PROBABILISTIC VERIFICATION AND ANALYSIS OF BIOPATHWAY DYNAMICS

SUCHEENDRA KUMAR PALANIAPPAN

NATIONAL UNIVERSITY OF SINGAPORE

2013

PROBABILISTIC VERIFICATION AND ANALYSIS OF BIOPATHWAY DYNAMICS

SUCHEENDRA KUMAR PALANIAPPAN

(*B.Eng, PESIT, India*)

A THESIS SUBMITTED
FOR THE DEGREE OF DOCTOR OF PHILOSOPHY IN COMPUTER SCIENCE
SCHOOL OF COMPUTING
NATIONAL UNIVERSITY OF SINGAPORE
2013

Declaration

I hereby declare that this thesis is my original work and it has been written by me in its entirety. I have duly acknowledged all the sources of information which have been used in the thesis.

This thesis has also not been submitted for any degree in any university previously.

Sucheendra kumar
Palaniappan
March 20, 2013

d

Acknowledgement

When I look back at the past few years of my doctoral studies, it has been nothing short of a roller coaster ride. I have seen my share of ups and downs, and they have all added to make the journey very memorable and enjoyable. In the process I have had a chance to meet, interact and work with a number of people who have and will continue to inspire me. I only wish I can be -atleast- in part, as awe-inspiring as them.

My deepest and most sincere gratitude goes out to Professor P. S. Thiagarajan. I have enjoyed his mentorship, advice and support at every stage of my PhD. I appreciate his patience, especially during the days when it was hard for me to get used to the pace of research. I truly admire his wisdom and enthusiasm for research, he will be someone I will always look up to where ever I go. I thank him for his continued financial support even after my scholarship expired.

Next, I would like to thank Dr.Blaise Genest, who has also been a constant source of guidance, advice and support. He is extremely friendly and someone who can be approached easily. Most of all, his passion for good research is contagious. I hope that I will get to meet and work with more people like him in the future. I would also like to convey my special thanks Dr.Akshay Sundaraman, he has been a good friend and mentor; I have learned a lot from. I thank Dr.Liu Bing for his support throughout my candidature.

I would like to thank Professor Ding Jeak Ling and her student Liu Qian Shania from the department of biological sciences for the collaboration, which contributed to a part of this thesis. I would like to thank Associate Professor David Hsu and Associate Professor Dong Jin Song for their valuable suggestions during my thesis proposal.

I would also extend my heartfelt thanks to Professor Limsoon Wong and Associate Professor Sung Wing Kin. I was fortunate to interact with Professor Wong during one of our projects, his diligence and quick response times never fail to amaze me. Professor Sung Wing Kin is also someone I look up to, he is there in the lab almost every day, discussing research problems and constantly mentoring his students in a very informal setting. I hope I can be like him once I step onto higher levels of my career.

In addition to these people who have played a crucial role in my journey, there have been numerous friends whom I met along the way. As they say “friendship doubles our joy and divides our grief”, I hope our friendships can go a long way. At the lab, among the former members, my special thanks go out to Joshua, Dr.Chiang and Dr.Sriganesh Srihari; they are pretty amazing. Thanks to Benjamin and Ah Fu for the fruitful collaboration, it was a breeze working with you guys. Special thanks to Wang Yue, I have learned a lot from him. Thanks to Jing Quan, he has been a great friend. Thanks to Chandana and Peiyong for showing what work life balance is. Special thanks to Michal, Ali, Javad, Hoang, Zhizhou, Kevin and Chern Han for all the great times. Many thanks to Haojun and Hufeng. I would like wish new members in the lab, Ramanathan, Ratul, Narmada and Charlie the best in whatever they do.

Outside lab, in school of computing, I have made great friends. First, I would like to thank Sudipta for being a good friend and exemplifying what a good researcher should be. He will continue to inspire me. Thanks to Manoranjan, Abhinav Dubey, Rajarshi, Manjunath, Satish, Prabhu, Bodhi, Sumanan, Malai, Padmanabha for being there. Special thanks to all other friends at school of computing.

Special thanks to Ramesh, Soneela, Aravind, Vamsi, Pradeep, Deepak, Souvik, Amit, Sujith. You have all been great support. Last, I would like to thank my family for being so patient and understanding. I realize that I may not have recalled all the people I owe my heartfelt thanks to. To everyone else whom I have forgotten due to my bad memory, my apologies; I thank you all.

Contents

Summary

Understanding the mechanisms by which biological processes function and regulate each other is crucial. Often, one studies these biological processes as a network of biomolecules interacting with each other through biochemical reactions. The dynamics of interaction among the various biomolecules determines the cellular functions and behavior. Hence, modeling and analyzing the dynamics of biochemical networks is crucial to the understanding of biological processes. *Computational Systems Biology* deals with the systematic application of computational methods to model and analyze such biochemical networks, which are often called biopathways.

Two main paradigms exist for modeling biopathways, the deterministic and the stochastic. In the deterministic approach ordinary differential equations (ODEs) are commonly used while in the stochastic approaches, Markov chains are common. Our focus is mainly on models that arise in stochastic settings. Our goal in the thesis is to use a formal verification technique called *probabilistic model checking* to verify and analyze the dynamics of stochastic models.

Model checking refers to the broad class of techniques to automatically evaluate if a system satisfies properties expressed as temporal logic formulas. Probabilistic model checking (PMC) deals with analysis and validation of systems which exhibit stochastic behavior. In the context of biological pathways, explicitly dealing with Markov chains is often infeasible due to the state space explosion problem. The results reported in [1, 2] shows that a probabilistic graphical model called dynamic Bayesian network (DBN) can be a more natural and succinct model to work with.

Consequently, our work concerns the analysis of DBN models of biopathways from a model checking point of view. Specifically, we first consider the problem of probabilistic model checking on DBNs based on probabilistic inference. However, exact inference is hard for large DBNs. To get around this, in the first part of the thesis, we present a new improved approximate inference method for DBNs called hybrid factored frontier. We then formulate, for DBNs, a new probabilistic temporal logic called bounded linear time probabilistic logic. We develop an –approximate– model checking framework based on

DBN inference algorithms. We then verify interesting dynamical properties of biological systems.

The second part of this thesis focuses on using another scalable probabilistic model checking approach called *statistical model checking* for calibration and analysis of ODE based models. The uncertainty concerning the initial states is modeled via a prior distribution over an interval of values. The noisiness and the cell-population-based nature of the experimental data are captured by the confidence level and strength of the statistical test. The experimental data as well as qualitative properties of the pathway are encoded as the specification formula in a temporal logic formalism. In this setting, we use optimized versions of statistical model checking algorithms for the task of parameter estimation. Specifically, we build a statistical model checking based parameter estimation framework by coupling it with standard global optimization techniques. Our results suggests that this framework is efficient, useful and scales well.

Finally, we apply our statistical model checking framework to build and calibrate an ODE model for the Toll like receptor (TLR) 3 and TLR7 pathways. We investigate specific crosstalk mechanisms which lead to synergy when the TLR3 and TLR7 receptors are stimulated together in a specific order and a specific time gap. Our analysis leads to interesting insights regarding the potential crosstalk mechanism.

List of Tables

List of Figures

Chapter 1

Introduction

Understanding “Life” has been a major scientific quest for mankind. Central to this quest is the study of basic unit of life, namely, the cell. The molecular composition of parts of a cell and how they function has been the fundamental question that biologists have been trying to answer over the past century. From DNA to RNAs, proteins etc., we now understand their chemical structure, basic functions and to a certain extent the mechanisms driving the key developmental and regulatory processes of life.

This has been possible, thanks to the rapid advancements in experimental technologies. A fitting example of the success of experimental biology is the human genome project. In the near future, one can get a human genome sequenced in a day for as little as US$1000 [3]. Similar technological advancements in other fronts are on the way. These technologies are producing vast amounts of data.

With all this data pouring in, we now have a good static picture of the different components and compositions of a cell along with their essential functions as documented in databases such as Gene ontology [4], BRENDA [5], PDB [6], Swiss-Prot [7], UniProt [8], TRANSFAC [9]. It is now crucial to study and understand the dynamic behavior of these components since they interact in complex yet coherent ways to perform biological functions. To achieve this, system level approaches to understanding biological systems is a basic requirement.

Henri Poincarè said , “The aim of science is not things themselves, as the dogmatists in their simplicity imagine, but the relations among things; outside these relations there is no reality knowable”. This captures the approach to be taken if new strides are to be made in our understanding of biological systems. For instance, it is well known that cancer is a complex disease, typically characterized by uncontrolled cellular growth. However, the mechanisms which decide the fate of a normal cells to become cancerous are so varied, complex, coordinated and systemic that studying components in isolation is unlikely to lead to an effective treatment [10]. Almost every human disease and biological process reflects this kind of systemic nature. The field of *Systems biology* stems from this need to understand biological processes as holistic dynamical systems. Its goal is to understand and analyze the behavior and interrelationships among functional biological systems [11].

Studying systems of such complexity requires a multidisciplinary approach. The field of *Computational Systems Biology* represents such efforts. It is at the intersection of computer science, engineering, mathematics, physics and biology. It primarily deals with building executable qualitative and quantitative mathematical models. It is concerned with developing efficient data structures, algorithms and formalisms for analyzing and visualizing the dynamics of biological processes[11]. These models, in addition to providing an understanding of the underlying mechanisms, can be used to predict system behavior under different conditions or perturbations. They can assist in designing better experiments. They also help by highlighting the gaps we have in our understanding. Furthermore, they can serve as repositories of our current knowledge of these systems. It is in this context the research in this thesis has been carried out.

1.1 Overview of the thesis

Biological processes are driven by networks of biochemical reactions. These networks are often termed biopathways. Different mathematical formulations have been used to model these pathways; biopathways are modeled and studied either as deterministic systems (such as ordinary differential equations (ODEs)) or stochastic systems (such as Markov chains). Our focus in this thesis will be on the class of models which arise in stochastic settings. In biological systems, stochasticity appears in different ways. Randomness, noise and uncertainty are central players in biological processes. Traditionally, in classical biology, these aspects were considered to be a nuisance. However, increasingly these aspects are considered important. In addition, experimental procedures are marred by limitations in technologies available for accurate observation and measurement of biomolecules. Hence, incorporating these aspects into modeling is crucial. For modeling stochastic biological processes, discrete time Markov chains (DTMC) and continuous time Markov chains (CTMC) serve as the core mathematical formalism. Two main issues exist in using these classes of models. First, in the context of systems biology models, the state space associated with these models is extremely large. Explicit representation of these systems is cumbersome and sometimes even impossible. In this context, the probabilistic graphical model called dynamic Bayesian networks (DBNs) offers attractive alternatives to succinctly represent pathway dynamics since they capture the probabilistic dynamics locally. In this thesis, one of our main focus will be DBNs.

The DBNs in our setting arise as approximations of the dynamics induced by a system of deterministic ordinary differential equations (ODE) which describe the signaling events of biochemical networks. The technique was developed in [12]. This approximation is derived by discretizing both the time and value domains, sampling the assumed set of initial states and using numerical integration to generate a large number of representative trajectories. Then based on the network structure and simple counting, the generated trajectories are stored compactly as a DBN. One can then analyze the biochemical network using the DBN. This approach scales well and has been used to aid biological studies [12, 1].

Formal verification, deals with the broad class of methods which deal with using mathematically rigorous techniques to prove or disprove that the system is "correct" with respect to intended properties specified in a formal language. Formal verification techniques chiefly comprise *Model checking* and *deductive verification*. They have been traditionally used in the context of hardware circuits, embedded and software systems which are safety critical [13]. Techniques from the domain of formal verification can be applied for automated analysis tasks in the context of biopathway models and hence provide a promising way to deal with model analysis. This thesis focuses on using a formal verification technique called probabilistic model checking (PMC) for analyzing the dynamics of stochastic biopathway models. The intended properties are specified in probabilistic temporal logics. The probabilistic model checker traverses the state space to quantitatively check if the stochastic model conforms to the properties.

Solving the PMC problem amounts to traversing the state space of the stochastic model, and computing the probability of the property to hold and comparing it with the threshold probability dictated by the temporal logic formula. Exact methods have a high time complexity and are suitable only for relatively small systems. In biological settings, the size of models is considerably larger than those that can be gracefully handled by exact methods. Hence, approximate methods for solving the problem need to be used. Our contributions in this thesis are towards this end.

As a key contribution of this thesis, we first consider the problem of probabilistic model checking on DBNs. Probabilistic model checking on DBNs is based on probabilistic inference. Exact probabilistic inference is infeasible for large DBNs, hence approximate algorithms are used. We present a major improvement to an existing inference algorithm called the factored frontier algorithm (FF). Next, we present a new probabilistic temporal logic and develop an approximate probabilistic model checking framework for DBNs. Both FF and our improved version of FF called hybrid factored frontier (HFF) play a crucial role in the solution of the associated model checking procedure.

A second class of approximate algorithms, called *Statistical model checking* works by sampling a set of simulation traces from the model. Each simulation trace is evaluated to determine if it satisfies the property, and the number of traces which satisfy the property are used to decide the solution of the PMC problem. These algorithms offer a promising approach to scale the applicability of PMC to large stochastic models. As a second major contribution of the thesis we present a statistical model checking based calibration framework for ODE models.

Finally, we apply our framework to construct and analyze a new ODE model for toll like receptor (TLR)3 and TLR7 signal transduction which play a crucial role in innate immune response. We use our statistical model checking framework to investigate cross talk mechanisms between these two pathways, which lead to synergistic immune response.

We now turn to a more detailed presentation of our contribution.

1.2 Research Contributions

1.2.1 Probabilistic model checking on DBNs

Markov chains of various kinds serve as the core mathematical formalism for modeling stochastic biological processes. However, in many of these settings, the probabilistic graphical model called dynamic Bayesian networks (DBNs) [14] can be a more appropriate model to work with. This is so since a DBN offers a factored and succinct representation of an underlying Markov chain. Here we look at DBNs from this standpoint.

Probabilistic inference on DBNs

A DBN has a finite set of random variables with each variable having a finite domain of values. The value of a variable at time t only depends on the values of its parents at time $t-1$. The probabilistic dynamics is captured by a Conditional Probability Table (CPT) associated with each variable at each time point. This table will specify how the value of a variable at t is conditioned by the values of its parent variables at time $t-1$. The global state of the system at time t is a tuple of values with each component denoting the value assumed by the corresponding variable at time t.

To analyze DBNs, one is interested in computing the marginal probability, i.e., the probability of a variable X taking value v at time t. To compute this exactly, we need to compute the joint probability distribution over global states at time t. This can be computed by propagating the joint distribution at time $t-1$ through the CPTs. Doing it exactly is infeasible for large DBNs [15]. Hence, approximate inference algorithms such as factored frontier (FF) algorithm [16] are used. Since the inference algorithm is approximate, it introduces errors in computing the probability distributions. To reduce these errors, we propose an improved inference algorithm, termed hybrid factored frontier (HFF) which is a parameterized extension of FF algorithm. The parameter acts as an tunable control between accuracy and effort. We show that HFF is a scalable and efficient algorithm in our setting with reduced errors. We also perform an error analysis of the HFF algorithm. Finally, we present experimental results using large DBN models to validate the improvements achieved by the HFF algorithm.

Probabilistic model checking based on probabilistic inference

We then formulate, for DBNs, a new probabilistic temporal logic called – bounded linear time probabilistic logic (BLTPL) – which allows us to express dynamic properties in terms of probability distributions. BLTPL can be considered as a probabilistic variant of Linear Time Temporal Logic (LTL) in which the atomic propositions represent marginal probabilities and are of the form $(X, v) \leq c$ or $(X, v) \geq c$ where X is a random variable corresponding to a node in the DBN, and c is a rational number in $[0, 1]$. The assertion $(X, v) \leq c$ says that the probability of the random variable X currently assuming the value v is less than c; similarly for the assertion $(X, v) \geq c$. The remaining operators of the logic are handled in the usual way. Semantically, BLTPL is similar to bounded LTL [13] in the sense the logic is interpreted over only a finite set of time points. In our logic, probability enters the picture only via atomic propositions. However, one can still express many interesting dynamical properties.

Next, we develop an approximate model checking framework based on the probabilistic inference algorithms on DBNs. We then use the developed algorithms to verify interesting dynamical properties of biological systems.

1.2.2 Statistical model checking based calibration of ODE models

Statistical model checking, as discussed before, relies on drawing repeated traces of the underlying stochastic system to statistically assert if a property holds. In the context of biological models, these algorithms can be improved for efficiency and can be suitably adapted to perform tasks such as model calibration of pathway models.

First, we show how statistical model checking can be used for analyzing ODE systems. We assume that the initial concentrations of the various species take their values according to a distribution (usually uniform) over a set of initial states, this is to account for the substantial cell-to-cell variability in the initial states[17]. In such a setting the vector fields defined by the ODE system will be a C^1 (continuously differentiable) function and hence one can assign a probability measure to the set of simulation traces that satisfy a dynamical property expressed as a bounded linear time temporal logic[18] formula.

Drawing simulation traces is an expensive task. Optimizing the generation and verification of these traces and using these algorithms for performing novel applications such as parameter estimation is important. We use an on-the-fly approach to perform statistical model checking where generation of the trace and model checking are performed together. Next, we formulate a statistical model checking based framework for parameter estimation of biopathway models. Specifically, we couple our statistical model checking algorithm with standard global optimization techniques to calibrate and analyze these systems. This approach has several advantages. First, both quantitative and qualitative knowledge (which can come from the literature or general observations about the system) can be utilized to calibrate the model. This is in contrast to traditional methods of pathway calibration which use only quantitative experimental time series data. The uncertainty concerning the initial states is modeled via a prior distribution over an interval of values that a variable can assume initially. The noisiness and the cell-population-based nature of the experimental data are captured by the confidence level and strength of the statistical test. It is a generic approach and can be applied in different model formalisms. Our results reported in chapter 7 and 8 suggest that our statistical model checking based framework is efficient, useful, and scales well.

Modeling and analysis of Toll like receptor pathway

We apply our calibration framework based on statistical model checking to model and analyze the signaling cascades involved in toll like receptor (TLR) pathways. These receptors are crucial players in innate immunity. They are among the key players driving immune system and are usually the first line of defense against external attacks (such as bacteria or viruses). Specifically, we construct an ODE based model of the TLR3 and TLR7 pathways and investigate potential cross talk mechanisms which lead to marked synergistic activation of immune response when these receptors are activated in a specific order and with a specific time gap. We use our statistical model checking based parameter estimation framework to estimate unknown parameters of the pathway. Next, we hypothesize and investigate three potential crosstalk mechanisms. Our initial analysis suggests that the cross talk mediated by the production of Type I interferons is the most promising candidate.

