. The objective is to guarantee reliable cell decisions concerning behavioral functions in uncertain and variable environments (Lauffenburger, 2000). This way, cellular systems can be robust to noise, and at the same time adaptive to external changes.

Hundreds of signaling pathways have been discovered. Commonly studied pathways include the mitogen-activated protein kinase (Schoeberl et al., 2002) and the Akt signaling pathways (Nicholson and Anderson, 2002). These two pathways have been implicated in several physiological processes such as growth and apoptosis. Other pathways include the various *Wnt* signaling pathways in developmental studies (Lee et al., 2003), and the NFκB signaling pathway which regulates immunity and inflammatory

responses (Gilmore, 2006).

The three types of pathways and their variants present the different views researchers have on the mechanisms of cell functions. They can be studied together to provide more insights on the mechanisms of cellular functions and regulation. Doi et al. (2006b) have combined elements of the gene regulatory network as well as signaling pathways to model and simulate a computational model of the p53 transcriptional activity using the formalism of Hybrid Functional Petri Nets. Miller et al. (2001) have also investigated the activity of the Wnt signaling pathway and multiple metabolic pathways in response to the thyroid hormone (T3) induced transcriptional program in the rat pituitary tumor cell line.

In our work, we will focus mainly on the signaling pathways, although the methods that will be presented in the following chapters are applicable to metabolic pathways and gene regulatory networks as well.

2.2 Pathway Modeling

In pathway models, the *state* of the system consists of the *concentration levels* of all the molecule species involved. The molecules in a pathway are assumed to be spatially homogeneous throughout the cell or the compartment such that the reaction rates can be expressed as functions of the concentrations of their participating molecules. In more advanced models, spatial inhomogeneities can be taken into account through the use of *partial differential equations*, where the derivatives are expressed with respect to several variables such as time and location in the cell.

For the rest of this section, we shall review some of the commonly used models in representing and capturing the dynamics of biological pathways. We first describe systems of ordinary differential equations and how the various rate equations can be formed within this framework. Next, we describe some of the *graphical* models that were originally developed in systems engineering but later used to model and simulate biological systems.

2.2.1 System of Ordinary Differential Equations

The use of systems of ordinary differential equations (ODEs) to model the dynamics of pathways is, by far, the most dominant approach in systems biology (Materi and Wishart, 2007). Using this formalism, a pathway is represented by a system of ODEs

$$\frac{dx_i}{dt} = f_i(\mathbf{x}(t), \mathbf{p}) \tag{2.1}$$

where dx_i/dt is the rate of change of the concentration level for molecule species x_i (typically a protein). The vector $\mathbf{x}(t)$ denotes the concentration levels of the molecules at time t. The set $\mathbf{p}$ are the rate parameters, many of which are usually unknown. The function f_i, typically nonlinear (but algebraic), encodes the form of the rate reaction that produces or consumes x_i. Except for simple systems, there are no analytical solutions to the system of equations. Most of them have to be integrated numerically using solvers such as Runge-Kutta (Hindmarsh, 1983) and LSODA (Petzold, 1983) in order to simulate their dynamics.

The form of the rate function f_i often follows certain principles in biochemistry. In this thesis, we describe two common ones that are based on chemical kinetics and enzyme theories.

Mass-action

An elementary reaction is one where the substrates react *directly* with one another to form the products in a single reaction step. Any complex chemical reaction can be expressed as a sequence of elementary reactions. The law of mass-action states that the rate of an elementary reaction is proportional to the concentration of the molecules involved. A pathway model where all the reactions are written in terms of these elementary mechanistic steps is known as a mass-action model.

A typical elementary reaction is shown in Equation 2.2. The substrates x_1 and x_2 bind reversibly to form a complex x_3 at a rate that is controlled by the forward and reverse rate constants k_1 and k_2.

$$x_1 + x_2 \underset{k_2}{\overset{k_1}{\rightleftharpoons}} x_3 \tag{2.2}$$

The system of ODEs that describes this simple reaction scheme is

$$\begin{aligned}
\frac{dx_1}{dt} &= k_2 x_3 - k_1 x_1 x_2 \\
\frac{dx_2}{dt} &= k_2 x_3 - k_1 x_1 x_2 \\
\frac{dx_3}{dt} &= k_1 x_1 x_2 - k_2 x_3
\end{aligned}$$

An important observation is that for reversible reactions, the ratio between the

product concentration and substrate concentration during equilibrium is constant. This ratio is known as the *equilibrium constant*. Denoted as K_{eq}, the equilibrium constant can also be computed as the ratio between the forward and the reverse rate constants.

$$K_{eq} = \frac{k_1}{k_2} \tag{2.3}$$

Under certain conditions, where the reactions reach equilibrium very rapidly, this constant can be used in place of the individual rate constants, reducing the number of differential equations and parameters at the expense of having more complex rate terms (Okino and Mavrovouniotis, 1998). Lee et al. (2003) have used this method to reduce the number of equations in their *Wnt* signaling pathway model from 15 equations down to only 7 equations.

On a general mass-action model, suppose we have n molecule species and m reactions, the rate v_j of reaction j can be specified by the following equation.

$$v_j = k_j \prod_{i=1}^{n} x_i^{g_{ij}} \tag{2.4}$$

where k_j is the rate constant for reaction j and g_{ij} is the reaction order of species i in reaction j. The rate of change for molecule x_i can then be represented by the mass balance equation

$$\frac{dx_i}{dt} = \sum_{j=1}^{m} c_{ij} v_j \tag{2.5}$$

In this equation, c_{ij} is the stoichiometric coefficient of species i in reaction j. Note that in basic elementary reactions, the reaction orders g and the stoichiometric coefficients c have the same values, although they are conceptually different. The stoichiometric coefficient refers to the quantitative relationship between reactants and products in a chemical reaction, while the reaction order is the number of reacting molecules whose concentration changes as a result of the reaction.

The stoichiometric coefficients can be arranged into a matrix. When analyzing an equilibrium system, where all the rates of change of the molecule species are 0, the *stoichiometric matrix* can provide a wealth of information and insights on the features of the pathways using techniques such as the *flux balance analysis*, *elementary flux modes* and *extreme pathways* (Kauffman et al., 2003; Klamt and Stelling, 2003).

A mass-action model allows pathways to be modeled systematically. One only has to consider all the elements in the pathway and their interactions. The form of

their rate equations is obvious. However, there will be in general too many unknown parameters. Another disadvantage is that, being too detailed, the model becomes very cumbersome to simulate and analyze. Schoeberl et al. (2002) have built the entire MAPK pathway using only mass-action kinetics. With the model, they form hypotheses about the robustness of the pathway to input ligand concentration, and identify the critical parameters determining signaling efficacy. However, their model is large, with 94 molecule species and 95 parameters controlling the various rate reactions. Most of these parameters were obtained from various literature sources, which may have incorporated assumptions that are no longer relevant to the reported one. On the other hand, to estimate all the parameters by fitting the model to experimental data will be a computationally intensive task, given the high level of detail. These problems become more prohibitive when other pathways are integrated to the MAPK pathway to study their cross-talk interactions.

To mitigate the modeling and simulation challenges due to the complexity or size of mass-action pathway models, simplifying assumptions can be made to reduce the number of pathway elements and reactions, while ensuring that the relevant dynamics are still being captured in the simplified models (Aldridge et al., 2006). A commonly used abstraction is to replace certain mass-action kinetics with the Michaelis-Menten approximation.

Michaelis-Menten

Most biological pathways have reactions that are enzyme-catalyzed. These enzymes are proteins that can immensely increase the rate of their target reactions. A key aspect is that the enzymes themselves are not being modified at the end of the reactions. Although the kinetics of such reactions can be modeled as a series of mass-action reactions, some of the intermediate products are unstable, e.g. the enzyme-substrate complex. Hence, measurements and determination of the rate parameters for these reactions are difficult. On the other hand, the parameters associated with the Michaelis-Menten kinetics are easier to measure under *in vitro* conditions.

The Michaelis-Menten equation is derived from the following scheme for enzyme-catalyzed reactions.

$$S + E \underset{k_{-1}}{\overset{k_1}{\rightleftharpoons}} ES \overset{k_2}{\rightarrow} E + P \tag{2.6}$$

where S is the substrate, E is the enzyme, and P is the product. ES is the intermediate complex formed when the enzyme E binds reversibly to the substrate S. The rate

constants k_1 and k_{-1} affect the forward and reverse reactions respectively, and k_2 is the catalytic constant. The degradation of product P back to the complex ES is considered negligible, and hence is not modeled. In addition, it is assumed that the binding of a substrate to an enzyme binding site does not affect the activity of any adjacent sites on the enzyme. Based on the scheme above, the system of ODEs for the dynamics is as shown below.

$$\frac{dS}{dt} = -k_1 S.E + k_{-1} ES \tag{2.7}$$
$$\frac{dES}{dt} = k_1 S.E - (k_{-1} + k_2) ES \tag{2.8}$$
$$\frac{dE}{dt} = -k_1 S.E + (k_{-1} + k_2) ES \tag{2.9}$$
$$\frac{dP}{dt} = k_2 ES \tag{2.10}$$

Quasi-Steady State Assumption

It is first assumed that the reversible conversion of E and S into the complex ES is much faster than decomposition of ES into E and P, i.e. $k_1, k_{-1} >> k_2$. This was later made more general in the quasi-steady state assumption by Briggs and Haldane (1925) which states that after an initial reaction period, the concentration of the intermediate complex ES remains constant, i.e. $dES/dt = 0$. This is justified only if the initial concentration of the substrate is much higher than the concentration of the enzyme.

Together with the conservation equation $E_{total} = E + ES$, an enzyme-catalyzed reaction can then be described by the following scheme:

$$S + E \xrightarrow{v} P + E \tag{2.11}$$

The rate of the reaction v can be expressed by the Michaelis-Menten equation

$$v = \frac{V_{\max} S}{K_m + S} \tag{2.12}$$

The constant $V_{\max}$ is the maximum rate of the reaction that can be attained when the enzyme is completely saturated with the substrate, i.e. $V_{\max} = k_2 E_{\text{total}}$. The constant K_m, also known as the *Michaelis constant*, is equal to the concentration of the substrate that yields half the maximal reaction rate. This constant can also be obtained from the following equation.

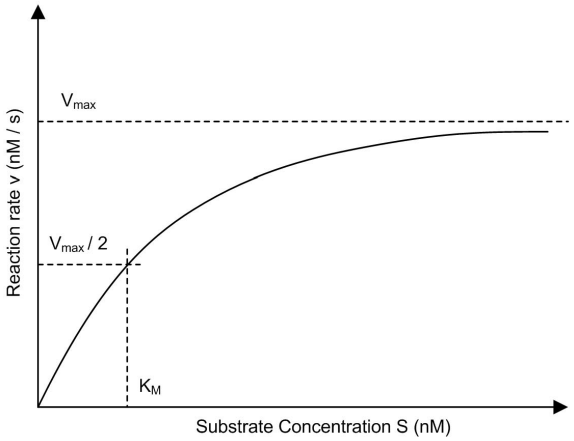


Figure 2.5: The graph of the initial reaction rates vs substrate concentration. The maximal rate is the constant $V_{\max}$, while the substrate concentration that yields the half-maximal rate is the Michaelis constant K_m. For low substrate concentration, v increases almost linearly with S while for high substrate concentration, the enzyme concentration level becomes the limiting factor instead.

$$K_m = \frac{k_{-1} + k_2}{k_1} \tag{2.13}$$

To obtain the parameters experimentally, one would only need to measure the initial reaction rates for different initial concentrations of the substrate and obtain the values directly from the plotted graph (Figure 2.5).

There are several other variants of the Michaelis-Menten equation that take into account influences by the surrounding molecules. These additional considerations have given rise to different forms such as the *reversible* Michaelis-Menten equations and equations with competitive and uncompetitive *inhibition*. One can refer to Klipp et al. (2005) for more details.

It is not always the case that the substrate is of a much higher concentration than the enzyme (Ciliberto et al., 2007). In such settings, the quasi-steady state assumption will fail. Another limitation of the quasi-steady state assumption is that, being an abstraction, it is unable to capture the true dynamical behavior of some pathways. Flach and Schnell (2006) have considered the assumption on a number of systems and have shown that some dynamics such as limit-cycle behaviors on linear feedback systems cannot be reproduced as compared to using mass-action models. Regardless, the Michaelis-Menten equation and its variants remain the most dominant abstraction which can be found in several kinetic models of bio-pathways. Many pathways represented as sys-

tems of ODEs have incorporated reactions that follow the mass-action law as well as Michaelis-Menten enzyme kinetics.

2.2.2 Boolean Networks

In a Boolean network, each gene or molecule species is modeled as a binary variable. For a pathway with n elements, the state of the system can be represented by an n-dimensional binary vector $\mathbf{x}$ such that each element x_i denotes the state of the ith element. To represent the dynamics, for each element i, we associate a boolean rule b_i. At time t, the state of the input variables to the boolean rule b_i will determine the next state of x_i at time $t+1$, i.e.

$$x_i(t+1) = b_i(\mathbf{x}(t)), 1 \leq i \leq n \tag{2.14}$$

An example of a Boolean network representation of a gene regulatory network is shown in Figure 2.6.

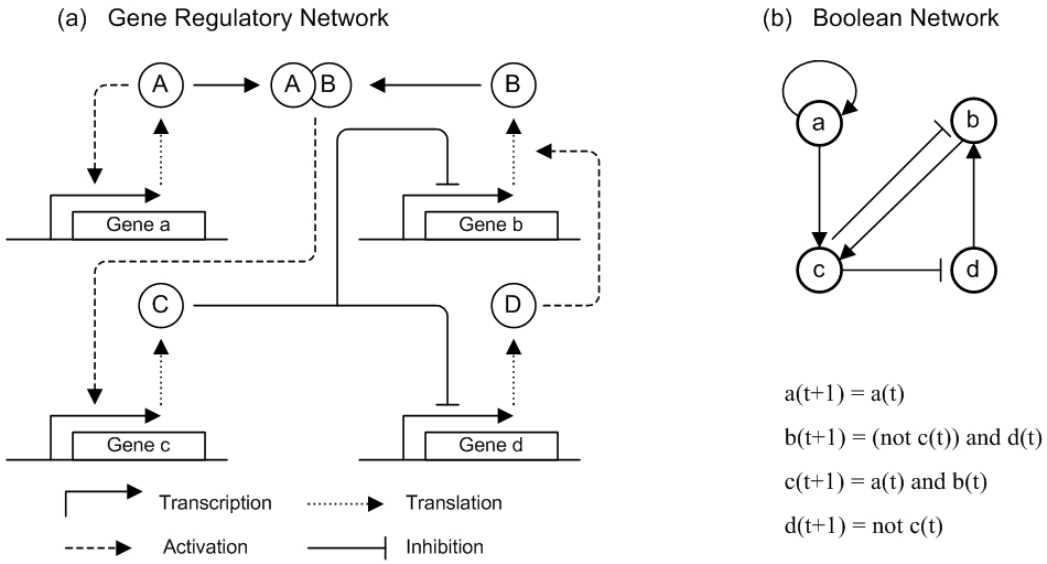


Figure 2.6: (a) An example of a simple gene regulatory network consisting of 4 genes. (b) The Boolean network representation and the corresponding Boolean rules.

Given a start state, the system can follow the Boolean rules in a *deterministic* manner, making it transit from one state to another state. The series of such states is known as a *trajectory*. If all the starting states are known, the set of resulting trajectories can be represented collectively as a state-transition diagram, which records all the possible behavior of the pathway. Using this diagram, one can study the global

behavior of the pathway. For instance, the presence of *cyclic attractors* (Kwon and Cho, 2007), i.e. a fixed cyclic sequence of states, corresponds to an oscillating dynamical behavior for the pathway. Emergence of point attractors, i.e. states which lead to themselves, can also aid in determining the conditions that will eventually lead to the steady state of the cellular system.

A main challenge in using Boolean networks for large scale modeling of bio-pathways is the determination of the pathway topology. There have been several algorithms designed to *reverse engineer* Boolean networks from steady state or time series gene expression data. *REVEAL* was one such algorithm developed for this purpose (Liang et al., 1998). However, the high levels of noise in experimental data and the huge data requirements for pathway inference have limited the practical use of Boolean networks (Gardner and Faith, 2005).

There have been several extensions to the basic Boolean network model. Schaub et al. (2007) used integer values to denote the finite states each gene or protein can take. They call this extended model *Qualitative networks* and used it to study the cross-talk interactions between the *Notch* and *Wnt* signaling pathways. Probabilistic features have also been added such that the new state of the element x_i is determined by the update function b_i, which is chosen from a set of functions with a certain probability. This is to enable uncertainties to be captured in the system. Note that these uncertainties arise not only from the reactions themselves, but also due to variances in the environment that are not explicitly modeled in the pathway.

2.2.3 Petri Nets

Petri nets are a fundamental model for specifying distributed discrete systems (Reisig, 1992). They offer an appealing visual notation which resembles the graphical diagrams often used by biologists to depict the biological entities and their interactions in the various pathways. In addition, the Petri net model has a precise semantics which fixes the meanings of the nodes and the arcs of the model as well as the dynamics.

A Petri net can be viewed as a bipartite graph with two kinds of nodes, usually called *places* and *transitions*. The places represent local states and the transitions represent local change-of-states. Entities called tokens are used to mark the places to specify the current distributed state of the system. The transitions, according to a firing rule associated with them, effect local transformations of the token distribution to model the system evolving from the current state to a new one. In the graphical representation, the places are drawn as circles, the transitions as boxes, and the tokens as small bullets placed inside the places. A standard firing rule is that if all the places

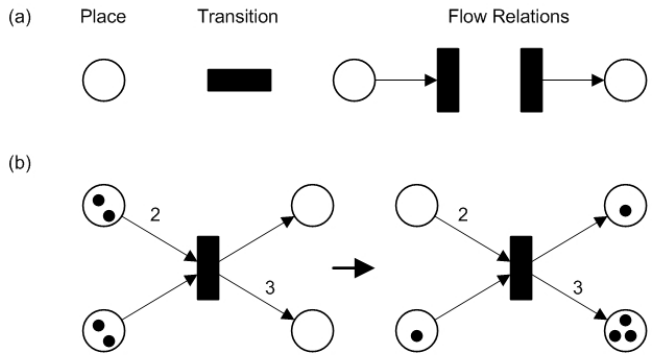


Figure 2.7: (a) The basic elements and connections of a Petri net model. (b) Change in markings of a Petri net due according to the firing rules.

pointing into the transition currently carry at least one token each, then the transition may fire. When it does so, one token is removed from each of its input places and one token is added to each of its output places. To improve modeling power, one can also associate weights with the arcs so that the firing of a transition can depend on, remove and add multiple tokens from its surrounding places. This is illustrated in Figure 2.7(b). There are a large number of variations of this basic model, and they have been deployed in a wide variety of application domains. A collection of such efforts in biological settings can be found in Chen and Hofestaedt (2003), Lee et al. (2006) and Voss et al. (2003).

Using Petri nets, different types of analyses on the biological pathways can be performed. Zevedei-Oancea and Schuster (2003) used Petri nets for the topological analysis of pathways, using features such as *boundedness* and *liveness* to reveal their behavior without the need for explicit simulation of the system. Heiner et al. (2004) have also used the concept of *T-invariants* in Petri net theory to establish the consistency of the apoptotic pathway. The T-invariant is a multi-set of transitions, the firing of which will reproduce a given system state. By identifying the T-invariants, they are able to verify the different conditions that will lead to apoptosis.

2.2.4 Hybrid Functional Petri Nets

The classical Petri net is a model of a discrete event system whereas a crucial aspect of bio-pathways is the various biochemical reactions, which are best specified as continuous differential equations. Indeed, both discrete and continuous features appear to be an integral part of fundamental biological processes. To account for this, several hybrid

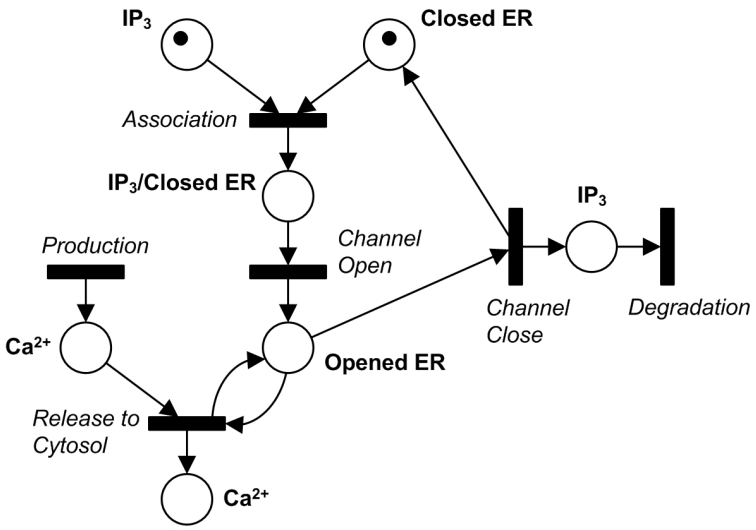


Figure 2.8: Example of a pathway depicting the release of Ca^{2+} into the cytosol being modeled as a Petri net.

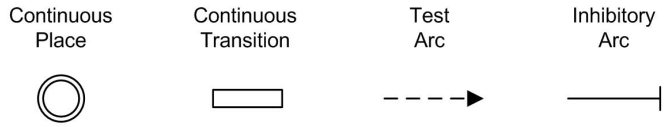


Figure 2.9: Additional elements for an HFPN model.

dynamical models have been proposed (Lincoln and Tiwari, 2004). In the setting of Petri nets, the hybrid version of interest to us is the Hybrid Functional Petri Net (HFPN) developed by Matsuno et al. (2003).

In an HFPN, places and transitions can be discrete or continuous. The marking associated with a continuous place can be a real number, which can change smoothly according to the speed assigned to the continuous transition(s) to which it serves as an input or output place. In addition, an edge can be one of three types: normal, inhibitory, or test arc. An inhibitory arc points from a place to a transition, and it specifies that the transition is inhibited from firing whenever a token is present in the place, or that the value of the place exceeds a certain threshold. A test arc from a place to a transition specifies that the transition can only fire if a token is present in the place, but the firing of the transition does not change the token count on this place. Figure 2.9 displays the HFPN elements that are not present in classical Petri nets.

We can also associate *weights* to the various arc types. For normal arcs connecting

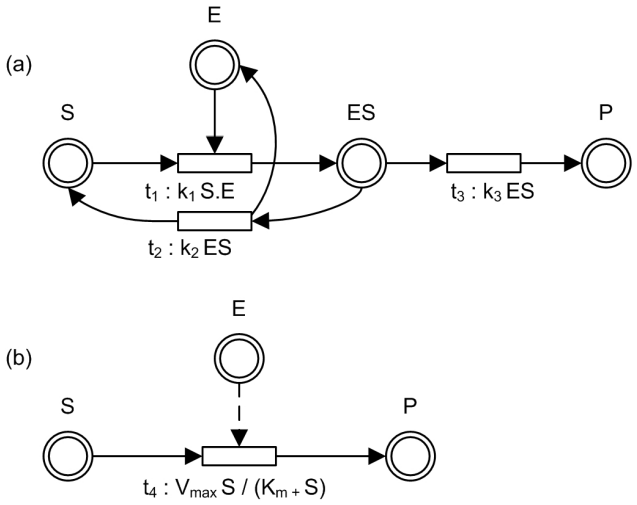


Figure 2.10: (a) HFPN representation of a biochemical reaction. Assuming quasi-steady state assumption, the model can be simplified into (b), where $V_{\max} = k_3 E$.

discrete places and transitions, their weights are integer values that denote the number of tokens that will be consumed or produced by the firing of the transitions. For normal arcs connecting continuous places and transitions, their weights represent the real-valued stoichiometric coefficients of the places for the transitions they are connected to. For test and inhibitory arcs, their weights are then the threshold values used when triggering or inhibiting the firing of the transitions.

For the enzyme-catalyzed reaction scheme in Equation 2.6, its HFPN representation is shown in Figure 2.10(a). Each molecule species is represented by a continuous place, and its concentration corresponds to the marking associated with that place. Each continuous transition is associated with the rate function of a reaction. Assuming the quasi-steady state assumption, the HFPN model can also be simplified using a test arc as shown in Figure 2.10(b).

HFPN as Graphical Representation of ODEs

A system of ODEs that describes a biological pathway can be represented as an HFPN model using only the continuous places and transitions, as well as the normal and test arcs. The discrete parts of the HFPN will not be needed.

Being a graphical formalism, the HFPN model is more amenable to structural analysis as compared to a system of ODEs. We will show informally how we can convert a system of ODEs into its HFPN representation. Consider, without loss of

generality, a mass-action model where the reaction rates are expressed according to Equation 2.4 and the rate of change of the molecular concentrations for each species is written as Equation 2.5.

For each variable x_i we assign a continuous place p_i. Similarly each rate equation v_j is assigned a continuous transition t_j. To connect the places and transitions by arcs, we look at the stoichiometric coefficients. Suppose there exists x_i and v_j such that $c_{ij} > 0$, then there will be an arc from t_j to p_i where the weight $W(t_j, p_i)$ is equal to the value of c_{ij}. For $c_{ij} < 0$, it will be a place-to-transition arc from p_i to t_j. The weight of the arc is the absolute value of c_{ij}. Read-only arcs will mainly appear in association with enzyme-catalyzed reactions described by Michaelis-Menten kinetics, where there is no consumption of the enzymes in the catalyzed reactions. Specifically, suppose there exists x_i and v_j such that $c_{ij} = 0$, and x_i is a term of the equation for v_j, then there is a read-only arc that connects the place p_i to t_j.

2.2.5 Hybrid Automata

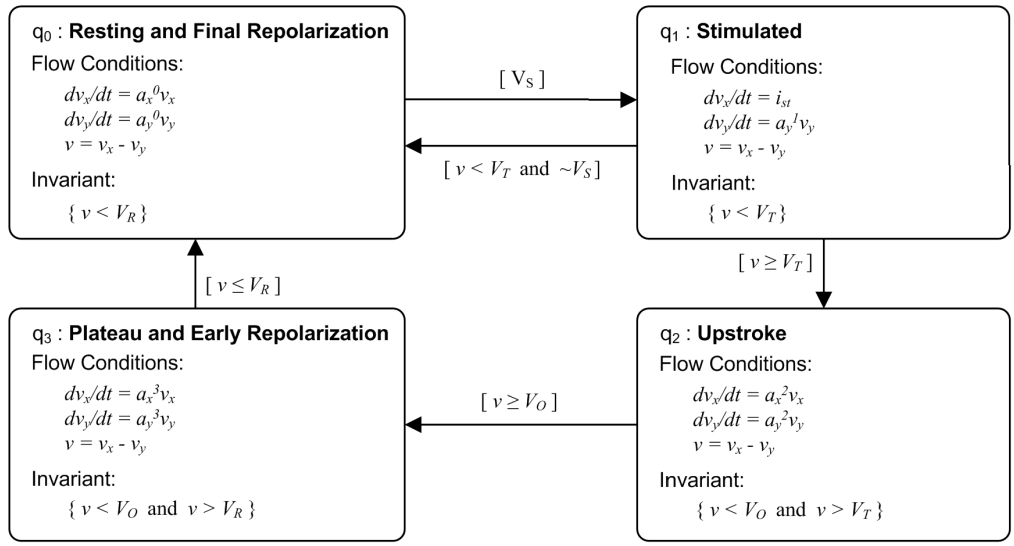
Hybrid automata is a specification language that is used to model the behavior of hybrid systems, which consist of discrete control computations in a continuous environment. Conventionally used to model embedded reactive systems such as engine control in avionics and automotive electronics, hybrid automata are increasingly being used to model the behavior of biological systems (Antoniotti et al., 2003; Casagrande et al., 2005; Mishra and Policriti, 2004).

A hybrid automaton has a finite set of states that are connected by transitions. The states capture the different *modes* of cellular behavior, such as the *run* and *tumble* modes in bacterial chemotaxis (Spiro et al., 1997). The behavior of the variables in each state is then governed by a set of ODEs. Each state is labeled by an invariant condition that must hold while the system is in that state, and each transition is labeled with a set of assignments guarded by events and jump conditions. Upon satisfying a jump condition, or the occurrence of an event, the system will switch from one mode to another. An example of a hybrid automaton depicting the cell excitation process of a squid giant axon is shown in Figure 2.11 (Ye et al., 2006).

A key aspect of a cell is its differing behaviors in various modes. This can be captured using hybrid automata. They can also be used to model phenomena such as cell differentiation, where the relevant pathways can have widely different dynamics, depending on which state the system is in.

Ghosh and Tomlin (2004) have applied hybrid automata to model inter-cellular Delta-Notch signaling in order to understand pattern-formation within several biolog-

(a)



(b)

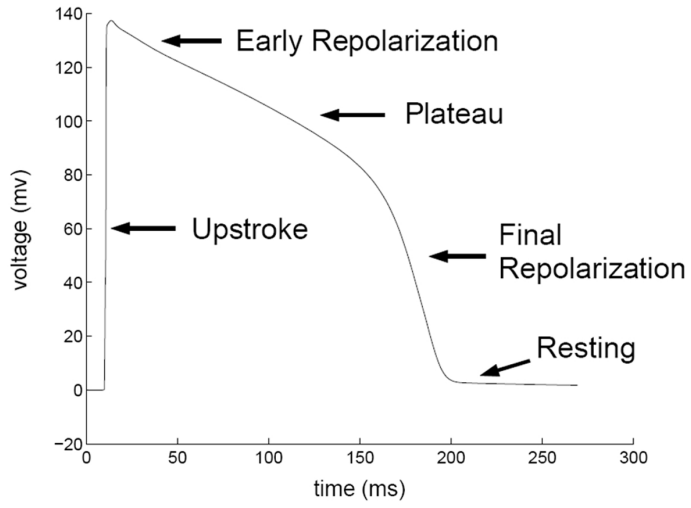


Figure 2.11: (a) Hybrid automaton of the Hodgkin-Huxley axon model with four states. In each state, v is the membrane voltage, v_x and v_y are the differential equations that control the voltage. V_S is the stimulation event and V_T, V_O and V_R are the threshold, overshoot and repolarization voltages respectively. (b) The dynamics of the excitation that is captured by the hybrid automaton.

ical systems. Ye et al. (2006) have also used hybrid automata to model the behavior of cardiac cells in the heart. Having abstracted the continuous behavior into discrete states, reachability analysis can be performed on the hybrid automata to determine whether a cell can assume certain behavior, given the initial conditions.

Several variants of the hybrid automata have been developed. Alur et al. (2001) have extended the hybrid automata by combining it with *agent-based* semantics. Instead of having the entire cellular system being represented by a single automaton, each protein or gene type has its own state-transition graph such that each of them behaves like an independent *agent*. The entire system can then be simulated as a population of concurrent agents, communicating with one another to influence the collective behavior of the system. Using such an approach, the authors have modeled the luminescence phenomenon in the bacterium *Vibrio fischeri*.