1.3 Outline of the thesis

The rest of this thesis is organized as follows.

In Chapter 2, we briefly discuss background material on modeling biological pathways, common techniques involved in pathway construction and analysis such as parameter estimation, sensitivity analysis and model checking.

Chapter 3 discusses Markov chains and dynamic Bayesian networks. This chapter also discusses how DBNs arise as approximate representations of bio pathway dynamics induced by a system of ODEs. They will serve as the main source of DBNs for all our case studies. However, the methods we develop in this thesis are applicable to DBNs in general.

Chapter 4 describes probabilistic inference on DBNs, and specifically discusses our improved inference method called hybrid factored frontier (HFF) algorithm.

Chapter 5 describes the basics of model checking, probabilistic model checking and discusses related work on the use of model checking in computational systems biology.

Chapter 6 presents our probabilistic temporal logic called bounded linear time probabilistic logic (BLTPL) and the probabilistic model checking framework based on the approximate inference algorithms for DBNs.

Chapter 7 discusses our work on using statistical model checking for parameter estimation of models that arise in the context of ODEs.

Chapter 8 discusses the application of our statistical model checking framework for modeling the toll like receptor pathway. We present our model for the TLR3 and TLR7 pathway, and hypothesize possible crosstalk mechanisms. We discuss some of our findings and the biological insights gained so far in the process.

Finally, Chapter 9 summarizes our main contributions in this thesis. We discuss the significance of the obtained results and also identify directions for future research.

1.4 Declaration

Major portions of this thesis are based on the following papers:

1. Sucheendra K. Palaniappan, S. Akshay, Blaise Genest, and P. S. Thiagarajan. *A hybrid factored frontier algorithm for dynamic Bayesian network models of biopathways.* In proceedings of the ninth international conference on computational methods in systems biology (CMSB), pages 35–44, New York, USA, 2011. ACM.

2. Sucheendra K Palaniappan, S Akshay, Bing Liu, Blaise Genest, and P S Thiagarajan. *A hybrid factored frontier algorithm for dynamic Bayesian networks with a biopathways application (expanded and improved version of the ninth international conference on computational methods in systems biology paper).* IEEE/ACM transactions on computational biology and bioinformatics / IEEE, ACM, 9(5):1352-1365, October 2012. PMID: 22529330.

3. Sucheendra K. Palaniappan and P. S. Thiagarajan. *Dynamic Bayesian networks: A factored model of probabilistic dynamics.* In Supratik Chakraborty and Madhavan Mukund, editors, automated technology for verification and analysis (ATVA), volume 7561 of Lecture Notes in Computer Science, pages 17–25. Springer, 2012.

4. Bing Liu, Andrei Hagiescu, Sucheendra K. Palaniappan, Bipasa Chattopadhyay, Zheng Cui, Weng-Fai Wong, and P. S. Thiagarajan. *Approximate probabilistic analysis of biopathway dynamics.* Bioinformatics, 28(11):1508–1516, June 2012.

5. Sucheendra K. Palaniappan, Benjamin M. Gyori, Bing Liu, David Hsu, P. S. Thiagarajan and Edmund M. Clarke. *Parameter estimation of biopathway models using statistical model checking.* Submitted.

6. Chuan H. Koh, Sucheendra K. Palaniappan, P. S. Thiagarajan, and Limsoon-Wong. *Improved statistical model checking methods for pathway analysis.* BMC Bioinformatics, 13(Suppl 17):S15, proceedings of 11th International conference on bioinformatics. Dec 2012.

Chapter 2

Preliminaries

Biological systems are composed of biomolecules whose complex yet coordinated actions leads to the numerous biological functions. We wish to reason about how these molecules work together at the systemic level to perform various biological functions. To systematically record and understand these interactions we construct models of biopathways.

In this chapter, we will briefly discuss biopathway modeling. First, we describe the main paradigms of modeling biopathways. Next, we discuss the typical modeling life cycle with emphasis on tasks such as model construction, model calibration, validation and analysis.

Biopathways can be broadly classified based on the biological functions they perform. *Gene regulatory networks* describe the regulatory interaction between genes in a cell. *Metabolic networks* describe chemical reactions involved in the production or breakdown of different metabolites which lead to energy production and storage in the cell. *Signaling pathways* describe reactions that occur with in a cell in response to external or internal stimuli. In the case of signaling pathways, the signal from the stimuli is carried by a cascade of proteins to the effector molecules which accordingly change the state of the cell. Our focus in this thesis will be on signaling pathways and their associated dynamics, although the methods developed through the thesis can be applied to other settings as well.

2.1 Biopathway modeling

A variety of mathematical models have been proposed for modeling signaling pathways. These models vary from being purely qualitative [19, 20, 21] to quantitative [22, 23] models. Model formulation can be purely deterministic, stochastic or a combination of both[24]. The choice of the modeling framework depends on the biological systems under study, the kind of experimental data available and the specific biological insights we hope to gain from the modeling exercise. The main formalisms for mathematical modeling include ODEs [25], partial differential equations (PDEs)[26], Boolean networks [27], Petri nets [28, 29], rule-based languages [30], process algebra [31, 32] etc.

2.1.1 Deterministic models

The most common paradigm for modeling biological systems are deterministic models based on Ordinary differential equations (ODEs). Given an initial state of the system its future states are uniquely determined by the underlying kinetics.

ODEs capture the concentration changes of different species through the reactions they take part in. The concentration of every molecular species is assumed to be continuous valued and its change over time is governed by a differential equation. The formulation is guided by the kinetic laws that govern each reaction [25]. Let us consider a pathway, comprising of a network of N species. We let each species be represented by X_i, $i \in [1 \ldots N]$. Let these N species, overall, participate in R reactions. Each reaction has an identifier Y_j, $j \in [1 \ldots R]$. Next, assuming that the reaction is confined in a constant volume V, let $n_{X_i}(t)$ denote the number of particles of species X_i at time t. We refer by $[X_i](t)$, the concentration of X_i at time t given by $n_{X_i}(t)/V$. With each reaction Y_j, we also associate a kinetic function f_j which represents the velocity of the reaction. Mass action kinetics is the simplest and most commonly used commonly used kinetic function. In this case the velocity of the reaction is proportional to the product of the reactant concentrations to the power of their corresponding molecularities. For instance, consider a reaction network consisting of five species as follows:

$$
\begin{aligned}
Y_1 &: A + 2B \rightarrow C \\
Y_2 &: C + D \rightarrow E
\end{aligned}
\tag{2.1}
$$

Here A and B are reactants, C denotes the formed product of reaction Y_1 which in turn interacts with D to form the final product E, f_1 and f_2 in this case will be $k1 * [A] * [B]^2$ and $k2 * [C] * [D]$ respectively. The quantity k_1, k_2 are called kinetic rate constants.

In some scenarios, several reactions may be lumped or assumptions about the relative speed or concentrations of the different species are made. This leads to more complex kinetic functions such as Michaelis Menten, ping-pong mechanisms or Hill reaction [33, 34] etc.

The set of coupled ODEs for the system consists of one equation for each of the variable X_i of the form

$$\frac{d[X_i]}{dt} = \sum_{j=1}^{R} (p_{ij} * f_j) \tag{2.2}$$

where $p_{ij} = 0$ if X_i does not participate in reaction Y_j, $p_{ij} = 1$ if X_i is a product in the reaction Y_j and $p_{ij} = -1$ if X_i is a reactant in the reaction. In the small example considered before, the corresponding system of ODEs will be:

$$
\begin{aligned}
\frac{d[A]}{dt} &= -k_1 \cdot [A] \cdot [B]^2 \\
\frac{d[B]}{dt} &= -k_1 \cdot [A] \cdot [B]^2 \\
\frac{d[C]}{dt} &= k_1 \cdot [A] \cdot [B]^2 - k_2 \cdot [C] \cdot [D] \\
\frac{d[D]}{dt} &= -k_2 \cdot [C] \cdot [D] \\
\frac{d[E]}{dt} &= k_2 \cdot [C] \cdot [D]
\end{aligned}
$$

Given a well-defined system of ODEs as discussed above, the initial values of the N species, the kinetic rate constants and suitable continuity assumptions, the solution to the system of ODEs will have a unique solution [35]. Hence, in principle, ODE based models can be used to get the temporal time profile of the system behavior by solving the corresponding system of ODEs. However, ODE systems which describe biopathway dynamics are usually high-dimensional and nonlinear and hence do not admit closed form solutions. Consequently, one must rely on numerical integration methods such as the Euler method, Runge-Kutta method[36] etc., to get approximate solutions. In addition, differential equations corresponding to biopathways are $stiff$ [37], i.e., the variables of the system of ODEs change at widely different scales. In such cases one has to use specialized stiff ODE solvers such as LSODA[38] , CVODE[39], ODEPACK[40] , ODEINT[41].

Formulating and solving ODEs, requires one to have a detailed knowledge about the mechanisms of the reactions, the value of rate constants etc. However, much of this information including many rate constant values will be unknown. Hence, restricted classes of ODEs which are derived from original ODEs by making several simplifying assumptions are often used. Examples include the $peicewise-multiaffine$ models which have been used to model gene regulatory models [42, 43]. The main advantage of these include, a simpler mathematical formalism, analysis even under parameter uncertainty, and in many cases the qualitative properties of solution are as good as ODEs[44, 45]. Another class of simplification of the original ODE formulation are the class of qualitative differential equations (QDE), used when quantitative knowledge about the system is limited. It has been used for qualitative reasoning in gene regulation studies[46, 47, 48].

2.1.2 Stochastic models

Deterministic approaches such as ODEs are applicable only when the number molecules of the different components are sufficiently high and that they are a part of a well-mixed solution. They ignore sources of noise which are inherent to biological systems.

Stochasticity manifests in biological system due to low concentration (particle numbers) of various species within a cell. Biomolecules which participate in processes such as transcription, translation, regulation of transcription etc., are in low copy numbers and hence small fluctuations can produce significant changes in the dynamics [49]. The concentration, localization, intrinsic state of these molecules also has an impact on the fate of the consequent processes they trigger [50]. In addition, cell-to-cell variability can occur due to random microscopic events in the cell which decide which reactions to occur and in what order [50].

Another consideration is that experimental procedures usually measure cell population data, each cell in the population may be in a slightly different state with respect to the concentration of different components, the onset of reactions, the surrounding micro environment in the cell etc. Modeling methods should factor in these aspects of the experimental data. A good example for this is reported in [17], where differences in the initial concentrations of various proteins regulating apoptosis was attributed to be the main cause of cell-to-cell variability in the timing and probability of cell death, it was shown to be the main reason that only a fraction of tumor cells were killed after exposure to chemotherapy[17].

A popular method for modeling stochastic systems is by the Chemical Master Equation(CME)[51]. The CME is a set of first order differential equations, which describe the time evolution of a well-mixed, homogeneous system in a way that takes into account the fact that number of molecules is known(and suitably low) and exhibit randomness in their dynamical behavior, the time evolution of the system is in terms of discrete stochastic events. The method accounts of the discreteness and stochasticity that is inherent in biological systems. The state of the system is defined as the number of molecules of each species at a particular time point. CME then considers the probability distribution over its possible states and tracks the time evolution of this distribution. Solving the CME is impractical due to resulting blow up in the state space even for relatively small systems. In fact the time evolution of CME can be described by a continuous time Markov chain (CTMC). So, to efficiently simulate the CME, Gillespie proposed the stochastic simulation algorithm (SSA) [51]. This method relies on carrying out large simulations the underlying stochastic system, until the resulting distribution of the state of the system approaches the distribution implied by the CME. This approach is also computationally expensive and many improvements to the original SSA have been proposed [52, 53, 54, 55].

Other formalism for analyzing stochastic models include process algebra based method such as Bio-PEPA[56, 32], Rule based formalisms such as κ[30] etc. Bio-PEPA is an extension of the stochastic process algebra framework PEPA, enhanced to handle biological networks. PEPA was originally used for performance analysis of concurrent systems. Models in Bio-PEPA represent a formal, compositional representation of the biological model. These models can be converted to a CTMC and analyzed numerically. Stochastic simulations such as SSA can also be carried out on these models.

The κ tool[30] uses a rule based modeling framework which views biological molecules as agents. The dynamics of the system is specified by a set of rules, which express the way these agents interact with each other. The set of rules fully specify the system. In fact, the κ model can be interpreted as a large and complex CTMC. Next, one analyzes them using stochastic simulations. The primary advantage of such rule based formalisms is that they overcome the combinatorial explosion in the number of species that arise especially during complex formation, localization of post translational modifications.

The PRISM tool[57] is a probabilistic model checker used for formal modeling and analysis of stochastic systems. It has also been used to model and analyze stochastic models of biopathways (which primarily arise as CTMCs[58, 59, 60, 61]). System models are described using a high-level state-based description language. In this language a system is described as the parallel composition of a set of *modules*. The PRISM model description is then translated into a CTMC, DTMC or Markov Decision Process (MDP). Properties are specified using PCTL (for DMTCs) or CSL (for CTMCs). In PRISM it is possible to either determine if a probability satisfies a given bound or obtain its actual value. There is also support for the specification and analysis of properties based on costs and rewards.

However, the primary concern in working with stochastic models is that of scalability and the resource intensive nature of computations. Performing stochastic simulations is slow even for small systems; hence considering practically large pathways is almost always intractable. The task of model calibration is also equally challenging for these class of systems.

2.2 Model construction

Model building and the associated analysis are important steps and we will discuss them in some detail in the current and following sections. Figure2.1 depicts the life cycle of building and analyzing a computational model.

Once we decide the scope of the modeling exercise, we build the structure of the model which incorporates our current understanding of the pathway. Resources such as

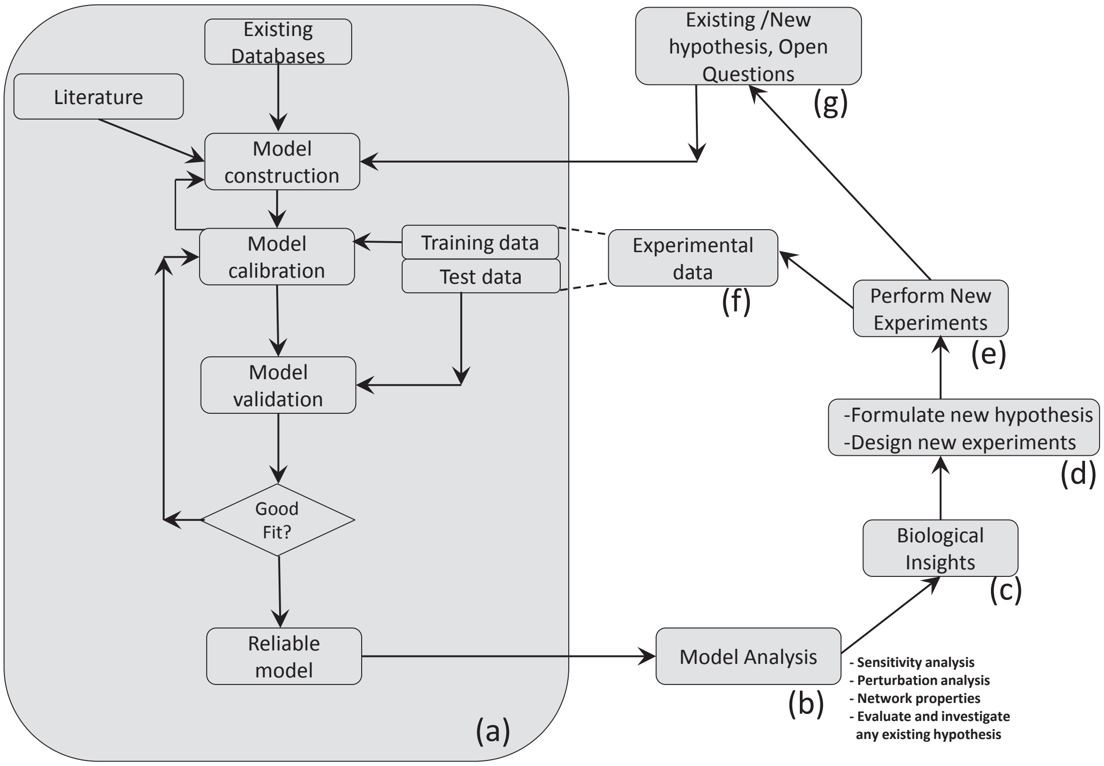


Figure 2.1: Life cycle of building a reliable computational model of Biopathways

existing literature about the pathway, databases such as Reactome [62], KEGG [63] etc., are used for the process. The initial structure also incorporates additional insights and domain knowledge by biologists. Next, a suitable modeling formalism is chosen to model the pathway.

2.3 Model calibration and validation

Once the structure of the pathway and a suitable modeling formalism has been decided, next, the task is to calibrate the model. Model calibration, often referred to as parameter estimation, deals with estimating unknown parameters of the model (depending on the chosen formalism). Unknown parameters usually include the kinetic reaction rate constants and initial concentration of reactants. The goal is to calibrate the model so that model predictions can reproduce the observations in experimental data. The available experimental data is usually divided into two parts, one is used for calibrating the model and the other is used to test the quality of estimated parameters. The problem is formulated as a mathematical optimization with the aim of minimizing (or maximizing) an *objective function*. The objective function gives a measure of difference (or similarity) between the experimental data and the model output. Parameter estimation is a resource intensive task since evaluating the goodness of fit for each parameter combination involves repeatedly simulating the underlying model. In large pathway models the search space can be high dimensional (owing to the large number of unknown parameters), and the objective function is non-linear and multi-modal.

The task of parameter estimation algorithms is to traverse the high dimensional parameter space to look for good parameter sets which can explain the experimental data. The major distinguishing feature of various optimization algorithms lies in the way they traverse the parameter space. They can be classified into *local* and *global* optimization methods. Local methods such as Levenberg-Marquardt [64, 65], Steepest Descent [66] and Hooke and Jeeves [67] have the advantage of converging fast, but usually suffer from the problem of settling in local minima. Global methods such as Genetic Algorithms (GA) [68], and Stochastic Ranking Evolutionary Strategy (SRES) [69] – although time consuming – guarantee a globally optimal solution in principle. A typical search procedure involves iteratively performing the following two steps until there is a good fit between model and experimental observations: 1) guess values of parameters based on the chosen optimization method 2) evaluate the objective function of the guessed parameters. Global optimization algorithms such as GA and SRES are known to perform well in the context of pathway models [70]. Given the dimensionality curse of parameter estimation, there has been some interesting work on de-compositional approaches for parameter estimation [71, 72].