In this section, we have reviewed some of the models that are commonly used to capture the dynamics of biological pathways. This is by no means an exhaustive review. The models described are largely deterministic, but they have stochastic extensions to cater for the probabilistic nature of certain processes. A key requirement for all the quantitative models presented is that their various rate parameters must be available before any meaningful simulation and analysis can be done.

2.3 Estimating the Pathway Parameters

For any modeling formalism, to experimentally determine all the parameters of a pathway model *in vitro* is a daunting task. In addition, *in vitro* measurements may not reflect the actual physiological conditions in the cell, affecting the accuracy of the measured parameters (Minton, 2001). A common approach to alleviate such issues is to estimate all the parameters by fitting the model to available data from steady state or time-series experiments.

Parameter estimation can be viewed as an optimization problem with differential-algebraic constraints. The input to the problem consists of the observed values of state variables at selected discrete time points. In the present setting, these are steady-state or time-series measurements of molecule concentration levels. The problem is to determine the values of the m parameters $\mathbf{p} \in \mathbb{R}_+^m$ (which we will often refer to as solutions) and all the unknown concentrations such that they minimize an objective function, typically of the form

$$J(\mathbf{p}) = \sum_{ijk} \frac{(x_{ij}(t_k, \mathbf{p}) - x_{ij}^{\exp}(t_k))^2}{w_{ij}} \tag{2.15}$$

subject to

$$\frac{d\mathbf{x}}{dt} = f(\mathbf{x}(t), \mathbf{p}) \tag{2.16}$$

$$h(\mathbf{x}, t) \geq 0 \tag{2.17}$$

$$\mathbf{p}^L \leq \mathbf{p} \leq \mathbf{p}^U \tag{2.18}$$

Here f is the set of differential constraints describing the system dynamics (equivalent to Equation 2.1). The function h is the state constraints for the variables $x \in \mathbf{x}$ for all $t \in [0, T]$. $\mathbf{p}^L$ and $\mathbf{p}^U$ are the lower and upper bound constraints on the set of parameters $\mathbf{p}$. The time points $t_k \in [0, T]$ denote the time instances where experimental data is available. The expression $x_{ij}(t_k, \mathbf{p})$ is the model predicted value of the variable x_i in experiment j at time point t_k using parameters $\mathbf{p}$. The term $x_{ij}^{\exp}(t_k)$ represents the corresponding experimental measurement of the same variable. w_{ij} is the weight that is used to normalize the contributions of each term to the objective function. This value is usually taken to be the maximum value of x_i in experiment j. This objective function is also sometimes referred to as the weighted residual square between data and simulation profile.

Given this problem formulation, a typical parameter estimation algorithm will have the following scheme (Mendes and Kell, 1998).

1 Set initial (possibly random) values for adjustable parameters (i.e. generate candidate solution)

2 Evaluate the objective function by simulating the model with the current parameter values. The output is usually referred to as the *score* of the parameter set/solution

3 If the stopping criteria is satisfied, go to step 6

4 Generate new guesses for the parameters

5 Repeat step 2

6 Terminate algorithm

The execution of a parameter estimation algorithm may be intuitively viewed as a *walk* in the parameter search space. Each point in the space will evaluate to a score. Collectively the set of points and their scores are referred to as the *solution landscape*.

Satisfying the stopping criteria means that a solution has been reached, though it might not be optimal. For some algorithms, this criterion is that the current solution evaluates to the lowest score as compared to its immediate neighboring solutions. This is also known as locating a *local minima*. If it can be shown that the score is the lowest for all possible parameter combinations, it will be considered a *global minima*. Other stopping criteria might include having executed a maximum number of iterations, or having the next few solutions that do not fare as well as the current one. After termination, the algorithm will return the best solution it has encountered.

Most algorithms differ in steps 1 and 4, and depending on the way the new guesses are being generated, the algorithms can be classified as *deterministic* or *stochastic* ones.

2.3.1 Deterministic Algorithms

The key feature of deterministic algorithms is *predictability*. Given the same starting values for the parameters, these algorithms will always converge to the same solution. For small models, deterministic methods are able to locate good solutions. They have been applied to estimate the parameters for a pathway that describes the cell cycle in frog eggs (Zwolak et al., 2005). This pathway is relatively small, with four molecule species and ten unknown parameters.

Most deterministic algorithms take a greedy approach when traversing the solution landscape. Suppose at iteration i, the current guess for the parameters is $\mathbf{p}_i \in \mathbb{R}_+^m$. A deterministic algorithm will compute the *gradient* of the solution landscape at that point, i.e. $\delta J(\mathbf{p}_i)/\delta \mathbf{p}_i$ to determine the direction to make for the next best guess.

An example of a deterministic method is the Levenberg-Marquardt algorithm (Gill et al., 1998). When the algorithm is in a region where the gradient is flat, it will take larger steps. Conversely, it will take smaller steps in a region where the gradient is steep. This allows it to save significant computational efforts in the flatter regions of the solution landscape.

Most deterministic algorithms are able to find solutions fairly quickly. However, for large biological pathway models, it has been shown that the solution landscape is very complex with possibly several local minima. For the deterministic algorithms, most of the solutions will thus be sub-optimal, with the algorithms getting trapped in local minima. One of the modification to the algorithms is to run them several times, using different starting points in the parameter space. However, their performance still does

not match up to stochastic methods, which we discuss next.

2.3.2 Stochastic Algorithms

Stochastic algorithms rely on probabilistic approaches for the updating of their parameter guesses. As such, these algorithms can only provide weak theoretical guarantees of convergence to the global minima. Stochastic algorithms are easy to implement and use. They either randomly select the next guess for the parameter values, or transform current solutions to produce new guesses. They are less susceptible to getting trapped in local, sub-optimal solutions.

Of particular interest are the *evolutionary computation* methods. These algorithms do not keep a single current guess of the parameter guesses. Rather, they keep a population of such solutions. Operations such as recombination, crossover and mutation are then performed on selected parents to produce the next *generation* of solutions.

Although weak in theoretical basis, evolutionary algorithms have been shown to work well in performing parameter estimation (Moles et al., 2003). We discuss below one of the most commonly used family of techniques - Evolutionary Strategies. Thereafter, we will also describe in detail one of its variants, the SRES algorithm.

Evolutionary Strategies

Evolutionary Strategies (ES) is a family of optimization techniques based on the ideas of adaptation and evolution (Beyer and Schwefel, 2002). It uses mutation, recombination and selection on a population of candidate solutions in order to evolve iteratively better solutions. In this algorithm, each iteration is known as a generation, and the solution for the current generation is then referred to as the parent solution, which will be used to produce the child solution for the next generation.

The ES algorithm maintains a group of μ parents, using which λ *children* will be generated as candidate solutions. Each child solution is obtained by a recombination of two parents using random crossover of the parameters, or taking the mid-points of their values. There are no specific guidelines on how recombination should be carried out, and its implementation is left to the designer of the ES algorithm.

After generating the children and scoring them, the best μ solutions from the combined set of parent and children solutions are selected to be the parents for the next generation. The entire process then repeats itself until no better solutions can be found, or if the algorithm has undergone a specified number of iterations. The pseudocode for $(\mu + \lambda)$-ES is shown in Algorithm 2.1.

Algorithm 2.1: $(\mu + \lambda)$-ES

```
begin
    Initialize parent population P_μ = {p_1, ..., p_μ}
    repeat
        S ← P_μ
        for i ← 1 to λ do
            Randomly select parents p_c1, p_c2 ∈ P_μ
            p_new ← Recombine(p_c1, p_c2)
            p_new ← Mutate(p_new)
            S ← S ∪ {p_new}
        end
        Sort(S)
        P_μ ← Select first μ from S
    until Stopping Criteria
end
```

Evolutionary Strategies with Stochastic Ranking

Evolution Strategies with stochastic ranking (SRES) is one of the best performing stochastic method in parameter estimation (Moles et al., 2003). This is a variant of the ES algorithm that assigns a ranking to the parent solutions in computing the current generation.

The SRES follows the scheme in Algorithm 2.1. However, it modifies the standard $(\mu + \lambda)$-ES algorithm in the Recombine() and Sort() steps. For assessing the fitness of a solution, it uses a scoring function which not only includes the standard objective function, but also a penalty component for solutions that fall within infeasible regions of the parameter space. This is summarized in the following equation.

$$\psi(\mathbf{p}) = J(\mathbf{p}) + r_g \phi(g(\mathbf{p})) \tag{2.19}$$

where ϕ is the penalty function for solutions that fall into infeasible regions. The term r_g is the penalty coefficient which controls the impact the penalty component has on the entire equation. This penalty coefficient may be variable, taking different values in each generation g. $J(\mathbf{p})$ is the standard objective function (Equation 2.15) for the candidate solution $\mathbf{p}$.

A unique feature of the SRES is that it does not necessarily use the penalty component for all comparisons. Rather, it ranks the solutions while taking into account the penalty component in a stochastic manner, hence the name stochastic ranking. Given a set of parent solutions, the SRES algorithm will sort them in a bubble-sort like

manner. For each pair-wise comparison, if any of the solutions is infeasible, the probability of including the penalty component during comparison is given by a variable P_f. If $P_f = 1$, it means that the algorithm will use only the objective function J for comparison, i.e. *under-penalization*, while $P_f = 0$ means that the penalty component will be included. Simulations have shown $P_f = 0.425$ as the optimal mix of balancing the dominance between the objective and penalty functions.

Instead of taking the mid-points of two randomly selected parent solutions (a typical recombination strategy), the SRES uses each parent solution to generate a number of child solutions. The parent and children solutions are then ranked according to the scheme described above, and the best μ solutions are taken to be the parents for the next generation. Again, the stopping criteria could be a maximum number of generations, or when no better solutions can be found.

Moles et al. (2003) have assessed the efficiency of the various optimization algorithms using a pathway parameter estimation problem as a case study. They compared several deterministic and stochastic algorithms, of which SRES outperforms all other methods that are being used. However, it consumes a significant amount of computational time.

Chapter 3

Pathway Model Decomposition

Pathway models are becoming larger and more complex (Weng et al., 1999). Hence their construction and especially their parameter estimation are becoming challenging. For large pathway models, it is natural to try to decompose them into smaller components and estimate the parameters for each component separately, thus reducing the computational complexity.

Most work on pathway model decomposition emphasize the identification of functional sub-networks using the structural features of the pathway. Schuster et al. (2002) have developed a decompositional approach based on the local connectivity of metabolites to decompose the metabolic map of *M. pneumoniae* into 19 sub-networks, using which they study the relevance of each individual modules. Nayak and De (2007) have also applied a decomposition algorithm on the MAPK and calcium signaling pathways of different organisms. They then carried out a comparative study on the modules of the different species. These approaches view the pathways as static maps and do not take into account their dynamics when identifying the individual modules.

Our decompositional approach considers the forms of the reactions that drive the flow of biomolecules. Specifically, we will use the HFPN formalism and its graphical features to derive the rules that enable us to decompose the pathway model into smaller, yet *simulatable* components. These components will enable parameter estimation to be made more manageable.

3.1 Components

The basic concept underlying our decompositional approach is that of *pathway components*. Broadly speaking, the component of a pathway model is a subset of its HFPN

graph such that given the appropriate input signals, initial conditions and parameter values, the component can be simulated. Simulation in this context refers to the notion that the values (concentration levels) of the places (biomolecules) can be known at any point of time either because they are given prior to the simulation, or they progress in accordance to the differential constraints put forth by Equation 2.1. In addition, given the same initial conditions, the adding of any more pathway elements to the component should not alter the kinetics of any of the elements in the component.

Definition 3.1 (HFPN)**.** Let $H = (P, T, F, W)$ be the underlying net of the HFPN where

- P is the set of places
- T is the set of transitions
- $F \subseteq (P \times T \times \{0, 1\}) \cup (T \times P)$ is the flow relation connecting places to transitions and vice versa. The place-to-transition arcs can be either normal (solid) or read-only (dashed). By convention $(p, t, 0) \in F$ will denote a read-only arc and $(p, t, 1) \in F$ will denote a normal arc. Naturally we require that whenever $(p, t, k) \in F$ then $(p, t, 1 - k) \notin F$
- $W : F \to \mathbb{R}^+$ is the function that assigns a *weight* to the flow relation, indicating the stoichiometric coefficient of the place for the transition it is connected to. For read-only arcs, their weights will be 0.

Definition 3.2 (Component)**.** A component of an HFPN model $H = (P, T, F, W)$ is a non-empty subset C of $P \cup T \cup F$ which satisfies the following closure conditions.

$$\text{If } t \in C \begin{cases} (t,p) \in F & \to \quad p \in C \text{ and } (t,p) \in C \\ (p,t,1) \in F & \to \quad p \in C \text{ and } (p,t,1) \in C \\ (p,t,0) \in F & \to \quad (p,t,0) \in C \end{cases} \tag{3.1}$$

$$\text{If } p \in C \begin{cases} (t,p) \in F & \to \quad t \in C \text{ and } (t,p) \in C \\ (p,t,1) \in F & \to \quad t \in C \text{ and } (p,t,1) \in C \\ (p,t,0) \in F & \to \quad (p,t,0) \in C \end{cases} \tag{3.2}$$

By the definition above, there may be *open-ended* read-only arcs, i.e. test arcs with no incoming places or outgoing transitions within the same component. For incoming test

arcs with places that belong to other components, those places will act as independent inputs during the simulation of the component (Figure 3.1).

The HFPN model itself is a component, and all other components are subsets of it. The components may also share some elements.

Definition 3.3 (Minimal Component)**.** Let $C \subseteq H$ be a component. Then C is said to be *minimal* if there does not exist a proper subset C' of C which is also a component.

With the notion of minimal components, we can make a few interesting observations. First, we develop some terminology. Suppose $x, y \in P \cup T$, then a path from x to y is a sequence of net elements $x_0, x_1, \ldots, x_n$ such that

(1) $x_0 = x$ and $x_n = y$

(2) For $0 \leq j < n$, if $x_j \in P$ then $(x_j, x_{j+1}, i) \in F$ or $(x_{j+1}, x_j) \in F$ with $i \in \{0, 1\}$. If $x_j \in T$ then $(x_j, x_{j+1}) \in F$ or $(x_{j+1}, x_j, i) \in F$ with $i \in \{0, 1\}$.

If $Z \subseteq P \cup T$ and $x, y \in Z$ and $\pi = x_0, x_1, \ldots, x_n$ is a path from x to y then π is said to be a path in Z if $x_j \in Z$ for $0 \leq j \leq n$. We note that if C is a minimal component then it must be connected in that for every $x, y \in C \cap (P \cup T)$ with $x \neq y$, there must be a path in C from x to y.

Let $x, y \in F$. Then a *solid path* from x to y is a sequence of net elements $x_0, x_1, \ldots, x_n$ such that

(1) $x_0 = x$ and $x_n = y$

(2) For $0 \leq j < n$, if $x_j \in P$ then $(x_j, x_{j+1}, 1) \in F$ or $(x_{j+1}, x_j) \in F$. If $x_j \in T$ then $(x_j, x_{j+1}) \in F$ or $(x_{j+1}, x_j, 1) \in F$

The notion of a solid path in $Z \subseteq P \cup T$ is defined in the obvious way.

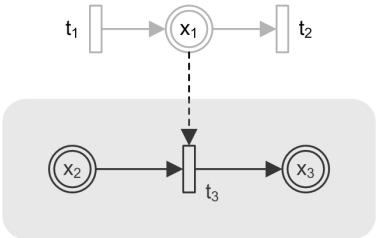


Figure 3.1: A component of a simple HFPN model consisting of the set $\{x_2, x_3, t_3, (x_1, t_3, 0), (x_2, t_3, 1), (t_3, x_3)\}$. The place x_1 is not in the component, but it serves as an independent input to transition t_3 during component simulation.

Fact 3.1. Let C be a minimal component and $(p, t, 0) \in F$ such that $p \in C$ and $t \in C$. Then there is a solid path from p to t in C.

Proof. Let C' be the least subset of C satisfying the following conditions

- $p \in C'$
- Suppose $p' \in C'$ and $t' \in C$ such that $(t', p') \in F$ then $t' \in C'$ and $(t', p') \in C'$.
- Suppose $p' \in C'$ and $t' \in C$ such that $(p', t', 1) \in F$ then $t' \in C'$ and $(p', t', 1) \in C'$.
- Suppose $t' \in C'$ and $p' \in C$ such that $(t', p') \in F$ then $p' \in C'$ and $(t', p') \in C'$.
- Suppose $t' \in C'$ and $p' \in C$ such that $(p', t', 1) \in F$ then $p' \in C'$ and $(p', t', 1) \in C'$.

Clearly, C' is a non-empty subset of C and hence $C - C'$ is a proper subset of C. We also note that $C' \cap F$ is a subset of $(P \times T \times \{1\}) \cup (T \times P)$. If $C - C'$ is an empty set then $C' = C$. This in turn implies that $t \in C'$ and hence, by the definition of C', the required solid path must exist in C. So suppose that $C'' = C - C'$ is non-empty. Then we claim that C'' must itself be a component. To see this, consider $p' \in C''$ and $(t', p') \in F$. Clearly $t' \in C$ since C is a component. Hence $t' \notin C''$ would imply that $t' \in C'$. But by the construction of C', this would imply that $p' \in C'$, contradicting $p' \in C'' = C - C'$. In a similar way, we can dispose off all other cases and conclude that C'' is indeed a component. But the minimality of C rules out $C - C'$ being a component.

□

Fact 3.2. Let C be a minimal component and $x, y \in C$ such that $x \neq y$. Then there exists a solid path in C from x to y.

Proof. Since C is minimal, then there must be a path $\pi = x_0, x_1, \ldots, x_n$ from x to y in C. We now proceed by induction on k, the number of read-only arcs encountered on π. In other words, for $0 \leq j < n$, let the size of the set $\{j \mid (x_j, x_{j+1}, 0) \in F \vee (x_{j+1}, x_j, 0) \in F\}$ be equals to k. If $k = 0$ then π itself is the required solid path. So assume $k > 0$. Let l be the least index such that $(x_l, x_{l+1}, 0) \in F$ or $(x_{l+1}, l_j, 0) \in F$. By Fact 3.1, there is a solid path $u_0, u_1, \ldots, u_m$ from x_l to x_{l+1} in C. Hence $\pi' = x_0, x_1, \ldots, x_l = u_0, u_1, \ldots, u_m = x_{l+1}, \ldots, x_n$ is also a path in C from x to y. Further, the number of read-only arcs encountered on π' is $k - 1$. The result now follows from the induction hypothesis. □

Fact 3.3. Let C and C' be two minimal components such that $C \neq C'$, then $C \cap C' \subseteq P \times T \times \{0\}$.

Proof. Suppose $C \cap C'$ contains a net element x. Since $C \neq C'$ there exists an element $y \in C - C'$. By Fact 3.2 there must exists a solid path π from x to y (in C). However, by the definition of a component $x \in C'$ would imply, by an easy induction on the length of π, that $y \in C'$ too. This is a contradiction. By a similar argument, we can also show that $C \cap C' \cap (P \times T \times \{1\}) = \phi$. □

For a component to be simulatable, all the transitions that affect the rate of change of the places must be within the same component. The observations above showed that other than read-only arcs, no elements are shared between the minimal components of an HFPN model. Each of them minimally satisfies the simulation requirement, and they can be simulated independently, given the proper inputs.
By identifying the minimal components of an HFPN model, we are able to induce a graph that depicts the topological arrangement of its minimal components, i.e. a *Component graph*.

Definition 3.4 (Component Graph)**.** The Component graph of H, denoted as CG_H is the directed graph (C, A) where

- C is the set of minimal components of H
- A is the least subset of $C \times C$ satisfying:
  Suppose $C', C'' \in C$ and there exists $(p, t, 0) \in F$ such that $p \in C' - C''$, $t \in C'' - C'$ and $(p, t, 0) \in C' \cap C''$, then $(C', C'') \in A$.

An example of the p53 pathway model (Doi et al., 2006a) being decomposed into its set of minimal components is illustrated in Figure 3.2. The components are connected to one another via test arcs (Fact 3.3). Taking each minimal component as a node, the HFPN can be represented as a component graph.

3.2 Decompositional Approach to Parameter Estimation

To estimate the parameters of a large signaling pathway, we first decompose it to obtain the set of minimal components and the component graph. Depending on the locations and availability of experimental data, we then order and group certain minimal components together. Parameter estimation is performed on such components in an iterative

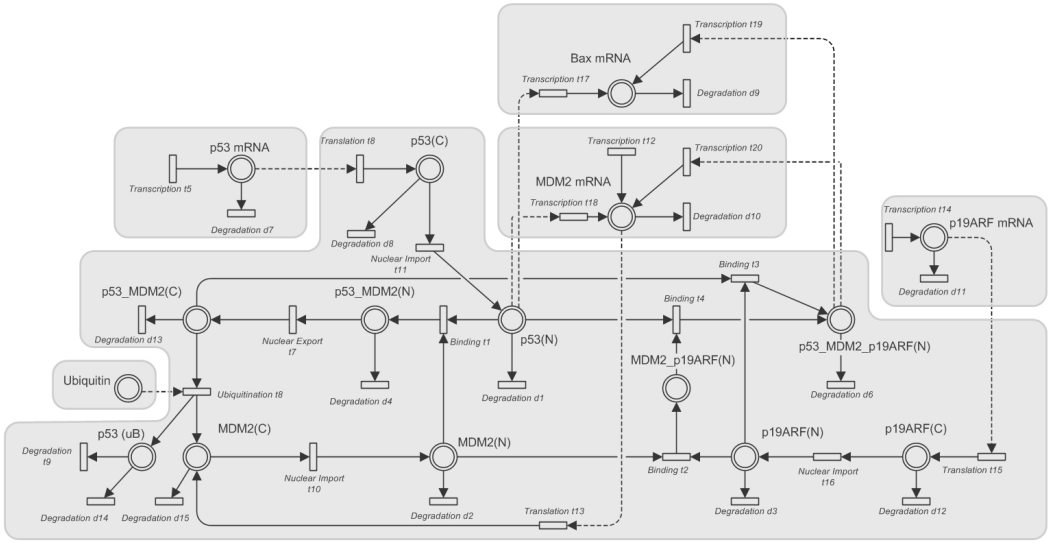


Figure 3.2: An example of the p53 pathway model being decomposed into its minimal components. Taking each minimal component (grey box) as a node, we can represent the HFPN as a component graph.

fashion. We select the smallest component that can be estimated and use the SRES algorithm to estimate its parameters. The simulation results and parameters of the estimated components are then stored for simulating the downstream components. This process is repeated until all the parameters have been estimated, or there is no more data that can be utilized. This way, the search space dimension is reduced and fewer computations are required in each iteration as we only need to simulate the component we are estimating.

3.2.1 Decomposition

The size and number of minimal components that can be obtained depends on the *coupling* of the pathway elements. In the worse case, all the pathway elements are dependent on one another so that the entire pathway is a minimal component. The computations to decompose the pathway is shown in Algorithms 3.1 and 3.2.

3.2.2 Grouping of the Components

Previously we have established the fact that given appropriate input signals, initial conditions and parameter values, a component can be simulated. However for the purpose of parameter estimation, this is a necessary but not sufficient condition. As

Algorithm 3.1: Component(e)

Data: An element $e \in P \cup T$
Result: The minimal component C which contains e
begin
  $C \leftarrow \{e\}$
  foreach $e' \in$ Neighbor(e) **do**
    if $e' \notin C$ **then**
      if $e \in T$ **then** `// e is a transition`
        if $(e, e') \in F$ **then**
          $C \leftarrow C \cup \{e', (e, e')\} \cup$ Component(e')
        else if $(e', e, 1) \in F$ **then**
          $C \leftarrow C \cup \{e', (e', e, 1)\} \cup$ Component(e')
        else if $(e', e, 0) \in F$ **then**
          $C \leftarrow C \cup \{(e', e, 0)\}$
        end
      else `// e is a place`
        if $(e', e) \in F$ **then**
          $C \leftarrow C \cup \{e', (e', e)\} \cup$ Component(e')
        else if $(e, e', 1) \in F$ **then**
          $C \leftarrow C \cup \{e', (e, e', 1)\} \cup$ Component(e')
        else if $(e, e', 0) \in F$ **then**
          $C \leftarrow C \cup \{(e, e', 0)\}$
        end
      end
    end
  end
  return C
end

Algorithm 3.2: Decompose(H)

Data: The HFPN model $H = (P, T, F, W)$
Result: The set of minimal components of H
begin
  $C \leftarrow \phi$
  foreach $e \in P \cup T$ **do**
    if $\forall C' \in C, e \notin C'$ **then**
      $C' \leftarrow$ Component(e)
      $C \leftarrow C \cup \{C'\}$
    end
  end
  return C
end

discussed in Section 2.3, parameter estimation requires the presence of experimental data in order to assess the goodness of fit for the current parameter guesses. In addition, the dynamics of the observed place must be influenced by at least one transition within the component, i.e. the place is not an independent input to the component. This leads us to the notion of *estimatable* components.

Definition 3.5 (Estimatable Component)**.** Let $C \subseteq H$ be a component. Then C is said to be *estimatable* if there exists a place $p \in C$ such that experimental data is available about the species associated with p, and there is a transition $t \in C$ such that $(p, t, 1) \in C$ or $(t, p) \in C$.

It is not always the case that every individual minimal component of a pathway can be estimated separately. Some of the minimal components need to be grouped together into estimatable components. Our approach does not fix a grouping and ordering of the components up-front. Instead, we group the estimatable components iteratively. In each iteration, we form an estimatable component, estimate its parameters, and update the HFPN model before grouping the next component.

To do so, we develop a *coloring* scheme for the components to denote the completeness of data and parameters available. We first assign *colors* to each minimal component of the component graph. We assume that we have experimental data that has been produced by K experiments, possibly conducted under different conditions. With each component we associate a K-dimensional color vector. Suppose the jth experiment produces time series data and/or steady state concentration data such that the associated component is estimatable. Then the component is defined to be *grey*. Otherwise it is defined to be *white*. For a component where all its parameters are known, it is defined to be *black*.

Suppose we have a colored component graph $CG_H = (C, A)$ with the structure shown in Figure 3.3. We first define the functions $\text{Col} : C \to \{\text{Black}, \text{Grey}, \text{White}\}$ and $\text{pred} : C \to \{C\}$. Col() maps a component to its corresponding color. For instance, $\text{Col}(C_6) = \text{Grey}$ and $\text{Col}(C_3) = \text{Black}$. The function pred() returns the set of *predecessor* components of a given component. Using the component graph as an illustration, $\text{pred}(C_5)$ will return the set $\{C_1, C_2, C_3, C_4, C_6\}$.

Without loss of generality, we assume that all the initial concentrations are known. For future cases where there are unknown initial concentrations for some molecule species, they can be treated as parameters to be estimated. We shall first discuss the case where the component graph is acyclic. Later on we will discuss how to apply our method on component graphs with cycles.

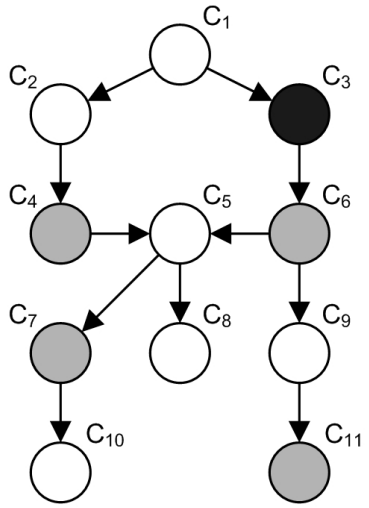


Figure 3.3: A colored component graph.

From the colored component graph, we pick a grey minimal component C'. Then the corresponding estimatable component - which we denote as C_{est} - is defined by the least set of components that satisfy the following conditions:

- $C' \subseteq C_{\text{est}}$
- If $C', C'', C''' \in CG_H$, $C' \subset C_{\text{est}}$, $(C'', C') \in A$,
  - $\text{Col}(C'') \neq \text{Black} \rightarrow C'' \subset C_{\text{est}}$
  - $\text{Col}(C'') = \text{Black}, C''' \in pred(C'')$ and $\text{Col}(C''') \neq \text{Black} \rightarrow C'' \subset C_{\text{est}}$

Like the decomposition algorithm, the above conditions specify a back tracing algorithm on the component graph to form an estimatable component. We start from a grey minimal component up to the first black component which has no non-black predecessor. For a component graph, there may be several estimatable components. We choose one that has the least number of unknown parameters while maximizing the amount of experimental data available. Thus, choosing an estimatable component can be a potentially non-trivial task. A number of strategies can be adapted to ease this task but we will not address them here. Instead, we shall simply choose one that has the smallest number of unknown parameters. After obtaining the estimatable component, we can use any available algorithms (Section 2.3) to estimate its parameters. Thereafter, we color it black and repeat the process. An example of the grouping and ordering of the components is shown in Figure 3.4

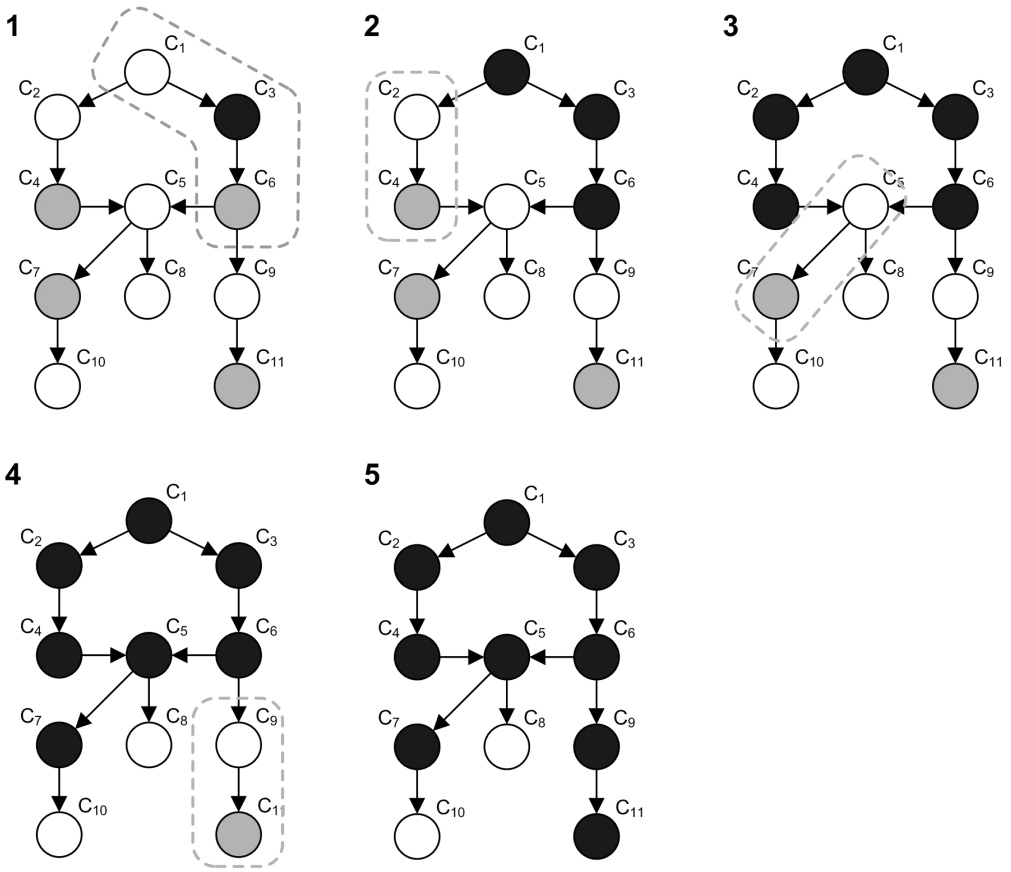


Figure 3.4: A possible grouping and estimation order, given a colored component graph. The white leaf components C_8 and C_{10} are not able to be estimated as there are no experimental data to provide the constraints for their parameters.

We can increase the computational efficiency by pre-computing traces for black components that do not have any non-black components as their predecessors. These black components appear as we perform parameter estimation using our decomposition approach on the pathway. Our aim is hence to color all the components in the component graph black, i.e. estimate for all the unknown parameters. Note that for leaf components that are white, there is no way to estimate their parameters since no data is available for fitting.

Cyclic Component Graphs

Our approach relies on the requirement of the component graph to be acyclic, since it also captures the simulation dependency between the individual components. If the graph is strongly connected, such that there exists a *directed* path between any two components in the component graph, none of the components can be simulated independently. This results in the entire pathway model having to be estimated as a whole - a situation no different from using current parameter estimation techniques.

This issue can be dealt with by a simple HFPN modification, taking into account the available experimental data. The basic intuition for this modification is: for places that act as inputs to read-only arcs, those with time series data can be approximated by a *spline* running through the data points. These splines can be instantiated as *pseudo*-place inputs to the read-only arcs during simulation. If the place and the connected transition belong to two separate components, the components are then decoupled. This process is repeated until we have an acyclic component graph, or there are no more such spline approximations.

3.3 Case Study

In this section, we present our decompositional approach to parameter estimation on the Akt-MAPK signaling pathway. This pathway is being studied by our collaborators - Dr Marie-Véronique Clément and her group in the Department of Biochemistry, NUS. They provided the pathway structure as well as the experimental data used for fitting the model.

3.3.1 The Akt Pathway

The serine/threonine kinase Akt plays an important role in the pro-survival signaling pathways. Triggered by various extracellular factors, the Akt can regulate the activities

of several downstream targets. Active Akt can sequester the pro-apoptotic protein Bad in the cytoplasm, such that its usual binding partner protein - Bcl-XL - is free to inhibit the activity of the cell-death protease caspase-9. In addition, Akt can also prevent the nuclear localization of the Forkhead transcription factors, promote the degradation of the kinase IκB and increase the activity of the NFκB transcription factor (Khwaja, 1999). As such, Akt has been recognized as a major factor that is involved in several types of cancer.

The activation of the Akt signaling pathway is a multi-step process (Bellacosa et al., 1998). When ligands such as fibroblast growth factors, epidermal growth factors and insulin bind to their respective membrane receptors, the cytosolic domains of the receptors will undergo conformational changes, allowing them to act as scaffolds for certain types of proteins in the cell. Phosphoinositide 3-kinase (PI3K) is one such protein that gets recruited. As a result, its catalytic subunit will be activated. The active PI3K will then phosphorylate the membrane phospholipid phosphatidylinositol-4,5-biphosphate (PIP_2) at the 3-OH position to form phosphatidylinositol-3,4,5-triphosphate (PIP_3). This is tightly regulated by the phosphatase and tensin homolog (PTEN), which removes the phosphate group from the same position. PIP_3 recruits Akt and the phosphoinositide-dependent kinase-1 (PDK1) to the plasma membrane, allowing the activation of Akt by PDK1. This activation of Akt is achieved by a sequential phosphorylation at its threonine residue 308 (Thr^{308}) and serine residue 473 (Ser^{473}) by PDK1 and an unknown kinase (currently named PDK2) respectively (Nicholson and

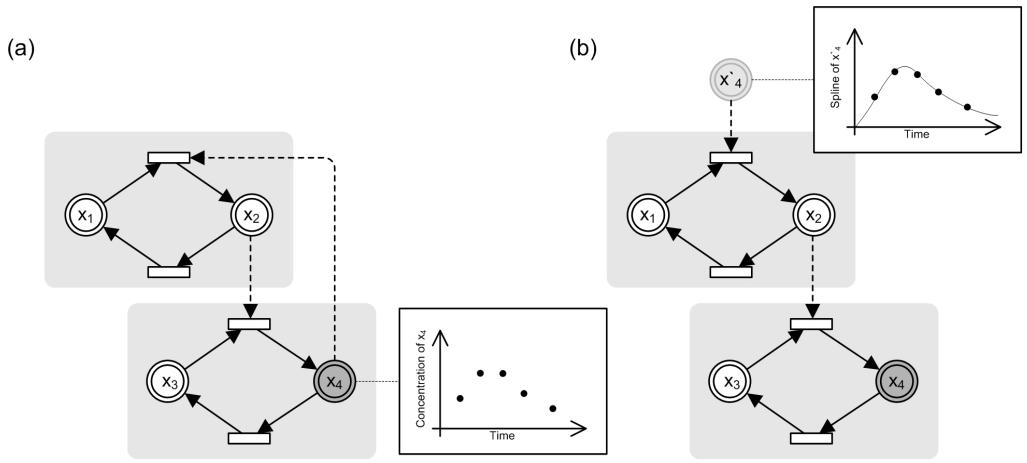


Figure 3.5: (a) A cyclic component graph. (b) Given that the place x_4 has time series experimental data, a spline can be fitted to it, producing the pseudo-place x'_4, which is used for the simulation of x_1 and x_2.

Anderson, 2002). Active Akt then further phosphorylates and activates its downstream targets such as the Forkhead transcription factor and glycogen synthase kinase 3β (GSK-3β).

An important molecule target for Akt signaling is Bad, a protein that regulates apoptosis (programmed cell death). Bad can bind to the anti-apoptotic proteins Bcl-2 and Bcl-XL, allowing the pro-apoptotic protein Bax to oligomerize at the mitochondria and promote the release of cytochrome c into the cytosol. This would lead to the activation of caspases and cell death. Upon phosphorylation by Akt at the Ser^{136} residue, Bad is sequestered in the cytosol by 14-3-3 proteins. Thus Bcl-2 and Bcl-XL can bind to Bax, hence preventing the release of cytochrome c and inhibiting apoptosis. Constitutive Akt signaling promotes cell survival and proliferation, leading to the formation of tumors.

MAPK Cross-talk

The significance of the Akt signaling pathway lies not only in the several cellular functions it regulates, but also in its interactions with other pathways (Heldin, 2001). The mitogen-activated protein kinase (MAPK) signaling cascade is one such pathway that is influenced by components of the Akt pathway.

The MAPK pathway is a highly conserved pathway that is linked to mitogenic responses and cell proliferation. It can be activated by a wide range of growth factors and hormones. Like the Akt, the active MAPK has several target molecules. Some of the signals that activate the Akt pathway can also activate the MAPK pathway. Upon activation, the tyrosine residues on the cytoplasmic domain of the cell membrane receptor is phosphorylated, serving as docking sites for proteins such as the growth factor receptor-bound protein 2 (Grb2) and the protein Son of Sevenless (SOS). The exchange factor SOS then replaces guanosine 5'-diphosphate (GDP) on the Ras protein with guanosine 5'-triphosphate (GTP), thus activating it. Activated Ras then binds to the protein Raf, triggering a wave of downstream phosphorylation where Raf activates the MAPK kinase (MEK) which in turn activates p44/42 MAPK (ERK).

Recent studies have shown that the Akt pathway can regulate the MAPK pathway by competitively phosphorylating Raf at Ser^{259} (Moelling et al., 2002), preventing further activation. However, Akt-MAPK regulation is not entirely inhibitory. PI3K has been shown to activate Raf via the intermediate protein Pak-1 (Chaudhary et al., 2000). Recently, Sato et al. (2004) have shown that PDK1 is involved in the activation of MEK. Moreover, our data also shows that the repression of PDK1 gene expression using small interference RNA (siRNA) will subsequently lead to a decrease in activated

MEK and ERK in a prostate cancer cell model (Teong and Clement, 2006). These phenomena lead to our hypothesis that PDK1 could be involved in the activation of MEK by phosphorylating it. The western blot data that depicts this phenomena is shown in Figure 3.6.

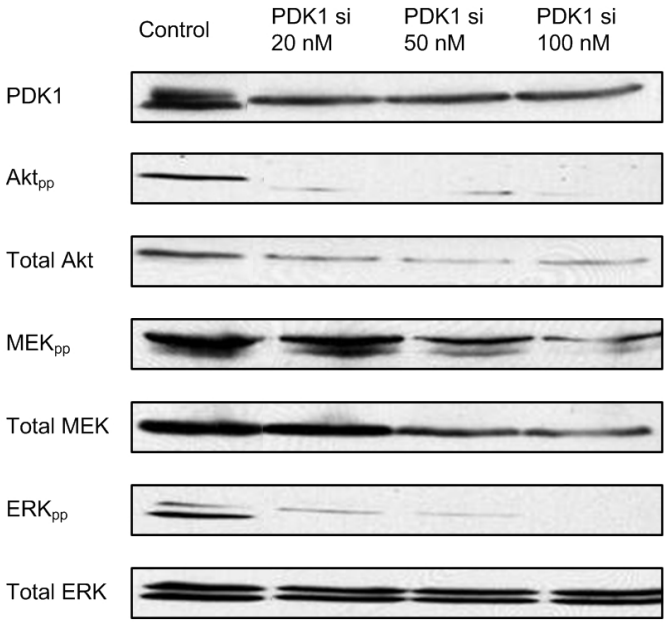


Figure 3.6: Steady-state western blot data for knock-down PDK1 experiment (Teong and Clement, 2006). Levels of PDK1 in the cells are reduced by adding PDK1 siRNA in increasing concentration - 20 nM, 50 nM and 100 nM. The total concentration of Akt, MEK and ERK remain relatively constant throughout. However their activities show a steady decrease, indicating some regulation of PDK1 on MEK and ERK.

The schematic describing the Akt pathway, its interactions with the MAPK pathway, and their downstream targets is shown in Figure 3.7. The hypothesized interaction is indicated by a bold arrow in the figure.

3.3.2 The HFPN Model of the Akt-MAPK Pathway

We modeled the Akt pathway and the MAPK pathway, as well as the hypothesized cross-talk interaction using the HFPN formalism. The full model is shown in Figure 3.8. The parameters associated with the transitions and their estimated values are shown in Table 3.1. Details of the experimental set-up and the initial conditions of the protein species are given in Appendix A. In Table 3.1, the four parameters K_{m10}, K_{m12}, k_{44} and k_{45} have values being taken from literature and are marked with an asterisk (*).

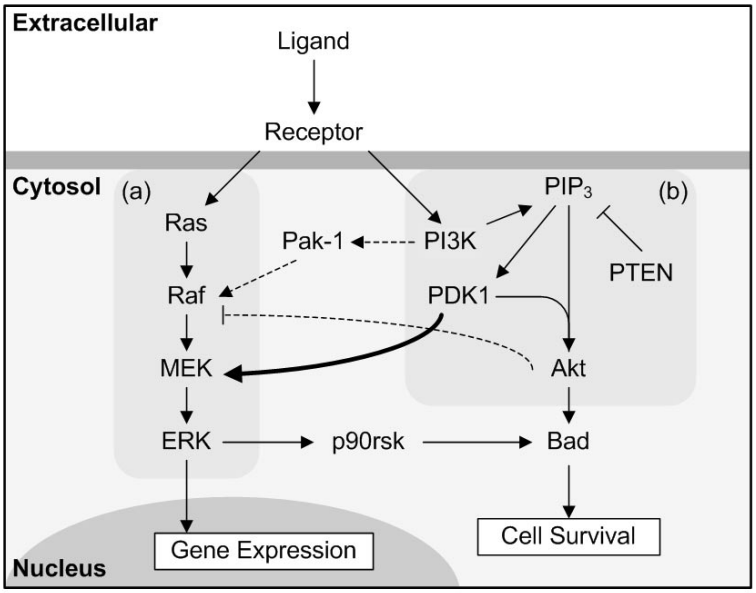


Figure 3.7: Schematic of the (a) MAPK pathway, (b) Akt pathway and their cross-talk interactions. Known cross-talk interactions are marked with dashed arrows while the hypothesized interaction is marked with a bold arrow.

The remaining parameters are estimated using our decompositional approach.

The MAPK pathway is depicted by Figures 3.8B, C and D. After activation by the cell membrane receptors, the protein Ras will catalyze the phosphorylation of Raf. Phosphorylated Raf, denoted as Raf_p will then phosphorylate MEK at two sites, forming the doubly phosphorylated MEK_{pp} (Figure 3.8C). Finally, ERK will be phosphorylated by MEK_{pp} in the same manner, as shown in Figure 3.8D to form ERK_{pp}. In our model, there are three possible cross-talk interactions: (i) Active PI3K can upregulate the phosphorylation of Raf via Pak-1 (Reactions 48 and 18). (ii) Akt can inhibit Raf activity (Reaction 20), and (iii) PDK1 can affect MEK phosphorylation (Reactions 22 and 25) by the hypothesized interactions. The protein PP2A is an ubiquitous phosphatase which reverses the action of several kinases in our pathway. Hence it is not considered as exclusively belonging to the Akt pathway or the MAPK pathway. Figure 3.8F models the activity of the Bcl2 family members, which include the proteins Bcl2, Bad and Bax. As mentioned in the previous section, these proteins play important roles in regulating apoptosis. It has also been shown that ERK can regulate Bad phosphorylation through the activation of the protein P90RSK, shown in Figure 3.8E.

Our model consists of 44 places and 51 transitions. It has 88 parameters, of which 84 are unknown and have to be estimated. For most of the transitions, their dynamics

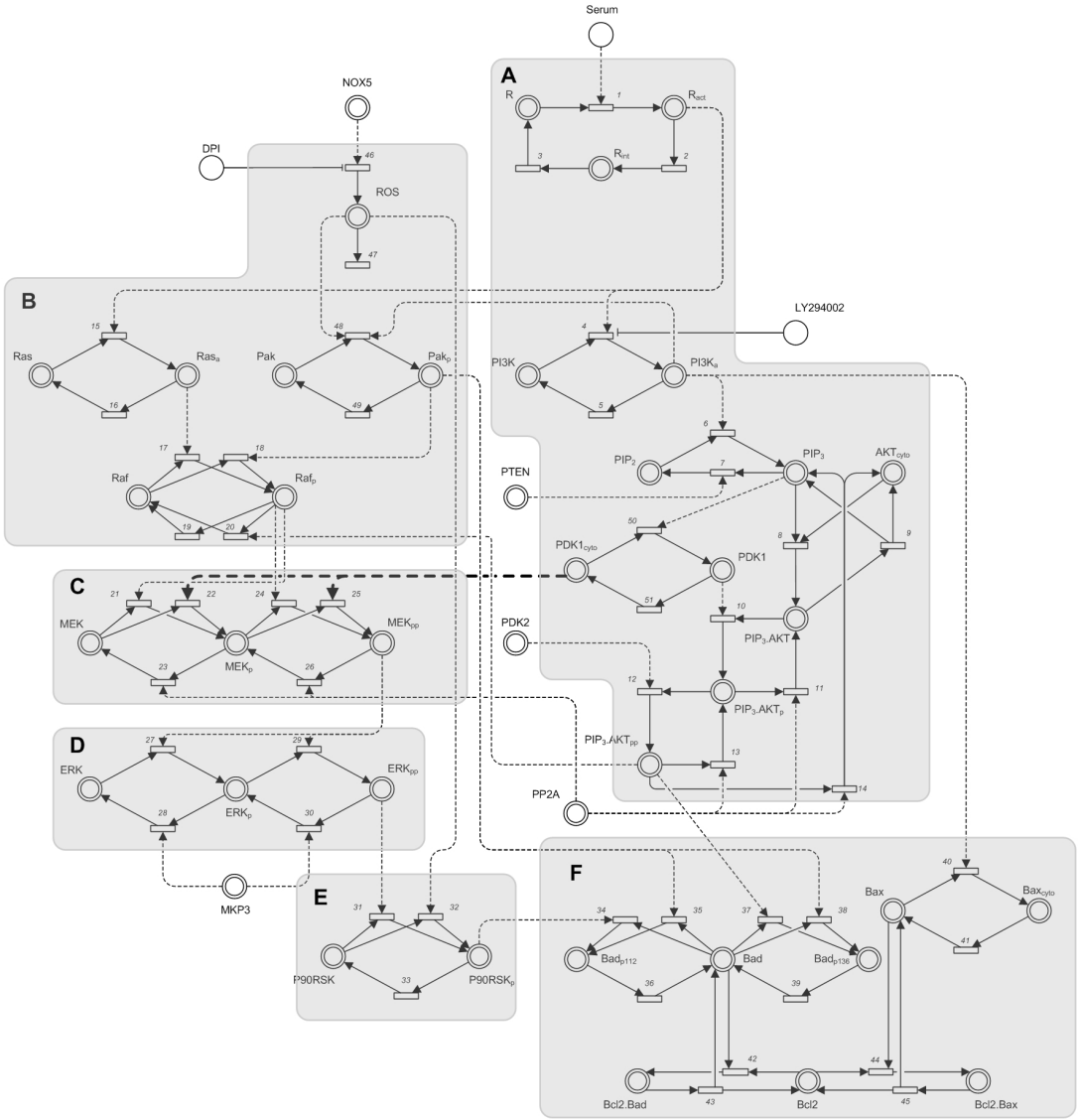


Figure 3.8: HFPN model of the Akt and MAPK pathways. The hypothesized cross-talk interaction is emphasized by bold test arcs. The grey boxes A - F represent the components after decomposition of the pathway and grouping the minimal components together based on the availability of experimental data. Certain places such as the PP2A are not included in any of the components as there are no dynamics, i.e. transitions, associated with them. Hence they can be taken as constants.

No	Rate Equation	Parameter	
1	$k_1[R]$	$k_1 = 0.01$	
2	$k_2[R_{act}]$	$k_2 = 0.002$	
3	$k_3[R_{int}]$	$k_3 = 0.001$	
4	$k_4[R_{act}][PI3K]/(K_{m4} + [PI3K])$	$k_4 = 0.3$	$K_{m4} = 78$
5	$V_5[PI3K_a]/(K_{m5} + [PI3K_a])$	$V_5 = 46.2$	$K_{m5} = 117$
6	$k_6[PI3K_a][PIP_2]/(K_{m6} + [PIP_2])$	$k_6 = 0.05$	$K_{m6} = 6170$
7	$k_7[PTEN][PIP_3]/(K_{m7} + [PIP_3])$	$k_7 = 5.5$	$K_{m7} = 80.9$
8	$k_8[PIP_3][Akt_{cyto}]$	$k_8 = 0.045$	
9	$k_9[PIP_3.Akt]$	$k_9 = 0.089$	
10	$k_{10}[PDK1][PIP_3.Akt]/(K_{m10} + [PIP_3.Akt])$	$k_{10} = 20$	$K_{m10} = 80000^*$
11	$k_{11}[PP2A][PIP_3.Akt_p]/(K_{m11} + [PIP_3.Akt_p])$	$k_{11} = 0.037$	$K_{m11} = 8800$
12	$k_{12}[PDK2][PIP_3.Akt_p]/(K_{m12} + [PIP_3.Akt_p])$	$k_{12} = 20$	$K_{m12} = 80000^*$
13	$k_{13}[PP2A][PIP_3.Akt_{pp}]/(K_{m13} + [PIP_3.Akt_{pp}])$	$k_{13} = 0.04$	$K_{m13} = 48000$
14	$k_{14}[PP2A][PIP_3.Akt_{pp}]/(K_{m14} + [PIP_3.Akt_{pp}])$	$k_{14} = 0.163$	$K_{m14} = 48000$
15	$k_{15}[R_{act}][Ras]/(K_{m15} + [Ras])$	$k_{15} = 50$	$K_{m15} = 20000$
16	$V_{16}[Ras_a]/(K_{m16} + [Ras_a])$	$V_{16} = 15000$	$K_{m16} = 7260$
17	$k_{17}[Ras_a][Raf]/(K_{m17} + [Raf])$	$k_{17} = 0.09$	$K_{m17} = 50$
18	$k_{18}[Pak_p][Raf]/(K_{m18} + [Raf])$	$k_{18} = 0.183$	$K_{m18} = 500$
19	$V_{19}[Raf_p]/(K_{m19} + [Raf_p])$	$V_{19} = 78$	$K_{m19} = 30$
20	$k_{20}[PIP_3.Akt_{pp}][Raf_p]/(Km_{20} + [Raf_p])$	$k_{20} = 0.1$	$Km_{20} = 13.2$
21	$k_{21}[Raf_p][MEK]/(K_{m21} + [MEK])$	$k_{21} = 5.6$	$K_{m21} = 7200$
22	$k_{22}[PDK1_{cyto}][MEK]/(K_{m22} + [MEK])$	$k_{22} = 0.04$	$K_{m22} = 2600$
23	$k_{23}[PP2A][MEK_p]/(K_{m23} + [MEK_p])$	$k_{23} = 0.45$	$K_{m23} = 1250$
24	$k_{24}[Raf_p][MEK_p]/(K_{m24} + [MEK_p])$	$k_{24} = 5.17$	$K_{m24} = 24500$
25	$k_{25}[PDK1_{cyto}][MEK_p]/(K_{m25} + [MEK_p])$	$k_{25} = 0.05$	$K_{m25} = 2150$
26	$k_{26}[PP2A][MEK_{pp}]/(K_{m26} + [MEK_{pp}])$	$k_{26} = 0.4$	$K_{m26} = 4316$
27	$k_{27}[MEK_{pp}][ERK]/(K_{m27} + [ERK])$	$k_{27} = 0.089$	$K_{m27} = 52000$
28	$k_{28}[MKP3][ERK_p]/(K_{m28} + [ERK_p])$	$k_{28} = 30$	$K_{m28} = 160$
29	$k_{29}[MEK_{pp}][ERK_p]/(K_{m29} + [ERK_p])$	$k_{29} = 0.0308$	$K_{m29} = 55000$
30	$k_{30}[MKP3][ERK_{pp}]/(K_{m30} + [ERK_{pp}])$	$k_{30} = 32$	$K_{m30} = 60$
31	$k_{31}[ERK_{pp}][P90RSK]/(K_{m31} + [P90RSK])$	$k_{31} = 0.0017$	$K_{m31} = 97.6$
32	$k_{32}[ROS][P90RSK]/(K_{m32} + [P90RSK])$	$k_{32} = 0.76$	$K_{m32} = 181$
33	$V_{33}[P90RSK_p]/(K_{m33} + [P90RSK_p])$	$V_{33} = 468$	$K_{m33} = 2.8$
34	$k_{34}[P90RSK_p][Bad]/(K_{m34} + [Bad])$	$k_{34} = 0.798$	$K_{m34} = 10$
35	$k_{35}[Pak_p][Bad]/(K_{m35} + [Bad])$	$k_{35} = 0.04$	$K_{m35} = 30000$
36	$V_{36}[Bad_{p112}]/(K_{m36} + [Bad_{p112}])$	$V_{36} = 821$	$K_{m36} = 43300$
37	$k_{37}[PIP_3.Akt_{pp}][Bad]/(K_{m37} + [Bad])$	$k_{37} = 0.397$	$K_{m37} = 20700$
38	$k_{38}[Pak_p][Bad]/(K_{m38} + [Bad])$	$k_{38} = 0.04$	$K_{m38} = 30000$
39	$V_{39}[Bad_{p136}]/(K_{m39} + [Bad_{p136}])$	$V_{39} = 821$	$K_{m39} = 43300$
40	$k_{40}[PI3K_a][Bax]/(K_{m40} + [Bax])$	$k_{40} = 0.0659$	$K_{m40} = 42000$
41	$k_{41}[Bax_{cyto}]$	$k_{41} = 0.0148$	
42	$k_{42}[Bad][Bcl2]$	$k_{42} = 0.0561$	
43	$k_{43}[Bcl2.Bad]$	$k_{43} = 0.0624$	
44	$k_{44}[Bax][Bcl2]$	$k_{44} = 0.002^*$	
45	$k_{45}[Bcl2.Bax]$	$k_{45} = 0.02^*$	
46	$k_{46}[NOX5]$	$k_{46} = 0.00038$	
47	$k_{47}[ROS]$	$k_{47} = 0.0155$	
48	$k_{48}[ROS][PI3K_p][Pak]/(K_{m48} + [Pak])$	$k_{48} = 0.14$	$K_{m48} = 482$
49	$V_{49}[Pak_p]/(K_{m49} + [Pak_p])$	$V_{49} = 83000$	$K_{m49} = 29100$
50	$k_{50}[PIP_3][PDK1_{cyto}]$	$k_{50} = 0.0007$	
51	$k_{51}[PDK1]$	$k_{51} = 0.98$	

Table 3.1: Rate reactions and their associated parameters. Concentration of the molecular species are denoted by the square braces [ ]. The Michaelis-Menton constants (K_m) are given in nM. The maximal rate constants (V) are expressed in nM.s^{-1}. The first order and second order rate constants (k) are given in s^{-1} and nM^{-1}.s^{-1} respectively.

are driven by Michaelis-Menton equations (e.g. Reactions 4, 5, 6). The rest are either association/dissociation reactions (e.g. Reactions 42, 43) or synthesis/degradation reactions (e.g. Reactions 46, 47) whose rates are governed by the mass-action laws. Each of these reactions has one or two parameters associated with them.

3.3.3 Results

We now describe our results on parameter estimation and on the hypothesized Akt-MAPK cross-talk mechanism (Section 3.3.1).

Parameter Estimation

Using the method described in the previous section, we performed parameter estimation for the Akt-MAPK pathway. Of the 44 places, 6 of them have observed data - PIP_3.Akt_{pp}, Raf_p, MEK_{pp}, ERK_{pp}, $P90RSK_p$ and Bad_{p112}. Data on PDK1 and PP2A was used as the initial conditions and hence was not considered part of the data used for fitting. The data for the estimation problem was obtained from 27 experiments, of which 10 provided time-series western blot data and 17 provided steady-state data. The 27 experiments were performed under 18 different initial conditions. We used data from 23 experiments as inputs to our parameter estimation procedure and reserved data from 4 time-series experiments to validate the results of estimation. The data reserved for validation was not revealed to the estimation procedure.

The Akt-MAPK pathways consist of 51 reactions and a total of 88 parameters, of which 84 are unknown. We assume that the initial conditions for all the places are known, and this is supported partly by experimental data. For other situations where not all the initial values are known, the missing protein concentrations can be treated as parameters to be estimated as well.

We ran the estimation procedure on a Pentium 4 PC with a 2.8GHz processor and 2.5GB memory. Using the back tracing algorithm from the observed places (Algorithms 3.1 and 3.2), our procedure decomposed the pathway into 6 components (Figure 3.8A-F). The average time to estimate the parameters for each component was about 3 hours, and the total time to estimate all the parameters was about 18 hours.

For comparison, we also tried to solve the same parameter estimation problem using the global approach; in other words we tried to estimate all the parameters together using the evolutionary search method. On the same computational platform, after running for 4 days, the global method failed to produce a set of parameters that can produce reasonable simulation results. See Figure 3.9 for a comparison.

An obvious measure to assess the accuracy of the parameters and the reliability of the parameter estimation method is the deviation of the simulation results from experimental data, i.e. Equation 2.15. Figure 3.10 shows the simulation results of both MEK and ERK activity and Figure 3.11 shows the Bad phosphorylation levels for the experiment where the cells are treated with diphenyleneiodonium (DPI), a NADPH oxidase inhibitor, in the presence of serum. Figure 3.10 shows that the match between the simulation results and experimental data is good, though not perfect. Given the limited, noisy data available and the high dimensionality of the search space, these results represent a reasonable first step. We expect that as we generate better data both in terms of quantity and quality, a better match will be obtained. Longer running time for the estimation procedure may also potentially help.

Among the results obtained, we indeed have cases in which the match between the simulation results and experimental data is not good. Figure 3.11 shows the simulation results and experimental data for phosphorylated Bad. There is a systematic, constant

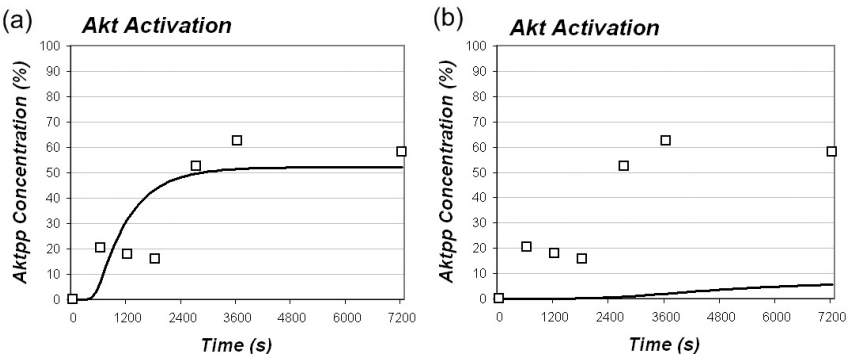


Figure 3.9: Comparison of the simulation results of Akt against experimental data using the parameters estimated with (a) the decompositional method and (b) the conventional method ('□' - experimental data points, '—' simulation profiles).

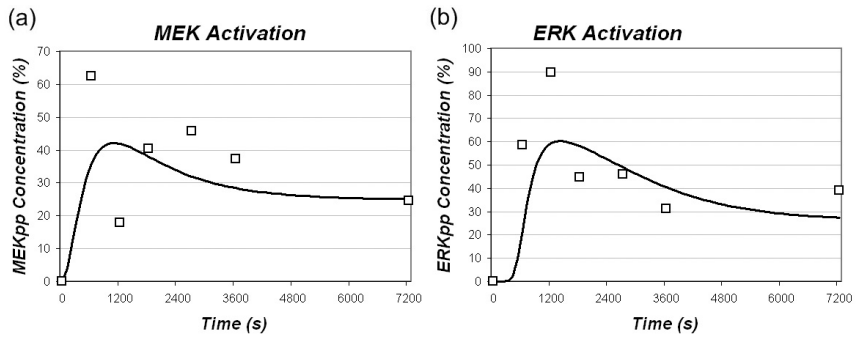


Figure 3.10: Simulation profiles of (a) MEK and (b) ERK activation levels.