Once the model is calibrated, it is subjected to model validation. In this step the model output is evaluated for goodness of fit with the test data (that was not used to train the model). If the fit is reasonably good, then we have a fairly accurate model using which further analysis tasks can be carried out. If the fit is not acceptable, then we continue another round of parameter estimation. This process continues till we can get reliable parameter estimates. Sometimes, we may not be able to get good parameter sets even after performing multiple rounds of parameter estimation, in which case we may need to go back to our original model structure and refine it by gathering more experimental or literature evidence about the structure and dynamics in close collaboration with biologists.

2.4 Model analysis

Once a reliable computational model has been built, next, one can perform various model analysis tasks using the model. Analysis methods such as *bifurcation analysis* [73], provides a framework to qualitatively analyze the dependence of qualitative behavior(such as oscillations) of the system on model parameters. It graphically describes the change in the behavior of a system when one or more model parameters are varied. Bifurcation points are points along the parameter space where there is switch in the desired behavior. It has been used in the context of biological systems for robustness analysis [73, 74, 75].

Another analysis method is *sensitivity analysis* which aims to study how changes in the kinetic rate constants or initial concentrations of species of the model affect the desired of dynamic behavior the model, either qualitatively or quantitatively.

Sensitivity analysis Sensitivity analysis deals with the study of how variations in parameters affect the dynamical behavior of the model. It helps in tasks such as robustness analysis, model reduction, optimal experimental design, drug target selection [76, 77, 78] etc. Sensitivity analysis methods can be classified into *local* and *global* methods. Local methods focus on assessing the effect of changes in individual parameters around their nominal values, locally [79, 80]. However, assessing changes locally can sometimes lead to misleading results. Global methods [81, 82], on the other hand, assess the importance of the parameters by varying them in a global manner. Various global methods have been recently applied on biological pathway models [83, 84, 85, 86]. These approaches, in general, work by drawing a representative set of samples from the parameter space, simulating the system for the chosen parameter sets, and deriving the global sensitivities of parameters by statistical analysis of the simulation results. For instance, Multi-parametric sensitivity analysis (MPSA) [87, 83], classifies the sampled parameter sets into *acceptable* and *unacceptable* classes based on a defined measure. Based on the distribution of elements in these classes it computes the sensitivity index.

Verification and analysis using formal methods

Getting meaningful biological insights from models is crucial. However, as the scale of these models increases, ensuring that models are in accordance with the current knowledge of the system and conform to experimental data are crucial. On the other

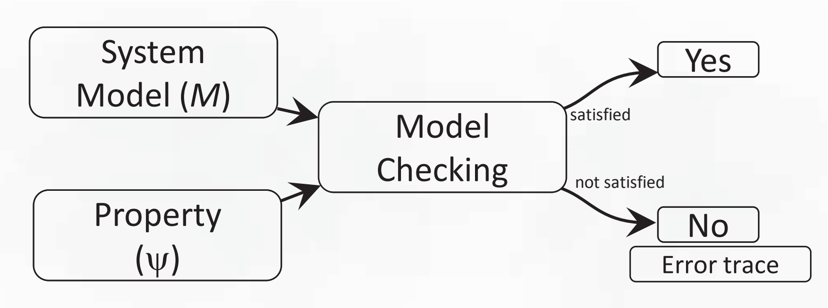


Figure 2.2: General model checking procedure

hand, modeling is essentially an iterative process, one may have to re-estimate some parameters, add new links to the model when new experimental data becomes available or if new hypotheses are to be incorporated into the model. At every stage of model construction and refinement there is a natural need for verifying these models to ensure that they are consistent with what is known about the system. In addition, for such large models, manual analysis of simulation output is increasingly difficult and is prone to interpretation error depending on the person analyzing the results. More importantly, instead of resorting to simulations, techniques which can look at all possible outcomes of the system behavior and reason about its properties are important.

Formal methods such as model checking provide an attractive approach for dealing with these issues. The basic idea is to formalize qualitative or quantitative system behavior into queries in a specification language - called temporal logics. These queries are then automatically processed using efficient algorithms to decide the extent to which the system conforms to them. There has been an increasing interest in using these approaches for analyzing biopathway dynamics[88, 89, 90, 57, 91, 92].

Model Checking refers to the broad class of techniques to automatically evaluate if a system model satisfies specific properties expressed as formulas in temporal logics. This method was initiated in the seminal work of Amir Pnueli [93] who proposed temporal logics as a formalism for specifying dynamic properties of computing systems which was followed by the technique of model checking, proposed independently by Clarke and Emerson [94] and Quellie and Sifakis [95]. Model checking has been widely used in domains of embedded systems, software engineering etc., to find critical bugs in hardware and software modules. These techniques have also been extended to analyze stochastic systems such as Markov chains, where they are studied under the umbrella of *probabilistic model checking.*

The main components of model checking procedure are as shown in figure 2.2

1. A model M of the system, represented as a state transition graph where the nodes (S) represent the possible states of the system and the edges ($T \subseteq S \times S$) represent possible transitions of the system from one state to another.

2. A labeling function L that labels each state in (S) with atomic propositions (AP) that hold in the state i.e, $L : S \mapsto 2^{AP}$;

3. The property to be checked (ψ) is expressed as formulas using temporal logics. These formulas are built using atomic propositions, propositional connectives and temporal operators.

4. A model checker which systematically explores the state space to verify if the property ψ holds for the model M.

The usefulness of model checking in systems biology is currently being emphasized [96]. It is suggested that in the future a library of model-checking queries that encode key behavioral features of a biological pathway will be built, which would be used as a yard stick to check the reliability of a model. It will enable testing any new model against these queries to assess its predictive power, a model that is consistent with all or most of the behavioral features in the library viewed as being reliable. Model checking has also been applied for model calibration and sensitivity analysis tasks [97, 98, 99, 100].

Chapter 3

Dynamic Bayesian Networks

In this chapter, we will begin by defining the notions of Markov chains, Bayesian networks and dynamic Bayesian networks (DBNs). Next, we will describe how DBNs arise as approximate representations of biopathway dynamics induced by a system of ODEs. This will form the basis for the material presented in the subsequent chapters.

3.1 Markov Chains

Consider a stochastic process $\{X_t,\, t = 0, 1, 2, 3...\}$. Assume that it takes values from a finite domain, say $S \in \{s_0, s_1, s_2, s_3...s_m\}$. Here t ranges over the time points of interest and $X_t = s_k$ indicates that the process is at a state s_k at time t. A Markov chain[101] can be defined as a stochastic process such that:

$P(X_{t+1} = s_j \mid X_t = s_i, X_{t-1} = s_{t-1}, \cdots, X_0 = s_0) = P(X_{t+1} = s_j \mid X_t = s_i) = p_{ij}$; $s_{t-1}, \cdots, s_0 \in S$; $i, j \in \{0, 1, 2...., m\}$ and $t \geq 0$.

The above expression says that the conditional probability of the stochastic process being at state s_j at time $t+1$ (X_{t+1}) given all its past states ($X_{t-1}, \cdots, X_0$) and current state(time X_t) is independent of all the past states and is given by p_{ij}. This is the Markov property. Whenever the process is in some state s_i at time t, it will transit to state s_j at time $t+1$ with a fixed probability p_{ij}, often referred to as transition probability. As a result, $p_{ij} \in [0, 1]$ and $\sum_j p_{ij} = 1$. We represent the transition probabilities using the matrix T of order $m \times m$, whose element T_{ij}=p_{ij}. An initial distribution λ^0 is specified over S at $t = 0$. The probability distribution λ^k over S at $t = k$ will be given by $(\lambda^0)T^k$.

3.2 Bayesian Networks

Bayesian networks (BN) ([14]) belong to a class of probabilistic graphical models consisting of a finite acyclic graphical graph $G_{\mathcal{B}} = (N, E)$ where N is the set of nodes, each node i representing a finite valued random variable X_i taking a value from domain V of cardinality K, for $1 \leq i \leq size(N)$. The set E of edges between nodes represent the local dependencies between nodes. Associated with every node X_i is a conditional probability table $C_i = P(X_i|Pa(X_i))$ where $Pa(X_i)$=$\{X_{j1}, X_{j2}....X_{jm}\}$ are the parents of the node X_i such that $\{j_k, i\} \in E$ for $1 \leq k \leq m$. The entries of C_i are of form $C_i(x_i \mid x_{j_1}, x_{j_2}...x_{j_m})$ where $x_i \in V$ and $(x_{j_1}, x_{j_2}...x_{j_m}) \in V^m$. Bayesian networks in essence encode and represent our assumptions about the conditional independence of variables in a distribution. In other words, Bayesian networks compactly maintain the joint distribution $P(X_1, X_2......X_l)$ in a factorized way as $\prod_{i=1}^{l} C_i$. Bayesian network have found wide applications including those in computational biology[102, 103], computer vision[104], gaming[105], information retrieval[106] etc.

3.3 Dynamic Bayesian Networks

Dynamic Bayesian Networks(DBNs) are a class of probabilistic graphical models which are used to model dynamical systems. They extend Bayesian networks to represent system behavior over time. DBNs have been extensively used in the fields of AI, computer vision, signaling processing [14],[107],[108]. They have also been used in computational biology to mainly model temporal data[109, 110]. A DBN consists of a finite set of random variables, with each variable taking a value from a finite domain V of cardinality K. The state of the system at a particular time point is given by the probability distribution of these random variables at particular time point. Formally, DBN $\mathcal{D}$ has an associated set of system variables $\mathcal{X} = \{X_1, X_2, \ldots, X_n\}$. It also has a discrete time domain $\mathcal{T} = \{0, 1, \ldots\}$ associated with it.

The structure of $\mathcal{D}$ consists of an acyclic directed graph $G_{\mathcal{D}} = (N, E)$ with $N = \mathcal{X} \times \mathcal{T}$. Thus there will be one node of the form X_i^t for each $t \in \mathcal{T}$ and each $i \in \{1, 2, \ldots, n\}$. The node X_i^t is to be viewed as a random variable that records the value assumed by the variable X_i at time t. The edge relation is derived by fixing the parenthood relation $PA : \mathcal{X} \to 2^{\mathcal{X}}$ over the system variables. Intuitively, $PA(X_i)$ is the set of system variables whose values at time t -probabilistically- influence the value assumed by X_i at time $t+1$. This crucial structural information is to be obtained from the application at hand and will often be readily available.

The map PA will in turn induce the map $Pa : N \to 2^N$ given by: $Pa(X_i^t) = 0$ if $t = 0$. For $t > 0$, $X_j^{t'} \in Pa(X_i^t)$ iff $t' = t-1$ and $X_j \in PA(X_i)$. The edge relation E is then given by: $(X_j^{t'}, X_i^t) \in E$ iff $X_j^{t'} \in Pa(X_i^t)$. The set $Pa(X_i^t)$ is also referred to as *parents* of variable X_i^t.

We consider a restricted class of DBNs in our discussion which are time-variant but have regular structure i.e the structure in terms of edges between variables across time points does not change, but the probabilistic relation between them changes. An example of such a DBN is shown in figure3.1.

Let i, j range over $\{1, 2, \ldots, n\}$. We denote by $\mathbf{X}$ the tuple $(X_1, \ldots, X_n)$. We let x_i, u_i, v_i to denote a value taken by X_i. They will be unrolled over a finite number of time points. Further, there will be no distinction between hidden and observable variables. To sum up, in our setting,

A *Dynamic Bayesian Network (DBN)* is a structure $\mathcal{D} = (\mathcal{X}, T, Pa, \{C_i^t\})$ where,

- T is a positive integer with t ranging over the set of time points $\{0, 1, \ldots, T\}$.

- $\mathcal{X} = \{X_i^t \mid 1 \leq i \leq n, 0 \leq t \leq T\}$ is the set of random variables. As usual, these variables will be identified with the nodes of the DBN. X_i^t is the instance of X_i at time slice t.

- (i) $Pa(X_i^0 = \emptyset)$ (ii) If $X_j^{t'} \in Pa(X_i^t)$ then $t' = t-1$. (iii) If $X_j^{t-1} \in Pa(X_i^t)$ for some t then $X_j^{t'-1} \in Pa(X_i^{t'})$ for every $t' \in \{1, 2, \ldots, T\}$. Thus the way nodes at the $(t-1)^{th}$ time slice are connected to nodes at the t^{th} time slice remains invariant as t ranges over $\{1, 2, \ldots, T\}$.

- C_i^t is the *Conditional Probability Table (CPT)* associated with node X_i^t specifying the probabilities $P(X_i^t \mid Pa(X_i^t))$. Suppose $Pa(X_i^t) = \{X_{j1}^{t-1}, X_{j2}^{t-1}, \ldots, X_{j_m}^{t-1}\}$ and $(x_{j_1}, x_{j_2}, \ldots, x_{j_m}) \in V^m$. Then we require, $\sum_{x_i \in V} C_i^t(x_i \mid x_{j_1}, x_{j_2}, \ldots, x_{j_m}) = 1$. Since the DBNs we discuss here are time-variant, in general C_i^t will be different from $C_i^{t'}$ if $t \neq t'$.

A state of the DBN at t will be a member of V^n, say $\mathbf{s} = (x_1, x_2, \ldots, x_n)$ specifying

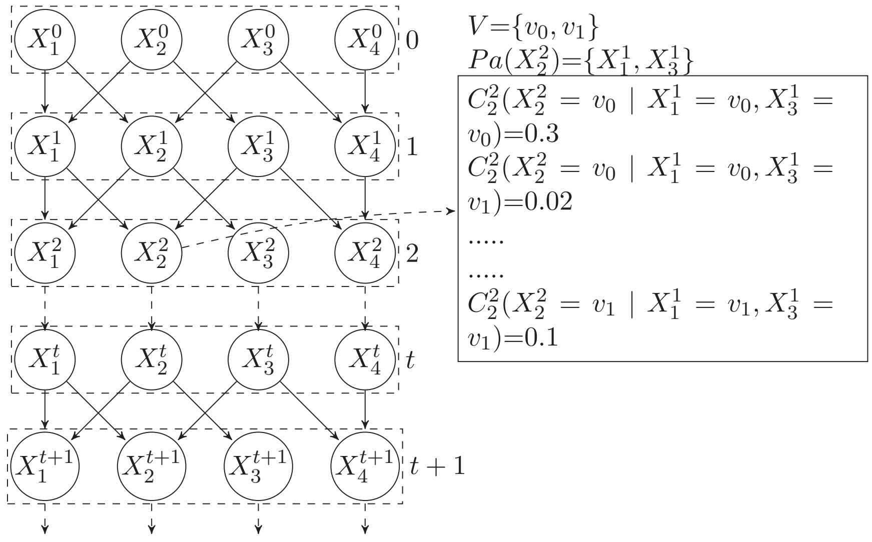

V={v_0, v_1}
$Pa(X_2^2)$={X_1^1, X_3^1}
$C_2^2(X_2^2 = v_0 \mid X_1^1 = v_0, X_3^1 = v_0)$=0.3
$C_2^2(X_2^2 = v_0 \mid X_1^1 = v_0, X_3^1 = v_1)$=0.02
.....
.....
$C_2^2(X_2^2 = v_1 \mid X_1^1 = v_1, X_3^1 = v_1)$=0.1
X_1^0 X_2^0 X_3^0 X_4^0 0
X_1^1 X_2^1 X_3^1 X_4^1 1
X_1^2 X_2^2 X_3^2 X_4^2 2
X_1^t X_2^t X_3^t X_4^t t
X_1^{t+1} X_2^{t+1} X_3^{t+1} X_4^{t+1} $t+1$

Figure 3.1: Example of a DBN

that $X_i^t = x_i$ for $1 \leq i \leq n$. This in turn stands for $X_i = x_i$ for $1 \leq i \leq n$ at t. Suppose $Pa(X_i^t) = \{X_{j_1}^{t-1}, X_{j_2}^{t-1}, \ldots, X_{j_m}^{t-1}\}$. Then a CPT entry of the form $C_i^t(x_i \mid x_{j_1}, x_{j_2}, x_{j_m}) = p$ says that if the system is in a state at $t-1$ in which $X_{j_l} = x_{j_l}$ for $1 \leq l \leq m$, then the probability of $X_i = x_i$ being the case at t is p. In this sense the CPTs specify the probabilistic dynamics locally. We define $\hat{i} = \{j \mid X_j \in PA(X_i)\}$ to capture Pa in terms of the corresponding indices.

In this thesis, $\mathbf{x}_I$ will denote a vector of values over the index set $I \subseteq \{1, 2, \ldots, n\}$. It will be viewed as a map $\mathbf{x}_I : I \to V$. We will often denote $\mathbf{x}_I(i)$ as $\mathbf{x}_{I,i}$ or just $\mathbf{x}_i$ if I is clear from the context. If $I = \{i\}$ and $\mathbf{x}_I(i) = x_i$, we will identify $\mathbf{x}_I$ with x_i. If I is the full index set $\{1, 2, \ldots, n\}$, we will simply write $\mathbf{x}$. Further, we denote by $\mathbf{X}^t$ the vector of random variables $(X_1^t, \ldots, X_n^t)$. Using these notations, we can write $C_i^t(x_i \mid \mathbf{u}_{\hat{i}}) = p$ to mean that p is the probability that $X_i = x_i$ at time t given that at time $t-1$, $X_{j_1} = \mathbf{u}_{j_1}, X_{j_2} = \mathbf{u}_{j_2}, \ldots, X_{j_m} = \mathbf{u}_{j_m}$ with $\hat{i} = \{j_1, j_2, \ldots, j_m\}$.

A primary task for analysis using DBNs is to Infer the probability distribution of the random variables is important, this is a crucial aspect of this thesis. We will discuss, in detail, the different probabilistic inference algorithms on DBNs in the following chapters.

As discussed before, our focus is on model checking DBN models which serve as succinct representations of Markov chains. In this section we describe how a rich class of DBNs arises as approximations of ODE dynamics. This method was developed in [12],[111].

3.4 Approximating ODE dynamics

Signaling pathways usually have external or internal stimuli triggering signaling proteins which then cascade these signals to downstream proteins and finally the signal reaches the effector protein which results in a biologically observable effect. The levels of these proteins play a crucial role in how the signal is transduced. The concentration levels of these proteins are recorded at specific time points. Experimental observations usually have limited precision owing to limitations in experimental technology. The data available from them are in the form of multiple repeats of the experiment, each having slightly different values due to experimental error or changes due to cell-cell variability. Sometimes the data may be available from different labs which are performed in slightly different conditions etc. Hence it is better to think of these species concentrations not as point values but being in discretized levels, the simplest being $High$ or low etc.