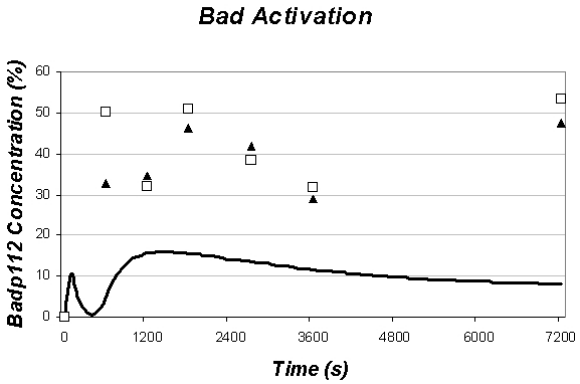


Figure 3.11: Simulation profile of Bad activation levels compared to experimental data. '□' - Data points for experiments where the cells are treated with DPI. '▲' - Data points for experiments with normal activation.

difference between them. From a static analysis of the model in Figure 3.8, the inhibition of the production of superoxide (ROS) by treatment with DPI will propagate downstream and we would expect the level of phosphorylated Bad at Ser_{112} to decrease (Figure 3.8F). However, the experimental data points for activated Bad in Figure 3.11 are similar to those from the cells which have not been treated with DPI. This difference could be due to unknown reactions and hence needs to be further investigated.

COPASI Implementation

To verify our approach and implementation of the decompositional method to parameter estimation, a separate study was carried out using the same pathway, decomposition rules and experimental data (Mehta, 2007). However, the simulation was performed using the open source software COPASI (Hoops et al., 2006). Parameter estimation of the individual components was performed using the SRES algorithm, which was included in COPASI.

Using the objective function in COPASI, the decompositional approach for parameter estimation managed to obtain an overall score of $45,335.3$, as compared to the score of 1.24071×10^7 when estimating for the entire pathway as a single component. This further verifies the efficacy of our approach for parameter estimation.

3.4 Discussion

In this chapter, we have introduced the concept of components for pathway models based on the HFPN formalism. Using the component graph, we are able to see the dependencies between the components. We then exploit this component graph and the locations of the experimental data to perform parameter estimation on its individual components. By doing so, we gain significant improvement in efficiency due to the reduction of the dimension of the parameter search space, as well as by simulating a smaller portion of the pathway at any given time. At the same time, we have the choice of using different optimization algorithms within a component, depending on the features of the component solution landscape.

However, in our approach, the order of estimation is non-deterministic, given that there is a choice to be made when deciding which component is to be estimated first. Hence the quality of the solutions may vary. In addition, due to noise in experimental data, and the multi-modal nature of the solution landscape, there is a possibility that the estimates we obtain from an estimatable component are sub-optimal globally. The main reason for this deficiency is that once we fixed the parameters for a component after estimation, we do not return to update it as the algorithm proceeds down the component graph. This may pose a problem as experimental data from the downstream components may possibly aid in constraining the search space for the upstream components. The next chapter addresses this issue by advocating a probabilistic approach that allows information from the components that are processed later in the algorithm to propagate throughout the component graph, updating all the other parameters in the process.

Chapter 4

Pathway Composition and Reconciliation of Parameter Estimates

In our decompositional approach, during grouping and selection (Section 3.2.2), we choose the component that has the least number of unknown parameters and performed parameter estimation on it. Thereafter we color the estimated component black and repeat the process. Once the parameters of a component have been fixed, they will be used for the estimation of subsequent components. Due to the noisy and multimodal nature of the solution landscape, one cannot assume that the problem has an *optimal substructure* property, i.e. the estimates for the individual components may not be the best globally.

During the grouping of the minimal components into larger ones, there might be components that overlap one another, i.e. sharing some parameters and molecule species. Again, we choose to estimate the shared parameters in the context of the component that has the least number of total unknown parameters. However, this strategy may not work for all cases as we do not know *a priori* which component has more reliable experimental data for parameter estimation. For instance, consider the pathway in Figure 4.1. Using our decompositional technique, the pathway can be decomposed into overlapping left and right components. The parameters associated with the transitions t_1 and t_2 are shared between the two components. Estimating the parameters of t_1 to t_4, followed by t_5 and t_6 may possibly result in differing estimates for the parameters of t_1 and t_2 when the order is switched. Alternatively, one can perform estimation on the overlapping components individually. However, this approach will

result in conflicting estimates for the shared parameters, with no principled way to reconcile or update them.

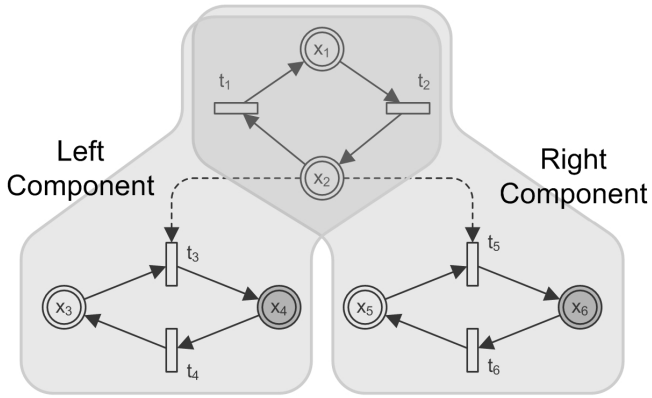


Figure 4.1: An example of a pathway that can be decomposed into two overlapping components. Rate parameters associated with the transitions t_1 and t_2 are shared between the two components. Grey places denote molecule species that have experimental data.

Overlapping pathway components may also result from the independent modeling of biological pathways. The pathway models being studied today are a result of cumulative work on several smaller sub-units. Pathways previously built and studied in isolation are now being pieced together (Christopher et al., 2004; Gat-Viks and Shamir, 2007; Klipp and Liebermeister, 2006) to form larger ones as new elements and cross-talk interactions are being discovered.

Introduction of new constraints and the need to update existing parameters may not be solely due to pathway composition. Information on the pathway dynamics is seldom available all at once. New experimental data will be added from time to time, where the need to validate testable hypotheses on the models leads to more experiments being performed. Hence the parameter values have to be updated such that they are consistent with the new data as well as with the old data. Aggravating this issue is the fact that the old data used to estimate the parameters may not be systematically archived and easily available. Hence, re-estimating the parameters of the pathway using all the data may not be possible. Even if the data is available, having additional data sets will increase the computational cost due to the need to satisfy more constraints per iteration of the estimation algorithm. Hence there is a need for a systematic way of integrating both pathway components as well as new experimental data.

4.1 Pathway Composition

In this chapter, we describe a probabilistic approach to pathway composition while maintaining global consistency of the parameters. We first represent each parameter by a probability distribution - or *belief* - over the interval between its upper and lower bounds. For each component, we associate a *factor graph* - a probabilistic graphical model - to represent the dependencies between the parameters. The dependencies, encoded as joint probability distributions are obtained by sampling the parameters of the component and scoring them against available experimental data. We compose the pathways together by fusing their factor graphs at the common nodes to form a larger composite factor graph. Finally, we apply the message passing algorithm of *belief propagation* on the composite factor graph to update the beliefs of the parameters. This algorithm will ensure that the constraints within a component will be propagated throughout the entire pathway such that upon convergence, the beliefs of all the parameters will be globally consistent with experimental data. In the following sections, we review factor graphs and belief propagation in the context of pathway models.

4.2 Factor Graphs

A factor graph is an undirected bipartite graph consisting of *factor nodes* and *variable nodes*. In the graphical form, the factor nodes and variable nodes are represented by rectangular and circular nodes respectively. The factor graph visually expresses the structure of *factorization*, such that each factor node represents a local function and each variable node represents a variable. An edge exists between a factor node and a variable node if and only if the variable is an argument of the function. A probability density function will be associated with each variable node. Similarly, a joint distribution is associated with a factor node between the variables that it is connected to.

As a simple example, suppose we have a function $g(x_1, x_2, x_3, x_4, x_5, x_6)$ of six variables, and g can be expressed as a product of five local functions, i.e.

$$g(x_1, x_2, x_3, x_4, x_5, x_6) = f_A(x_1) f_B(x_2, x_3) f_C(x_1, x_4, x_6) f_D(x_2, x_5) f_E(x_5, x_6) \quad (4.1)$$

The factor graph that encodes this factorization is shown in Figure 4.2.

Factor graphs in systems biology have largely been used to infer and model gene regulatory networks from microarray gene expression data. Each variable node represents

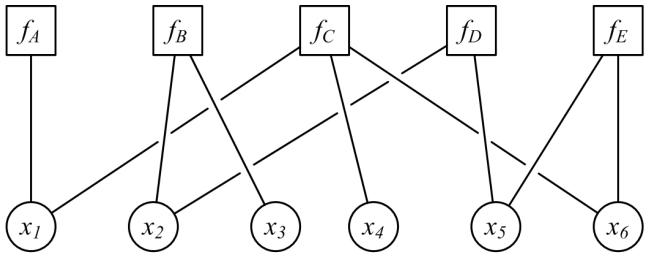


Figure 4.2: A simple factor graph example for the product $f_A(x_1)\ f_B(x_2, x_3)\ f_C(x_1, x_4, x_6)\ f_D(x_2, x_5)\ f_E(x_5, x_6)$.

a gene and each factor node represents the regulatory relationship between a group of genes. Gat-Viks et al. (2005) have used factor graphs to model a pathway that includes diverse biological entities such as mRNAs, proteins and metabolites. They have applied their technique on two of the pathways in *S. Cerevisiae* to integrate expression data with prior biological knowledge on the pathways. In addition, they have also used factor graphs to refine and expand signaling pathways (Gat-Viks and Shamir, 2007). Other applications of factor graphs include the building of self-correcting pathway maps (Rzhetsky et al., 2006), and the construction of physical network models (Yeang et al., 2004).

4.2.1 Factor Graph representation of Biological Pathways

In this thesis, we use factor graphs to represent the rate parameters and their dependencies. In any pathway model, there will be dependencies between its parameters. For instance, consider a reversible mass action reaction that is in steady state. The forward rate parameter will be correlated to the reverse rate parameter, given a particular equilibrium concentration for the substrates and products.

In the present setting, the dynamics of the pathway is described by a system of ODEs where the ith equation has the form $dx_i/dt = f_i(\mathbf{x}(t), \mathbf{p})$. Each variable node in the factor graph representation will represent a parameter, and each factor node will represent a function f_i. For cases where there are several individual experiments, each data set will have its own factor graph *instance*. For convenience, we will denote the factor node of equation f_i for experiment j as F_{ij}. When we are not referring to any experiments in particular, we will simply write it as F_i. Denoting the variable node representing parameter p as V_p, an edge exists between a variable node V_p and factor node F_i if the parameter $p \in \mathbf{p}$ is present in the function f_i. This rule effectively captures all the *direct* parameter dependencies given the rate equations.

This way of deriving the factor graph may result in a set of disconnected factor sub-graphs. Consider the simple pathway model below (Figure 4.3a). The parameters k_1 and k_2 are both present in equations f_1 and f_2. Similarly, k_3 and k_4 are both present in f_3 and f_4. However, there are no direct dependencies relating k_1, k_2 to k_3 and k_4. Hence, the resulting factor graph of the pathway has two separate sub-graphs (Figure 4.3b).

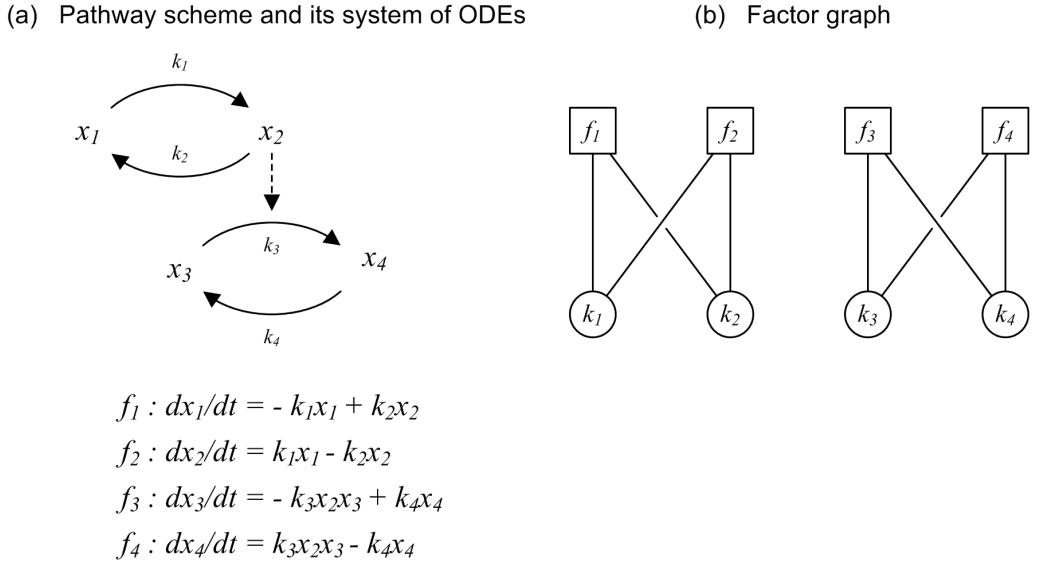


Figure 4.3: (a) A simple pathway and its system of ODEs. (b) The factor graph representation capturing only direct parameter dependencies, resulting in two disjoint factor graphs.

Due to the molecule species x_2, there is an *indirect* dependency between the parameters k_3, k_4 and both k_1 and k_2. Hence, the factor graph should reflect this by having a path between the nodes V_{k_1}, V_{k_2}, V_{k_3} and V_{k_4}. There are two ways to address this issue.

Back tracing

One way is to extend the rule for connecting the nodes of a factor graph to include the following: An edge exists between the factor node F_i and the variable node V_k if some x_j appears in f_i and k is an argument of f_j. The pathway in Figure 4.3 will then have the factor graph representation shown in Figure 4.4a.

By recursively back tracing the dependencies and adding more edges, one can obtain a valid factor graph. However, for a typical pathway, this factor graph will be

fully connected, i.e. all the factor nodes will be connected to all the variable nodes. Since the factor graph is a probabilistic model, such a graph will have several *multivariate distributions* that will be associated with its nodes, increasing its complexity. Hence, tracing the dependencies by one-step will be sufficient in capturing the relevant dependencies that are required for propagating the parameter constraints later on.

Molecule Species as Variable Nodes

Alternatively, we can choose to represent some of the molecule species by variable nodes - similar to the parameters - to bridge the disconnected sub-graphs. Suppose we denote the set of neighboring nodes of a factor graph element n (factor node or variable node) as Neighbor(n). If there exists $F_i \in F$ and $F_j \in F$ where x_i is a term in the function f_j, and there is no path between any elements of Neighbor(F_i) and Neighbor(F_j), then we can capture the dependencies by having a variable node V_{x_i} for x_i, with the edges (F_i, V_{x_i}) and (F_j, V_{x_i}). The factor graph representation of the pathway in Figure 4.3a derived with this additional rule is shown in Figure 4.4b.

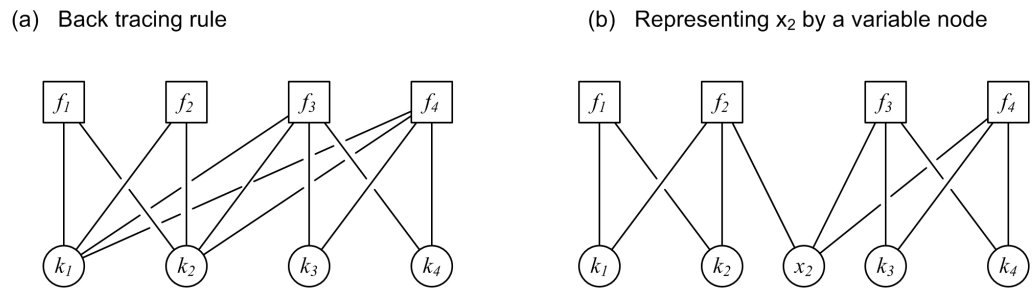


Figure 4.4: (a) Connecting disjoint factor sub-graphs by the one-step back tracing rule. (b) Representing the molecule species x_2 by a variable node to connect the factor graphs.

The choice of deciding which method to use for capturing parameter dependencies in disconnected factor graphs depends on how the pathway is decomposed during parameter estimation. As a guide, if the disconnected factor sub-graphs are derived from the same component, a back trace rule can be applied. Otherwise, the molecule species (typically an enzyme) that connects the two components can be represented as a variable node in order to allow their factor graphs to be fused together later on.

4.3 The Compatibility Function

With each factor node F_i, we associate a compatibility function ψ_i over the parameters that are connected to it. This compatibility function is defined as a joint probability distribution over the connected parameters. Since we are dealing with a dynamical system with no closed-form solutions, this distribution has to be built up by *sampling* the parameter space. A variety of sampling techniques can be used.

For each factor node F_i, the combinations of parameter values are scored using the following objective function

$$J(\mathbf{p}_i) = \sum_{x_j \in \mathbf{x}_{\text{obs}}, k} \frac{(x_{jk}(t, \mathbf{p}) - x_{jk}^{\text{exp}}(t))^2}{w_{jk}} \tag{4.2}$$

where $x_{jk}^{\text{exp}}(t)$ is the kth experimental data point of variable x_j and $x_{jk}(t, \mathbf{p})$ is the corresponding predicted value generated by simulating the ODEs using the sampled values of the parameters $\mathbf{p}$. The set $\mathbf{p}_i$ consists of the parameters that are connected to the factor node F_i. w_{jk} is the weight, typically the maximal value of x_j, used to normalize its contribution to the objective function. This objective function is similar to Equation 2.15, except that the elements of set $\mathbf{x}_{\text{obs}}$ depends on whether or not the molecule species corresponding to F_i is observed.

The scores are then converted into probabilities by the following equation

$$\psi_i(\mathbf{p}_i) = \frac{1}{z_i} e^{\frac{-\mu_i J(\mathbf{p}_i)}{\max J(\mathbf{p}_i)}} \tag{4.3}$$

where z_i is a normalizing constant to ensure that the probabilities sum up to 1, and μ_i is a scaling factor, which is usually set to a value between 10 and 20, depending on the accuracy of the experimental data. Varying the scaling factor affects the sensitivity of the compatibility function to noise in the data. The intuition behind the conversion function is that the larger the difference between the predicted and experimental values, the smaller the probability for that particular parameter combination that produces it. These distributions encode the dependencies between the parameters with respect to experimental data, and they are immutable.

4.3.1 Local Scores and Global Scores

In a pathway model, we can define two types of scores - *local* and *global*, with respect to the set of data from a single experiment. A local score is obtained by applying the objective function (Equation 4.2) on a single observed molecule species. On the other

hand, a global score is obtained by applying the same objective function, but on all the observed molecule species in the pathway. For a factor node F_i whose molecule species x_i is observed, its objective function will be derived using a *local* score, i.e. $\mathbf{x}_{\text{obs}} = \{x_i\}$. Otherwise, the *global* score will be used.

A factor graph G is a factorization of the probabilities of the system (Kschischang et al., 2001). Hence G can be expressed as a product of the conversion equations $\psi_i(\mathbf{p}_i)$ such that:

$$P(G) = \prod_{F_i \in F} \psi_i(\mathbf{p}_i) \tag{4.4}$$

where F is the set of factor nodes. Since each factor node F_i represents the ODE function f_i, which in turn determines the dynamics of molecule species x_i, we can divide the factor nodes into *observed* and *unobserved* nodes. We denote them by the sets F_{obs} and F_{unobs} respectively, such that $F_{\text{obs}} \cup F_{\text{unobs}} = F$ and $F_{\text{obs}} \cap F_{\text{unobs}} = \phi$. Hence, Equation 4.4 can be written as

$$P(G) = \prod_{F_i \in F_{\text{obs}}} \psi_i(\mathbf{p}_i) \prod_{F_j \in F_{\text{unobs}}} \psi_j(\mathbf{p}_j) \tag{4.5}$$

Expanding the equations ψ_i and ψ_j yields the following

$$P(G) = \alpha \prod_{F_i \in F_{\text{obs}}} e^{-\gamma_i J(\mathbf{p}_i)} \prod_{F_j \in F_{\text{unobs}}} e^{-\gamma_j J(\mathbf{p}_j)} \tag{4.6}$$

where the constant $\gamma_i = \mu_i / \max J(\mathbf{p}_i)$. The normalizing constants can be brought out and written as the single constant α.

For the observed factor nodes, their objective functions J can be further expanded into the weighted mean-square difference over the molecule species they represent. For the unobserved ones, they can be expressed as the global mean-square difference between all the observed molecule species.

$$P(G) = \alpha \prod_{F_i \in F_{\text{obs}}} e^{-\gamma_i \sum_k \frac{(x_{ik} - x_{ik}^{\text{exp}})^2}{w_{ik}}} \prod_{F_j \in F_{\text{unobs}}} e^{-\gamma_j \sum_i \sum_k \frac{(x_{ik} - x_{ik}^{\text{exp}})^2}{w_{ik}}} \tag{4.7}$$

$$P(G) = \alpha e^{-\sum_i \gamma_i \sum_k \frac{(x_{ik} - x_{ik}^{\text{exp}})^2}{w_{ik}}} \prod_{F_j \in F_{\text{unobs}}} e^{-\gamma_j \sum_i \sum_k \frac{(x_{ik} - x_{ik}^{\text{exp}})^2}{w_{ik}}} \tag{4.8}$$

Pushing all the constants γ_i and γ_j out into a single term γ, we have

$$P(G) = \alpha e^{-\gamma \sum_{ik} \frac{(x_{ik} - x_{ik}^{\exp})^2}{w_{ik}}} \tag{4.9}$$

Equation 4.9 is the same as using the objective function (Equation 2.15) for the entire pathway to produce a joint probability distribution between all the parameters. Finding the most likely parameter assignment (highest probability) is then equivalent to minimizing the objective function. This justifies not only our use of local and global scores in building the compatibility functions, but also our approach of factorizing the pathway into their respective factors nodes.

4.4 Parameter Beliefs

In the previous section, we have discussed the compatibility functions that are associated with the factor nodes. For the variable nodes, they are attached with the beliefs of the parameters they represent. For the unknown parameters, their beliefs are uniformly distributed as it is equally likely for them to assume any value within their allowable ranges. Known parameters can have *a priori* distributions to reflect prior knowledge.

Previously we have mentioned that variable nodes can also represent molecule species. Unlike parameters, each being able to assume a single value, the molecule species have dynamics associated with them. For each initial condition, a molecule species will have a *trace* associated with it. However, we can assume the existence of a hashing function that maps each trace to a unique index. The beliefs of the molecule species is then a probability distribution over the range of such indices.

By encoding the dependencies of the parameters in the factor nodes and the parameter beliefs in the variable nodes, we can then use belief propagation on the factor graph to propagate the constraints in the factor nodes throughout the graph. During each iteration, the beliefs of the parameters are updated such that they gradually assume a distribution that takes into account the information from all the constraints present in the pathway, i.e. global consistency.

4.5 Belief Propagation

Belief propagation is a probabilistic inference technique (Pearl, 1988), and is sometimes referred to as the *sum-product* or the *max-product* algorithm. It is a message passing scheme that has been used in several other areas such as statistical physics, computer vision, and error-correcting coding theory, albeit under different names.

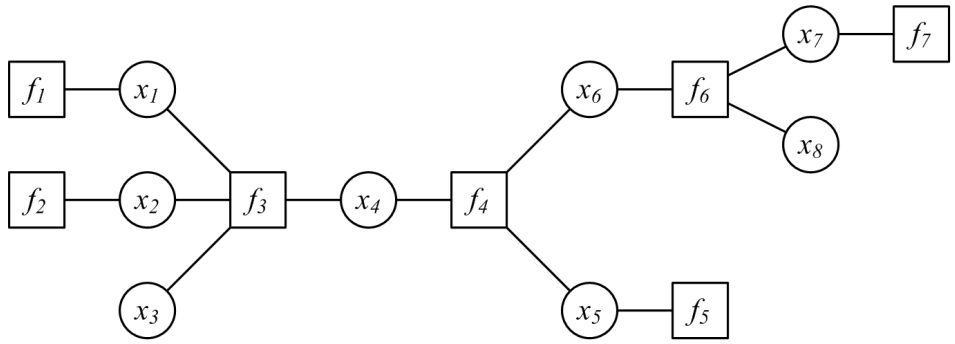


Figure 4.5: Example of a simple factor graph where the global function can be factorized into a product of local functions.

To have an intuition on how belief propagation works, consider the simple example of a factor graph as shown in Figure 4.5. Its global function $g(x_1, \ldots, x_8)$ can be factorized into a product of local functions:

$$\begin{aligned} g(x_1, \ldots, x_8) &= f_1(x_1) f_2(x_2) f_3(x_1, x_2, x_3, x_4) \\ &\quad f_4(x_4, x_5, x_6) f_5(x_5) f_6(x_6, x_7, x_8) f_7(x_7) \end{aligned} \tag{4.10}$$

Suppose one is interested to find the marginal probability of the variable x_4. This can be achieved by *summing over* all the other variables.

$$P(x_4) = \sum_{x_1, x_2, x_3, x_5, x_6, x_7, x_8} g(x_1, \ldots, x_8) \tag{4.11}$$

By substituting Equation 4.10 into Equation 4.11 and pushing all the sums in, we can obtain the following expression:

$$\begin{aligned} P(x_4) &= \Big(\textstyle\sum_{x_1, x_2, x_3} f_3(x_1, x_2, x_3, x_4) f_1(x_1) f_2(x_2) \Big). \\ &\quad \Big(\textstyle\sum_{x_5, x_6} f_4(x_4, x_5, x_6) f_5(x_5) . \Big(\textstyle\sum_{x_7, x_8} f_6(x_6, x_7, x_8) f_7(x_7) \Big) \Big) \end{aligned} \tag{4.12}$$

Equation 4.12 specifies an ordering of summary operations performed using only the probability distributions in the local factor and variable nodes. For instance, the term $\sum_{x_1, x_2, x_3} f_3(x_1, x_2, x_3, x_4) f_1(x_1) f_2(x_2)$ can be computed by a pointwise multiplication of the distributions at F_1 with V_{x_1}, F_2 with V_{x_2}, followed by another pointwise multiplication of their results (together with V_{x_3}) with F_3. The marginal probability

of x_4 from these distributions can then be obtained by summing up all the variables other than x_4. The same operation can be applied on $\sum_{x_7,x_8} f_6(x_6, x_7, x_8)f_7(x_7)$, before multiplying it with $\sum_{x_5,x_6} f_4(x_4, x_5, x_6)f_5(x_5)$. These sets of operations can be realized on the factor graph as a series of *message-passing* steps starting from the nodes $F_1, F_2, V_{x_3}, F_5, F_7, V_{x_8}$ and converging at the node V_{x_4}.

By repeating the same message passing steps for all the variable nodes, one can obtain the marginal distributions for all the variables in the factor graph, i.e. propagating the beliefs between nodes of the factor graph. This is also known as the *sum-product* algorithm.

4.5.1 Belief Propagation on Factor Graphs

On a factor graph, we denote the passing of a message from factor node F_i to variable node V_j as $m_{F_iV_j}(V_j)$ and a message in the opposite direction as $m_{V_jF_i}(F_i)$. These messages are probability distributions of the variable nodes. Informally, a variable node collects messages from its $n-1$ neighbors and performs a pointwise multiplication with its own belief, hence updating it. Thereafter, it will send this updated belief to its nth factor node for further processing. A factor node, upon receiving messages from its $m-1$ neighboring variable nodes, will be able to *infer* the belief of its mth neighbor node. This is done by performing a pointwise multiplication of all its incoming messages with its joint probability distribution table, and marginalizing on all but the mth variable. This belief is then propagated to that variable node for it to update its belief. Note that unlike the variable nodes, the joint distributions of the factor nodes are not changed in the process.

Formally, a message from factor node F_i to variable node V_j is defined as:

$$m_{F_iV_j}(V_j) = \sum_{V_a \in N(F_i)/V_j} \Psi_i \prod_{V_a \in N(F_i)/V_j} m_{V_aF_i}(F_i) \tag{4.13}$$

and a message from variable node V_j to factor node F_i is:

$$m_{V_jF_i}(F_i) = \alpha\phi_j(V_j) \prod_{F_b \in N(V_j)/F_i} m_{F_bV_j}(V_j) \tag{4.14}$$

where $N(n)$ denotes the set of nodes neighboring node n and α is a normalizing constant. $\phi_j(V_j)$ is the current local belief of variable node V_j and it gets updated after each iteration.

4.5.2 Loopy Belief Propagation

On singly connected factor graphs, belief propagation is an exact inference algorithm, i.e. the beliefs of the nodes will converge to the correct marginals in a number of iterations that is equal to the diameter of the graph (Pearl, 1988). However, when used to model biological pathways, one seldom gets a loop-free graph. The variant of belief propagation that works on such graphs is known as *loopy belief propagation*. Although there is no theoretical guarantee that this algorithm will converge, it has been shown that it often does (Murphy et al., 1999). In addition, loopy belief propagation also gives a good approximation to the correct marginals (Murphy et al., 1999; Weiss and Freeman, 1999).

Algorithm 4.1 describes loopy belief propagation as applied on a factor graph $G = (F, V, E)$ where F is the set of factor nodes, V is the set of variable nodes, and $E \subseteq (F \times V)$ are the edges connecting them. For each iteration, the algorithm has two phases. In the first phase, it processes the factor nodes, computing the beliefs of their neighboring variable nodes by collating and multiplying (Section 4.5.1) the variable nodes' current beliefs. These new beliefs are kept in message queues, to be used for updating the variable nodes during the second phase of the iteration.

Algorithm 4.1: Loopy Belief Propagation

input: $G = (F, V, E)$
begin
  repeat
    foreach $F_i \in F$ **do**
      foreach $V_j \in N(F_i)$ **do**
        $m_{F_iV_j}(V_j) \leftarrow \sum_{V_a \in N(F_i)/V_j} \Psi_i \prod_{V_a \in N(F_i)/V_j} \phi_a(V_a)$
        Push $m_{F_iV_j}(V_j)$ into Queue$[V_j]$
      end
    end
    foreach $V_j \in V$ **do**
      while Queue$[V_j] \neq \phi$ **do**
        $m \leftarrow$ Pop from Queue$[V_j]$
        $\phi_j(V_j) \leftarrow \phi_j(V_j) \times m$
      end
      $\phi_j(V_j) \leftarrow \alpha\phi_j(V_j)$
    end
  until Convergence *or* Max Iterations
end

4.5.3 Max-Product and Sum-Product

Previously we have discussed belief propagation as the sum-product algorithm. Given a probabilistic graphical model, the sum-product algorithm will give us the posterior marginal probability distributions of the variables upon convergence. In parameter estimation, we aim to find the most likely assignment instead. Hence we can replace the *sums* in Equations 4.13 and 4.14 by a *max* operator. This is also referred to as the max-product algorithm (Kschischang et al., 2001).

For this class of algorithms, the distributions upon convergence are the *maximum a posteriori* (MAP) probabilities for the variables. Assignment based on this distribution yields the most probable values for the variable nodes. By representing the biological pathway as a factor graph, and performing the max-product belief propagation, the assignment of values to the parameters (variable nodes) based on the final probability distribution can then be reported as the estimated values.