In addition, ODEs describing these processes are usually nonlinear due to the nature of the kinetic laws governing the reactions. Except for the toy examples, the ODEs system will also be high dimensional. Hence, closed-form solutions will not be obtainable. One must instead resort to repeated large scale numerical simulations to perform tasks such as parameter estimation, validation and sensitivity analysis. Further, only a small amount of noisy data of limited precision will be available to support model calibration and validation.

With this motivation, we describe the discrete approximation of biological pathway dynamics modeled using ODEs [112, 12, 111]. To formalize notations, let the biologically relevant time points of interest be $\{0, 1, \ldots, T\}$. Next we assume that we are interested in the concentrations of the different species involved in the pathway (referred to as variables from now on) only in terms of their relative levels and not as point values. Let us assume that the pathway has n variables (species) of interest, denoted by $y_1, y_2 ... y_n$ and m kinetic rate constants of interest denoted by $r_1, r_2 ... r_m$ respectively.

We are specifically interested in the dynamics of these pathways. Figure 3.2(a) shows a simple network consisting of 3 reactions; a pair of reversible reactions and one irreversible reaction. The dynamics of such a network can be modeled as a system of ODEs; as discussed before, there one equation of the form $\frac{dy_i}{dt} = f(\mathbf{y}, \mathbf{r})$ for each

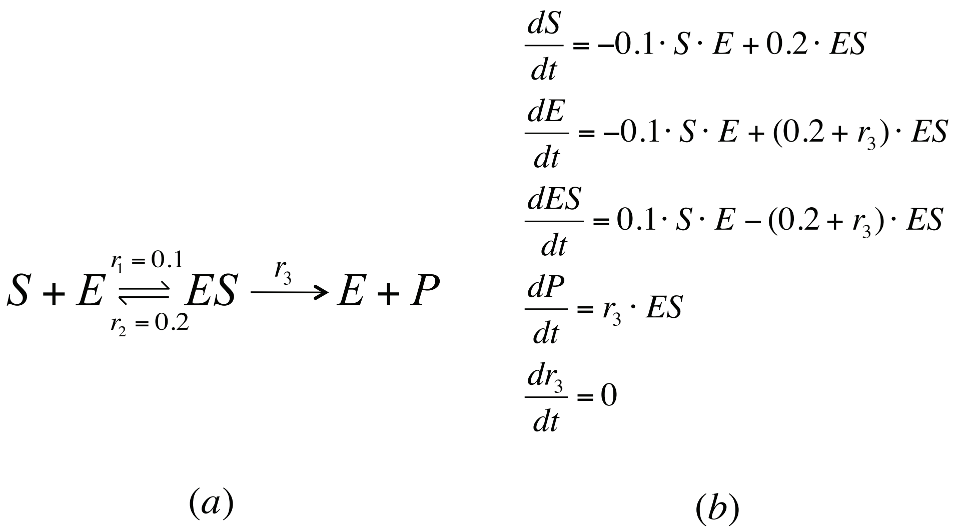

$r_1 = 0.1$
$r_2 = 0.2$
r_3
$S + E \rightleftharpoons ES \longrightarrow E + P$
$\frac{dS}{dt} = -0.1 \cdot S \cdot E + 0.2 \cdot ES$
$\frac{dE}{dt} = -0.1 \cdot S \cdot E + (0.2 + r_3) \cdot ES$
$\frac{dES}{dt} = 0.1 \cdot S \cdot E - (0.2 + r_3) \cdot ES$
$\frac{dP}{dt} = r_3 \cdot ES$
$\frac{dr_3}{dt} = 0$
(a)
(b)

Figure 3.2: (a) The enzyme catalytic reaction network. (b) The ODE model

molecular species y_i, with f describing the kinetics of the reactions that produce and consume y_i, while $\mathbf{y}$ is the set (vector) of molecular species taking part in these reactions and $\mathbf{r}$ are the rate constants associated with these reactions. The speed of each reaction will be determined by the kinetic law governing this reaction. The rate constants specify the relative speed and affinity of the different reaction components. In figure 3.2(b), we have assumed that the kinetics of all three reactions is governed by the mass law [25] which states that the rate at which a reaction proceeds is directly proportional to the current concentration levels of the reactants taking part in the reaction. Thus the rate at which the forward reaction produces the enzyme-substrate complex ES from the substrate S and the enzyme E is directly proportional to the current concentrations of E and S. Further, the *rate constant* for this reaction, is given to be 0.1 in this example. This produces the term $0.1 \times S \times E$ in the equation for S which will capture the rate at which S is being depleted due to the forward reaction. Similarly the term $0.2 \times ES$ will capture the rate which S is being produced by the reverse reaction where we are given that the rate constant for this reaction is 0.2.

The range of values of each variable y_i is partitioned into $|\mathcal{I}_i|$ intervals where $\mathcal{I}_i = \{[v_i^{min}, v_i^1),\ [v_i^1, v_i^2),\ \ldots,\ [v_i^{L_i-1}, v_i^{max}]\}$ denotes the set of these intervals. We discretize the range of each parameter r_j (in total m of them) into $|\mathcal{I}_{r_j}|$ intervals where $\mathcal{I}_{r_j} = \{[v_{r_j}^{min}, v_{r_j}^1),\ [v_{r_j}^1, v_{r_j}^2),\ \ldots,\ [v_{r_j}^{L_{r_j}-1}, v_{r_j}^{max}]\}$. The set defined by $\mathcal{I} = \cup_{i=1}^{n} \mathcal{I}_i \cup \cup_{j=1}^{m} \mathcal{I}_{r_j}$ will be called the **discretization**. The discretization and flow induced by the systems of ODEs induces a discrete time Markov chain ($\mathcal{MC}$). Let v_{y_i} be a real number in the range of y_i. We define $[v_{y_i}]$ as the interval in which v_{y_i} falls. Similarly, let k_{r_j} be a real number in the range of r_j, we define $[k_{r_j}]$ as the interval in which k_{r_j} falls in. Next, for the vector defining all the species and kinetic parameters represented by $\mathbf{s} = (v_{y_1}, v_{y_2}, \ldots, v_{y_n}, k_{r_1}, k_{r_2}, \ldots, k_{r_m})$, we define the interval vector - referred to as a **discrete state** - as $[\mathbf{s}] = ([v_{y_1}], [v_{y_2}], \ldots, [v_{y_n}], [k_{r_1}], \ldots, [k_{r_m}])$. In our Markov chain, a state is defined as - $\mathcal{MC}$-state - is a pair $(\mathbf{s'}, t)$, where $\mathbf{s'}$ is a discrete state and $t \in \{0, 1, \ldots, T\}$. Next, we define the probability of a discrete state $\mathbf{s'}$ at time point t as $Pr(\mathbf{s'}, t) = \mathbf{P}^t(\{\mathbf{s'} \mid \mathbf{s'} \in I_1 \times I_2 \times \ldots I_n \times I_{r_1} \ldots \times I_{r_m}\})$, where $\mathbf{P}^t$ is the probability distribution at time t over the $\sigma-$ algebra pertaining to the flow induced by the set of ODEs assuming that the initial values of the variables of the ODEs are uniformly distributed within a hypercube $I_1^0 \times I_2^0 \times \ldots I_n^0 \times I_{r_1}^0 \ldots \times I_{r_m}^0$, $\mathbf{s'} = (I_1, I_2, \ldots I_n, I_{r_1} \ldots, I_{r_m})$; here I_i, I_i^0, $I_{r_j}^0$ and I_{r_j} will represent an interval belonging to $\mathcal{I}_i$ and $\mathcal{I}_{r_j}$ and $i \in \{1 \ldots n\}$ and $j \in \{1 \ldots m\}$. For more technical details, we refer the reader to [112].

An $\mathcal{MC}$-state, $(\mathbf{s'}, t)$ is feasible iff $Pr(\mathbf{s'}, t) > 0$. Next, the transition relation between $\mathcal{MC}$-states is denoted as $\rightarrow$, it is defined as : $(\mathbf{s'}, t) \rightarrow (\mathbf{s''}, t')$ iff $t = t' - 1$, both the states should be feasible and the states $(\mathbf{s'}, t)$ and $(\mathbf{s''}, t')$ should be reachable by the flow induced by ODEs. Having defined the states of the Markov chain and the transition relation, next, let us now look at the transition probabilities of the Markov chain. Let E and F denote the event that the system is in the state $(\mathbf{s'}, t)$ and in $(\mathbf{s''}, t')$, $t' = t + 1$, both the states being feasible. Let $E \cap F$ be the joint event the system is at the $(\mathbf{s'}, t)$ and $(\mathbf{s''}, t')$ at $t' = t + 1$. Consequently, the transition probability $Pr((\mathbf{s'}, t) \rightarrow (\mathbf{s''}, t')) = Pr(F|E) = Pr(E \cap F)/Pr(E)$. Refer to [112] for more information. We can now define the Markov chain, $\mathcal{MC}$, as $(\mathcal{S}, T_{\mathcal{S}})$, where $\mathcal{S}$ is the set of $\mathcal{MC}$-states and $T_{\mathcal{S}}$ is the transition probability matrix where entries correspond to the probability of transitioning between any two $\mathcal{MC}$-states $\in \mathcal{S}$.

$\mathcal{MC}$ cannot be explicitly computed for ODEs since they typically do not have closed form solutions and that it is large. Thus, one can only compute approximations of $\mathcal{MC}$. To do so, we can simulate the system by sampling the initial state many times according to the assumed prior distribution, determine through numerical integration the $\mathcal{MC}$-states as well as the transitions along this simulated trajectory. Then through a simple counting process involving the generated trajectories, the Markov chain can be computed as an

$$\frac{dS}{dt} = -0.1 \cdot S \cdot E + 0.2 \cdot ES$$

$$\frac{dE}{dt} = -0.1 \cdot S \cdot E + (0.2 + r_3) \cdot ES$$

$$\frac{dES}{dt} = 0.1 \cdot S \cdot E - (0.2 + r_3) \cdot ES$$

$$\frac{dP}{dt} = r_3 \cdot ES$$

$$\frac{dr_3}{dt} = 0$$

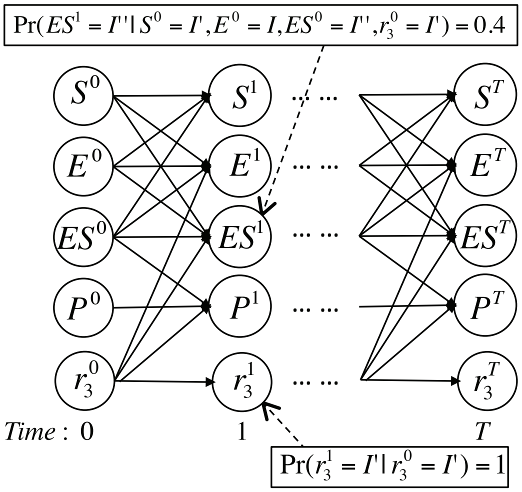

$\Pr(ES^1 = I'' \mid S^0 = I', E^0 = I, ES^0 = I'', r_3^0 = I') = 0.4$
S^0
E^0
ES^0
P^0
r_3^0
S^1
E^1
ES^1
P^1
r_3^1
S^T
E^T
ES^T
P^T
r_3^T
$Time: 0$
1
T
$\Pr(r_3^1 = I' \mid r_3^0 = I') = 1$

Figure 3.3: DBN approximation of the ODE

approximation of $\mathcal{MC}$. In the worst case, the number of states in this approximated Markov chain will be $O(K^{n+m})$ where K is $max\{|\mathcal{I}_i|, 1 \leq i \leq n; |\mathcal{I}_{r_j}|, 1 \leq j \leq m\}$. As a result, for many biological pathways, it will be too large. For instance for the pathway models we consider, each having about 30 proteins, whose values are each discretized into 5 intervals, the number of potential states are of the order of 5^{30} even across a single time step, which is too large to be represented and analyzed explicitly.

3.4.1 The DBN representation of ODE dynamics

The main observation that leads to a compact representation of the Markov chain introduced in the last section, is that we can factorize the Markov chain, $\mathcal{MC}$, by exploiting the structure information in ODEs and representing it compactly as a time variant DBN. First, we specify a random variable Y_i for each variable y_i of the ODE model. Next, for each unknown rate constant r_j , we add one random variable R_j. Since we have m unknown parameters, each time slice of the DBN will consist of $n + m$ nodes, one for each of the random variables. Across every time slice, the node Y_k^{t-1} will be in $Pa(Y_i^t)$ iff $k = i$ or y_k appears in the equation for $d(y_i)/dt$. Further, the node R_j^{t-1} will be in $Pa(Y_i^t)$ iff r_j appears in the equation for $d(y_i)/dt$. On the other hand R_j^{t-1} will be the only parent of the node R_j^t. Figure 3.3 shows the transformation of the ODE. In this example, we have assumed that r_3 is the only unknown rate constant.

Suppose $Pa(Y_i^t) = \{Z_1^{t-1}, Z_2^{t-1}, \ldots, Z_k^{t-1}\}$. Then a CPT entry of the form $C_i^t(I \mid I_1, I_2, \ldots, I_k) = p$ says that p is the probability of the value of y_i falling in the interval I at time t, given that the value of Z_j was in I_j for $1 \leq j \leq k$. The probability p is calculated through simple counting. Suppose N is the total number of generated trajectories. We first record, the number of trajectories whose value of Z_j falls in the interval I_j simultaneously for each $j \in \{1, 2, \ldots, k\}$ at time $t - 1$. Suppose this number is J. We then determine for how many of these J trajectories, the value of Y_i falls in the interval I at time t. suppose this number is J', then p is set to be $\frac{J'}{J}$ (It should now be clear why $C_i^t(I \mid I_1, I_2, \ldots, I_k)$ will be in general different from $C_i^{t'}(I \mid I_1, I_2, \ldots, I_k)$ if $t \neq t'$). If r_j is an unknown rate constant, in the CPT of R_j^t we will have $P(R_j^t = I_{r_j} \mid R_j^{t-1} = I'_{r_j}) = 1$ if $I = I'$ and $P(R_j^t = I_{r_j} \mid R_j^{t-1} = I'_{r_j}) = 0$ otherwise. This is because the sampled initial value of r_j does not change during numerical integration. Suppose r_j appears on the right hand side of the equation for y_i and $Pa(Y_i^t) = \{Z_1^{t-1}, Z_2^{t-1}, \ldots, Z_\ell^{t-1}\}$ with $Z_\ell^{t-1} = R_j^{t-1}$. Then for each choice of interval values for nodes other than R_j^{t-1} in $Pa(Y_i^t)$ and for each choice of interval value $\widehat{I}_{r_j}$ for r_j there will be an entry in the CPT of Y_i^t of the form $P(y_i^t = I \mid Z_1^{t-1} = I_1, Z_2^{t-1} = I_2, \ldots, R_j^{t-1} = \widehat{I}_{r_j}) = p$. This is so since we will sample for all possible initial interval values for r_j. In this sense the CPTs record the approximated dynamics for all possible combinations of interval values for the unknown rate constants. These features are illustrated in figure 3.3 for the unknown rate constant r_3. For more details, we refer the reader to [112]. Once the DBN approximation has been constructed, tasks such as parameter estimation and sensitivity analysis can be carried out efficiently using standard DBN inferencing algorithms [12].

Chapter 4

Inference on Dynamic Bayesian Networks

4.1 Introduction

Probabilistic graphical models such as DBNs -as we discussed in the previous chapters- solve the problem of succinctly representing high dimensional probabilistic dynamics. However, the time complexity of inferring the probability distribution of states at a given time point in these models is still exponential in the size of the network [15]. This chapter focuses on probabilistic inference algorithms on DBNs. Specifically, the focus is on computing the marginal probability distribution of random variables.

We first discuss existing inference algorithms for DBNs. Next, we present our improved inference algorithm, termed hybrid factored frontier (HFF). We provide experimental results to validate the scalability and efficiency of HFF. These inference algorithms will play a crucial role in the model checking algorithms described later. First we look at exact inference for DBNs.

Exact probabilistic inference

Using notations developed in Chapter 3, the joint probability distribution $P(X_1^t, X_2^t, \ldots, X_n^t)$ describes the possible states of the system at time point t. In other words, $P(\mathbf{X}^t = \mathbf{x})$ is the probability that the system will reach the state $\mathbf{x}$ at t. Starting from $P(\mathbf{X}^0)$ at time 0, given by $P(\mathbf{X}^0 = \mathbf{x}) = \prod_i C_i^0(\mathbf{x}_i)$ probabilistic inference aims to compute $P(X_1^t, \ldots, X_n^t)$ for a any given time point t. We can compute this exactly using the CPTs to inductively compute this:

$$P(\mathbf{X}^t = \mathbf{x}) = \sum_{\mathbf{u}} \Big(\prod_i C_i^t(\mathbf{x}_i \mid \mathbf{u}_{\hat{i}}) \Big) P(\mathbf{X}^{t-1} = \mathbf{u}) \tag{4.1}$$

with $\mathbf{u}$ ranging over V^n.

Since $|V| = K$, the number of possible states at t is K^n. Hence explicitly computing and maintaining the probability distributions is feasible only if n is small or if the underlying graph of the DBN falls apart into many disjoint components. Neither restriction is realistic and hence one needs approximate ways to maintain $P(\mathbf{X}^t)$ compactly and compute it efficiently.

Two main deterministic approximate algorithms include the factored frontier algorithm (FF)[16], the Boyen Koller algorithm (BK)[15] [113]. These algorithms maintain the joint probability distributions approximately; such approximate distributions are usually called belief states. In BK, a belief state is maintained compactly as a product of the probability distributions of independent clusters of variables. This belief state is then propagated *exactly* at each step through the CPTs. Then the new belief state is compacted again into a product of the probability distributions of the clusters. This is in contrast to FF algorithm which maintains a belief state as a product of the marginal distributions of the individual variables. Instead of computing first the new belief state as done by BK, the FF algorithm computes the new marginal distributions directly via the propagation of the current marginal distributions through the CPTs. Finding the right set of clusters in BK is important for improved results, and if the cluster size is large, inference is still infeasible. Moreover, for our application both BK and FF have drawbacks.