On loop free graphs, this algorithm converges to a unique probability distribution. For loopy belief propagation in the max-product form, it has also been shown that the assignment based on the *MAP* distributions is guaranteed to be more optimal than all the other assignments in a particularly large region around that assignment (Weiss and Freeman, 2001). In some factor graphs, this region can be exponentially large.

(a) Sum-product operation

x_2 \ x_1	0	1	2	3
0	0.10	0.04	0.02	0.01
1	0.07	0.05	0.18	0.03
2	0.08	0.01	0.05	0.12
3	0.11	0.06	0.05	0.02

$p(x_1) = \alpha \Sigma_{x2} f(x_1, x_2)$
$= \langle 0.36, 0.16, 0.30, 0.18 \rangle$

(b) Max-product operation

x_2 \ x_1	0	1	2	3
0	0.10	0.04	0.02	0.01
1	0.07	0.05	0.18	0.03
2	0.08	0.01	0.05	0.12
3	0.11	0.06	0.05	0.02

$p(x_1) = \alpha \max_{x2} f(x_1, x_2)$
$= \langle 0.23, 0.13, 0.38, 0.26 \rangle$

Figure 4.6: Difference between the (a) Sum-product operation and the (b) Max-product operation. Assignment using sum-product will give $x_1 = 0$ as the most likely value, while the max-product operation will give $x_1 = 2$ instead.

4.6 Composition

Instead of simply performing the estimation algorithm at the component level, we first derive the factor graphs from each individual component. Using the available experimental data in each component, we build up the joint probability distribution tables by sampling the parameters of each component separately. Note that between two neighboring components that do not share any parameters, their factor graphs will each have a variable node that represents the molecule species which connects the two components together.

To compose the factor graphs together, we fuse them at their *common variable nodes* (Figure 4.7), and performing a pointwise multiplication to the current beliefs of those variable nodes. Thereafter, we simply apply the belief propagation algorithm on the composite factor graph. This will enable constraints from one component to propagate to the other regardless of order, hence updating the beliefs of their parameters. Similarly, the pointwise multiplication of the shared parameters will enable reconciliation of their beliefs due to the different components.

4.7 Data Integration

Throughout this chapter, we have discussed the building of the compatibility functions by sampling the parameters with available experimental data. However, for the same molecule species, not all the data needs to be used at the same time. Instead, for each set of experimental data, we can derive a separate *factor graph instance* from the pathway component. Again, for integration, we only need to fuse the factor graph instances at their common variable nodes, and using belief propagation to reconcile the estimates, hence taking into account all the available experimental data (Figure 4.8). Note that the variable nodes representing the same molecule species, but for different data sets, are considered different nodes. Unlike the variable nodes which represent the pathway parameters, these variable nodes will *not* be fused together during the composition of the factor graphs.

When a new set of data is made available, it can be integrated by deriving the factor graph instance using only that data set, followed by fusing it with the current available factor graph and using belief propagation to update the beliefs of the parameters. This is in contrast with conventional parameter estimation techniques where all the data - new and old - will have to be used to re-estimate the parameters.

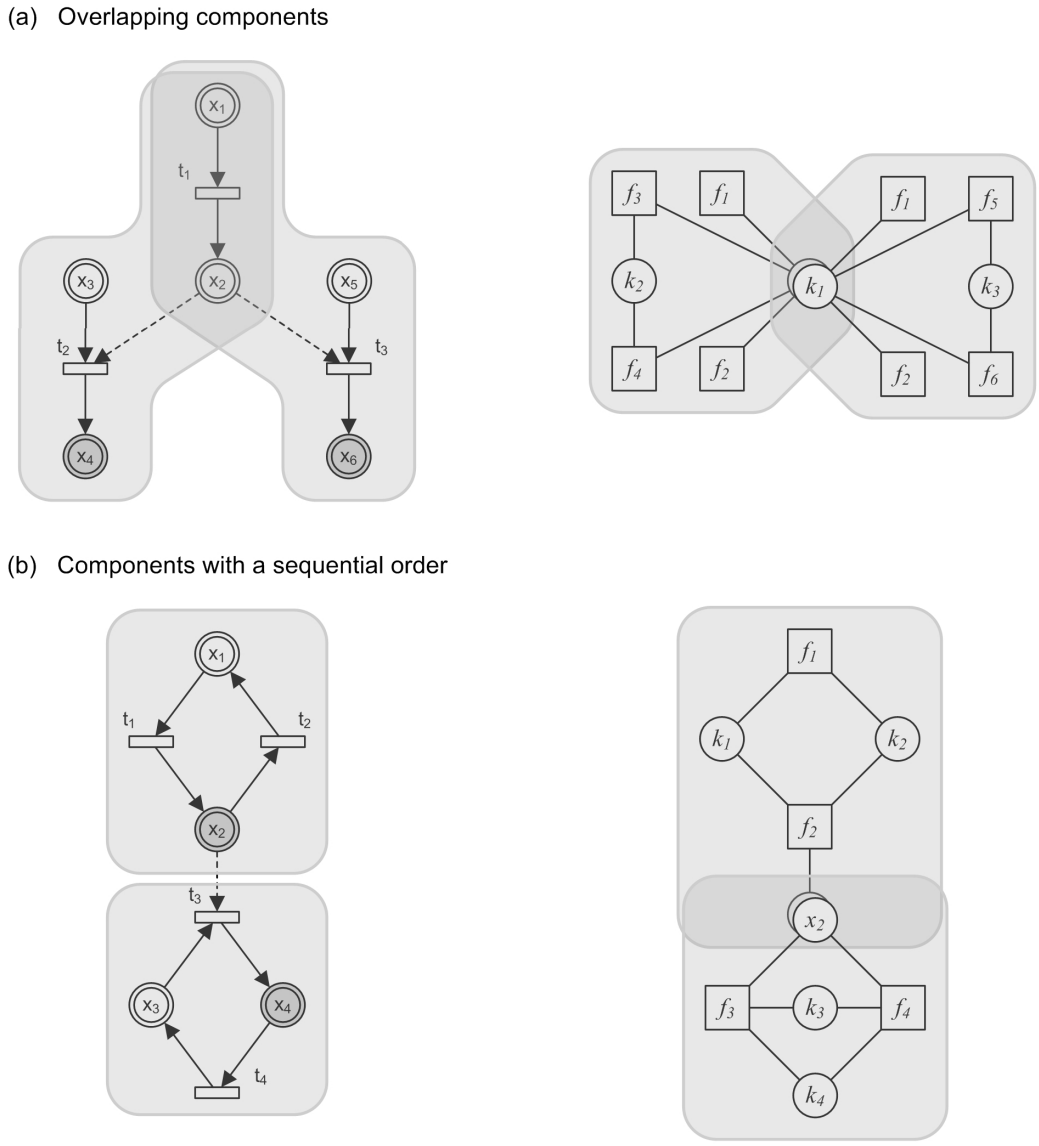


Figure 4.7: Examples of pathway components in their HFPN forms and their corresponding factor graphs. Grey places denote molecule species that have observed data. Each transition t_i has a parameter k_i associated with it. (a) Two overlapping components and their factor graphs. The factor graphs are derived using the one-step back tracing rule. They share the same parameter k_1. Note that there are two instances of f_1 and f_2, each from the different components. (b) Non-overlapping components. They are connected by the molecule x_2, which is represented by a variable node in their factor graphs. This variable node will be fused together during composition.

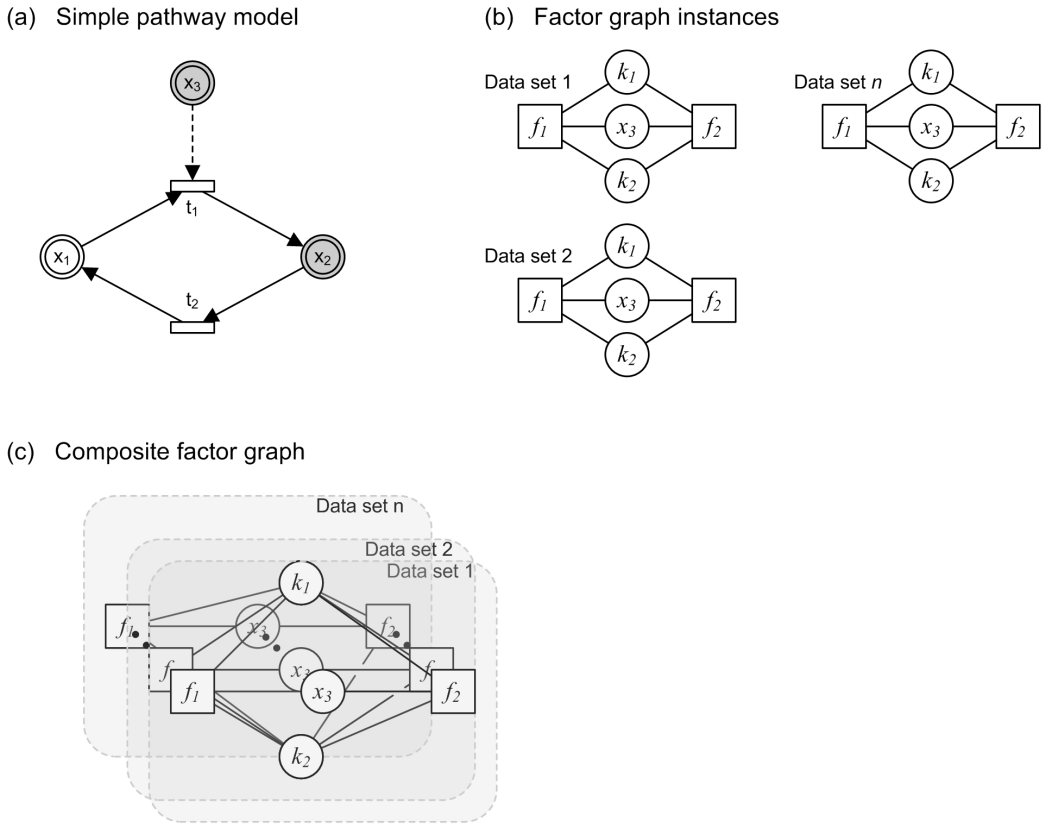
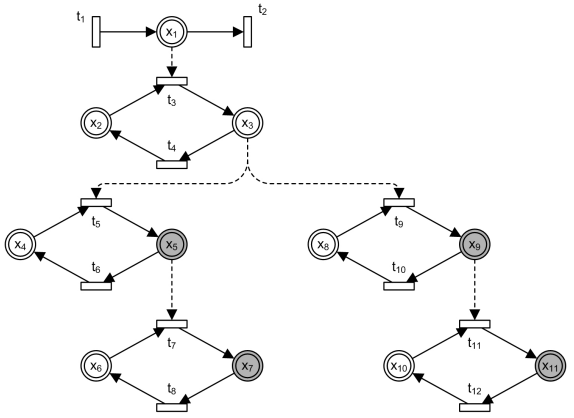


Figure 4.8: (a) Example of a simple pathway with the observed variables x_2 and x_3. (b) The factor graph instances, each corresponding to a single data set. (c) Fusing the factor graph instances at the common variable nodes k_1 and k_2. Variable node x_2 cannot be fused together as it represents traces from different initial conditions.

4.8 A Small Case Study: Branching Signaling Pathway

To test our approach of belief propagation, we first applied it to a synthetic pathway model that exhibits branching, a typical feature found in signaling pathways (Figure 4.9). Our algorithm is implemented in C++. It works in the discrete domain by dividing each dimension of the parameter space into finite partitions. Belief propagation will then give us the most likely domains, or the *maximum a posteriori* (MAP) partitions. However, in order to assess the performance of our approach, we have to fine-tune the estimates to obtain point values for the parameters. We did so by applying the Levenberg-Marquardt algorithm (Gill et al., 1998) starting from the mid-points of the MAP partitions.

We constructed a branched pathway with 11 molecule species and 12 kinetic rate constants, with their "nominal values" falling in the interval [0.0, 1.0]. For discretization, we divided each parameter domain into 10 equally-sized partitions. Simulation of this model yielded the synthetic data we used *in lieu* of experimental data, and then the nominal rate constants were set aside. We allowed synthetic time series data at 20 discrete time points to be made available for 5 of the molecules - x_1, x_5, x_7, x_9 and x_{11}. The 12 parameters were unknown and have to be estimated. This is a reasonable approximation of the parameter estimation problem in actual settings.



$$
\begin{aligned}
dx_1/dt &= k_1 - k_2 x_1 \\
dx_2/dt &= k_4 x_3 - k_3 x_1 x_2 \\
dx_3/dt &= k_3 x_1 x_2 - k_4 x_3 \\
dx_4/dt &= k_6 x_5 - k_5 x_3 x_4 \\
dx_5/dt &= k_5 x_3 x_4 - k_6 x_5 \\
dx_6/dt &= k_8 x_7 - k_7 x_5 x_6 \\
dx_7/dt &= k_7 x_5 x_6 - k_8 x_7 \\
dx_8/dt &= k_{10} x_9 - k_9 x_3 x_8 \\
dx_9/dt &= k_9 x_3 x_8 - k_{10} x_9 \\
dx_{10}/dt &= k_{12} x_{11} - k_{11} x_{10} x_9 \\
dx_{11}/dt &= k_{11} x_{10} x_9 - k_{12} x_{11}
\end{aligned}
$$

Figure 4.9: Schema of the pathway model and the system of ODEs that defines it.

Using the decompositional approach in the previous chapter, the pathway can be broken down into multiple overlapping components. We will consider components C1 and C2, which consist of the variable sets $\{x_1, x_2, x_3, x_4, x_5, x_6, x_7\}$ and $\{x_1, x_2, x_3, x_8, x_9, x_{10}, x_{11}\}$ respectively (Figure 4.10a). We estimated the parameters by sampling the components separately, and then we reconciled their values by belief propagation

on the combined factor graph (Figure 4.10b). As an alternative, we also applied our previous method by estimating the parameters for C1 followed by the remaining ones in C2 (Figure 4.10c - Scheme S1) and vice versa (Figure 4.10d - Scheme S2). We compared the efficiency and quality of our results with other optimization algorithms - Levenberg-Marquardt (LM), Evolutionary Strategies with stochastic ranking (SRES) and Genetic Algorithm (GA). All simulations were performed on an Intel Pentium M processor with 1 GB memory. Fine-tuning of the estimates, and the running of other optimization techniques were performed using the open source software COPASI (Hoops et al., 2006). We scored the resulting parameters obtained from all the algorithms using the weighted sum-of-squares difference between the experimental data and the corresponding simulation trace. The results of the simulations are summarized in Table 4.1.

4.8.1 Results and Discussion

The measure of quality for this set of results is not the closeness of the estimated parameters to the nominal ones. Rather, it is the score, which is the weighted sum-of-squares difference between the simulated concentration traces (generated using those parameters) and the synthetic experimental data. We can see from the results (Table 4.1) that belief propagation out-performs all the other algorithms both in terms of efficiency (requiring 10440 evaluations, where each evaluation is a complete numerical simulation of a single component) as well as quality (with the lowest score of 0.0119). Due to discretization, our implementation will only give us the best partition within each parameter domain. To simulate the model, we need point values and we obtain this by using the mid-points of the MAP partitions as a starting point for fine-tuning using the LM algorithm.

It is interesting to note that the decompositional schemes S1 and S2 (Figure 4.10c and 4.10d) provide better results than SRES, GA and LM. This is largely due to the lower dimensionality of the components. However, the quality varies depending on the order in which the components C1 and C2 are considered. This variation is a result of prematurely fixing the estimates of the parameters in one component when there could be better solutions when taking into account the entire pathway. The issues of ordering and choosing the components were largely left unaddressed in our previous chapter but belief propagation fills this gap nicely by allowing each of the components to be estimated separately, and later combining their estimates by its message passing scheme.

Besides requiring fewer model evaluations, the additional runtime incurred for mes-

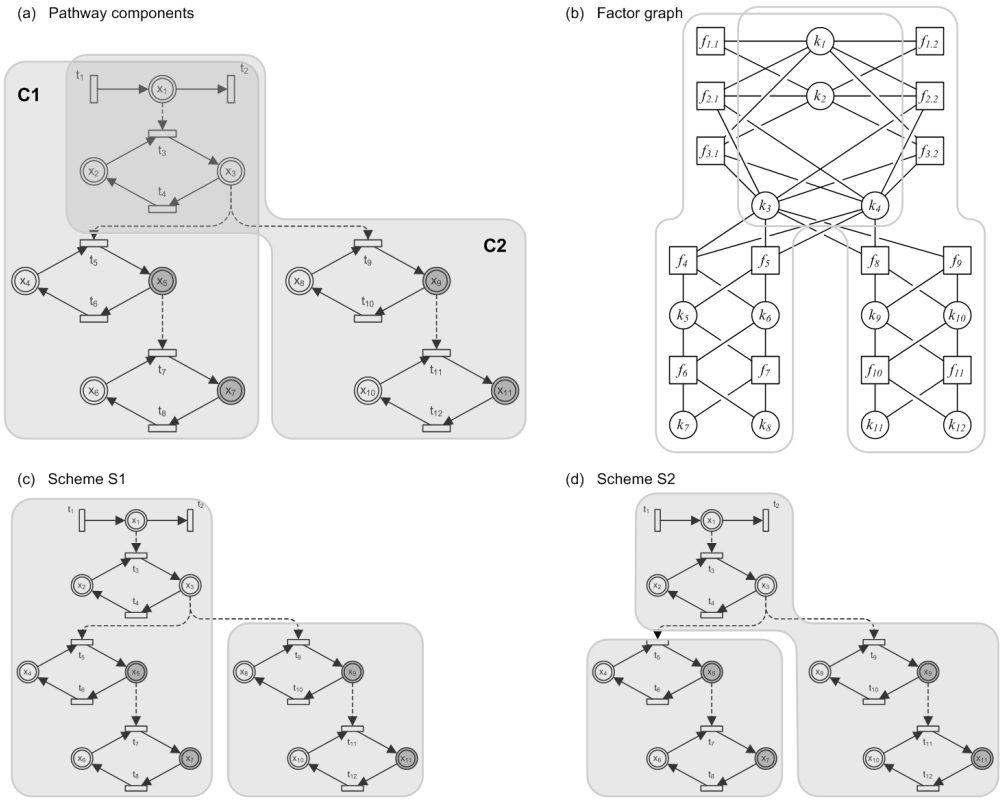


Figure 4.10: (a) Overlapping components C1 and C2. (b) Factor graph derived from C1 and C2, with the left factor graph corresponding to C1 and the right one, C2. They are composed together via the common variable nodes k_1, k_2, k_3 and k_4. (c) Scheme S1, where the left component is estimated first, followed by the remaining parameters in the right component. (d) Scheme S2 with a reverse order of estimation.

Parameters	**Nominal Values**	**BP/LM**	**S1**	**S2**	**LM**	**GA**	**SRES**
k_1	0.625	0.6368	0.6915	0.4635	0.3991	0.3035	0.5031
k_2	0.228	0.2317	0.2609	0.1311	0.7912	$1.2e^{-138}$	0.1629
k_3	0.112	0.4007	0.0739	0.0565	0.3236	0.3997	0.1020
k_4	0.96	0.1917	0.3264	0.7327	1	0.6247	0.6063
k_5	0.579	0.1089	0.9317	1	0.5592	0.2361	0.6859
k_6	0.312	0.2269	0.8134	0.3240	0.3779	0.3815	0.4444
k_7	0.628	0.6235	0.6349	0.7405	0.5877	0.6680	0.8878
k_8	0.104	0.0983	0.1657	0.1224	0.3032	0.0481	0.5638
k_9	0.04	0.0075	0.0456	0.1184	0.3310	0.0192	0.0710
k_{10}	0.286	0.1836	0.5523	0.8324	0.7980	0.4360	0.8602
k_{11}	0.624	0.6317	0.6551	0.7041	0.2847	0.4132	0.5773
k_{12}	0.88	0.8896	0.9468	0.8388	0.9402	0.5318	0.7079
Search Parameters	–	**Iter**: 200 **Tol**: $1e^{-5}$	**Gen**: 200 **Pop** : 20	**Gen**: 200 **Pop** : 20	**Iter**: 200 **Tol**: $1e^{-5}$	**Gen**: 400 **Pop**: 40	**Gen**: 400 **Pop**: 40
Score	–	**0.0119**	**0.4138**	**2.3498**	**269.061**	**4.75115**	**2.5103**
Evaluations	–	5905	47820	47820	177	15258	95814

Table 4.1: Comparison of belief propagation (BP/LM) with the original decompositional approach (S1 and S2) and with other optimization techniques (LM, SRES, GA). The zeroth, first and second order rate constants are given in $\mathrm{nM.s^{-1}}$, $\mathrm{s^{-1}}$ and $\mathrm{nM^{-1}.s^{-1}}$ respectively. Search parameters are specific to their individual algorithms for them to run: **Iter** = Maximum number of iterations; **Tol** = Tolerance; **Gen** = Number of generations; **Pop** = Population size. An evaluation is a complete numerical simulation of the ODEs using current parameter estimates. The **score** for the parameters is the weighted sum-of-squares difference between the synthetic experimental data and the corresponding simulation trace generated using those parameters

sage passing itself is quadratic in the size of the pathway. The size of the joint probability distribution table of the factor node, on the other hand, is exponential in the number of variable nodes it is connected to. Hence, it is the degree of the factor graph, rather than the pathway size, that will be the limiting factor for the performance of belief propagation. It will be interesting to determine if there is a reasonable upper bound on the connectivity of the factor nodes for biological pathways.

Chapter 5

An Integrated Case Study

In this chapter, we apply all the techniques presented in the previous chapters to a large pathway model that is based on the Akt-MAPK signaling pathway. We decompose the pathway into its set of minimal components, and with a suitable grouping scheme, form the components to be estimated. We estimate the parameters for the components individually and combine their factor graphs together during pathway composition and reconciliation. In addition, we also incorporate incremental data integration. Unlike conventional parameter estimation algorithms, where there is a need to re-estimate the parameters using *all* the experimental data, we form the factor graph instance using only the new data, and update the beliefs of the parameters through belief propagation. We will show how such an approach not only achieve a reasonable level of running time for every new set of data, but also maintains the overall consistency of the parameter estimates.

5.1 Pathway Model

The pathway used in this chapter is a synthetic one that is patterned after the Akt-MAPK signaling pathway presented in Section 3.3.1. We choose a synthetic model so as to generate enough pseudo-experimental data to use *in lieu* of actual data, which is still very sparse and extremely noisy. In this pathway, we simplified a few processes and used only mass-action forms for the rate equations, while keeping the main interactions. The simulation set-up is similar to the example in Section 4.8. However, the scope is much larger, involving more pathway elements, synthetic data, and a more complex pathway topology.

In this pathway, we have 36 molecule species and 42 unknown rate parameters. The

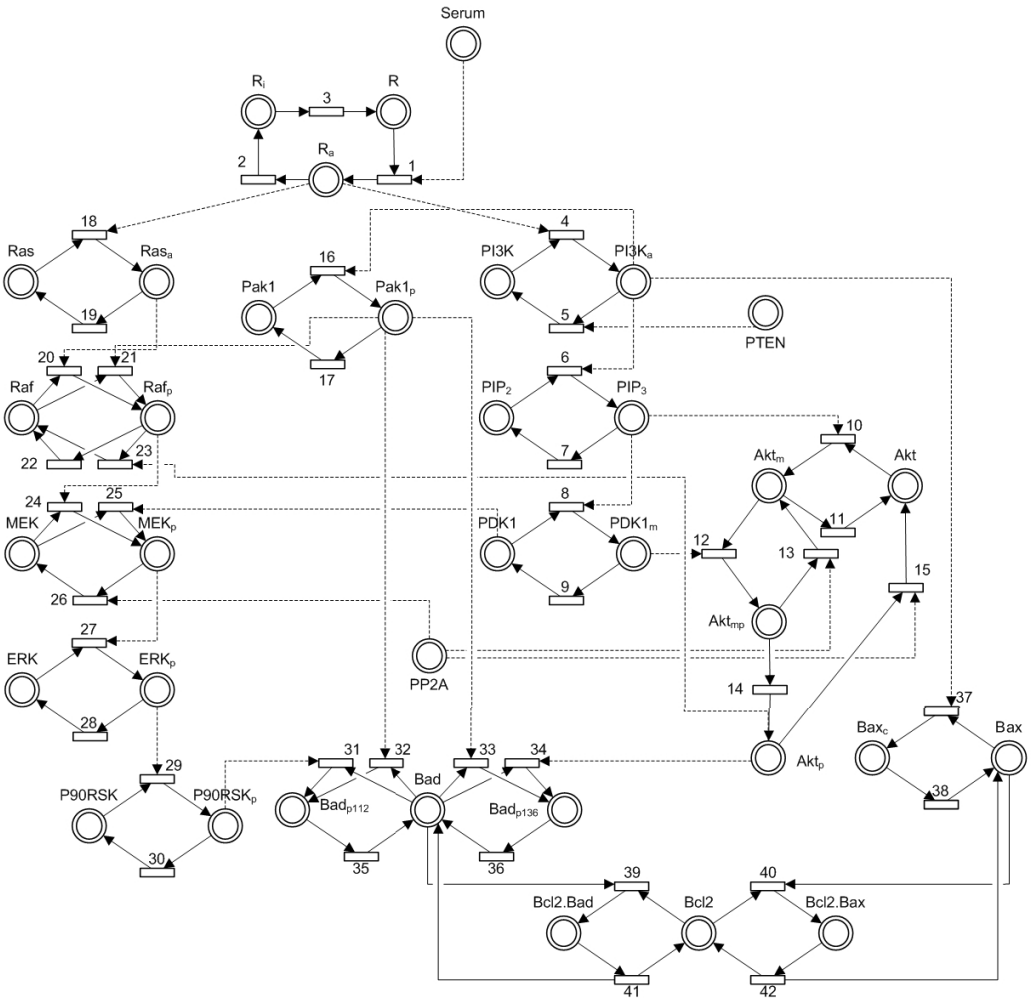


Figure 5.1: The HFPN model of the Akt-MAPK pathway.

HFPN model of the pathway is shown in Figure 5.1 and the rate equations are given in Table 5.1. The nominal values of the rate parameters are set to fall within the interval [0.0, 1.0] and the molecular concentrations of the species are within the interval [0.0, 5.0] nM. We will only present data for 13 of the molecule species at discrete time points. There will be 6 independent data sets from different initial conditions, and they will be supplied to the algorithm in an incremental fashion.

No	Rate Equation	No	Rate Equation
1	$k_1[R]$	22	$k_{22}[Raf_p]$
2	$k_2[R_a]$	23	$k_{23}[Akt_p][Raf_p]$
3	$k_3[R_i]$	24	$k_{24}[Raf_p][MEK]$
4	$k_4[R_a][PI3K]$	25	$k_{25}[PDK1][MEK]$
5	$k_5[PTEN][PI3K_a]$	26	$k_{26}[MEK_p][PP2A]$
6	$k_6[PI3K_a][PIP_2]$	27	$k_{27}[MEK_p][ERK]$
7	$k_7[PIP_3]$	28	$k_{28}[ERK_p]$
8	$k_8[PIP_3][PDK1]$	29	$k_{29}[ERK_p][P90RSK]$
9	$k_9[PDK1_m$	30	$k_{30}[P90RSK_p]$
10	$k_{10}[PIP_3][Akt]$	31	$k_{31}[P90RSK_p][Bad]$
11	$k_{11}[Akt_m]$	32	$k_{32}[Pak1_p][Bad]$
12	$k_{12}[PDK1_m][Akt_m]$	33	$k_{33}[Pak1_p][Bad]$
13	$k_{13}[PP2A][Akt_{mp}]$	34	$k_{34}[Akt_p][Bad]$
14	$k_{14}[Akt_{mp}]$	35	$k_{35}[Bad_{p112}]$
15	$k_{15}[PP2A][Akt_p]$	36	$k_{36}[Bad_{p136}]$
16	$k_{16}[PI3K_a][Pak1]$	37	$k_{37}[PI3K_a][Bax]$
17	$k_{17}[Pak1_p]$	38	$k_{38}[Bax_c]$
18	$k_{18}[R_a][Ras]$	39	$k_{39}[Bad][Bcl2]$
19	$k_{19}[Ras_a]$	40	$k_{40}[Bax][Bcl2]$
20	$k_{20}[Ras_a][Raf]$	41	$k_{41}[Bcl2.Bad]$
21	$k_{21}[Pak1_p][Raf]$	42	$k_{42}[Bcl2.Bax]$

Table 5.1: Rate equations of the various reactions in the pathway. Concentration of the molecule species are denoted in [ ].

5.1.1 Minimal Components

We first decomposed the pathway into its set of minimal components. Some of these minimal components are not estimatable. Nonetheless, they are important in forming the component graph, using which we can see their dependencies. Based on the locations of experimental data, we later grouped these minimal components into their estimatable components for conversion into factor graphs.

The 13 molecule species that have experimental data are - $PI3K_a$, $PDK1_m$, Akt_p, Bax, $Pak1_p$, Ras_a, Raf_p, MEK_p, ERK_p, $P90RSK_p$, Bad_{p112}, Bad_{p136} and Bcl2. Using

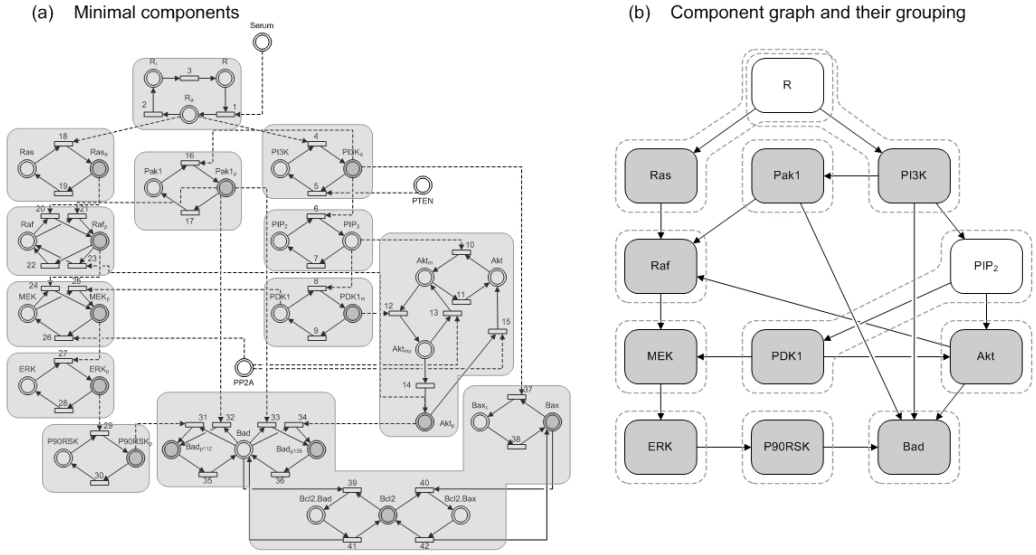


Figure 5.2: (a) The HFPN model of the pathway showing the minimal components (grey boxes). Places with experimental data are colored grey. (b) The component graph of the pathway. Minimal components are grouped together within dashed boxes, and will be used for the derivation of their factor graphs.

the grouping technique (Section 3.2.2), we are able to obtain 10 components. The minimal components and their groupings are shown in Figure 5.2. Two of the components overlap each other and share the common parameters k_1, k_2 and k_3.

5.1.2 Factor Graph Representation

To convert each component into a factor graph, we follow the scheme presented in Section 4.2.1. For molecule species that act as border elements between components, we represent them in the factor graph as variable nodes. Otherwise, the back tracing method is used (for instance the R.PI3K component). This is due to the fact that the domain size for the indices representing molecular traces is much larger than the domain size of the parameter values. Hence, we utilize the one-step back tracing rule as much as possible in order to avoid increasing the size of the joint probability distribution tables at the factor nodes. The factor graphs for the individual components are shown in Figure 5.3.

Note that the components for different data sets need not be the same. As an example, consider the R.PI3K component in Figure 5.3. In our case study, the lack of data for the molecules R, R_a and R_i makes it necessary to group them with the

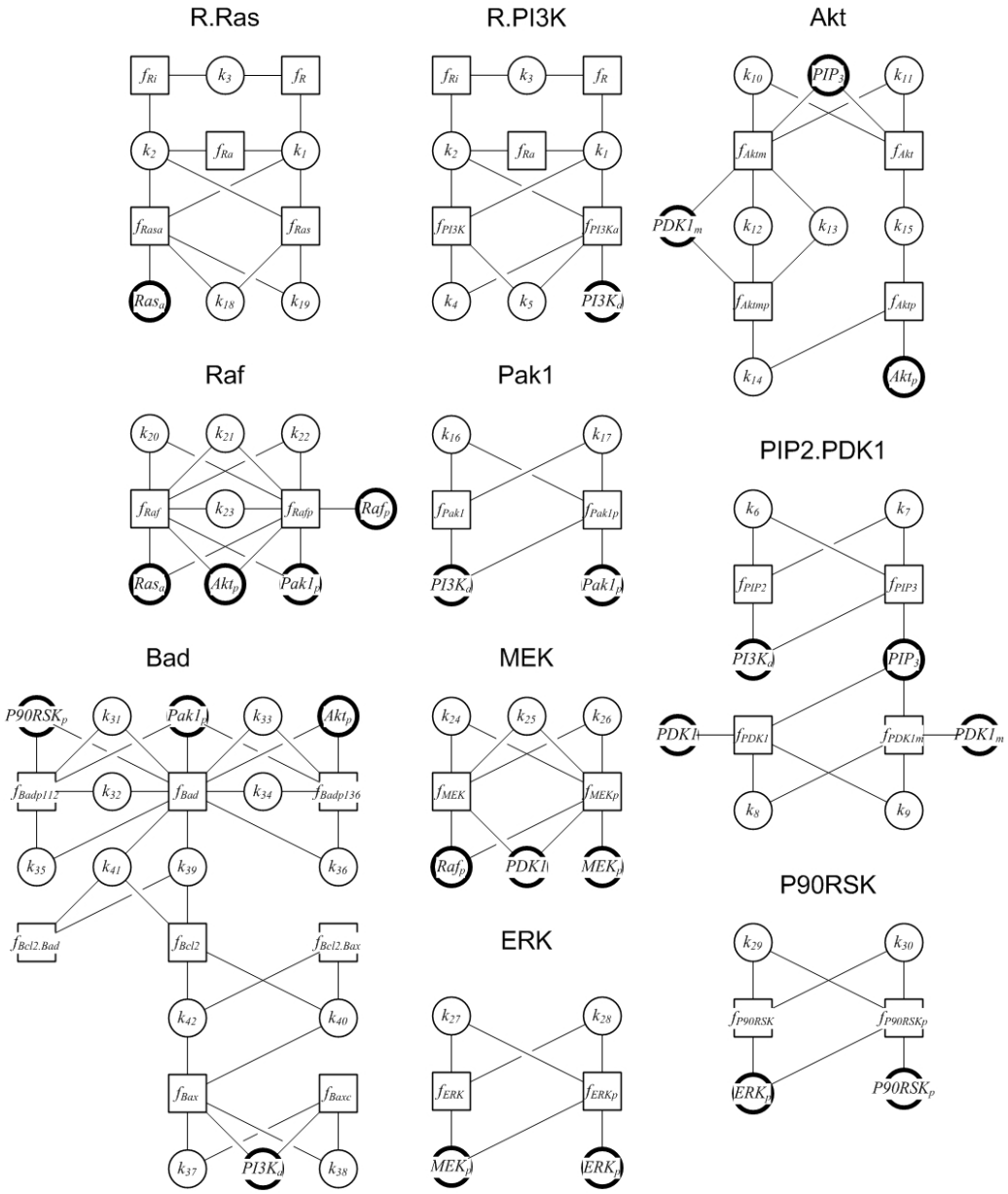


Figure 5.3: The factor graphs of each individual component. The variable nodes that model the molecule species are drawn as darker nodes.

minimal component PI3K. Hence we use the one-step back trace to form the factor graph instance for this component. However, the presence of data for the minimal component R in another data set may result in a slightly different structure for the factor graph instance. Nonetheless, fusing both factor graphs at their common variable nodes later on will still enable the beliefs from the different data sets to reconcile with each other.

5.2 Extensions to the Algorithm

In the course of implementing our approach for the building and parameter estimation of large pathway models, we have also extended the basic algorithm to circumvent some of the technical hurdles.

5.2.1 Representing Traces

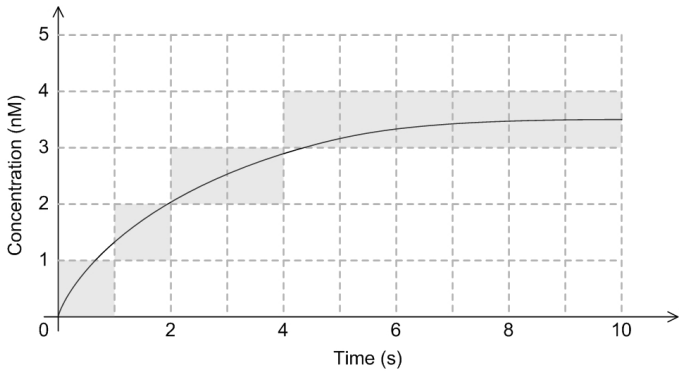
For each variable node representing a parameter, there is an associated probability distribution which we have discretized into 10 equal partitions within the interval [0.0, 1.0]. However, variable nodes representing molecule species cannot be discretized in this manner as each trace is not a single value, but a series of molecule concentration levels over time.

We handle this issue by first discretizing the concentration-time domain into finite partitions, i.e. dividing the time domain into 10 equal partitions and the concentration domain into 5 equal partitions. Hence a molecule trace running through this discrete space can be written as a quinary (base 5) number which can be then be converted into an integer index (Figure 5.4).

On the other hand, we can also obtain a trace given the index by doing the reverse, i.e. converting the index into the quinary number (for our example), and fitting a smooth curve through all the mid points of the shaded partitions in the concentration-time domain. However, due to discretization, some accuracy will be lost.

Given such a representation for the molecule traces, one can ignore the ordering of components when building the joint probability distribution tables of their factor graphs. During the sampling of a component which takes its input value from molecule species represented by a variable node, one can simply pick a value from the trace-index domain and generate its corresponding molecule trace for simulation.

However, we observed that simulation of a pathway model with all the possible parameter combinations will only produce a finite family of molecule traces. Hence,



$$\begin{aligned} \text{Quinary Representation} &= 0122333333_5 \\ \text{Index} &= (0 \times 5^0) + (1 \times 5^1) + \ldots + (3 \times 5^9) \\ &= 7324055 \end{aligned}$$

Figure 5.4: The discretized concentration-time domain for a molecule trace. The shaded partitions are the regions which the trace has visited, using which we can compute the index for the trace.

associating the variable nodes representing molecule species with distributions that span across the entire trace-index domain may be redundant.

In order to make use of our observation, we exploit the structure of the component graph to induce a partial order on the components. Sampling the parameters of the upstream components will then generate a set of *seen* trace-indices for the molecule species. In addition, for each seen trace-index, we keep a *cache* of the trace with the lowest score. During the sampling of the downstream components, we sample from the seen trace-indices and use the traces in the cache during simulation.

5.2.2 Directed Sampling of the Parameter Space

Within a component, we can employ a variety of sampling techniques when building up the joint probability distribution tables.

Previously, we have used a uniform sampling scheme for building the joint probability distributions of the components. For each component, we assign its parameters random values within their allowable ranges, and simulate the component to obtain the score for that parameter combination. With limited computational resources, this *random walk* may not be able to capture all the relevant dependencies between the parameters. Instead, we implement a *directed sampling* approach, where the current

parameters and their scores from the objective function will aid in generating the next set of values for the parameters.

In a component, our directed sampling approach is based on the Evolution Strategies with stochastic ranking (SRES) algorithm (Runarsson and Yao, 2000). In our implementation, we keep a population of 20 parameter sets, each of them to be given a score using the component global score of the objective function - Equation 4.2. The sampling process itself follows the scheme presented in Section 2.3.2. The only difference is that each discarded parameter set will still have its score being recorded as a probability in the joint probability distribution table of the factor nodes. This is in contrast with the standard SRES algorithm, where the worse performing solution set gets discarded without any forms of *book-keeping*. Our method allows the algorithm to spend computational power sampling the more probable regions in the parameter space rather than simply performing a random walk in the space.

5.3 Data Integration

To demonstrate the data integration aspect of our approach, we estimate the parameters using 6 synthetic data sets obtained from different initial conditions. The different initial conditions are summarized in Table 5.2.

Using our approach, we derive the factor graphs of the individual components and build their distribution tables by sampling using only the first set of data. Thereafter, we compose the pathway components by fusing their factor graphs and reconcile the parameter beliefs by belief propagation. Due to our discretization of the parameter domain, belief propagation will only return the MAP partitions for the parameters. We then use the mid-points of these partitions as starting points for fine-tuning. The fine-tuning is done using the Levenberg-Marquardt algorithm - a local optimizer - which is implemented in the open source software COPASI (Hoops et al., 2006). The *final score* upon optimization is then reported as the performance of our simulation. This process is repeated for 10 times and the factor graph (with all its distributions) for the best performing simulation run is kept and stored.

To integrate the new set of data, we derive another factor graph instance for the pathway components and build the distribution tables by sampling the parameters with only the new data set. During composition, the factor graph instances are not only combined together to form the factor graph of the pathway, they are also fused with the factor graph instance from the first data set. The parameter beliefs are then reset back to uniform distributions before performing belief propagation on the combined factor

graph. This is to ensure that equal weightage is given to the two data sets, instead of starting the propagation with *a priori* distributions. Again, this process is repeated for 10 times, and the best performing factor graph is stored.

	Initial Condition					
Molecule Species	**1**	**2**	**3**	**4**	**5**	**6**
Serum	**1.0**	**1.0**	**1.0**	**1.0**	**1.0**	**1.0**
PP2A	**5.0**	**5.0**	**5.0**	**5.0**	**1.0**	**5.0**
R	**5.0**	**5.0**	**5.0**	**5.0**	**5.0**	**2.0**
R_a	0.0	0.0	0.0	0.0	0.0	0.0
R_i	0.0	0.0	0.0	0.0	0.0	0.0
PI3K	**5.0**	**5.0**	**5.0**	**5.0**	**5.0**	**5.0**
$PI3K_a$	0.0	0.0	0.0	0.0	0.0	0.0
PIP_2	**5.0**	**5.0**	**5.0**	**5.0**	**5.0**	**5.0**
PIP_3	0.0	0.0	0.0	0.0	0.0	0.0
PDK1	**5.0**	**5.0**	**1.0**	**5.0**	**5.0**	**5.0**
$PDK1_m$	0.0	0.0	0.0	0.0	0.0	0.0
Akt	**5.0**	**1.0**	**5.0**	**5.0**	**5.0**	**5.0**
Akt_m	0.0	0.0	0.0	0.0	0.0	0.0
Akt_{mp}	0.0	0.0	0.0	0.0	0.0	0.0
Akt_p	0.0	0.0	0.0	0.0	0.0	0.0
Pak1	**5.0**	**5.0**	**5.0**	**5.0**	**5.0**	**5.0**
$Pak1_p$	0.0	0.0	0.0	0.0	0.0	0.0
Ras	**5.0**	**5.0**	**5.0**	**5.0**	**5.0**	**5.0**
Ras_a	0.0	0.0	0.0	0.0	0.0	0.0
Raf	**5.0**	**5.0**	**5.0**	**5.0**	**5.0**	**5.0**
Raf_p	0.0	0.0	0.0	0.0	0.0	0.0
MEK	**5.0**	**5.0**	**5.0**	**5.0**	**5.0**	**5.0**
MEK_p	0.0	0.0	0.0	0.0	0.0	0.0
ERK	**5.0**	**5.0**	**5.0**	**5.0**	**5.0**	**5.0**
ERK_p	0.0	0.0	0.0	0.0	0.0	0.0
P90RSK	**5.0**	**5.0**	**5.0**	**5.0**	**5.0**	**5.0**
$P90RSK_p$	0.0	0.0	0.0	0.0	0.0	0.0
Bad	**5.0**	**5.0**	**5.0**	**5.0**	**5.0**	**5.0**
Bad_{p112}	0.0	0.0	0.0	0.0	0.0	0.0
Bad_{p136}	0.0	0.0	0.0	0.0	0.0	0.0
Bax	0.0	0.0	0.0	0.0	0.0	0.0
Bax_c	**5.0**	**5.0**	**5.0**	**5.0**	**5.0**	**5.0**
Bcl2	**5.0**	**5.0**	**5.0**	**5.0**	**5.0**	**5.0**
Bcl2.Bad	0.0	0.0	0.0	0.0	0.0	0.0
Bcl2.Bax	0.0	0.0	0.0	0.0	0.0	0.0
PTEN	**1.0**	**1.0**	**1.0**	**0.01**	**1.0**	**1.0**

Table 5.2: The set of initial conditions. Values are given in nM. Condition 1 is the normal scenario. The other conditions are obtained by perturbing the initial concentrations of the molecule species, indicated in the table as grey colored cells.

As a comparison, we perform the same data integration using the conventional parameter estimation algorithms - SRES and Genetic Algorithm (GA). Since these two techniques offer no principled way of updating the parameters, introduction of new data set will involve re-estimating the parameters using all the experimental data, both old and new. In that respect, each iteration will require more model evaluations, since there are additional initial conditions to simulate for each set of parameter guesses.

5.4 Results and Discussion

Our algorithm is implemented in C++. All simulations were performed on an Intel Core 2 Duo 1.66 GHz processor with 1 Gb memory. The software COPASI was used for the fine-tuning of the parameter estimates, as well as performing conventional parameter estimation using SRES and GA. These methods perform estimation on the entire pathway with no decomposition. The results of the simulations are summarized in Tables 5.3 and 5.4. The performance of our approach is assessed by two metrics - *quality* and *efficiency*.

The quality of the solution is measured by the objective function (Equation 2.15). Since the fine-tuning process, SRES and GA are performed using COPASI, the objective function implemented within that software is used. For each algorithm, we run the algorithm 10 times. The best score, mean and standard deviation are all presented in Table 5.3. Details of the simulation results can be found in Appendix B.

Number of	BP			SRES			GA		
data sets	Best	Mean	Std Dev	Best	Mean	Std Dev	Best	Mean	Std Dev
1	**0.133**	1.749	1.836	**0.849**	1.58	0.531	**2.32**	5.52	2.95
2	**0.154**	1.962	2.579	**1.12**	1.86	0.601	**4.26**	6.60	2.01
3	**0.154**	1.384	0.907	**1.82**	2.48	0.528	**4.73**	10.1	4.47
4	**0.382**	3.182	2.695	**2.90**	4.69	1.40	**4.44**	10.9	4.97
5	**0.496**	1.457	1.682	**3.47**	6.59	3.17	**18.3**	36.9	13.3
6	**0.456**	1.149	1.102	**4.24**	6.79	1.95	**15.9**	27.8	12.3

Table 5.3: Comparison of the performance of belief propagation (BP) as compared to Evolution Strategies with stochastic ranking (SRES) and genetic algorithms (GA). For each algorithm, we show the best, mean and standard deviation of the scores out of 10 independent runs.

The efficiency of the algorithm is measured by the number of model evaluations (Table 5.4). Since our probabilistic approach only requires the sampling of new experimental data during integration, the number of model evaluations is kept relatively constant with each additional data set. The slight increase in the number of evaluations

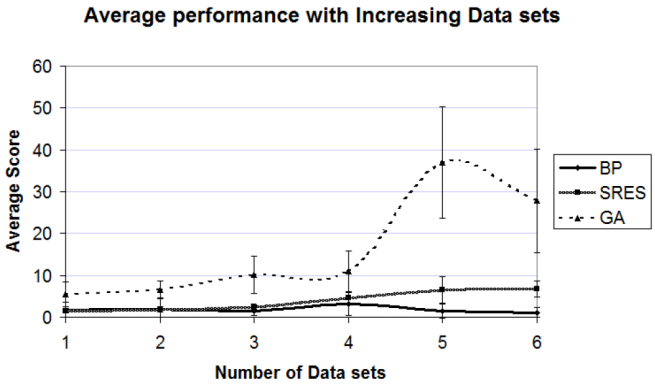


Figure 5.5: The average score and standard deviation for the various algorithms with increasing number of data sets.

is due to the fine-tuning process. On the other hand, SRES and GA do not store the results of prior computation. Hence the number of evaluations increases linearly with the number of data sets.

Number of data sets	BP	SRES	GA
1	12,466	11,944	9,494
2	13,221	23,888	19,108
3	13,488	35,832	28,464
4	13,893	47,776	38,160
5	14,146	59,720	47,560
6	14,955	71,664	58,560

Table 5.4: Comparison of the number of evaluations used by the algorithms in obtaining their best solutions.

Figure 5.5 shows the graph of the average performance for the different algorithms with increasing data set. Interestingly, when the number of experiments is small, SRES is able to produce extremely good estimates. However as the number of experiments exceeds 4 initial conditions, its average performance starts to degrade. The belief propagation approach, on the other hand, is able to maintain a relatively stable, and in some cases, slightly improved performance.

It should be noted that belief propagation itself will have an impact on the running time of the algorithm. We have mentioned previously that the complexity of message passing is quadratic in the size of the factor graph (Section 4.8.1). For our Akt-MAPK example with 6 sets of experimental data, this level of complexity is still manageable. However, as the number of experimental data increases, we foresee that data integration

in the manner presented will be limited by the belief propagation process (This issue is also faced by conventional techniques such as SRES and GA). However, it can be alleviated by storing only the *posterior* beliefs of the parameters instead of the factor graphs after a certain number of experimental data sets has been used. The beliefs have summarized all the current dependencies, and can be used as *a priori* distributions for the factor graphs of new data sets.

Chapter 6

Conclusion

In this chapter, we summarize the contributions made in this thesis and discuss the directions that can be further explored.

6.1 Our Contributions

Our main contribution in this thesis is a methodology for parameter estimation of large pathway models. Our approach also allows the addition of new experimental data while maintaining global consistency of the parameter estimates.

For a large pathway model represented as an HFPN model, we exploit its structure and break it down into its set of minimal components (Chapter 3). By laying out the minimal components as a component graph, we are able to trace out the simulation dependencies between them. Thereafter, based on the locations of experimental data, the minimal components can be grouped together into larger components, after which an estimation order can be imposed upon them. This approach significantly reduces the parameter search space.

In Chapter 4, we consider some of the major issues that arise from the decompositional approach. Parameters in upstream components, once fixed, cannot be updated when experimental data from downstream components indicate the existence of better estimates for those parameters. In some cases, the ordering of the components is non-deterministic (possibly due to overlapping components) and hence the quality of the parameter estimates vary depending on the order of estimation (for instance see Section 4.8.1). We handle this situation by first capturing the dependencies between the parameters in a component by a factor graph. The dependencies themselves are implemented as joint probability distributions obtained by sampling the parameters with

experimental data. Finally we compose the components together to form the complete pathway. All the parameter estimates are updated by fusing the factor graphs and passing their local constraints throughout the composite factor graph via belief propagation.

The advantage of using a factor graph to capture the local dependencies within a component is that there is no longer a need to order the components for parameter estimation. Although we might still impose one for technical reasons (see Section 5.2.1), it is not a requirement. In addition, the technique of fusing the factor graphs together followed by belief propagation can be applied to the integration of new experimental data. This is crucial for pathway modeling as it is not always the case that all the experimental data are available at the same time. They are usually produced in an incremental fashion.

In Chapter 5, we show how all the concepts in this thesis can be applied in the parameter estimation of a synthetic pathway model. The performance of our approach, as compared to the other algorithms SRES and GA, show the stability of our approach when integrating new experimental data.

6.2 Future Work

Currently, we represent the beliefs of the parameters as discrete probability distributions. The advantage is that such a representation is easy to implement. However, accuracy may be lost due to the discretization process. The computational complexity for belief propagation on discrete distributions also increases exponentially with the connectivity of the factor nodes. Here, we suggest some means to improve the performance and accuracy of belief propagation.

In addition, we discuss some of the directions that can be further explored using our probabilistic approach as the basic framework.

Sparse Representation

Within the joint probability distribution tables of the factor nodes, not all entries contain meaningful information. Given a factor node and its set of connected parameters, there could be several parameter combinations that are *not* feasible, i.e. having a probability that is close to 0. For a factor node that has a high proportion of such entries, one can implement the distribution table as a sparse matrix.

Pal et al. (2006) have implemented a version of belief propagation using sparse discrete distribution. Instead of representing the belief of each parameter as an array,

they store it as a finite list of probabilities such that the *compressed* form does not diverge from the true distribution by a given threshold limit. Although they have not extended this sparse representation to the conditional probability distributions in their model, it is possible to employ a similar technique to reduce the space and time requirements of belief propagation on factor graphs.

Continuous Distribution

Another possible extension to our approach is to implement the beliefs and dependencies as continuous distributions. These distributions can either be *Gaussian distributions* or *mixed-kernel distributions* (Sudderth et al., 2003).

Using continuous distributions, especially Gaussian ones, the domain of each parameter can be made wider as one only needs to store the mean and the variance for each parameter. Dependencies can be implemented as multivariate covariance matrices, which are much smaller than our current joint discrete distributions. In addition, the updates and operations of belief propagation can be formulated in terms of the means and covariance matrices, making it possible to propagate the beliefs even for a large factor graph model.

The correctness of belief propagation on large Gaussian graphical models of arbitrary topology has been studied by Weiss and Freeman (1999) and we postulate that similar methods can be used to enhance the applicability of our approach.

Loopy Belief Propagation

Other than modifying the representation of the beliefs and parameter dependencies, the loopy belief propagation scheme itself provides several opportunities for further improvements.

Ideally, the *shape* of the final belief of a pathway parameter should be an indication of the model sensitivity to that parameter, i.e. a flat distribution would imply that the model is not sensitive to that parameter since any value it assumes gives the same overall score. However, our current implementation of the loopy belief propagation algorithm is such that it will eventually converge to a relatively *well-defined* distribution.

The reason why the *MAP* probabilities have such well-defined shapes is that messages get propagated indefinitely around the factor graph, such that the beliefs of the parameters get amplified over time. Although this effect can still give us a good approximation for the assignment of values to the parameters, pertinent information related to the shape of the distribution is lost. Currently we handle this by setting a limit to

the number of iterations for belief propagation. This limit is proportional to the size of the factor graph.

Other issues when using loopy belief propagation are the presence of (i) *cycling errors*, where the old information is mistaken as new, and (ii) *convergence errors*, where the variable nodes adjacent to a factor node get their beliefs from another common factor node. The impact of these two errors on the final beliefs in the loopy belief propagation scheme has been briefly studied by Bolt and van der Gaag (2004).

Model Evaluation

Parameter estimation of bio-pathways has always been an under-constrained problem where the number of unknown parameters exceeds the information content provided by experimental data. The probabilistic approach to representing and estimating pathway parameters as put forth in this thesis can then form the foundation for futher understanding the issue of model evaluation.

As mentioned previously, the *a posteriori* distributions of the parameters indicate not only the most probable value, but also the level of uncertainty associated with them. For large bio-pathway models, the data may provide sufficient information only on some parts of the pathway. By studying the individual parameter distributions, biologists can then determine which molecule species require more measurement, and which components can be subjected to model reduction.

At present, our work does not tackle the issues of model overfitting, comparison and evaluation in the context of data assimilation considered in Nagasaki et al. (2006) and Tasaki et al. (2006). However, given our probabilistic framework, we should be able to adopt the Bayesian approach to model comparison and address these issues as well.

Apart from obtaining globally consistent parameter estimates, it will also be interesting to repeatedly perform our algorithm, and analyze the distributions of the estimated parameters. Doing so may reveal features concerning the structure of bio-pathways, as well as the adequacy of experimental data. They may also provide some means for model comparison and evaluation. To this end, we have performed some additional simulations in response to comments by reviewers.

6.3 Composition and Updating Pathway Models

In this thesis, the components were produced mainly as a result of decomposition. However, this need not always be the case. As mentioned in Chapter 4, pathway com-

ponents can arise from independent studies on different pathways. By standardizing the methods of capturing the relationships between the parameters and bordering molecule species, composition of dynamical pathway models can potentially be achieved without the need to re-estimate the parameters using all the experimental data.

Furthermore, new players and new interactions are continuously being discovered, modifying the pathway structure itself. By decomposing the pathway into smaller components, and allowing their factor graphs to be built independently, only the factor graphs of the components affected by the new interactions need to be *re-sampled*. The beliefs of the other parameters can then be updated as the new constraints are propagated to other parts of the pathway.

While the methods presented in this thesis are targeted specifically at the parameter estimation problem, it provides a framework that allows the systematic modeling, updating and integration of large biological pathway models.

Appendix A

Supplement Data for Chapter 3

The data in this appendix shows all the initial conditions and experimental data used for the parameter estimation of the Akt-MAPK signaling pathway. The perturbations in the initial conditions are highlighted in grey colored cells. The molecule species Serum, DPI and LY294002 are implemented as discrete places, and hence only assume the values 0 or 1 to indicate whether they are absent or present respectively. The concentrations of the other molecule species are given in nM.

A.1 Knockdown Experiments

The following experiments show the effects of knocking down Akt and PDK1 on the activity of certain molecule species. The proteins Akt and PDK1 are knocked down by introducing small interfering RNAs (siRNA) to the cells. siRNA are a class of short double stranded RNA molecules that regulates the concentration levels of the target proteins by interfering with the translation of their gene transcripts.

Experiment 1.1 Control

Initial Condition

Species	Concentration	Species	Concentration	Species	Concentration
Serum	1	Ras	18900	PP2A	150
R	80	Ras_a	0	MKP3	2.4
R_{act}	0	DPI	0	P90RSK	5
R_{int}	0	NOX5	2000	$P90RSK_p$	0
PI3K	100	ROS	200	Bad	100
$PI3K_a$	0	Pak	500	Bad_{p112}	0
PIP_2	7000	Pak_p	0	Bad_p136	0
PIP_3	0	Raf	66.4	Bcl	100
Akt_{cyto}	200	Raf_p	0	Bcl2.Bad	0
PIP_3.Akt	0	MEK	36500	Bcl2.Bax	0
$PIP_3.Akt_p$	0	MEK_p	0	Bax	0
$PIP_3.Akt_{pp}$	0	MEK_{pp}	0	Bax_{cyto}	100
$PDK1_{cyto}$	1000	ERK	34900	LY294002	0
PDK1	0	ERK_p	0	PTEN	0.1
PDK2	3	ERK_{pp}	0		

Experimental Data

Time(s)	0	3600
$PIP_3.Akt_{pp}$	0	138.2
Raf_p	0	64.47
MEK_{pp}	0	29966
ERK_{pp}	0	18811
Bad_{p112}	0	84.5

Table A.1: Control experiment.

Experiment 1.2 Akt siRNA 20 nM

Initial Condition

Species	Concentration	Species	Concentration	Species	Concentration
Serum	1	Ras	18900	PP2A	150
R	80	Ras_a	0	MKP3	2.4
R_{act}	0	DPI	0	P90RSK	5
R_{int}	0	NOX5	2,000	$P90RSK_p$	0
PI3K	100	ROS	200	Bad	100
$PI3K_a$	0	Pak	500	Bad_{p112}	0
PIP_2	7000	Pak_p	0	Bad_p136	0
PIP_3	0	Raf	66.4	Bcl	100
Akt_{cyto}	**32.13**	Raf_p	0	Bcl2.Bad	0
PIP_3.Akt	0	MEK	36500	Bcl2.Bax	0
$PIP_3.Akt_p$	0	MEK_p	0	Bax	0
$PIP_3.Akt_{pp}$	0	MEK_{pp}	0	Bax_{cyto}	100
$PDK1_{cyto}$	1000	ERK	34900	LY294002	0
PDK1	0	ERK_p	0	PTEN	0.1
PDK2	3	ERK_{pp}	0		

Experimental Data

Time(s)	0	3600
$PIP_3.Akt_{pp}$	0	15.68
Raf_p	0	51.53
MEK_{pp}	0	4781.5
ERK_{pp}	0	21742.7
Bad_{p112}	0	44

Table A.2: Treatment with 20 nM Akt siRNA.

Experiment 1.3 Akt siRNA 50 nM

Initial Condition

Species	Concentration	Species	Concentration	Species	Concentration
Serum	1	Ras	18900	PP2A	150
R	80	Ras_a	0	MKP3	2.4
R_{act}	0	DPI	0	P90RSK	5
R_{int}	0	NOX5	2000	$P90RSK_p$	0
PI3K	100	ROS	200	Bad	100
$PI3K_a$	0	Pak	500	Bad_{p112}	0
PIP_2	7000	Pak_p	0	Bad_p136	0
PIP_3	0	Raf	66.4	Bcl	100
Akt_{cyto}	**24.93**	Raf_p	0	Bcl2.Bad	0
PIP_3.Akt	0	MEK	36500	Bcl2.Bax	0
$PIP_3.Akt_p$	0	MEK_p	0	Bax	0
$PIP_3.Akt_{pp}$	0	MEK_{pp}	0	Bax_{cyto}	100
$PDK1_{cyto}$	1000	ERK	34900	LY294002	0
PDK1	0	ERK_p	0	PTEN	0.1
PDK2	3	ERK_{pp}	0		

Experimental Data

Time(s)	0	3600
$PIP_3.Akt_{pp}$	0	14.61
MEK_{pp}	0	16972.5
ERK_{pp}	0	27012.6
Bad_{p112}	0	69

Table A.3: Treatment with 50 nM Akt siRNA.