FF is attractive in terms of its simplicity and computational effort but unlike the case of BK, it lacks a rigorous error analysis. More importantly, FF can exhibit significant errors. As for BK, apart from the need to compute the next belief state exactly -which can be computationally expensive- its performance depends on how one clusters the variables. Identifying the right set of clusters is a difficult problem. There seems to be no efficient techniques for doing this with guaranteed performance. One could avoid the problem of identifying clusters by just using singleton clusters (the so called fully factored BK algorithm). However, this can also lead to significant errors. This sets the motivation for our work. In specific, we propose an improved parameterized algorithm called hybrid factored frontier algorithm(HFF)[114] which attempts to bridge some of the gaps in previous algorithms. Next, we will discuss the FF algorithm in detail, since our HFF algorithm is based on it. We will follow this up with a description of our improved HFF algorithm and the corresponding error analysis.

4.2 The Factored Frontier algorithm

As discussed before exact inference on DBNs is infeasible for large DBNs. One must use approximate methods, here we will focus on a simple and efficient approximate algorithm called the Factored Frontier (FF) algorithm [16]. FF maintains and propagates joint probability distributions $\Pr(X_1^t, X_2^t, \ldots, X_n^t)$ in an approximate fashion. Approximate probability distributions will be called belief states and denoted by B, B^t etc. Exact probability distributions will be denoted by P, P^t etc. Formally, a belief state B is a map from $V^n \to [0,1]$ such that $\sum_{\mathbf{u}\in V^n} B(\mathbf{u}) = 1$. Thus a belief state is just a probability distribution but it will be convenient to linguistically separate them.

The FF algorithm uses marginal functions to represent belief states. A marginal function is a map $M : \{1,\ldots,n\} \times V \to [0,1]$ such that $\sum_{v\in V} M(i,v) = 1$ for each i. In what follows, u, v will range over V while $\mathbf{u}$ and $\mathbf{v}$ will range over V^n. A belief state B induces the marginal function M_B via $M_B(i,v) = \sum_{\mathbf{u}|\mathbf{u}_i=v} B(\mathbf{u})$. On the other hand, from a marginal function M, one can obtain a belief state B_M via $B_M(\mathbf{u}) = \prod_i M(i,\mathbf{u}_i)$. From the above definitions it follows that for a marginal function M, we have $M_{B_M} = M$. That is, for any i, v, $M_{B_M}(i,v) = \sum_{\mathbf{u}|\mathbf{u}_i=v} B_M(\mathbf{u}) = \sum_{\mathbf{u}|\mathbf{u}_i=v} \prod_j M(j,\mathbf{u}_j) = \prod_j \sum_{\mathbf{u}|\mathbf{u}_i=v} M(j,\mathbf{u}_j) = \left(\prod_{j|j\neq i} \sum_{\mathbf{u}_j} M(j,\mathbf{u}_j)\right) \cdot M(i,v) = M(i,v)$. On the other hand, for a belief state B, unless $B = B_M$, we may have $B_{M_B} \neq B$.

For a DBN $\mathcal{D} = (\mathcal{X}, T, Pa, \{C_i^t\})$ recall that $\hat{i} = \{j \mid X_j \in PA(X_i)\}$ captures the set of indices of the parents of i. In what follows, $V_{\hat{i}}$ will denote the tuple of values defined by $\hat{i}$. Thus, with a slight abuse of notation, $\mathbf{u}, \mathbf{v}$ will be used to denote $|\hat{i}|$-dimensional vectors of values over V.

Given a DBN $\mathcal{D} = (\mathcal{X}, T, Pa, \{C_i^t\})$, FF computes inductively a sequence M^t of marginal functions as:

- $M^0(i,u) = C_i^0(u)$,
- $M^t(i,u) = \sum_{\mathbf{v}\in V_{\hat{i}}}[\prod_{j\in\hat{i}} M^{t-1}(j,\mathbf{v}_j)]C_i^t(u \mid \mathbf{v})$.

It is easy to check that these are indeed marginal functions, i.e., $\sum_{u\in V} M^t(i,u) = 1$ for all t and i. Thus FF maintains B^t, the belief state at t, compactly via the marginal function M^t. More precisely, $B^t(\mathbf{u}) = \prod_j M^t(j,\mathbf{u}_j) = B_{M^t}(\mathbf{u})$.

Let $t \geq 1$. Suppose that the DBN transforms the belief state B^{t-1} into the new belief state $\widehat{B}^t$. In other words, $\widehat{B}^t$ is the belief state obtained by performing $t-1$ steps of FF and exact computation at the t^{th} step. Then by Equation (4.1), we have:

$$\widehat{B}^t(\mathbf{x}) = \sum_{\mathbf{u}} B^{t-1}(\mathbf{u})\Big(\prod_i C_i^t(\mathbf{x}_i \mid \mathbf{u}_{\hat{i}})\Big) \tag{4.2}$$

However, the t^{th} step of FF computes directly the marginal function M^t, which then represents the new belief state at time t as $B^t = B_{M^t}$. In general, $B^t \neq \widehat{B}^t$, that is, the belief state B^t represented via M^t is an approximation of the belief state $\widehat{B}^t$ as defined above. However, the computation of M^t is itself accurate in the following sense.

Proposition 1. *For all $t \in \{1,\ldots,T\}$, $M^t(i,v) = M_{\widehat{B}^t}(i,v)$ for each i and v.*

Proof. For $t > 0$, we have:

$$\begin{aligned}
M_{\widehat{B}^t}(i,v) &= \sum_{\mathbf{v}\mid\mathbf{v}_i=v} \widehat{B}^t(\mathbf{v}) \\
&= \sum_{\mathbf{v}\mid\mathbf{v}_i=v}\sum_{\mathbf{u}}\prod_j B^{t-1}(\mathbf{u})(C_j^t(\mathbf{v}_j \mid \mathbf{u}_{\hat{j}})) \\
&\quad \text{(by Equation (4.2))} \\
&= \sum_{\mathbf{u}} B^{t-1}(\mathbf{u}) \sum_{\mathbf{v}\mid\mathbf{v}_i=v}\prod_j (C_j^t(\mathbf{v}_j \mid \mathbf{u}_{\hat{j}})) \\
&= \sum_{\mathbf{u}} B^{t-1}(\mathbf{u})\Big(\sum_{\mathbf{v}_n} C_n^t(\mathbf{v}_n \mid \mathbf{u}_{\hat{n}})\Big)\ldots \\
&\quad \Big(C_i^t(v \mid \mathbf{u}_{\hat{i}})\Big)\ldots\Big(\sum_{\mathbf{v}_1} C_1^t(\mathbf{v}_1 \mid \mathbf{u}_{\hat{1}})\Big) \\
&= \sum_{\mathbf{u}} B^{t-1}(\mathbf{u})\Big(C_i^t(v \mid \mathbf{u}_{\hat{i}})\Big)
\end{aligned}$$

The last of the above equalities follows since each of the summands within the expression add up to 1. Now, using $B^{t-1}(\mathbf{u}) = \prod_k M^{t-1}(k,\mathbf{u}_k)$ and splitting the above summation, we obtain:

$$
\begin{aligned}
M_{\widehat{B}^t}(i,v) &= \sum_{\mathbf{u}\in V_{\widehat{i}}} \sum_{\mathbf{u}\notin V_{\widehat{i}}} \prod_{k} M^{t-1}(k,\mathbf{u}_k)\Big(C_i^t(v \mid \mathbf{u}_{\widehat{i}})\Big) \\
&= \sum_{\mathbf{u}\in V_{\widehat{i}}} \prod_{k\in\widehat{i}} M^{t-1}(k,\mathbf{u}_k)\Big(C_i^t(v \mid \mathbf{u}_{\widehat{i}})\Big) \\
&\quad \sum_{\mathbf{u}\notin V_{\widehat{i}}} \prod_{k\notin\widehat{i}} M^{t-1}(k,\mathbf{u}_k)
\end{aligned}
$$

$$
\begin{aligned}
&= \sum_{\mathbf{u}\in V_{\widehat{i}}} \prod_{k\in\widehat{i}} M^{t-1}(k,\mathbf{u}_k)\Big(C_i^t(v \mid \mathbf{u}_{\widehat{i}})\Big) \\
&\quad \prod_{k\notin\widehat{i}} \sum_{\mathbf{u}_k} M^{t-1}(k,\mathbf{u}_k) \\
&= \sum_{\mathbf{u}\in V_{\widehat{i}}} \prod_{k\in\widehat{i}} M^{t-1}(k,\mathbf{u}_k)\Big(C_i^t(v \mid \mathbf{u}_{\widehat{i}})\Big) \\
&= M^t(i,v)
\end{aligned}
$$

The second factor above is just a product of 1's (by the definition of marginals) and the proposition follows. □

As B^0 is accurate by definition, M^1 will also be accurate but not necessarily B^1. Let the Marginal distribution $(M^t(i))$ computed for each variable i at time t by FF be the set comprising elements $M^t(i,u)$ for $u \in V$. FF generates in one sweep the sequence of (approximate) marginal distribution vectors $(M^0(1), M^0(2), \ldots, M^0(n))$ $(M^1(1), M^1(2), \ldots, M^1(n))$ $\ldots$ $(M^T(1), M^T(2), \ldots, M^T(n))$ (for convenience we have assumed that all the rate constants are known). The time complexity of FF is $O(T \cdot n \cdot K^{d+1})$ where $|V| = K$ and d is the maximum over the number of parents that a node can have. Usually d will be much smaller than n and in this sense FF is efficient since its time complexity is linear in n.

4.3 Hybrid Factored Frontier algorithm

It is important to consider improved algorithms for inference in DBNs, we propose the Hybrid factored frontier (HFF) algorithm to this effect. HFF maintains the current belief state as a hybrid entity; for a small number of global states called *spikes*, their current probabilities are maintained. The probability distribution over the remaining states is represented, as in FF, as a product of the marginal probability distributions. The key insight underlying this idea is that when the error produced by one step of the inference algorithm is large for a global state, then either the probability of this state or its estimate must itself be high. If such states are chosen to be the spikes then since the total probability is bounded by 1, the number of spikes at each time point must be small. The main technical component of HFF is to explicitly identify and approximately compute the probabilities of the spikes.

A pleasing feature of HFF is that it is a parameterized version of FF with σ, the number of spikes, being the parameter. When $\sigma = 0$, we get FF and when $\sigma = N$ where N is the total number of global states, we get the exact inference algorithm. Thus by tuning σ, one can gain control over the error behavior. We have derived the single step error bound for HFF, which then also leads to an error analysis for FF. We show that the worst case one step error of HFF is lower than that of FF. The time complexity of HFF is $O(n \cdot (\sigma^2 + K^{D+1}))$ where n is the number of nodes in the DBN, σ is the number of spikes, K is the maximum number of values that a random variable (associated with each node) can assume and D is the maximum number of parents that a node can have. This compares favorably with the time complexity of FF which is $O(n \cdot K^{D+1})$. Since the running time of HFF is *linear* in n, it scales well in terms of network size. The factor D is determined by the maximum number of reactions that a species takes part in as a product or reactant. For most of the networks we have encountered, D is much smaller than n.

A simple but crucial observation is that whenever the error $max_{\mathbf{u} \in V^n}\{|\widehat{B}^t(\mathbf{u}) - B^t(\mathbf{u})|\}$ incurred by FF at step $t > 0$ (ignoring the error made in the previous steps) is large for some $\mathbf{u}$ then $M^t(i, \mathbf{u}_i)$ is large for *every* i. This is so since, $M^t(j, \mathbf{u}_j) = M_{\widehat{B}^t}(j, \mathbf{u}_j) \geq \max(\widehat{B}^t(\mathbf{u}), B^t(\mathbf{u}))$, which follows from Proposition 1 and the definition of marginals.

A second important observation is that there can only be a few instances of $\mathbf{u}$ such that $M^t(i, \mathbf{u}_i)$ is large for every i. For instance, there can be only one such $\mathbf{u}$ if we want $M^t(i, \mathbf{u}_i) > \frac{1}{2}$ for every i. Hence, by computing $\widehat{B}^t(\mathbf{u})$ for a small subset of V^n for which M^t is high for all dimensions and maintaining it explicitly, one can hope to reduce the one step error incurred FF and hence the overall error too. This is the intuition underlying the HFF algorithm.

4.3.1 The Hybrid Factored Frontier algorithm

The overall structure of HFF is as follows. Starting with $t = 0$, we inductively compute and maintain the tuple $(M^t, S^t, B^t_H, \alpha^t)$, where:

- M^t is a marginal function.
- $S^t \subseteq V^n$ is a set of tuples called *spikes*.
- $B^t_H : V^n \to [0,1]$ is a function s.t. $B^t_H(\mathbf{u}) = 0$ if $\mathbf{u} \notin S^t$ and $\sum_{\mathbf{u}\in S^t} B^t_H(\mathbf{u}) < 1$.
- $\alpha^t = \sum_{\mathbf{u}\in S^t} B^t_H(\mathbf{u})$.

This hybrid state $(M^t, S^t, B^t_H, \alpha^t)$ represents the following belief state B^t:

$$B^t(\mathbf{u}) = B^t_H(\mathbf{u}) + (1-\alpha^t)\prod_i M^t_H(i,\mathbf{u}_i), \text{ where}$$

$$M^t_H(i,v) = [M^t(i,v) - \sum_{\{\mathbf{u}\in S^t \mid \mathbf{u}_i = v\}} B^t_H(\mathbf{u})]/(1-\alpha^t)$$

The first component of $B^t(\mathbf{u})$ is the probability mass $B^t_H(\mathbf{u})$ of the spike (if $\mathbf{u}$ is not a spike, $B^t_H(\mathbf{u}) = 0$). The second component is the product of (uniformized) marginals $M^t_H(i, v)$, as in FF. Notice that we need to use M^t_H rather than M^t since the cumulative weight of the contribution made by the spikes needs to be discounted from M^t. The coefficient $(1 - \alpha^t)$ must be used first to ensure that M^t_H is a marginal function, and second to ensure that B^t is a belief state, as will be demonstrated subsequently.

The HFF algorithm

We initialize with $M^0 = C^0$, $S^0 = \emptyset$, $B^0_H = \mathbf{0}$ and $\alpha^0 = 0$ and fix a parameter σ. This σ will be the number of spikes we choose to maintain. It is a crucial parameter as our results will show. We inductively compute $(M^{t+1}, S^{t+1}, B^{t+1}_H, \alpha^{t+1})$ from $(M^t, S^t, B^t_H, \alpha^t)$ as follows.

Step 1: We first compute M^{t+1} as:

$$\begin{aligned} M^{t+1}(i,x) = \sum_{\mathbf{u}\in S^t}[B_H^t(\mathbf{u}) \times C_i^{t+1}(x \mid \mathbf{u}_{\hat{i}})] \\ +(1-\alpha^t)\Big(\sum_{\mathbf{u}_{\hat{i}}}[\prod_{j\in\hat{i}} M_H^t(j,\mathbf{u}_j)\times C_i^{t+1}(x\mid \mathbf{u}_{\hat{i}})]\Big) \end{aligned}$$

Step 2: We next compute a set S^{t+1} of at most σ spikes using M^{t+1}. We want to consider as spikes $\mathbf{u} \in V^n$ where $M^{t+1}(i, \mathbf{u}_i)$ is large for every i. To do so, we find a constant η^{t+1} such that $M^{t+1}(i, \mathbf{u}_i) \geq \eta^{t+1}$ for every i for a subset of V^n containing σ elements and for all other $\mathbf{u'}$, there exists i with $M^{t+1}(i, \mathbf{u}'_i) < \eta^{t+1}$. We compute η^{t+1} via binary search. First we fix the precision with which we want to compute η^{t+1} to be ξ. We have found $\xi = 10^{-6}$ to be a good choice. For this choice there will be at most 20 iterations of the loop described below. The search for η^{t+1} proceeds as follows:

```
• $\eta_1 = 0$ and $\eta_2 = 1$.
• While $\eta_2 - \eta_1 > \xi$ do
    1. $\eta = \frac{\eta_1+\eta_2}{2}$.
    2. Determine the set of values $U_i$ such that $v \in U_i$ iff $M^{t+1}(i, v) > \eta$.
    3. Set $a_i$ to be the cardinality of $U_i$.
    4. If $\prod_i (a_i) > \sigma$ then $\eta_1 = \eta$; otherwise $\eta_2 = \eta$
• endwhile
• Return $\eta^{t+1} = \eta_2$ and $S^{t+1} = \prod_i U_i$
```

Step 3: Finally, we compute $B_H^{t+1}(\mathbf{u})$ for each $\mathbf{u}$ in S^{t+1} as follows, by only taking into account the contribution of the current spikes.

$$B_H^{t+1}(\mathbf{u}) = \sum_{\mathbf{v}\in S^t}(B^t(\mathbf{v}) \times \prod_i C_i^{t+1}(\mathbf{u}_i \mid \mathbf{v}_{\hat{i}}))$$

End of Algorithm

As in the case of FF, we denote by $\widehat{B}^{t+1}$ the belief state obtained from B^t through an exact step of the DBN:

$$\widehat{B}^{t+1}(\mathbf{u}) = \sum_{\mathbf{v}\in V^n} (B^t(\mathbf{v}) \times \prod_i C_i^{t+1}(\mathbf{u}_i \mid \mathbf{v}_{\widehat{i}}))$$

Notice that $B_H^{t+1}(\mathbf{u}) \leq \widehat{B}^{t+1}(\mathbf{u})$ for all $\mathbf{u}$. We recall that T is the number of time points, σ the number of spikes, n the number of variables, V is the set of values with $K = |V|$, and D be the maximum in-degree of the DBN graph.

Theorem 2. *HFF has the following properties.*

1. *if $\sigma = 0$, the HFF algorithm is the same as FF and if $\sigma = K^n$, it is the exact algorithm.*

2. *$M^t(i,v) = M_{\widehat{B}^t}(i,v)$ for every v. Further, B^t is a belief state while M_H^t and M^t are marginal functions, for every t.*

3. *The time complexity of HFF is $O(T \cdot n \cdot (\sigma^2 + K^{D+1}))$.*

Proof. $\sigma = 0$ implies that the set of spikes $S^t = \emptyset$ for all t. This implies that $\alpha^t = 0$ and the computation done by HFF is the same as FF. If $\sigma = K^n$, then $S^t = V$ for all t and $\alpha^t = 1$ (of course, M_H^t is then not computed). Thus, this boils down to perform exact inferencing. We have now established part (1).