Experiment 1.4 Akt siRNA 100 nM

Initial Condition

Species	Concentration	Species	Concentration	Species	Concentration
Serum	1	Ras	18900	PP2A	150
R	80	Ras_a	0	MKP3	2.4
R_{act}	0	DPI	0	P90RSK	5
R_{int}	0	NOX5	2000	$P90RSK_p$	0
PI3K	100	ROS	200	Bad	100
$PI3K_a$	0	Pak	500	Bad_{p112}	0
PIP_2	7000	Pak_p	0	Bad_p136	0
PIP_3	0	Raf	66.4	Bcl	100
Akt_{cyto}	**20.43**	Raf_p	0	Bcl2.Bad	0
PIP_3.Akt	0	MEK	36500	Bcl2.Bax	0
$PIP_3.Akt_p$	0	MEK_p	0	Bax	0
$PIP_3.Akt_{pp}$	0	MEK_{pp}	0	Bax_{cyto}	100
$PDK1_{cyto}$	1000	ERK	34900	LY294002	0
PDK1	0	ERK_p	0	PTEN	0.1
PDK2	3	ERK_{pp}	0		

Experimental Data

Time(s)	0	3600
$PIP_3.Akt_{pp}$	0	19.90
MEK_{pp}	0	14235
ERK_{pp}	0	39786
Bad_{p112}	0	114.1

Table A.4: Treatment with 100 nM Akt siRNA.

Experiment 1.5 PDK1 siRNA 20 nM

Initial Condition

Species	Concentration	Species	Concentration	Species	Concentration
Serum	1	Ras	18900	PP2A	150
R	80	Ras_a	0	MKP3	2.4
R_{act}	0	DPI	0	P90RSK	5
R_{int}	0	NOX5	2000	$P90RSK_p$	0
PI3K	100	ROS	200	Bad	100
$PI3K_a$	0	Pak	500	Bad_{p112}	0
PIP_2	7000	Pak_p	0	Bad_p136	0
PIP_3	0	Raf	66.4	Bcl	100
Akt_{cyto}	200	Raf_p	0	Bcl2.Bad	0
PIP_3.Akt	0	MEK	36500	Bcl2.Bax	0
$PIP_3.Akt_p$	0	MEK_p	0	Bax	0
$PIP_3.Akt_{pp}$	0	MEK_{pp}	0	Bax_{cyto}	100
$PDK1_{cyto}$	**207.8**	ERK	34900	LY294002	0
PDK1	0	ERK_p	0	PTEN	0.1
PDK2	3	ERK_{pp}	0		

Experimental Data

Time(s)	0	3600
$PIP_3.Akt_{pp}$	0	40.8
Raf_p	0	65.072
MEK_{pp}	0	34857.5
ERK_{pp}	0	5269.9
Bad_{p112}	0	68

Table A.5: Treatment with 20 nM PDK1 siRNA.

Experiment 1.6 PDK1 siRNA 50 nM

Initial Condition

Species	Concentration	Species	Concentration	Species	Concentration
Serum	1	Ras	18900	PP2A	150
R	80	Ras_a	0	MKP3	2.4
R_{act}	0	DPI	0	P90RSK	5
R_{int}	0	NOX5	2000	$P90RSK_p$	0
PI3K	100	ROS	200	Bad	100
$PI3K_a$	0	Pak	500	Bad_{p112}	0
PIP_2	7000	Pak_p	0	Bad_p136	0
PIP_3	0	Raf	66.4	Bcl	100
Akt_{cyto}	200	Raf_p	0	Bcl2.Bad	0
PIP_3.Akt	0	MEK	36500	Bcl2.Bax	0
$PIP_3.Akt_p$	0	MEK_p	0	Bax	0
$PIP_3.Akt_{pp}$	0	MEK_{pp}	0	Bax_{cyto}	100
$PDK1_{cyto}$	**178**	ERK	34900	LY294002	0
PDK1	0	ERK_p	0	PTEN	0.1
PDK2	3	ERK_{pp}	0		

Experimental Data

Time(s)	0	3600
$PIP_3.Akt_{pp}$	0	95.6
MEK_{pp}	0	39785
ERK_{pp}	0	3629.6
Bad_{p112}	0	48.2

Table A.6: Treatment with 50 nM PDK1 siRNA.

Experiment 1.7 PDK1 siRNA 100 nM

Initial Condition

Species	Concentration	Species	Concentration	Species	Concentration
Serum	1	Ras	18900	PP2A	150
R	80	Ras_a	0	MKP3	2.4
R_{act}	0	DPI	0	P90RSK	5
R_{int}	0	NOX5	2000	$P90RSK_p$	0
PI3K	100	ROS	200	Bad	100
$PI3K_a$	0	Pak	500	Bad_{p112}	0
PIP_2	7000	Pak_p	0	Bad_p136	0
PIP_3	0	Raf	66.4	Bcl	100
Akt_{cyto}	200	Raf_p	0	Bcl2.Bad	0
PIP_3.Akt	0	MEK	36500	Bcl2.Bax	0
$PIP_3.Akt_p$	0	MEK_p	0	Bax	0
$PIP_3.Akt_{pp}$	0	MEK_{pp}	0	Bax_{cyto}	100
$PDK1_{cyto}$	**234.1**	ERK	34900	LY294002	0
PDK1	0	ERK_p	0	PTEN	0.1
PDK2	3	ERK_{pp}	0		

Experimental Data

Time(s)	0	3600
$PIP_3.Akt_{pp}$	0	23.2
MEK_{pp}	0	26426
ERK_{pp}	0	872.5
Bad_{p112}	0	43.5

Table A.7: Treatment with 100 nM PDK1 siRNA.

A.2 Treatment Experiments

In these sets of experiments, the cells are treated with different combinations of the chemicals diphenyleneiodonium (DPI), LY294002 and PP2A siRNA. DPI is a NADPH oxidase inhibitor that reduces the production of reactive oxygen species (ROS) and LY294002 is a chemical that inhibits the activation of PI3K. Some of the experiments are also performed in the absence of Serum.

Experiment 2.1 - Control

Initial Condition

Species	Concentration	Species	Concentration	Species	Concentration
Serum	1	Ras	18900	PP2A	150
R	80	Ras_a	0	MKP3	2.4
R_{act}	0	DPI	0	P90RSK	5
R_{int}	0	NOX5	2000	$P90RSK_p$	0
PI3K	100	ROS	200	Bad	100
$PI3K_a$	0	Pak	500	Bad_{p112}	0
PIP_2	7000	Pak_p	0	Bad_p136	0
PIP_3	0	Raf	66.4	Bcl	100
Akt_{cyto}	200	Raf_p	0	Bcl2.Bad	0
PIP_3.Akt	0	MEK	36500	Bcl2.Bax	0
$PIP_3.Akt_p$	0	MEK_p	0	Bax	0
$PIP_3.Akt_{pp}$	0	MEK_{pp}	0	Bax_{cyto}	100
$PDK1_{cyto}$	1000	ERK	34900	LY294002	0
PDK1	0	ERK_p	0	PTEN	0.1
PDK2	3	ERK_{pp}	0		

Experimental Data

Time(s)	0	3600
$PIP_3.Akt_{pp}$	0	114.8
MEK_{pp}	0	19965.5
ERK_{pp}	0	27361.6
Bad_{p112}	0	36.6
$P90RSK_p$	0	0.975

Table A.8: Control experiment.

Experiment 2.2 PP2A siRNA

Initial Condition

Species	Concentration	Species	Concentration	Species	Concentration
Serum	1	Ras	18900	PP2A	**41.44**
R	80	Ras_a	0	MKP3	2.4
R_{act}	0	DPI	0	P90RSK	5
R_{int}	0	NOX5	2000	$P90RSK_p$	0
PI3K	100	ROS	200	Bad	100
$PI3K_a$	0	Pak	500	Bad_{p112}	0
PIP_2	7000	Pak_p	0	Bad_p136	0
PIP_3	0	Raf	66.4	Bcl	100
Akt_{cyto}	200	Raf_p	0	Bcl2.Bad	0
PIP_3.Akt	0	MEK	36500	Bcl2.Bax	0
$PIP_3.Akt_p$	0	MEK_p	0	Bax	0
$PIP_3.Akt_{pp}$	0	MEK_{pp}	0	Bax_{cyto}	100
$PDK1_{cyto}$	1000	ERK	34900	LY294002	0
PDK1	0	ERK_p	0	PTEN	0.1
PDK2	3	ERK_{pp}	0		

Experimental Data

Time(s)	0	3600
$PIP_3.Akt_{pp}$	0	150
MEK_{pp}	0	41318
ERK_{pp}	0	20870.2
Bad_{p112}	0	33.7
$P90RSK_p$	0	0.445

Table A.9: Treatment with PP2A siRNA.

Experiment 2.3 No Serum

Initial Condition

Species	Concentration	Species	Concentration	Species	Concentration
Serum	**0**	Ras	18900	PP2A	150
R	80	Ras_a	0	MKP3	2.4
R_{act}	0	DPI	0	P90RSK	5
R_{int}	0	NOX5	2000	$P90RSK_p$	0
PI3K	100	ROS	200	Bad	100
$PI3K_a$	0	Pak	500	Bad_{p112}	0
PIP_2	7000	Pak_p	0	Bad_p136	0
PIP_3	0	Raf	66.4	Bcl	100
Akt_{cyto}	200	Raf_p	0	Bcl2.Bad	0
PIP_3.Akt	0	MEK	36500	Bcl2.Bax	0
$PIP_3.Akt_p$	0	MEK_p	0	Bax	0
$PIP_3.Akt_{pp}$	0	MEK_{pp}	0	Bax_{cyto}	100
$PDK1_{cyto}$	1000	ERK	34900	LY294002	0
PDK1	0	ERK_p	0	PTEN	0.1
PDK2	3	ERK_{pp}	0		

Experimental Data

Time(s)	0	3600
$PIP_3.Akt_{pp}$	0	85.2
MEK_{pp}	0	8614
Bad_{p112}	0	9.7
$P90RSK_p$	0	1.52

Table A.10: No Serum.

Experiment 2.4 No Serum and PP2A siRNA

Initial Condition

Species	Concentration	Species	Concentration	Species	Concentration
Serum	**0**	Ras	18900	PP2A	**33.48**
R	80	Ras_a	0	MKP3	2.4
R_{act}	0	DPI	0	P90RSK	5
R_{int}	0	NOX5	2000	$P90RSK_p$	0
PI3K	100	ROS	200	Bad	100
$PI3K_a$	0	Pak	500	Bad_{p112}	0
PIP_2	7000	Pak_p	0	Bad_p136	0
PIP_3	0	Raf	66.4	Bcl	100
Akt_{cyto}	200	Raf_p	0	Bcl2.Bad	0
PIP_3.Akt	0	MEK	36500	Bcl2.Bax	0
$PIP_3.Akt_p$	0	MEK_p	0	Bax	0
$PIP_3.Akt_{pp}$	0	MEK_{pp}	0	Bax_{cyto}	100
$PDK1_{cyto}$	1000	ERK	34900	LY294002	0
PDK1	0	ERK_p	0	PTEN	0.1
PDK2	3	ERK_{pp}	0		

Experimental Data

Time(s)	0	3600
$PIP_3.Akt_{pp}$	0	107.6
MEK_{pp}	0	24783.5
Bad_{p112}	0	18.3
$P90RSK_p$	0	0.62

Table A.11: Treatment with PP2A siRNA and no Serum.

Experiment 2.5 DPI Treatment

Initial Condition

Species	Concentration	Species	Concentration	Species	Concentration
Serum	1	Ras	18900	PP2A	150
R	80	Ras_a	0	MKP3	2.4
R_{act}	0	DPI	**1**	P90RSK	5
R_{int}	0	NOX5	2000	$P90RSK_p$	0
PI3K	100	ROS	200	Bad	100
$PI3K_a$	0	Pak	500	Bad_{p112}	0
PIP_2	7000	Pak_p	0	Bad_p136	0
PIP_3	0	Raf	66.4	Bcl	100
Akt_{cyto}	200	Raf_p	0	Bcl2.Bad	0
PIP_3.Akt	0	MEK	36500	Bcl2.Bax	0
$PIP_3.Akt_p$	0	MEK_p	0	Bax	0
$PIP_3.Akt_{pp}$	0	MEK_{pp}	0	Bax_{cyto}	100
$PDK1_{cyto}$	1000	ERK	34900	LY294002	0
PDK1	0	ERK_p	0	PTEN	0.1
PDK2	3	ERK_{pp}	0		

Experimental Data

Time(s)	0	3600
$PIP_3.Akt_{pp}$	0	74.8
MEK_{pp}	0	11935.5
ERK_{pp}	0	1047
Bad_{p112}	0	20.3
$P90RSK_p$	0	0.93

Table A.12: Treatment with DPI.

Experiment 2.6 DPI Treatment and PP2A siRNA

Initial Condition

Species	Concentration	Species	Concentration	Species	Concentration
Serum	1	Ras	18900	PP2A	**63.2**
R	80	Ras_a	0	MKP3	2.4
R_{act}	0	DPI	**1**	P90RSK	5
R_{int}	0	NOX5	2000	$P90RSK_p$	0
PI3K	100	ROS	200	Bad	100
$PI3K_a$	0	Pak	500	Bad_{p112}	0
PIP_2	7000	Pak_p	0	Bad_p136	0
PIP_3	0	Raf	66.4	Bcl	100
Akt_{cyto}	200	Raf_p	0	Bcl2.Bad	0
PIP_3.Akt	0	MEK	36500	Bcl2.Bax	0
$PIP_3.Akt_p$	0	MEK_p	0	Bax	0
$PIP_3.Akt_{pp}$	0	MEK_{pp}	0	Bax_{cyto}	100
$PDK1_{cyto}$	1000	ERK	34900	LY294002	0
PDK1	0	ERK_p	0	PTEN	0.1
PDK2	3	ERK_{pp}	0		

Experimental Data

Time(s)	0	3600
$PIP_3.Akt_{pp}$	0	181.8
MEK_{pp}	0	37814
ERK_{pp}	0	3455.1
Bad_{p112}	0	14.8
$P90RSK_p$	0	3.995

Table A.13: Treatment with DPI and PP2A siRNA.

Experiment 2.7 LY294002 Treatment

Initial Condition

Species	Concentration	Species	Concentration	Species	Concentration
Serum	1	Ras	18900	PP2A	150
R	80	Ras_a	0	MKP3	2.4
R_{act}	0	DPI	0	P90RSK	5
R_{int}	0	NOX5	2000	$P90RSK_p$	0
PI3K	100	ROS	200	Bad	100
$PI3K_a$	0	Pak	500	Bad_{p112}	0
PIP_2	7000	Pak_p	0	Bad_p136	0
PIP_3	0	Raf	66.4	Bcl	100
Akt_{cyto}	200	Raf_p	0	Bcl2.Bad	0
PIP_3.Akt	0	MEK	36500	Bcl2.Bax	0
$PIP_3.Akt_p$	0	MEK_p	0	Bax	0
$PIP_3.Akt_{pp}$	0	MEK_{pp}	0	Bax_{cyto}	100
$PDK1_{cyto}$	1000	ERK	34900	LY294002	**1**
PDK1	0	ERK_p	0	PTEN	0.1
PDK2	3	ERK_{pp}	0		

Experimental Data

Time(s)	0	3600
$PIP_3.Akt_{pp}$	0	19.6
MEK_{pp}	0	18578.5
ERK_{pp}	0	14553.3
Bad_{p112}	0	1.1

Table A.14: Treatment with LY297002.

Experiment 2.8 LY294002 Treatment and PP2A siRNA

Initial Condition

Species	Concentration	Species	Concentration	Species	Concentration
Serum	1	Ras	18900	PP2A	**2.73**
R	80	Ras_a	0	MKP3	2.4
R_{act}	0	DPI	0	P90RSK	5
R_{int}	0	NOX5	2000	$P90RSK_p$	0
PI3K	100	ROS	200	Bad	100
$PI3K_a$	0	Pak	500	Bad_{p112}	0
PIP_2	7000	Pak_p	0	Bad_p136	0
PIP_3	0	Raf	66.4	Bcl	100
Akt_{cyto}	200	Raf_p	0	Bcl2.Bad	0
PIP_3.Akt	0	MEK	36500	Bcl2.Bax	0
$PIP_3.Akt_p$	0	MEK_p	0	Bax	0
$PIP_3.Akt_{pp}$	0	MEK_{pp}	0	Bax_{cyto}	100
$PDK1_{cyto}$	1000	ERK	34900	LY294002	**1**
PDK1	0	ERK_p	0	PTEN	0.1
PDK2	3	ERK_{pp}	0		

Experimental Data

Time(s)	0	3600
$PIP_3.Akt_{pp}$	0	21.6
MEK_{pp}	0	23652
ERK_{pp}	0	5549.1
Bad_{p112}	0	9.4

Table A.15: Treatment with LY297002 and PP2A siRNA.

Experiment 2.9 DPI and LY294002 Treatment

Initial Condition

Species	Concentration	Species	Concentration	Species	Concentration
Serum	1	Ras	18900	PP2A	150
R	80	Ras_a	0	MKP3	2.4
R_{act}	0	DPI	**1**	P90RSK	5
R_{int}	0	NOX5	2000	$P90RSK_p$	0
PI3K	100	ROS	200	Bad	100
$PI3K_a$	0	Pak	500	Bad_{p112}	0
PIP_2	7000	Pak_p	0	Bad_p136	0
PIP_3	0	Raf	66.4	Bcl	100
Akt_{cyto}	200	Raf_p	0	Bcl2.Bad	0
PIP_3.Akt	0	MEK	36500	Bcl2.Bax	0
$PIP_3.Akt_p$	0	MEK_p	0	Bax	0
$PIP_3.Akt_{pp}$	0	MEK_{pp}	0	Bax_{cyto}	100
$PDK1_{cyto}$	1000	ERK	34900	LY294002	**1**
PDK1	0	ERK_p	0	PTEN	0.1
PDK2	3	ERK_{pp}	0		

Experimental Data

Time(s)	0	3600
$PIP_3.Akt_{pp}$	0	23
MEK_{pp}	0	5438.5
ERK_{pp}	0	418.8
Bad_{p112}	0	7.6

Table A.16: Treatment with DPI and PP2A siRNA.

Experiment 2.10 DPI, LY294002 Treatments and PP2A siRNA

Initial Condition

Species	Concentration	Species	Concentration	Species	Concentration
Serum	1	Ras	18900	PP2A	**9.75**
R	80	Ras_a	0	MKP3	2.4
R_{act}	0	DPI	**1**	P90RSK	5
R_{int}	0	NOX5	2000	$P90RSK_p$	0
PI3K	100	ROS	200	Bad	100
$PI3K_a$	0	Pak	500	Bad_{p112}	0
PIP_2	7000	Pak_p	0	Bad_p136	0
PIP_3	0	Raf	66.4	Bcl	100
Akt_{cyto}	200	Raf_p	0	Bcl2.Bad	0
PIP_3.Akt	0	MEK	36500	Bcl2.Bax	0
$PIP_3.Akt_p$	0	MEK_p	0	Bax	0
$PIP_3.Akt_{pp}$	0	MEK_{pp}	0	Bax_{cyto}	100
$PDK1_{cyto}$	1000	ERK	34900	LY294002	**1**
PDK1	0	ERK_p	0	PTEN	0.1
PDK2	3	ERK_{pp}	0		

Experimental Data

Time(s)	0	3600
$PIP_3.Akt_{pp}$	0	34
MEK_{pp}	0	15804.5
ERK_{pp}	0	139.6
Bad_{p112}	0	1.9

Table A.17: Treatment with DPI, LY294002 and PP2A siRNA.

A.3 Time Series Data

These sets of experiments show the time-series data of certain molecule species under the various treatments.

Experiment 3.1 Control

Initial Condition

Species	Concentration	Species	Concentration	Species	Concentration
Serum	1	Ras	18900	PP2A	150
R	80	Ras_a	0	MKP3	2.4
R_{act}	0	DPI	0	P90RSK	5
R_{int}	0	NOX5	2000	$P90RSK_p$	0
PI3K	100	ROS	200	Bad	100
$PI3K_a$	0	Pak	500	Bad_{p112}	0
PIP_2	7000	Pak_p	0	Bad_p136	0
PIP_3	0	Raf	66.4	Bcl	100
Akt_{cyto}	200	Raf_p	0	Bcl2.Bad	0
PIP_3.Akt	0	MEK	36500	Bcl2.Bax	0
$PIP_3.Akt_p$	0	MEK_p	0	Bax	0
$PIP_3.Akt_{pp}$	0	MEK_{pp}	0	Bax_{cyto}	100
$PDK1_{cyto}$	1000	ERK	34900	LY294002	0
PDK1	0	ERK_p	0	PTEN	0.1
PDK2	3	ERK_{pp}	0		

Experimental Data

Time(s)	0	600	1200	1800	2700	3600	7200
$PIP_3.Akt_{pp}$	0	4	43.4	2.2	33.8	54.8	250
MEK_{pp}	0	21863.5	19308.5	12227.5	13870	14052.5	13176.5
ERK_{pp}	0	22126.6	30014	7329	12424.4	13401.6	18636.6
Bad_{p112}	0	32.7	34.4	46.2	41.7	28.8	47.5
$P90RSK_p$	0	3.845	3.025	0.28	0.62	1.715	0.74

Table A.18: Control experiment.

Experiment 3.2 No Serum

Initial Condition

Species	Concentration	Species	Concentration	Species	Concentration
Serum	**0**	Ras	18900	PP2A	150
R	80	Ras_a	0	MKP3	2.4
R_{act}	0	DPI	0	P90RSK	5
R_{int}	0	NOX5	2000	$P90RSK_p$	0
PI3K	100	ROS	200	Bad	100
$PI3K_a$	0	Pak	500	Bad_{p112}	0
PIP_2	7000	Pak_p	0	Bad_p136	0
PIP_3	0	Raf	66.4	Bcl	100
Akt_{cyto}	200	Raf_p	0	Bcl2.Bad	0
PIP_3.Akt	0	MEK	36500	Bcl2.Bax	0
$PIP_3.Akt_p$	0	MEK_p	0	Bax	0
$PIP_3.Akt_{pp}$	0	MEK_{pp}	0	Bax_{cyto}	100
$PDK1_{cyto}$	1000	ERK	34900	LY294002	0
PDK1	0	ERK_p	0	PTEN	0.1
PDK2	3	ERK_{pp}	0		

Experimental Data

Time(s)	0	600	1200	1800	2700	3600	7200
$PIP_3.Akt_{pp}$	0	59.8	92.2	17.6	25	12.6	89.8
MEK_{pp}	0	29419	15549	14527	14016	10329.5	13030.5
ERK_{pp}	0	18427.2	9108.9	2094	4188	418.8	5444.4
Bad_{p112}	0	24.6	27.2	24.4	32.2	16.7	17.9
$P90RSK_p$	0	3.64	1.82	0.165	0.315	0.225	0.21

Table A.19: Time series data with no Serum.

Experiment 3.3 LY294002 Treatment

Initial Condition

Species	Concentration	Species	Concentration	Species	Concentration
Serum	1	Ras	18900	PP2A	150
R	80	Ras_a	0	MKP3	2.4
R_{act}	0	DPI	0	P90RSK	5
R_{int}	0	NOX5	2000	$P90RSK_p$	0
PI3K	100	ROS	200	Bad	100
$PI3K_a$	0	Pak	500	Bad_{p112}	0
PIP_2	7000	Pak_p	0	Bad_p136	0
PIP_3	0	Raf	66.4	Bcl	100
Akt_{cyto}	200	Raf_p	0	Bcl2.Bad	0
PIP_3.Akt	0	MEK	36500	Bcl2.Bax	0
$PIP_3.Akt_p$	0	MEK_p	0	Bax	0
$PIP_3.Akt_{pp}$	0	MEK_{pp}	0	Bax_{cyto}	100
$PDK1_{cyto}$	1000	ERK	34900	LY294002	**1**
PDK1	0	ERK_p	0	PTEN	0.1
PDK2	3	ERK_{pp}	0		

Experimental Data

Time(s)	0	600	1200	1800	2700	3600	7200
$PIP_3.Akt_{pp}$	0	5.4	7.2	0	0	0	0
MEK_{pp}	0	20586	22082.5	13468.5	11607	14235	10293
ERK_{pp}	0	22685	32212.7	4188	14274.1	4502.1	7538.4
Bad_{p112}	0	30.5	17.6	21.4	21	11.7	8.3
$P90RSK_p$	0	2.99	3.17	0.23	0.285	0.41	0.125

Table A.20: Time series data with LY294002 treatment.

Experiment 3.4 No Serum and LY294002 Treatment

Initial Condition

Species	Concentration	Species	Concentration	Species	Concentration
Serum	**0**	Ras	18900	PP2A	150
R	80	Ras_a	0	MKP3	2.4
R_{act}	0	DPI	0	P90RSK	5
R_{int}	0	NOX5	2000	$P90RSK_p$	0
PI3K	100	ROS	200	Bad	100
$PI3K_a$	0	Pak	500	Bad_{p112}	0
PIP_2	7000	Pak_p	0	Bad_p136	0
PIP_3	0	Raf	66.4	Bcl	100
Akt_{cyto}	200	Raf_p	0	Bcl2.Bad	0
PIP_3.Akt	0	MEK	36500	Bcl2.Bax	0
$PIP_3.Akt_p$	0	MEK_p	0	Bax	0
$PIP_3.Akt_{pp}$	0	MEK_{pp}	0	Bax_{cyto}	100
$PDK1_{cyto}$	1000	ERK	34900	LY294002	**1**
PDK1	0	ERK_p	0	PTEN	0.1
PDK2	3	ERK_{pp}	0		

Experimental Data

Time(s)	0	600	1200	1800	2700	3600	7200
$PIP_3.Akt_{pp}$	0	3.6	8.6	0	0	0	0
MEK_{pp}	0	17155	17374	11132.5	10110.5	11388	11278.5
ERK_{pp}	0	18741.3	18392.3	2059.1	1535.6	698	1151.7
Bad_{p112}	0	19	16.9	24	11.9	9.6	4.7
$P90RSK_p$	0	2.225	1.755	0.08	0.22	0.075	0.06

Table A.21: Time series data with LY294002 treatment and no Serum.

Experiment 3.5 DPI Treatment

Initial Condition

Species	Concentration	Species	Concentration	Species	Concentration
Serum	1	Ras	18900	PP2A	150
R	80	Ras_a	0	MKP3	2.4
R_{act}	0	DPI	**1**	P90RSK	5
R_{int}	0	NOX5	2000	$P90RSK_p$	0
PI3K	100	ROS	200	Bad	100
$PI3K_a$	0	Pak	500	Bad_{p112}	0
PIP_2	7000	Pak_p	0	Bad_p136	0
PIP_3	0	Raf	66.4	Bcl	100
Akt_{cyto}	200	Raf_p	0	Bcl2.Bad	0
PIP_3.Akt	0	MEK	36500	Bcl2.Bax	0
$PIP_3.Akt_p$	0	MEK_p	0	Bax	0
$PIP_3.Akt_{pp}$	0	MEK_{pp}	0	Bax_{cyto}	100
$PDK1_{cyto}$	1000	ERK	34900	LY294002	0
PDK1	0	ERK_p	0	PTEN	0.1
PDK2	3	ERK_{pp}	0		

Experimental Data

Time(s)	0	600	1200	1800	2700	3600	7200
$PIP_3.Akt_{pp}$	0	33.8	66.2	7	69.2	17.8	167.8
MEK_{pp}	0	22776	6497	14709.5	16680.5	13578	8979
ERK_{pp}	0	20451.4	31305.3	15600.3	16019.1	10819	13645.9
Bad_{p112}	0	50.4	31.9	51	38.4	31.8	53.5
$P90RSK_p$	0	3.935	2.615	0.22	0.675	0.195	0.21

Table A.22: Time series data with DPI treatment.

Experiment 3.6 No Serum and DPI Treatment

Initial Condition

Species	Concentration	Species	Concentration	Species	Concentration
Serum	**0**	Ras	18900	PP2A	150
R	80	Ras_a	0	MKP3	2.4
R_{act}	0	DPI	**1**	P90RSK	5
R_{int}	0	NOX5	2000	$P90RSK_p$	0
PI3K	100	ROS	200	Bad	100
$PI3K_a$	0	Pak	500	Bad_{p112}	0
PIP_2	7000	Pak_p	0	Bad_p136	0
PIP_3	0	Raf	66.4	Bcl	100
Akt_{cyto}	200	Raf_p	0	Bcl2.Bad	0
PIP_3.Akt	0	MEK	36500	Bcl2.Bax	0
$PIP_3.Akt_p$	0	MEK_p	0	Bax	0
$PIP_3.Akt_{pp}$	0	MEK_{pp}	0	Bax_{cyto}	100
$PDK1_{cyto}$	1000	ERK	34900	LY294002	0
PDK1	0	ERK_p	0	PTEN	0.1
PDK2	3	ERK_{pp}	0		

Experimental Data

Time(s)	0	600	1200	1800	2700	3600	7200
$PIP_3.Akt_{pp}$	0	33.2	34.4	2.2	67.8	7.6	25.6
MEK_{pp}	0	22812.5	7628.5	15074.5	14782.5	12629	7884
ERK_{pp}	0	14343.9	24081	732.9	2792	2163.8	1081.9
Bad_{p112}	0	33.1	27.2	40.6	25.6	23.6	23.8
$P90RSK_p$	0	3.54	1.625	0.09	0.195	0.03	0.07

Table A.23: Time series data with DPI treatment and no Serum.

A.4 Validation Data Set

These sets of data are used for the validation of our parameter estimation algorithm. They are not shown to the algorithms during the estimation process itself. Rather, the model is simulated with the estimated parameters to produce protein concentration profiles, which are then compared with the validation data sets to assess the quality of the estimated parameters.

Experiment 4.1 Control

Initial Condition

Species	Concentration	Species	Concentration	Species	Concentration
Serum	1	Ras	18900	PP2A	150
R	80	Ras_a	0	MKP3	2.4
R_{act}	0	DPI	0	P90RSK	5
R_{int}	0	NOX5	2000	$P90RSK_p$	0
PI3K	100	ROS	200	Bad	100
$PI3K_a$	0	Pak	500	Bad_{p112}	0
PIP_2	7000	Pak_p	0	Bad_p136	0
PIP_3	0	Raf	66.4	Bcl	100
Akt_{cyto}	200	Raf_p	0	Bcl2.Bad	0
PIP_3.Akt	0	MEK	36500	Bcl2.Bax	0
$PIP_3.Akt_p$	0	MEK_p	0	Bax	0
$PIP_3.Akt_{pp}$	0	MEK_{pp}	0	Bax_{cyto}	100
$PDK1_{cyto}$	1000	ERK	34900	LY294002	0
PDK1	0	ERK_p	0	PTEN	0.1
PDK2	3	ERK_{pp}	0		

Experimental Data

Time(s)	0	600	1200	1800	2700	3600	7200
$PIP_3.Akt_{pp}$	0	40.8	35.8	31.6	104.8	124.8	116
MEK_{pp}	0	15330	11862.5	7519	7008	8066.5	7190.5
ERK_{pp}	0	37692	35249	24499.8	27222	23034	17868.8
Bad_{p112}	0	64.6	75.2	78.6	78.5	82.5	83
$P90RSK_p$	0	3.185	3	1.8	1.715	1.55	0.1505

Table A.24: Control experiment.