We prove that for all $t \geq 1$, if B^{t-1} is a belief state and M^{t-1}, M_H^{t-1} are marginals, then $M^t = M_{\widehat{B}^t}$ and B^t is a belief state and M^t, M_H^t are marginals. We thus obtain Part (2) by induction on t, using the fact that B^0 is a belief state and M^0, M_H^0 are marginals by definitions. For $t \geq 0$, let $M^t(i,v), M_H^t(i,v)$ be marginals and B^t be a belief state. Then at $t+1$, let us start by proving $M_{\widehat{B}^{t+1}}(i,v) = M^{t+1}(i,v)$. The first step is the same as in Proposition 1:

$$\begin{aligned} M_{\widehat{B}^{t+1}}(i,v) &= \sum_{\mathbf{v}\mid\mathbf{v}_i=v} \widehat{B}^{t+1}(\mathbf{v}) \\ &= \sum_{\mathbf{u}} B^t(\mathbf{u}) \Big(C_i^{t+1}(v \mid \mathbf{u}_{\widehat{i}}) \Big) (\text{by Equation (4.2)}) \end{aligned}$$

Now however, the definition of B^t is different for HFF and so we have from (4.3) above:

$$\begin{aligned} M_{\widehat{B}^{t+1}}(i,v) = \sum_{\mathbf{u}} B_H^t(\mathbf{u}) \Big(C_i^{t+1}(v \mid \mathbf{u}_{\widehat{i}}) \Big) \\ +(1-\alpha^t) \sum_{\mathbf{u}} \Big(\prod_{k=1}^{n} M_H^t(k,\mathbf{u}_k) \Big) \Big(C_i^{t+1}(v \mid \mathbf{u}_{\widehat{i}}) \Big) \end{aligned} \tag{4.3}$$

But if $\mathbf{u} \notin S^t$, then $B_H^t(\mathbf{u}) = 0$. Further splitting the second term as in Proposition 1, we obtain:

$$M_{\widehat{B}^{t+1}}(i, v) = \sum_{\mathbf{u} \in S^t} B_H^t(\mathbf{u}) \Big(C_i^{t+1}(v \mid \mathbf{u}_{\widehat{i}}) \Big)$$

$$+ (1 - \alpha^t) \sum_{\mathbf{u} \in V_{\widehat{i}}} \sum_{\mathbf{u} \notin V_{\widehat{i}}} \prod_k M_H^t(k, \mathbf{u}_k) \Big(C_i^{t+1}(v \mid \mathbf{u}_{\widehat{i}}) \Big)$$

$$= \sum_{\mathbf{u} \in S^t} B_H^t(\mathbf{u}) \Big(C_i^{t+1}(v \mid \mathbf{u}_{\widehat{i}}) \Big) + (1 - \alpha^t)$$

$$\sum_{\mathbf{u} \in V_{\widehat{i}}} \prod_{k \in \widehat{i}} M_H^t(k, \mathbf{u}_k) \Big(C_i^{t+1}(v \mid \mathbf{u}_{\widehat{i}}) \Big) \sum_{\mathbf{u} \notin V_{\widehat{i}}} \prod_{k \notin \widehat{i}} M_H^t(k, \mathbf{u}_k)$$

$$= \sum_{\mathbf{u} \in S^t} B_H^t(\mathbf{u}) \Big(C_i^{t+1}(v \mid \mathbf{u}_{\widehat{i}}) \Big) + (1 - \alpha^t)$$

$$\sum_{\mathbf{u} \in V_{\widehat{i}}} \prod_{k \in \widehat{i}} M_H^t(k, \mathbf{u}_k) \Big(C_i^{t+1}(v \mid \mathbf{u}_{\widehat{i}}) \Big) \prod_{k \notin \widehat{i}} \sum_{\mathbf{u}_k} M_H^t(k, \mathbf{u}_k)$$

$$= \sum_{\mathbf{u} \in S^t} B_H^t(\mathbf{u}) \Big(C_i^{t+1}(v \mid \mathbf{u}_{\widehat{i}}) \Big) + (1 - \alpha^t)$$

$$\sum_{\mathbf{u} \in V_{\widehat{i}}} \prod_{k \in \widehat{i}} M_H^t(k, \mathbf{u}_k) \Big(C_i^{t+1}(v \mid \mathbf{u}_{\widehat{i}}) \Big) \times 1$$

$$= M^{t+1}(i, v)$$

In the step above, $\sum_{\mathbf{u}_k} M_H^t(k, \mathbf{u}_k) = 1$ follows from our inductive hypothesis that M_H^t is a marginal. Now, we will prove the remainder of this part, i.e., M^{t+1}, M_H^{t+1} are marginals and B^{t+1} is a belief state. For all i, again from $\alpha^t = \sum_{\mathbf{u} \in S^t} B_H^t(\mathbf{u})$ and

$\sum_{v \in V} C_i^{t+1}(v \mid \mathbf{u}_{\hat{i}}) = 1$, we have:

$$\begin{aligned}
\sum_{v \in V} M^{t+1}(i,v) &= \sum_{\mathbf{u} \in S^t} [\sum_{v \in V} C_i^{t+1}(v \mid \mathbf{u}_{\hat{i}}) \times B_H^t(\mathbf{u})] \\
&\quad +(1-\alpha^t)\Big(\sum_{\mathbf{u}_{\hat{i}}} [\sum_{v \in V} C_i^{t+1}(v \mid \mathbf{u}_{\hat{i}}) \times \prod_{j \in \hat{i}} M_H^t(j, \mathbf{u}_j)]\Big) \\
&= \sum_{\mathbf{u} \in S^t} B_H^t(\mathbf{u}) + (1-\alpha^t) \sum_{\mathbf{u}_{\hat{i}}} \prod_{j \in \hat{i}} M_H^t(j, \mathbf{u}_j) \\
&= \alpha^t + (1-\alpha^t) \prod_{j \in \hat{i}} \sum_{\mathbf{u}_j} M_H^t(j, \mathbf{u}_j) = 1
\end{aligned}$$

Now, using the above and $\alpha^{t+1} = \sum_{\mathbf{u} \in S^{t+1}} B_H^{t+1}(\mathbf{u})$ (assuming $\alpha^{t+1} \neq 1$), we have:

$$\begin{aligned}
&\sum_{v \in V} M_H^{t+1}(i,v) = \\
&\Big(\sum_{v \in V} M^{t+1}(i,v) - \sum_{v \in V} \sum_{\mathbf{u} \in S^{t+1} | \mathbf{u}_i = v} B_H^{t+1}(\mathbf{u})\Big) \times \frac{1}{1-\alpha^{t+1}} \\
&= \Big(1 - \sum_{\mathbf{u} \in S^{t+1}} B_H^{t+1}(\mathbf{u})\Big) \frac{1}{1-\alpha^{t+1}} = 1
\end{aligned}$$

$$\begin{aligned}
\sum_{\mathbf{u} \in V^n} B^{t+1}(\mathbf{u}) &= \sum_{\mathbf{u} \in V^n} B_H^{t+1}(\mathbf{u}) \\
&\quad + (1-\alpha^{t+1}) \sum_{\mathbf{u} \in V^n} \prod_i M_H^{t+1}(i, \mathbf{u}_i) \\
&= \sum_{\mathbf{u} \in S^{t+1}} B_H^{t+1}(\mathbf{u}) + (1-\alpha^{t+1}) \times 1 = 1
\end{aligned}$$

It now follows easily that for any i, v, $1 \geq M^{t+1}(i,v) \geq 0$ and $1 \geq M_H^{t+1}(i,v)$. It remains to prove that $M_H^{t+1}(i,v) \geq 0$, that is $M^{t+1}(i,v) \geq \sum_{\mathbf{u} \in S^{t+1} | \mathbf{u}_i = v} B_H^{t+1}(\mathbf{u})$. As $B_H^{t+1}(\mathbf{u}) \leq \widehat{B}^{t+1}(\mathbf{u})$ for all $\mathbf{u}$, we have,

$$\begin{aligned}
\sum_{\mathbf{u} \in S^{t+1} | \mathbf{u}_i = v} B_H^{t+1}(\mathbf{u}) &\leq \sum_{\mathbf{u} \in S^{t+1} | \mathbf{u}_i = v} \widehat{B}^{t+1}(\mathbf{u}) \\
&= M_{\widehat{B}^{t+1}}(i,v) = M^{t+1}(i,v)
\end{aligned}$$

which completes the proof of part(2).

Turning to part (3), we note that at each time point, the step 1 of HFF has the same complexity as FF together with the spikes contributing: $O(K \cdot n \cdot (K^D + \sigma))$. Step 2 makes at most $K \times n$ comparisons for each iteration of the loop and there are only a constant number of iterations of the loop. Thus the complexity of this step per time point is $O(K \times n)$. Step 3 computes for each spike, $B^t(u)$ from the values of $B^t_H(u)$ and $M^t(i, u)$ which takes $O(Kn + \sigma n)$. Then, we sum over all the spikes the value computed by multiplying n values of the CPT which takes $O(\sigma \times n)$. Thus, this step overall takes $O(\sigma K n + n\sigma^2)$. Hence the overall time complexity of HFF is the sum of all these quantities which is $O(T \cdot n \cdot (K^{D+1} + K\sigma + \sigma^2))$ which is bounded by $O(T \cdot n \cdot (K^{D+1} + \sigma^2))$. □

HFF gathers in one sweep -just as FF does- the required information about the belief states. However, it can take more time than FF depending on the number of spikes but the added complexity is only quadratic in the number of spikes.

4.3.2 Error analysis

It is easy to see that with each time slice t of the DBN one can associate a stochastic matrix $\mathcal{T}_t$. This stochastic matrix will capture the transformation of probability distributions effected by the n CPTs associated with the time slice t as dictated by Equation 4.1. In particular, we will have $P(X^t) = \mathcal{T}_t(P(X^{t-1}))$.

We now denote the cumulative error at t as Δ^t and define it to be: $\Delta^t = max_{\mathbf{u} \in V^n}(|P(X^t = \mathbf{u}) - B^t(u)|)$. Towards deriving an upper bound for Δ^t, we first note that Markov chain theory (for instance, using the Dobrushin's coefficient, see chapter 6.7 in [115]) guarantees the following:

Theorem 3. *Let $\mathcal{T}$ be an n-dimensional stochastic matrix. Then for two probability distributions A, B, we have $||\mathcal{T}(A) - \mathcal{T}(B)||_\infty \leq \beta_{\mathcal{T}} ||A - B||_\infty$ where $0 \leq \beta_{\mathcal{T}} \leq 1$ is a constant that depends only on $\mathcal{T}$.*

$\beta_{\mathcal{T}}$ is called the *contraction factor*. In what follows we shall write β^t for the contraction factor associated with $\mathcal{T}_t$ and set $\beta = \underset{t}{max}\ \beta^t$.

An implicit assumption in what follows is that $\beta < 1$. As pointed out in [15] this is a very reasonable assumption since it fails for the extreme case where the variables are completely decoupled and are independent. The case studies we report in section 4.4 also easily satisfy this assumption. When $\beta < 1$, due to the theorem above the maximum error will reduce by a factor of β at each step as we step through t starting from $t = 0$. Hence the *cumulative* error will stabilize rapidly.

Now following a reasoning similar to [15] we shall show that Δ^t can be bounded by $\epsilon_0(\sum_{j=0}^{t}\beta^j)$ where ϵ_0 is the maximum one step error given by: $\epsilon_0 = \underset{t}{max}||B^t - \mathcal{T}_t(B^{t-1})||_\infty$. Notice that $\mathcal{T}_t(B^{t-1})$ was denoted as $\widehat{B}^t$ in previous subsections.

Lemma 4. $\Delta^t \leq \epsilon_0(\sum_{j=0}^{t}\beta^j)$. *Further if* $\beta < 1$, *we have* $\Delta^t \leq \frac{\epsilon_0}{1-\beta}$.

Proof. By definition of overall error and the above stated property of Markov chains,

$$\begin{aligned}\Delta^t &= |B^t - P(X^t)| \\ &\leq |B^t - \mathcal{T}_t(B^{t-1})| + |\mathcal{T}_t(B^{t-1}) - \mathcal{T}_t(P(X^{t-1}))| \\ &\leq \epsilon_0 + \beta_t\Delta^{t-1}\end{aligned}$$

Then by recursively computing the second factor, we obtain,

$$\begin{aligned}\Delta^t &\leq \epsilon_0 + \beta_t\epsilon_0 + \beta_t\beta_{t-1}\epsilon_0 + \ldots + (\beta_t\beta_{t-1}\cdots\beta_1)\epsilon_0 \\ &\leq \epsilon_0(\sum_{j=0}^{t}\beta^j)\end{aligned}$$

Further if $\beta < 1$, we have:

$$\Delta^t \leq \epsilon_0(\sum_{j=0}^{t}\beta^j) \leq \epsilon_0(\sum_{j=0}^{\infty}\beta^j) = \frac{\epsilon_0}{1-\beta}$$

□

We note that $\sum_{j=0}^{t}\beta^j$ depends only on the DBN. Hence, theoretically comparing the error behaviors of FF and HFF amounts to comparing their single step errors. To do so, we shall next analyze single step error of FF followed by that of HFF.

Recall that for FF, $B^t = B_{M_{\widehat{B}^t}}$. Thus the one-step error incurred by FF at step t is $max_{\mathbf{u}\in V^n}\{|\widehat{B}^t(\mathbf{u}) - B_{M_{\widehat{B}^t}}(\mathbf{u})|\}$. We can bound this from above by ϵ_0 where : $\epsilon_0 = max\{|B(\mathbf{u}) - B_{M_B}(\mathbf{u})|\}$ with B ranging over the set of all possible belief states and $\mathbf{u}$ ranging over V^n. It turns out that ϵ_0 can be made arbitrarily close to 1 as n, the number of variables, tends to ∞. To see this, fix $0 < \delta < 1$ and consider the belief state B

defined by $B(\mathbf{u}) = 1 - \delta$, $B(\mathbf{u}') = \delta$ for some $\mathbf{u}, \mathbf{u}' \in V^n$ such that for all i, $\mathbf{u}_i \neq \mathbf{u}_j$ and $B(\mathbf{v}) = 0$ for all $\mathbf{v} \in V^n \setminus \{\mathbf{u}, \mathbf{u}'\}$. Then, $M_B(i, \mathbf{u}_i) = 1 - \delta$ for all $i \in \{1, \ldots, n\}$ and so $B_{M_B}(\mathbf{u}) = (1-\delta)^n$. As a result we have $\epsilon_0 = max|B - B_{M_B}| \geq (1-\delta) - (1-\delta)^n$ which tends to $1 - \delta$ as n tends to ∞. Now if we choose δ to be close to 0, $1 - \delta$ is close to 1. Thus ϵ_0 can be made as close to 1 as we want, with n tending to ∞. We found that the cumulative errors made by FF can be large in practice too as shown in the next section.

Notice that for HFF too we have $B^t = B_{M_{\widehat{B}^t}}$, and its one step error can be bound by ϵ_0. However, the spikes can be used to bound the single step error of HFF more precisely as follows:

Claim 1. *The one step error made by HFF is bounded by* $\hat{\epsilon}_0$ *with* $\hat{\epsilon}_0 \leq \min\{(1-\alpha), \eta\}$, *where* $\alpha = \min_t(\alpha^t)$ *and* $\eta = \max_t(\eta^t)$.

(Proof sketch). If α is large, then the value of $\widehat{B}^t(\mathbf{u}) \leq 1 - \alpha$ for $\mathbf{u} \notin S^t$. Also, as $B^t(\mathbf{u}) \leq \widehat{B}^t(\mathbf{u})$, we have $\widehat{B}^t(\mathbf{u}) - B^t(\mathbf{u}) \leq 1 - \alpha$. Finally, if η is small, then by construction for all $\mathbf{u} \notin S^t$, $M^{t+1}(i, v) \leq \eta$ for some i with $\mathbf{u}_i = v$, and hence $\widehat{B}^t(\mathbf{u}) \leq M^{t+1}(i, v) \leq \eta$. Also, $\widehat{B}^t(\mathbf{u}) - B^t(\mathbf{u}) \leq \eta \times \sum_{\mathbf{v} \notin S^t} \prod_i C_i^{t+1}(\mathbf{u}_i \mid \mathbf{v}_{\hat{i}})) \leq \eta$ for $\mathbf{u} \in S^t$. □

Thus, the worst case error for HFF with at least two spikes (implying $\eta < 1/2$) is smaller than for FF. Taking more spikes will increase α and decrease η , reducing the worst case error. Experiments in the next section show that the practical accuracy is also improved as we increase the number of spikes.

4.4 Experimental evaluation

We have implemented our algorithm in C++. The experiments reported here were carried out on an Opteron 2.2Ghz processor, with 3GB memory. The algorithms were evaluated on five DBN models of biochemical networks: the small enzyme catalytic reaction network shown in figure 3.2 for initial experimentation, the EGF-NGF pathway [116] under (a) EGF-stimulation (b) NGF-stimulation (c) co-stimulation of EGF and NGF, and the Epo mediated ERK signaling pathway. The ODE model for the EGF-NGF pathway was obtained from the BioModels database [117] and the Epo mediated ERK signaling pathway from [118]. For all these models, there were no unknown parameters and this enabled us to focus on the main issue of evaluating the performance of HFF. The DBNs

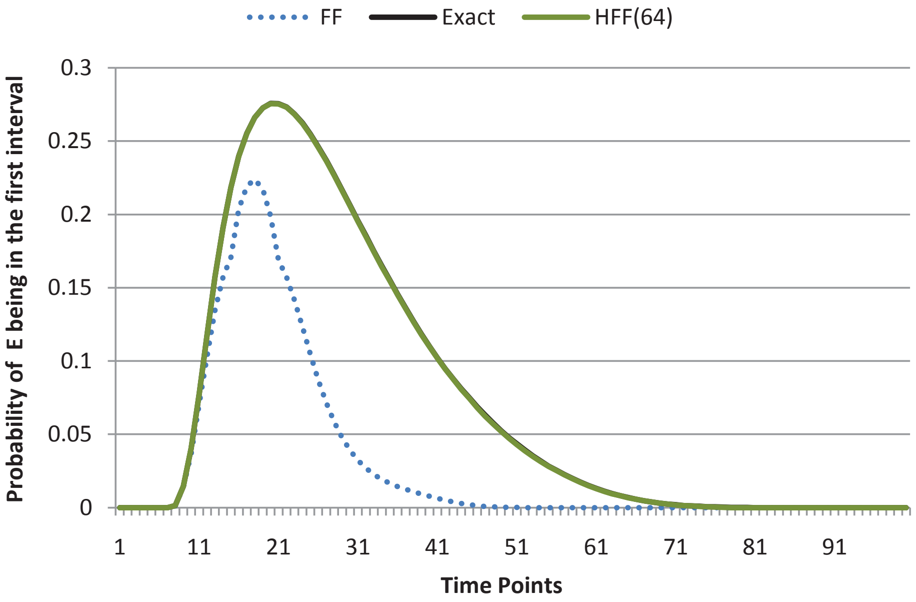


Figure 4.1: Marginal probability of E being in the interval $[0, 1)$, $M^t(E \in [0, 1))$

were constructed using the method presented in previous chapters [12]. To improve the quality of the approximations for the large pathway models, we constructed the DBNs using the *equation based subinterval sampling* method explained in more detail later. In what follows, we highlight the main findings of our experiments.

4.4.1 Enzyme catalytic kinetics

For initial validation, we started with the enzyme catalytic reaction network shown in figure 3.2 which has only 4 species/equations and 3 rate constants. The value space of each variable was divided into 5 equally wide intervals ($\{[0, 1), [1, 2), \ldots, [4, 5]\}$). We assumed the initial distributions of variables to be uniform over certain intervals. We then fixed the time horizon of interest to be 10 minutes and divided this interval evenly into $[0, 1, \ldots, 100]$ time points. The conditional probability tables associated with each node of the DBN were filled by generating 10^6 trajectories by direct random sampling over the initial states [12].

This being a small example, we could compute the marginal distributions for each species exactly. We ran FF and HFF(σ) with various choices of σ, the number of spikes. The resulting estimates were then compared against the exact marginals. We also ran the fully factored version of BK (which we call BK in this section), using the implementation provided in the Bayes Net Toolbox of MATLAB [119].

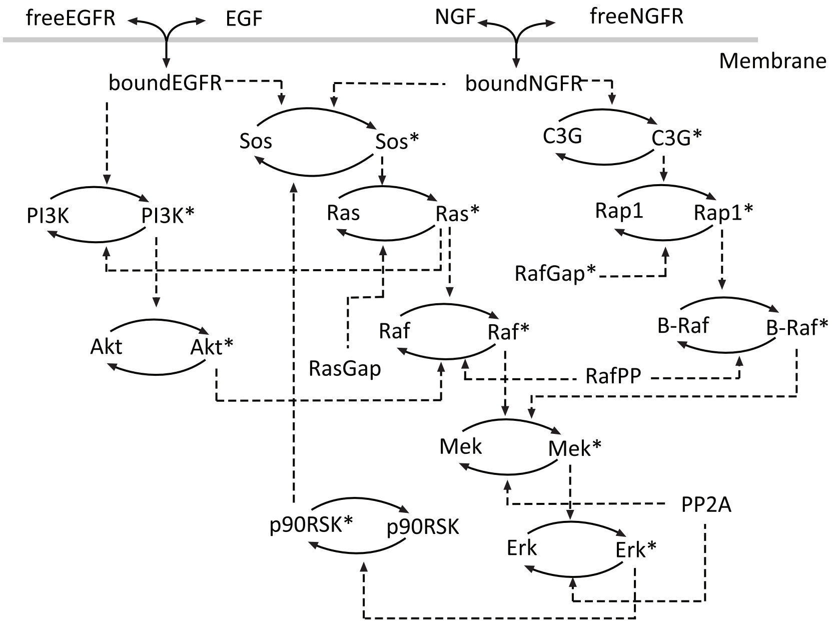


Figure 4.2: EGF-NGF pathway

In what follows we report the errors in terms of the absolute difference between the marginal probabilities computed by the exact and approximate methods. Thus if we say the error is 0.15 then this means that the actual marginal probability was p and the marginal probability computed by the approximate algorithm was p' with $|p - p'| = 0.15$.

Even for this small network, FF and BK deviated from some of the exact marginals by as much as 0.169. Figure 4.1 shows the profile of the marginal distribution of E (the enzyme) assuming a value in the first interval as computed by FF, BK, HFF(64) and the exact method. The profiles of exact and HFF(64) were almost the same while FF and BK (whose curve practically coincides with that of FF and is hence not shown) make noticeable errors. The computation times for all the algorithms were negligible. The maximum error incurred for the 4 species taken over all the interval values and all time points was 0.169 for FF and 0.024 for HFF(16) and 0.003 for HFF(64). Further, the number of errors greater than 0.1 taken over all the species, intervals and time points reduced from 72 for FF to 0 for HFF(16). Finally, we compute the L1 error across all marginals - per time point - between exact marginals and the one's compute by different approximation algorithms. Figure 4.4 shows the plots of L1 error between the various algorithms across every time point, it shows that the L1 error is high for FF compared to HFF which further reduced with increasing number of spikes.

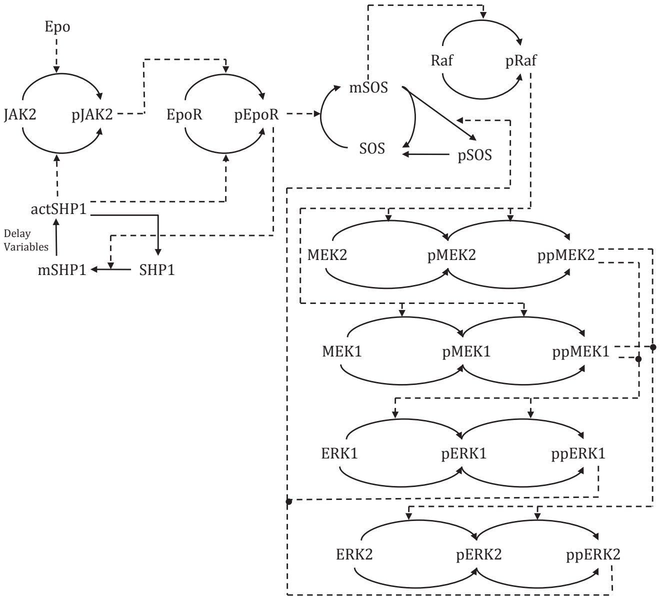


Figure 4.3: Epo mediated ERK Signaling pathway

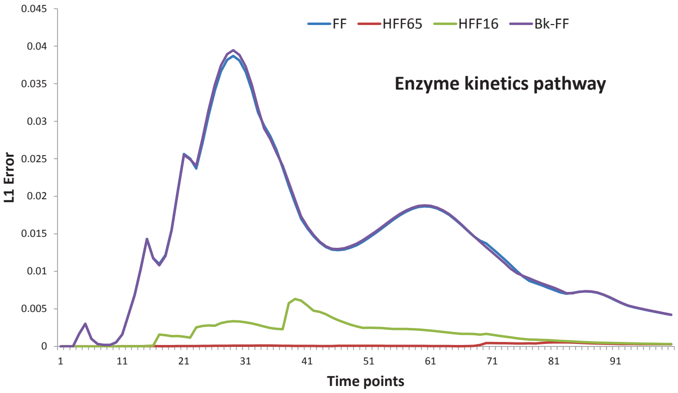


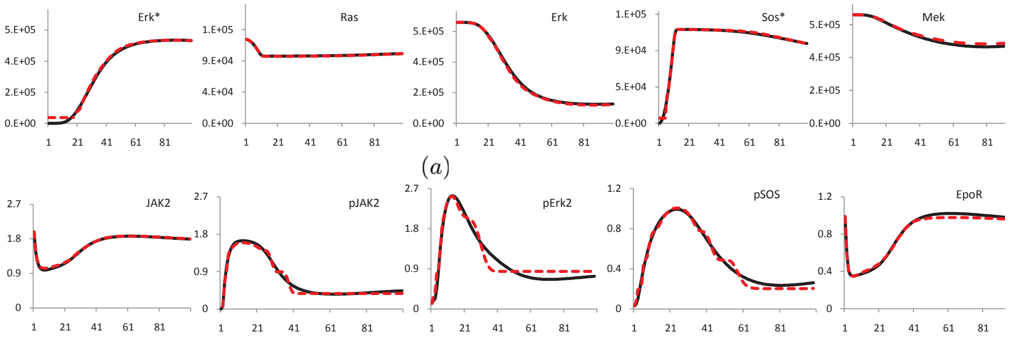
Figure 4.4: L1 error vs time points : Enzyme catalytic pathway

4.4.2 The large pathway models

As explained before, during the construction of the DBN we assume that the initial values are distributed along certain predefined intervals of a variable's value space. The vector of initial states for large systems will hence be high dimensional. To ensure that the ODE dynamics is well explored, one needs to draw a large number of representative trajectories. Naive direct sampling where we randomly pick values from the initial intervals vector cannot ensure that all parts of the initial states region are sufficiently probed. Hence we used a more sophisticated sampling method called *equation based subinterval sampling* which is a variant of the method proposed in [12]. Suppose the ODE equation for the variable x_i involves variables x_j and x_k. We then subdivide the initial intervals of the variables x_i, x_j and x_k into J finer subintervals. Then for *every combination of subintervals* say, (I_i, I_j, I_k), we pick H samples each of which will have its x_i-value falling in I_i, x_j-value falling in I_j and its x_k-value falling in I_k while the values for the other variables are picked randomly from within their initial intervals. This ensures a coverage of at least H samples for every combination of the subintervals of the variables governing each equation which in turn ensures that ODE dynamics is being explored systematically along each dimension at least. In general, if an equation has R variables on its right hand side, and there are n equations and H is the required degree of coverage per equation, we pick $n \cdot H \cdot J^{R+1}$ samples.

To assess the quality of the constructed DBNs in terms of the original ODE dynamics, we used Monte Carlo integration to generate random trajectories from the prior (initial states distribution) using the ODE. We then computed the average values of each variable at the time points $0 \leq t \leq T$. We term the resulting time series for each variable as a *nominal profile*. We then used marginal probability values derived from the DBN approximation to compute expected values as follows $E(M^t(i,u)) = \sum_{u=u_j}(M^t(i,u_j) \cdot L)$, where L is the mid-point of the interval u_j. For each variable, the resulting time series of expected values was compared with its nominal profile. For all the models studied below the quality of the DBN approximation measured this way was high. Due to space limitations, the comparison plots will be shown in what follows here only for a few chosen species in the case of the NGF stimulated EGF-NGF pathway and the Epo mediated ERK pathway.

Finally, for the DBNs arising from EGF-NGF pathway and Epo mediated ERK



(*a*)

(*b*)

Figure 4.5: Comparison of ODE dynamics with DBN approximation. Solid black line represents nominal ODE profiles and dashed red lines represent the DBN simulation profiles for (a) NGF stimulated EGF-NGF Pathway (b) Epo mediated ERK pathway

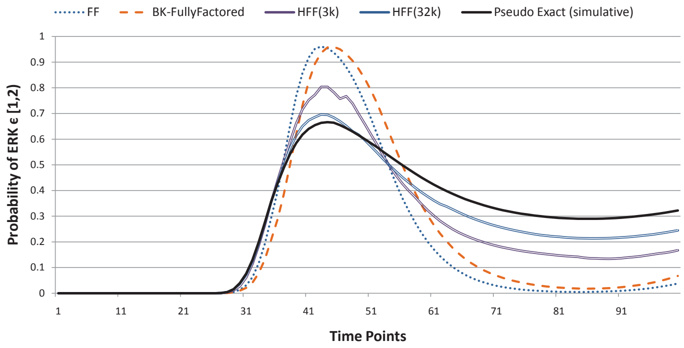


Figure 4.6: Marginal probability of Erk being in the interval $[1, 2)$, $M^t(Erk \in [1, 2))$, under NGF-stimulation

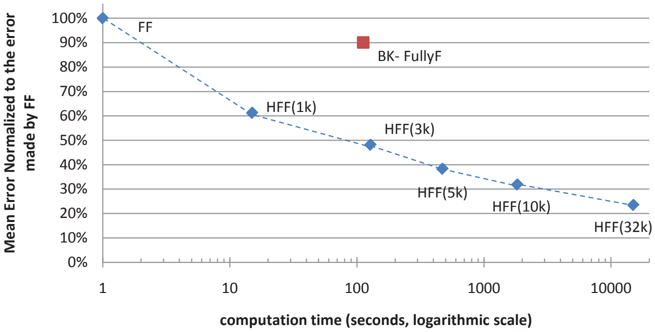


Figure 4.7: Normalized mean error for $M^t(Erk \in [1, 2))$ under NGF-stimulation.

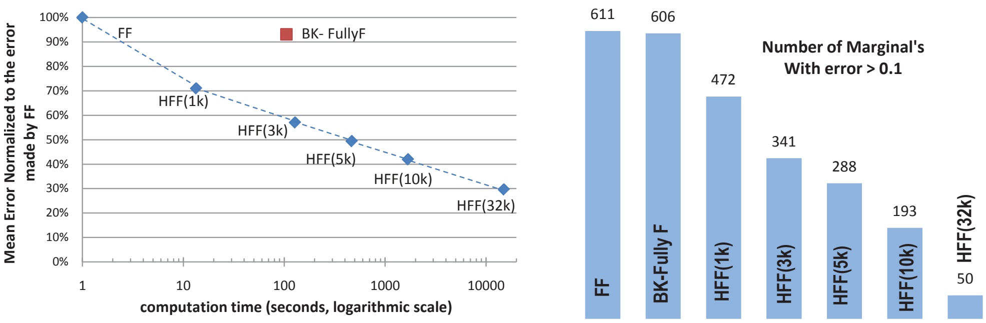


Figure 4.8: (a) Normalized mean errors over all marginals, (b) Number of marginals with error greater than 0.1: NGF-stimulation

pathway, exact inference is infeasible due to the large sizes of the corresponding DBNs. To get around this, we used simulation based inferencing of the DBN to obtain an estimate of the exact marginal distribution. These marginals were used -in place of exact marginals- as benchmarks to compare the performance of the various algorithms. Here again we compared HFF(σ) for various choices of σ with FF and BK. We discuss towards the end of this section the performance of the clustered version of BK. In what follows, we write HFF(cK) to mean the HFF(σ) with $\sigma = c \cdot 1000$.

The EGF-NGF pathway

The EGF-NGF pathway describes the behavior of PC12 cells under multiple stimulations. In response to EGF stimulation they proliferate but differentiate into sympathetic neurons in response to NGF stimulation. This phenomenon has been intensively studied [120] and the network structure of this pathway is as shown in figure 4.2. The ODE model of this pathway [117] consists of 32 differential equations and 48 associated rate constants (estimated from multiple sets of experimental data as reported in [117]).

To construct the three DBNs arising out of EGF, NGF and co-stimulation, we divided as before the value domains of the variables into 5 equally wide intervals and assumed the initial distributions to be uniformly distributed over some of these intervals. The time horizon of each model was set at 10 minutes which was evenly divided into 100 time points. To fill up the conditional probability tables, we used the equation based subinterval sampling. We subdivided each of the initial states into 4 subintervals. 2.1 million trajectories were generated to get a coverage of 500 per combination of the subinterval. As shown in figure 4.5(a), the quality of the approximations relative to the

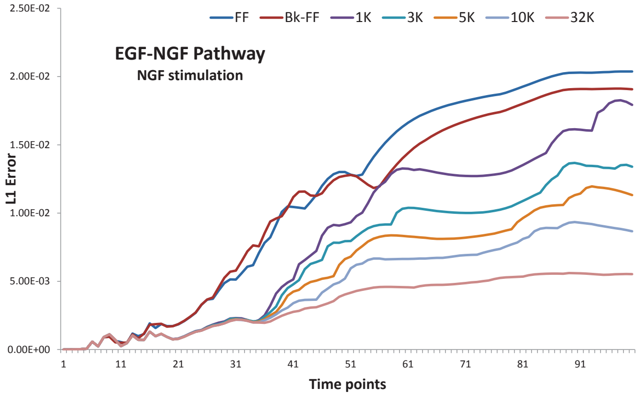


Figure 4.9: L1 error vs time points : NGF-stimulation

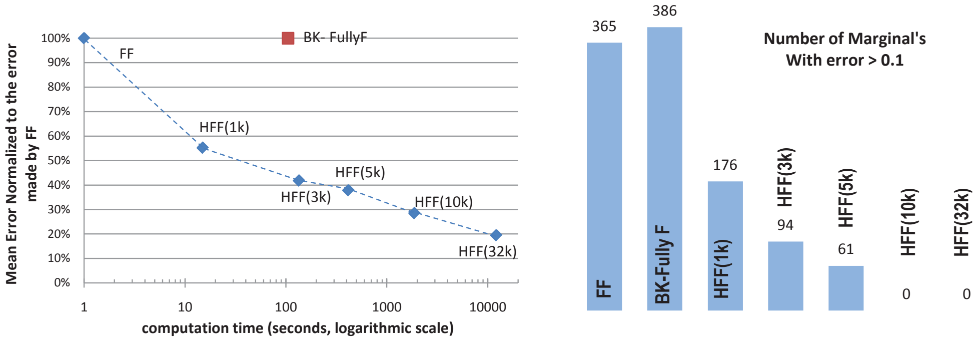


Figure 4.10: (a) Normalized mean error over all marginals (b) Number of marginals with error greater than 0.1: EGF- stimulation

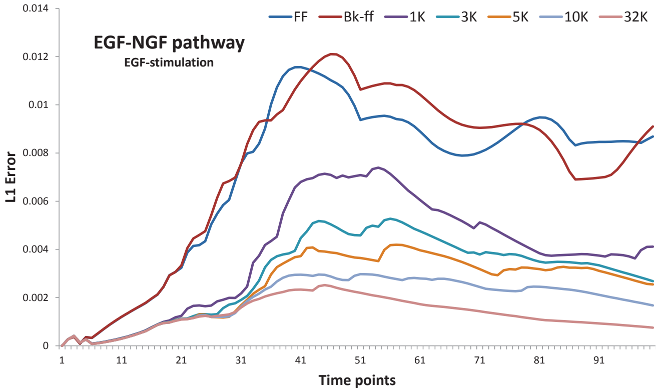


Figure 4.11: L1 error vs time points : EGF-stimulation

original ODE dynamics was high. Once we had the DBNs, we ran FF, BK and HFF(σ) for various choices of σ.

For the DBN obtained for the pathway under NGF-stimulation, for 6 of the 32 species there were significant differences between FF and BK on one hand and HFF on the other, including some biologically important proteins such as *Sos* and *Erk*. In figure 4.6, we show for *Erk*, the marginal probability of the concentration falling in the interval $[1, 2)$ at various time points as computed by FF, BK, HFF($3K$) and HFF($32K$) as well as the pseudo-exact marginals obtained via massive Monte Carlo simulations. We observe that HFF tends to the exact values as the number of spikes increases.