Experiment 4.2 No Serum

Initial Condition

Species	Concentration	Species	Concentration	Species	Concentration
Serum	**0**	Ras	18900	PP2A	150
R	80	Ras_a	0	MKP3	2.4
R_{act}	0	DPI	0	P90RSK	5
R_{int}	0	NOX5	2000	$P90RSK_p$	0
PI3K	100	ROS	200	Bad	100
$PI3K_a$	0	Pak	500	Bad_{p112}	0
PIP_2	7000	Pak_p	0	Bad_p136	0
PIP_3	0	Raf	66.4	Bcl	100
Akt_{cyto}	200	Raf_p	0	Bcl2.Bad	0
PIP_3.Akt	0	MEK	36500	Bcl2.Bax	0
$PIP_3.Akt_p$	0	MEK_p	0	Bax	0
$PIP_3.Akt_{pp}$	0	MEK_{pp}	0	Bax_{cyto}	100
$PDK1_{cyto}$	1000	ERK	34900	LY294002	0
PDK1	0	ERK_p	0	PTEN	0.1
PDK2	3	ERK_{pp}	0		

Experimental Data

Time(s)	0	600	1200	1800	2700	3600	7200
$PIP_3.Akt_{pp}$	0	67.6	73.2	67.4	49.4	26	0
MEK_{pp}	0	8723.5	2445.5	1168	1131.5	1277.5	803
ERK_{pp}	0	24185.7	11237.8	1116.8	1012.1	1989.3	2896.7
Bad_{p112}	0	34.8	34.1	33.5	34.2	40	40
$P90RSK_p$	0	1.82	0.4	0.085	0.03	0.04	0.07

Table A.25: Validation time series data with no Serum.

Experiment 4.3 LY294002 Treatment

Initial Condition

Species	Concentration	Species	Concentration	Species	Concentration
Serum	1	Ras	18900	PP2A	150
R	80	Ras_a	0	MKP3	2.4
R_{act}	0	DPI	0	P90RSK	5
R_{int}	0	NOX5	2000	$P90RSK_p$	0
PI3K	100	ROS	200	Bad	100
$PI3K_a$	0	Pak	500	Bad_{p112}	0
PIP_2	7000	Pak_p	0	Bad_p136	0
PIP_3	0	Raf	66.4	Bcl	100
Akt_{cyto}	200	Raf_p	0	Bcl2.Bad	0
PIP_3.Akt	0	MEK	36500	Bcl2.Bax	0
$PIP_3.Akt_p$	0	MEK_p	0	Bax	0
$PIP_3.Akt_{pp}$	0	MEK_{pp}	0	Bax_{cyto}	100
$PDK1_{cyto}$	1000	ERK	34900	LY294002	**1**
PDK1	0	ERK_p	0	PTEN	0.1
PDK2	3	ERK_{pp}	0		

Experimental Data

Time(s)	0	600	1200	1800	2700	3600	7200
$PIP_3.Akt_{pp}$	0	27.8	2	0	0	0	0
MEK_{pp}	0	14600	10950	6570	3321.5	5329	2701
ERK_{pp}	0	34900	23906.5	17275.5	13366.7	17973.5	5584
Bad_{p112}	0	53.5	38.3	12.4	7	2.2	8.18
$P90RSK_p$	0	2.8	2.075	1.22	0.21	0.64	0.35

Table A.26: Validation time series data with LY294002 treatment.

Experiment 4.4 No Serum and LY294002 Treatment

Initial Condition

Species	Concentration	Species	Concentration	Species	Concentration
Serum	**0**	Ras	18900	PP2A	150
R	80	Ras_a	0	MKP3	2.4
R_{act}	0	DPI	0	P90RSK	5
R_{int}	0	NOX5	2000	$P90RSK_p$	0
PI3K	100	ROS	200	Bad	100
$PI3K_a$	0	Pak	500	Bad_{p112}	0
PIP_2	7000	Pak_p	0	Bad_p136	0
PIP_3	0	Raf	66.4	Bcl	100
Akt_{cyto}	200	Raf_p	0	Bcl2.Bad	0
PIP_3.Akt	0	MEK	36500	Bcl2.Bax	0
$PIP_3.Akt_p$	0	MEK_p	0	Bax	0
$PIP_3.Akt_{pp}$	0	MEK_{pp}	0	Bax_{cyto}	100
$PDK1_{cyto}$	1000	ERK	34900	LY294002	**1**
PDK1	0	ERK_p	0	PTEN	0.1
PDK2	3	ERK_{pp}	0		

Experimental Data

Time(s)	0	600	1200	1800	2700	3600	7200
$PIP_3.Akt_{pp}$	0	25.8	5	0	0	0	0
MEK_{pp}	0	6789	813.95	146	328.5	638.75	452.6
ERK_{pp}	0	23941.4	11342.5	453.7	349	237.32	446.72
Bad_{p112}	0	36.5	31.3	24	25.6	18.4	7.3
$P90RSK_p$	0	1.505	0.365	0	0.06	0.13	0.085

Table A.27: Validation time series data with LY294002 treatment and no Serum.

Appendix B

Supplement Data for Chapter 5

The data in this appendix shows the results of the simulations performed using the three algorithms - belief propagation (BP), Evolution Strategies with stochastic ranking (SRES) and genetic algorithms (GA). For each number of datasets, the simulations are performed 10 times, and the best scoring result is highlighted in grey colored cell. The data sets are introduced incrementally, such that for belief propagation, the factor graph and its sampled distributions that produce the best result for the current datasets are used when the information from the next dataset is being integrated. For SRES and GA, since they are not able to store the results of previous computations, they have to re-compute the estimates using all the available datasets.

Belief propagation

Simulation	1 Dataset	2 Datasets	3 Datasets	4 Datasets	5 Datasets	6 Datasets
1	1.395	1.122	2.558	1.726	1.006	0.881
2	0.814	0.400	1.626	8.142	0.582	0.729
3	2.818	1.167	1.370	3.334	0.582	**0.456**
4	2.623	4.180	2.014	0.864	0.783	0.806
5	6.269	8.357	0.618	2.734	**0.496**	0.636
6	0.245	2.670	**0.154**	**0.382**	0.508	3.923
7	0.420	0.944	2.750	1.673	0.531	0.601
8	1.611	**0.154**	0.260	1.844	5.642	0.504
9	1.159	0.418	0.862	3.347	3.196	2.247
10	**0.133**	0.206	1.632	7.775	1.243	0.703
Mean	1.749	1.962	1.384	3.182	1.457	1.149
Std Dev	1.836	2.579	0.907	2.695	1.682	1.102

Table B.1: Simulation results using belief propagation (BP).

Evolution Strategies with stochastic ranking

Simulation	1 Dataset	2 Datasets	3 Datasets	4 Datasets	5 Datasets	6 Datasets
1	**0.849**	1.641	3.054	5.092	**3.465**	5.343
2	1.743	2.202	1.937	6.528	4.299	**4.245**
3	1.527	1.518	2.670	6.235	11.947	9.745
4	1.766	2.017	2.355	4.067	4.656	6.455
5	2.465	**1.119**	2.383	3.677	8.940	10.405
6	2.371	1.329	**1.817**	4.213	3.542	6.286
7	1.455	2.604	1.956	6.785	5.498	7.362
8	1.426	1.926	3.316	4.161	9.496	5.516
9	1.144	1.285	3.082	**2.903**	10.026	7.046
10	1.035	2.970	2.251	3.230	3.986	5.525
Mean	1.578	1.861	2.482	4.689	6.586	6.793
Std Dev	0.531	0.602	0.528	1.397	3.171	1.954

Table B.2: Simulation results using Evolution Strategies with stochastic ranking (SRES).

Genetic algorithm

Simulation	1 Dataset	2 Datasets	3 Datasets	4 Datasets	5 Datasets	6 Datasets
1	6.032	6.008	7.213	10.496	20.411	24.586
2	10.777	**4.258**	6.770	9.497	54.784	21.239
3	3.048	8.988	11.711	21.697	**18.331**	24.548
4	7.116	8.047	10.548	8.113	52.305	20.463
5	5.345	7.660	19.561	6.749	37.165	40.041
6	2.902	4.493	**4.737**	14.064	37.194	**15.881**
7	9.858	5.387	10.907	10.317	50.507	20.184
8	4.455	5.859	14.273	8.548	43.510	52.843
9	3.377	5.145	5.453	**4.440**	29.617	40.399
10	**2.323**	10.192	10.050	15.463	25.540	17.665
Mean	5.523	6.604	10.122	10.938	36.937	27.785
Std Dev	2.948	2.006	4.474	4.968	13.270	12.276

Table B.3: Simulation results using genetic algorithm (GA).

Bibliography

Aldridge, B. B., Burke, J. M., Lauffenburger, D. A., and Sorger, P. K. (2006). Physicochemical modeling of cell signaling pathways. *Nature Cell Biology*, 8(11):1195–1203.

Alur, R., Belta, C., Ivančić, F., Kumar, V., Mintz, M., Pappas, G. J., Rubin, H., and Schug, J. (2001). Hybrid modeling and simulation of biomolecular networks. In *Proceedings of the 4th International Workshop on Hybrid Systems: Computation and Control*, volume 2034 of *Lecture Notes in Computer Science*, pages 19–32.

Antoniotti, M., Mishra, B., Piazza, C., Policriti, A., and Simeoni, M. (2003). Modeling cellular behavior with hybrid automata: Bisimulation and collapsing. In *Computational Methods in Systems Biology*, volume 2602 of *Lecture Notes in Computer Science*, pages 57–74.

Bansal, M., Belcastro, V., Ambesi-Impiombato, A., and di Bernardo, D. (2007). How to infer gene networks from expression profiles. *Molecular Systems Biology*, 3(78):1–10.

Bellacosa, A., Chan, T. O., Admed, N. N., Datta, K., Malstrom, S., Stokoe, D., McCormick, F., Feng, J., and Tsichlis, P. (1998). Akt activation by growth factors is a multi-step process: the role of the PH domain. *Oncogene*, 17:313–325.

Beyer, H.-G. and Schwefel, H.-P. (2002). Evolution strategies - A comprehensive introduction. *Natural Computing*, 1:3–52.

Bhalla, U. S. and Iyengar, R. (1999). Emergent properties of networks of biological signaling pathways. *Science*, 283:381–387.

Bolouri, H. and Davidson, E. H. (2002). Modeling DNA sequence-based *cis*-regulatory gene networks. *Developmental Biology*, 246:2–13.

Bolt, J. H. and van der Gaag, L. C. (2004). On the convergence error in loopy propagation. *Proceedings of the 16th Belgium-Netherlands Conference on Artificial Intelligence*, pages 267–274.

Bourqui, R., Cottret, L., Lacroix, V., Auber, D., Mary, P., Sagot, M.-F., and Jourdan, F. (2007). Metabolic network visualization eliminating node redundance and preserving metabolic pathways. *BMC Systems Biology*, 1(29).

Briggs, G. E. and Haldane, J. B. S. (1925). A note on the kinetics of enzyme action. *Biochemical Journal*, 19:338–339.

Casagrande, A., Mysore, V., Piazza, C., and Mishra, B. (2005). Independent dynamics hybrid automata in systems biology. *Proceedings of the 1st International Conference on Algebraic Biology*, 10(9):61–73.

Cascante, M., Boros, L. G., Comin-Anduix, B., de Atauri, P., Centelles, J. J., and Lee, P. W. N. (2002). Metabolic control analysis in drug discovery and disease. *Nature Biotechnology*, 20:243–249.

Chaudhary, A., King, W. G., Mattaliano, M. D., Frost, J. A., Diaz, B., Morrison, D. K., Cobb, M. H., Marshall, M. S., and Brugge, J. S. (2000). Phosphatidylinositol 3-kinase regulates Raf1 through Pak phosphorylation of serine 338. *Current Biology*, 10(9):551–554.

Chen, M. and Hofestaedt, R. (2003). Quantitative Petri net model of gene regulated metabolic networks in the cell. *In Silico Biology*, 3(3):347–365.

Chen, R. E. and Thorner, J. (2005). Systems biology approaches in cell signaling research. *Genome Biology*, 6(235).

Christopher, R., Dhiman, A., Fox, J., Gendelman, R., Haberitcher, T., Kagle, D., Spizz, G., Khalil, I. G., and Hill, C. (2004). Data-driven computer simulation of human cancer cell. *Annals of the New York Academy of Sciences*, 1020(1):132–153.

Ciliberto, A., Capuani, F., and Tyson, J. J. (2007). Modeling networks of coupled enzymatic reactions using the total quasi-steady state approximation. *PLoS Computational Biology*, 3(3):463–472.

Davidson, E. H., Rast, J. P., Oliveri, P., Ransick, A., Calestani, C., Yuh, C.-H., Minokawa, T., Amore, G., Hinman, V., Arenas-Mena, C., Otim, O., Brown, C. T., Livi, C. B., Lee, P. Y., Revilla, R., Schilstra, M. J., Clarke, P. J. C., Rust, A. G., Pan, Z., Arnone, M. I., Lee, R., Cameron, R. A., McClay, D. R., Hood, L., and Bolouri, H. (2002). A provisional regulatory gene network for specification of Endomesoderm in the sea urchin embryo. *Developmental Biology*, 246(1):162–190.

Doi, A., Nagasaki, M., Matsuno, H., and Miyano, S. (2006a). Simulation-based validation of the p53 transcriptional activity with Hybrid Functional Petri net. *In Silico Biology*, 6(1-2):1–13.

Doi, A., Nagasaki, M., Ueno, K., Matsuno, H., and Miyano, S. (2006b). A combined pathway to simulate CDK-dependent phosphorylation and ARF-dependent stabilization for p53 transcriptional activity. *Genome Informatics*, 17(1):112–123.

Flach, E. H. and Schnell, S. (2006). Use and abuse of the quasi-steady-state approximation. *IEE Proceedings - Systems Biology*, 153(4):187–191.

Friedman, N. (2004). Inferring cellular networks using probabilistic graphical models. *Science*, 303:799–805.

Friedman, N., Linial, M., Nachman, I., and Pe'er, D. (2000). Using bayesian networks to analyze expression data. *Journal of Computational Biology*, 7(3-4):601–620.

Gadkar, K. G., Gunawan, R., and III, F. J. D. (2005). Iterative approach to model identification of biological networks. *BMC Bioinformatics*, 6:155.

Gardner, T. S. and Faith, J. J. (2005). Reverse-engineering transcription control networks. *Physics of Life Reviews*, 2:65–88.

Gat-Viks, I. and Shamir, R. (2007). Refinement and expansion of signaling pathways: The osmotic response network in yeast. *Genome Research*, 17(3):358–367.

Gat-Viks, I., Tanay, A., Raijman, D., and Shamir, R. (2005). The factor graph network model for biological systems. In *Proceedings of the 9th International Conference on Research in Computational Molecular Biology*, volume 3500 of *Lecture Notes in Computer Science*, pages 31–47.

Ghosh, R. and Tomlin, C. (2004). Symbolic reachable set computation of piecewise affine hybrid automata and its application to biological modelling: Delta-Notch protein signalling. *Systems Biology*, 1(1):170–183.

Gill, P. E., Murray, W., and Wright, M. H. (1998). *Practical optimization*. Academic Press.

Gilmore, T. D. (2006). Introduction to NF-κB: players, pathways, perspectives. *Oncogene*, 25:6680–6684.

Gonzalez, O. R., Küper, C., Jung, K., Jr, P. C. N., and Mendoza, E. (2006). Parameter estimation using simulated annealing for S-System models of biochemical networks. *Bioinformatics*, 23(4):480–486.

Heiner, M., Koch, I., and Will, J. (2004). Model validation of biological pathways using Petri nets - demonstrated for apoptosis. *BioSystems*, 75:15–28.

Heldin, C.-H. (2001). Signal transduction: multiple pathways, multiple options for therapy. *Stem Cells*, 19(4):295–303.

Hindmarsh, A. C. (1983). ODEPACK, a systematized collection of ODE solvers. *Scientific Computing*, pages 55–64.

Holme, P., Huss, M., and Jeong, H. (2003). Subnetwork hierarchies of biochemical pathways. *Bioinformatics*, 19(4):532–538.

Hoops, S., Sahle, S., Gauges, R., Lee, C., Pahle, J., Simus, N., Singhal, M., Xu, L., Mendes, P., and Kummer, U. (2006). COPASI - a COmplex PAthway SImulator. *Bioinformatics*, 22(24):3067–3074.

Ihekwaba, A. E. C., Broomhead, D. S., Grimley, R. L., Benson, N., and Kell, D. B. (2004). Sensitivity analysis of parameters controlling oscillatory signaling in the NF-κB pathway: the roles of IKK and IκBα. *Systems Biology*, 1(1):93–103.

Jacobson, M. and McCarthy, N. (2002). *Apoptosis: The molecular biology of programmed cell death*. Oxford University Press, 1st edition.

Kauffman, K. J., Prakash, P., and Edwards, J. S. (2003). Advances in flux balance analysis. *Current Opinion in Biotechnology*, 14:491–496.

Kel, A., Voss, N., Jauregui, R., Kel-Margoulis, O., and Wingender, E. (2005). Beyond microarrays: Finding key transcription factors controlling signal transduction pathways. *BMC Bioinformatics*, 7(Supp 2):S13.

Keller, G. (2005). Embryonic stem cell differentiation: emergence of a new era in biology and medicine. *Genes and Development*, 19:1129–1155.

Khwaja, A. (1999). Akt is more than just a Bad kinase. *Nature*, 401:33–34.

Kim, S. Y., Imoto, S., and Miyano, S. (2003). Inferring gene networks from time series microarray data using dynamic bayesian networks. *Briefings in Bioinformatics*, 4(3):228–235.

Kitano, H. (2002). Computational systems biology. *Nature*, 420:206–210.

Klamt, S. and Stelling, J. (2003). Two approaches for metabolic pathway analysis? *Trends in Biotechnology*, 21(2):64–69.

Klipp, E., Herwig, R., Kowald, A., Wierling, C., and Lehrach, H. (2005). *Systems biology in practice: concepts, implementation and application*. Wiley-Vch, 1st edition.

Klipp, E. and Liebermeister, W. (2006). Mathematical modeling of intracellular signaling pathways. *BMC Neuroscience*, 7(Suppl 1):S10.

Koh, G., Teong, H. F. C., Clément, M.-V., Hsu, D., and Thiagarajan, P. S. (2006). A decompositional approach to parameter estimation in pathway modeling: a case study of the Akt and MAPK pathways and their crosstalk. *Bioinformatics*, 22(14):e271–e280.

Krieger, C. J., Zhang, P., Mueller, L. A., Wang, A., Paley, S., Arnaud, M., Pick, J., Rhee, S. Y., and Karp, P. D. (2004). MetaCyc: a multiorganism database of metabolic pathways and enzymes. *Nucleic Acids Research*, 32:D438–D442.

Kschischang, F. R., Frey, B. J., and Loeliger, H.-A. (2001). Factor graphs and the sum-product algorithm. *IEEE Transactions on Information Theory*, 47(2):498–519.

Kwon, Y.-K. and Cho, K.-H. (2007). Boolean dynamics of biological networks with multiple coupled feedback loops. *Biophysical Journal*, 92:2975–2981.

Lauffenburger, D. A. (2000). Cell signaling pathways as control modules: Complexity for simplicity? *PNAS*, 97(10):5031–5033.

Lee, D.-Y., Zimmer, R., Lee, S. Y., and Park, S. (2006). Colored Petri net modeling and simulation of signal transduction pathways. *Metabolic Engineering*, 8:112–122.

Lee, E., Salic, A., Küger, R., Heinrich, R., and Kirschner, M. W. (2003). The roles of APC and Axin derived from experimental and theoretical analysis of the Wnt pathway. *PLoS Biology*, 1(1):116–132.

Lee, S. J., Lee, D.-Y., Kim, T. Y., Kim, B. H., Lee, J., and Lee, S. Y. (2005). Metabolic engineering of Escherichia coli for enhanced production of succinic acid, based on genome comparison and in silico gene knockout simulation. *Applied and Environmental Microbiology*, 71(12):7880–7887.

Liang, S., Fuhrman, S., and Somogyi, R. (1998). Reveal, a general reverse engineering algorithm for inference of genetic network architectures. *Pacific Symposium on Biocomputing*, 3:18–29.

Lincoln, P. and Tiwari, A. (2004). Symbolic systems biology: hybrid modeling and analysis of biological networks. In *Hybrid Systems: Computation and Control*, volume 2993 of *Lecture Notes in Computer Science*, pages 660–672.

Mahdavi, A., Davey, R. E., Bhola, P., Yin, T., and Zandstra, P. W. (2007). Sensitivity analysis of intracellular signaling pathway kinetics predicts targets for stem cell fate control. *PLoS Computational Biology*, 3(7):1257–1267.

Margolin, A. A., Nemenman, I., Basso, K., Wiggins, C., Stolovitzky, G., Favera, R. D., and Califano, A. (2006). ARACNe: An algorithm for the reconstruction of gene regulatory networks in a mammalian cellular context. *BMC Bioinformatics*, 7(Supp 1).

Materi, W. and Wishart, D. S. (2007). Computational systems biology in drug discovery and development: Methods and application. *Drug Discovery Today*, 12(7/8):295–303.

Matsuno, H., Li, C., and Miyano, S. (2006). Petri net based descriptions for systematic understanding of biological pathways. *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences*, E89-A(11):3166–3174.

Matsuno, H., Tanaka, Y., Aoshima, H., Doi, A., Matsui, M., and Miyano, S. (2003). Biopathways representation and simulation on hybrid functional Petri net. *In Silico Biology*, 3(3):389–404.

Mehta, K. (2007). Personal communication.

Mendes, P. and Kell, D. B. (1998). Non-linear optimization of biochemical pathways: applications to metabolic engineering and parameter estimation. *Bioinformatics*, 14(10):869–883.

Miller, L. D., Park, K. S., Guo, Q. M., Alkharouf, N. W., Malek, R. L., Lee, N. H., Liu, E. T., and Cheng, S.-Y. (2001). Silencing of Wnt signaling and activation of multiple metabolic pathways in response to Thyroid hormone-stimulated cell proliferation. *Molecular and Cellular Biology*, 21(19):6626–6639.

Minton, A. P. (2001). The influence of macromolecular crowding and macromolecular confinement on biochemical reactions in physiological media. *The Journal of Biological Chemistry*, 276(14):10577–10580.

Mishra, B. and Policriti, A. (2004). Systems biology and automata. In *Proceedings of the 3rd Workshop on Computation of Biochemical Pathways and Genetic Networks*. Springer Verlag.

Moelling, K., Schad, K., Bosse, M., Zimmermann, S., and Schweneker, M. (2002). Regulation of Raf-Akt cross-talk. *Journal of Biological Chemistry*, 277(34):31099–31106.

Moles, C. G., Mendes, P., and Banga, J. R. (2003). Parameter estimation in biochemical pathways: A comparison of global optimization methods. *Genome Research*, 13(11):2467–2474.

Murphy, K. P., Weiss, Y., and Jordan, M. I. (1999). Loopy belief propagation for approximate inference: An empirical study. In *Proceedings of the 15th Annual Conference on Uncertainty in Artificial Intelligence (UAI-99)*, pages 467–47. Morgan Kaufmann.

Nagasaki, M., Yamaguchi, R., Yoshida, R., Imoto, S., Doi, A., Tamada, Y., Matsuno, H., Miyano, S., and Higuchi, T. (2006). Genomic data assimilation for estimating hybrid function Petri net from time-course gene expression data. *Genome Informatics*, 17(1):46–61.

Nayak, L. and De, R. K. (2007). An algorithm for modularization of MAPK and calcium signaling pathways: Comparative analysis among different species. *Journal of Biomedical Informatics*, 40:726–749.

Nicholson, K. M. and Anderson, N. G. (2002). The protein kinase B/Akt signalling pathway in human malignancy. *Cellular Signalling*, 14:381–395.

Nurse, P. (2003). Systems biology: understanding cells. *Nature*, 424:883.

Oda, K., Matsuoka, Y., Funahashi, A., and Kitano, H. (2005). A comprehensive pathway map of epidermal growth factor receptor signaling. *Molecular Systems Biology*, 1:doi:10.1038/msb4100014.

Ogata, H., Goto, S., Sato, K., Fujibuchi, W., Bono, H., and Kanehisa, M. (1999). KEGG: Kyoto encyclopedia of genes and genomes. *Nucleic Acids Research*, 27(1):29–34.

Okino, M. S. and Mavrovouniotis, M. L. (1998). Simplification of mathemical models of chemical reaction systems. *Chemical Reviews*, 98(2):391–408.

Pal, C., Sutton, C., and McCallum, A. (2006). Sparse forward-backward using minimum divergence beams for fast training of conditional random fields. *IEEE International Conference on Acoustic, Speech and Signal Processing*, 5:581–584.

Papin, J. A. and Palsson, B. O. (2004). Topological analysis of mass-balanced signaling networks: a framework to obtain network properties including crosstalk. *Journal of Theoretical Biology*, 227:283–297.

Pearl, J. (1988). *Probabilistic reasoning in intelligent systems: Networks of plausible inference*. Morgan Kaufmann Publishers, 2nd edition.

Petzold, L. (1983). Automatic selection of methods for solving stiff and nonstiff systems of ordinary differential equations. *SIAM Journal on Scientific and Statistical Computing*, 4:136–148.

Pozarowski, P., Holden, E., and Darzynkiewicz, Z. (2006). Laser scanning cytometry: principles and applications. *Methods in Molecular Biology*, 319:165–192.

Raeymaekers, L. (2002). Dynamics of boolean networks controlled by biologically meaningful functions. *Journal of Theoretical Biology*, 218:331–341.

Raman, K., Rajagopalan, P., and Chandra, N. (2005). Flux balance analysis of Mycolic Acid pathway: Targets for anti-tubercular drugs. *PLoS Computational Biology*, 1(5):e46.

Reisig, W. (1992). *A primer in Petri net design*. Springer Verlag.

Runarsson, T. P. and Yao, X. (2000). Stochastic ranking for constrained evolutionary optimization. *IEEE Transactions on Evolutionary Computation*, 4(3):284–294.

Rzhetsky, A., Zheng, T., and Weinreb, C. (2006). Self-Correcting Maps of Molecular Pathways. *PLoS ONE*, 1(1):e61.

Sato, S., Fujita, N., and Tsuruo, T. (2004). Involvement of 3-phosphoinositide-dependent protein kinase-1 in the MEK/MAPK signal transduction pathway. *Journal of Biological Chemistry*, 279(32):33759–33767.

Schaub, M. A., Henzinger, T. A., and Fisher, J. (2007). Qualitative networks: a symbolic approach to analyze biological signaling networks. *BMC Systems Biology*, 1(4).

Schilling, C. H., Schuster, S., Palsson, B. O., and Heinrich, R. (1999). Metabolic pathway analysis: basic concepts and scientific applications in the post-genomic era. *Biotechnology Progress*, 15:296–303.

Schoeberl, B., Eichler-Jonsson, C., Gilles, E. D., and Müller, G. (2002). Computational modeling of the dynamics of the MAP kinase cascade activated by surface and internalized EGF receptors. *Nature Biotechnology*, 20:370–375.

Schuster, S., Pfeiffer, T., Moldenhauer, F., Koch, I., and Dandekar, T. (2002). Exploring the pathway structure of metabolism: decomposition into subnetworks and application to Mycoplasma pneumoniae. *Bioinformatics*, 18(2):351–361.

Segers, V. F. M. and Lee, R. T. (2008). Stem-cell therapy for cardiac disease. *Nature*, 451:937–942.

Shapiro, H. M. (2003). *Practical flow cytometry*. Wiley-Liss, 4th edition.

Shmulevich, I., Dougherty, E. D., and Zhang, W. (2002). From boolean to probabilistic boolean networks as models of genetic regulatory networks. *Proceedings of the IEEE*, 90(11):1778–1792.

Spiro, P. A., Parkinson, J. S., and Othmer, H. G. (1997). A model of excitation and adaptation in bacterial chemotaxis. *PNAS*, 94:7263–7268.

Sudderth, E. B., Ihler, A. T., Freeman, W. T., and Willsky, A. S. (2003). Nonparametric belief propagation. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 1:605–612.

Tasaki, S., Nagasaki, M., Oyama, M., Hata, H., Ueno, K., Yoshida, R., Higuchi, T., Sugano, S., and Miyano, S. (2006). Modeling and estimation of dynamic EGFR pathway by data assimilation approach using time series proteomic data. *Genome Informatics*, 17(2):226–228.

Teong, H. F. C. and Clement, M. V. (2006). Unpublished manuscript.

Teusink, B., Passarge, J., Reijenga, C. A., Esgalhado, E., van der Weijden, C. C., Schepper, M., Walsh, M. C., Bakker, B. M., van Dam, K., Westerhoff, H. V., and Snoep, J. L. (2000). Can yeast glycolysis be understood in terms of in vitro kinetics of the constituent enzymes? Testing biochemistry. *European Journal of Biochemistry*, 267:5313–5329.

Thieffry, D. and Sarkar, S. (1998). Forty years under the central dogma. *Trends in Biochemical Sciences*, 23(8):312–316.

Voit, E. O. (2000). *Computational analysis of biochemical systems - a practical guide for biochemists and molecular biologists*. Cambridge University Press.

Voit, E. O. and Almeida, J. (2004). Decoupling dynamical systems for pathway identification from metabolic profiles. *Bioinformatics*, 20(11):1670–1681.

Voss, K., Heiner, M., and Koch, I. (2003). Steady state analysis of metabolic pathways using Petri nets. *In Silico Biology*, 3(3):367–387.

Weiss, Y. and Freeman, W. T. (1999). Correctness of belief propagation in Gaussian graphical models of arbitrary topology. In *NIPS*, pages 673–679.

Weiss, Y. and Freeman, W. T. (2001). On the optimality of solutions of the max-product belief propagation algorithm in arbitrary graphs. *IEEE Transactions on Information Theory*, 47(2):736–744.

Weng, G., Bhalla, U. S., and Iyengar, R. (1999). Complexity in biological signaling systems. *Science*, 284:92–96.

Wilson, K. and Walker, J. (2005). *Principles and techniques of biochemistry and molecular biology*. Cambridge University Press, 6th edition.

Ye, P., Entcheva, E., Smolka, S. A., True, M. R., and Grosu, R. (2006). Hybrid automata as a unifying framework for modeling excitable cells. *Annual International Conference of the IEEE Engineering in Medicine and Biological Society*, 1:4151–4154.

Yeang, C.-H., Ideker, T., and Jaakkola, T. (2004). Physical network models. *Journal of Computational Biology*, 11(2-3):243–262.

Zevedei-Oancea, I. and Schuster, S. (2003). Topological analysis of metabolic networks based on Petri net theory. *In Silico Biology*, 3(3):323–345.

Zwolak, J. W., Tyson, J. J., and Watson, L. T. (2005). Parameter estimation for a mathematical model of the cell cycle in frog eggs. *Journal of Computational Biology*, 12(1):48–63.