To measure the overall error behavior, noting that HFF always did better than FF, we fixed the error incurred by FF as the base (100%) and normalized all other errors relative to this base. Under this regime, the relationship between computation time and normalized mean error for *Erk*'s value to fall in $[1, 2)$ is shown in figure 4.7. We observe that the mean error reduces to 22% for HFF($32K$) at the cost of approximately 10^4 seconds increase in running time. For HFF(σ) the errors did not decrease linearly as the number of spikes were increased. This is to be expected since the probability mass captured by the additional spikes will be less than what is captured by the initial spikes.

Overall, the maximum error over *all* the marginals ($32 \times 5 \times 100 = 16000$ data points) reduced from 0.42 for FF to 0.3 for HFF($3K$) and to 0.12 for HFF($32K$). The normalized mean error over all marginals went down to 60% for HFF($3K$) and 30% for HFF($32K$) as shown in figure 4.8(a) which also displays the corresponding computation times. Further, when we computed the *number of marginals* with errors greater than 0.1, we found that this number reduced to about half for HFF($3K$) and by more than a factor of 10 for HFF($32K$) compared to FF as shown in figure 4.8(b). We also compute the L1 error across all marginals - per time point - between exact marginals and the one's compute by different approximation algorithms. Figure 4.9 shows the plots of L1 error between the various algorithms across every time point, it shows that the L1 error is high for FF compared to HFF which further reduced with increasing number of spikes.

For the DBN obtained for the pathway under EGF-stimulation we found similar results. Overall, the maximum error over *all* the marginals reduced from 0.35 for FF to 0.14 for HFF($3K$) and to 0.07 for HFF($32K$). The normalized mean error over all marginals went down to 40% for HFF($3K$) spikes and 20% for HFF($32K$) spikes as shown

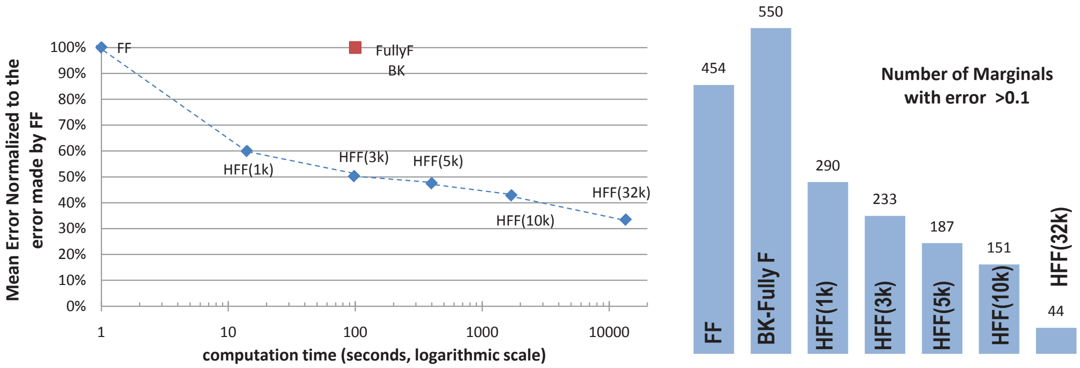


Figure 4.12: (a) Normalized mean error over all marginals (b) Number of marginals with error greater than 0.1: EGF-NGF Co-stimulation

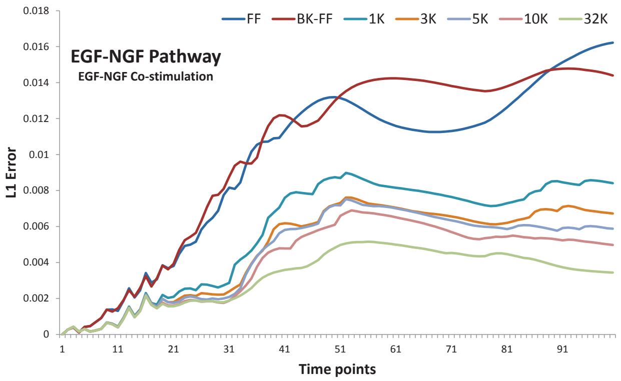


Figure 4.13: L1 error vs time points : EGF-NGF Co-stimulation

in figure 4.10(a) which also displays the corresponding computation times. Further, when we computed the number of marginals with errors greater than 0.1, we found that this number reduced by more than a factor of 4 for HFF($3K$) and to 0 for HFF($32K$) as shown in figure 4.10(b). Similar results were obtained for the DBN describing the dynamics of the EGF-NGF pathway under co-stimulation of both NGF and EGF as shown in figure 4.12. Figures 4.11 and 4.13 show the plots of L1 error between the various algorithms across every time point for the DBNs obtained for EGF- stimulation and EGF-NGF-co-stimulation respectively, it shows that the L1 error is high for FF compared to HFF which further reduced with increasing number of spikes.

The Epo mediated ERK pathway

Next we considered the DBN model of Epo mediated ERK signaling pathway as shown in figure 4.3. *Erk* and its related kinase isoforms play a crucial role in cell proliferation,

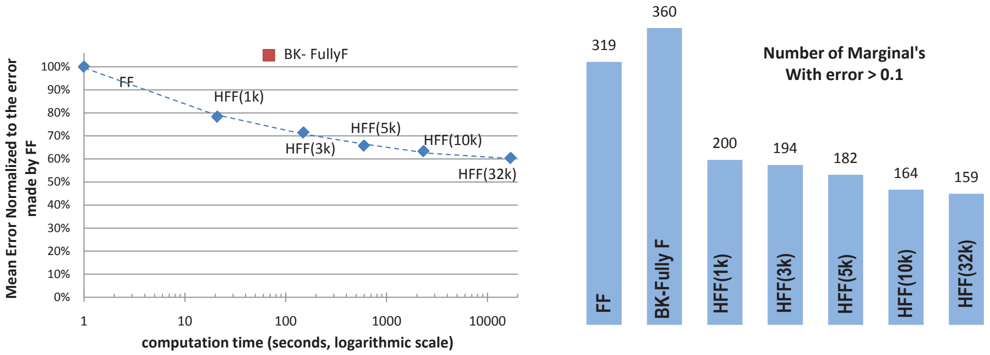


Figure 4.14: (a) Normalized mean errors over all marginals, (b) Number of marginals with error greater than 0.1: Epo stimulated ERK pathway

differentiation and survival. This pathway describes the effect of these isoforms on the Epo (cytokine) induced ERK cascade. The ODE model of this pathway [116] consists of 32 differential equations and 24 associated rate constants. To construct the DBN, we divided the value domain of variables into 5 intervals. Here the interval sizes for variables were not all kept equal. For 23 species that have very low basal concentration level, we set the first interval of the corresponding variables to be smaller ($\sim 20\%$) compared to the other 4 intervals (equal sized). The rest 9 variables all have equal sized intervals as before. Time horizon was fixed at 60 minutes which was then divided into 100 time points. We constructed the DBN using equation based subinterval sampling. As figure 4.5(b) indicates, the quality of the approximation relative to the original ODE dynamics was again high. We then ran FF, BK and HFF(σ) for various choices of σ.

FF and BK were quite accurate for many of the species. However, for some species such as *JAK2*, phosphorylated *EpoR, SHP1* and *mSHP1* etc. which are biologically relevant, FF and BK incurred a max error of 0.49. On the other hand, HFF($3K$) incurred a max error of 0.45 while HFF($32K$) incurred a max error of 0.31. The normalized mean error over all marginals went down to $\sim 70\%$ for HFF($3K$) and $\sim 60\%$ for HFF($32K$) as shown in figure 4.14(a). Further, when we computed the number of marginals with errors greater than 0.1, we found that this number reduced by around half for HFF($32K$) compared to FF as shown in figure 4.14(b).

4.4.3 Comparison with clustered BK

An important component of the BK algorithm is the grouping of the variables into clusters. The idea is to choose the clusters in such a way that there is not much interaction between

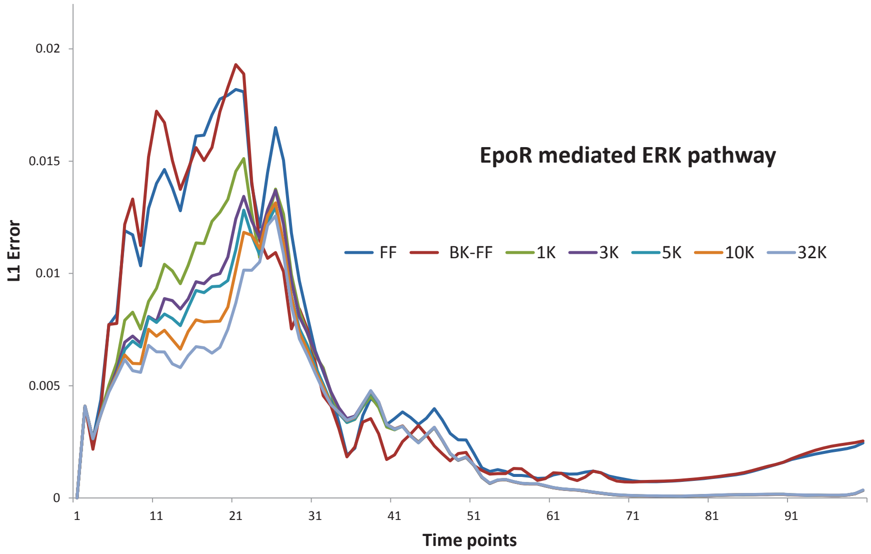


Figure 4.15: L1 error vs time points : Epo stimulated ERK pathway

variables belonging to two different clusters. When this is done well, BK can also perform well. However, choosing the right clusters seems to be a difficult task. The easy option, namely, the fully factored BK in which each cluster is a singleton performs in our case studies as badly (or well) as FF.

We tried to gain a better understanding of BK augmented with non-trivial clusters by using the structure of the pathway to come up with good clusters. A natural way to form 2-clusters seemed to be to pair together the activated (phosphorylated) and inactivated (dephosphorylated) counterparts of a species in the pathway. For the EGF-NGF pathway, this clustering indeed reduced overall errors compared to FF and HFF($3K$). However, we found that HFF(σ) with $\sigma > 5000$ outperformed this version of BK. We did not consider bigger clusters for two reasons: first, when we tried to increase the sizes and the number of clusters in different ways, BK ran out of the 3GB memory. Second, there seemed to be no biological criterion using which one could improve the error performance of BK.

For the Epo mediated ERK pathway too we tried similar clustering. Here the natural clusters were of size 3. Unfortunately, the results were as bad as for fully factored BK. HFF, even with $1K$ spikes ($\sigma = 1000$) was able to perform better than this clustered version of BK. This suggests that the clusters we chose were not the right ones. Hence in our setting, a clustered version of BK that performs well in terms of the computational resources required and the errors incurred appears to be difficult to realize.

4.5 Discussion

In this chapter we have described our improved probabilistic inference algorithm, HFF, for DBNs. HFF algorithm reduces errors made by approximate algorithms such as FF by maintaining a small number of full dimensional state vectors called spikes, whose probabilities are maintained at each time slice in addition to maintaining and propagating belief states in a factored form. By tuning the number of spikes, one can gain accuracy at the cost of increased but polynomial (quadratic) computational cost. We have used large DBNs to illustrate the improvements achieved by our algorithm in comparison with FF. We have also shown that HFF is more practical than algorithms such as BK in our setting. The following chapters will focus on probabilistic model checking.

Chapter 5

Probabilistic Model Checking

In this chapter we will discuss the basics of probabilistic model checking. They refer to the class of formal verification techniques for automated analysis of probabilistic systems. We will first describe model checking in a setting where probabilities do not arise. This will be followed by discussion of its counterpart for probabilistic systems. We will then follow it up with a discussion on the application of model checking to computational systems biology. It will set the background for our contributions to this topic presented in the subsequent chapters.

5.1 Models

First we discuss Kripke structures [121], which are commonly used to describe finite state models. Next, we discuss common probabilistic models such as discrete time Markov chains (DTMC) and Continuous time Markov chains (CTMC).

5.1.1 Kripke structures

A Kripke structure, used to describe a finite state model, can be formally defined as a tuple , $\boldsymbol{K} = < S,s_{init},T,L >$ where

- S is a finite set of states;
- s_{init} is the initial state;
- $T \subseteq S \times S$ is a transition relation between states such that $\forall\ s \in S$, $\exists\ s' \in S$ such that $(s, s') \in T$;

- $L : S \mapsto 2^{AP}$, where L is a labeling function that labels each state $s \in S$ with the set of atomic propositions that are *true* in that state;

For probabilistic systems which usually modeled as Markov chains, we use variants of Kripke structure. The transition relation T is replaced with a stochastic transition relation R, which comprise of either transition probabilities (known as Discrete Time Markov Chains (DTMCs)) or transition rates (Continuous Time Markov Chains (CTMCs)).

5.1.2 DTMC, CTMC

A labeled DTMC [122] can be defined as a tuple $< S$,s_{init},R,$L >$ where

- S are the finite set of states;
- $s_{init} \in S$ is the initial starting state;
- $R : S \times S \rightarrow$ [0,1] is a transition probability function such that $R(s, s')$ is the probability of moving from s to s' where $s, s' \in S$ and $\sum_{s' \in S} R(s, s') = 1$ for all s $\in S$.
- $L : S \rightarrow 2^{AP}$, where L is a labeling function that labels each state with the set of atomic propositions(AP) that are *true* in that state.

Hence, a DTMC can be considered as a Kripke structure where the transition across states is augmented with probabilities i.e, if the system is in state $s \in S$ at time t, it stays there for one unit of time and jumps to state $s' \in S$ at time $t + 1$ with probability $R(s, s')$, regardless of its history up to and including time $t - 1$. A transition from state s to s' can only take place if $R(s, s') > 0$. Each state $s \in S$ is labeled with atomic propositions. We define a *path* in the DTMC to be a finite execution (of length k) of the DTMC starting from s_{init}, where each subsequent state $s' \in S$ is decided according to R. The probability of a path $s_{init}, s_1 .... s_i ... s_k$ where $s_i, s_k \in S$ is 1 if $k = init$ or $=$ $R(s_{init}, s_1) \times ... \times R(s_{i-1}, s_i) ... \times R(s_{k-1}, s_k)$ otherwise. The probability space consist of all *paths* starting at s_{init} and of length $k + 1$.

A labeled CTMC [123] follows a similar definition to that of DTMC, the only difference is that a CTMC allows modeling of continuous time. The edges carry probabilistic timing information. This means that state changes in a CTMC can occur at arbitrary time unlike at fixed time interval in a DTMC. Instead of the transition state probability matrix in DTMCs, a rate matrix R' is defined, which gives the rates $R'(s, s')$ at which transitions occur between each pair of states $s, s' \in S$. If $R'(s, s') = 0$ then no transition from state s to s' is possible, else if $R'(s, s') > 0$, then $1-e^{-R'(s,s').t}$ denotes the probability of moving from state s to s' within t time units. DTMC and CTMC models have are used in the context of biological systems [92, 124] for modeling and analysis of biopathway dynamics.

5.2 Temporal logics

Temporal logics are formalisms used to describe the set of properties about system behavior. The set of *temporal operators* describe the implicit time ordering between events of the system. There exist many different temporal logic formalisms which differ based on the model to be analyzed and the desired expressive power of the formalism. The choice of temporal logic formalism is crucial, since the complexity of verification depends on it. Temporal logics may be differentiated into categories depending on the systems they are used to reason about. They can be either probabilistic, non-probabilistic or be considered in linear time, branched time setting etc. Examples of non-probabilistic temporal logics include Linear Time Temporal Logic (LTL) which considers models where time is modeled along a single path, Computation Tree Logic (CTL) which considers time modeled as a tree representing the different paths the system could take. Probabilistic counterparts include PCTL which is a probabilistic extension of CTL, PLTL which is a probabilistic extension of LTL. To illustrate the ideas of these temporal logics, we will discuss LTL and probabilistic CTL (PCTL) in the following:

Linear Time Temporal Logic (LTL)

LTL [93] was first proposed by Amir Pnueli in the context of verification of programs. It is used to express properties along paths of the system.

Syntax of LTL Let's assume that $AP = \{A_1, ...A_n\}$ be the set of atomic propositions. Formulas in LTL are built from AP along with propositional logic connectives $\{\vee, \sim\}$ and temporal operators $\mathbf{O}$,$\cup$. Given the set AP, a LTL formula is inductively defined as:

- *true*, *false* are LTL formulas;

- $\forall\ A_i \in AP$, A_i is a LTL formula;

- If ψ is a LTL formula then $\sim \psi$ is an LTL formula;

- If ψ, ψ' are LTL formula then $\psi \vee \psi'$ is a LTL formula;

- If ψ, ψ' are LTL formula then so are $\mathbf{O}(\psi)$, $\psi \cup \psi'$.

Semantics of LTL $\forall i$ such that $i \in (0, 1, 2......)$, let π_i denote the sequence of states s_i, s_{i+1}, s_{i+2}... in a path π, we denote an LTL formula ψ holds in the path starting at state s_i by $\pi_i \models \psi$. The relation $\pi_i \models \psi$ is defined as follows:

- $\pi_i \models$ *true*, $\pi_i \not\models$ *false*;

- If $\psi \in AP$, $\pi_i \models \psi$ iff $s_i \models \psi$ (ψ is $true$ at s_i);

- $\pi_i \models \sim \psi$ iff $\pi_i \not\models \psi$;

- $\pi_i \models \psi \vee \psi'$ iff $\pi_i \models \psi$ or $\pi_i \models \psi'$;

- $\pi_i \models \mathbf{O}(\psi)$ iff $\pi_{i+1} \models \psi$;

- $\pi_i \models \psi \cup \psi'$ iff there exists a j, $j \geqslant i$ such that $\pi_j \models \psi'$ and $\forall\ k$, $i \leqslant k < j$, $\pi_k \models \psi$.

The derived propositional operators such as $\wedge$, $\implies$, $\equiv$ and the temporal operators **G** (always from now), **F**(sometime in the future) follow from basic operators through the following relation, $\psi \wedge \psi' = \sim (\sim \psi \vee \sim \psi')$, $(\psi \implies \psi') = (\sim \psi \vee \psi')$, $(\psi \equiv \psi') = (\psi \implies \psi' \wedge \psi' \implies \psi)$, $\mathbf{F}(\psi) = true \cup \psi$, $\mathbf{G}(\psi) = \sim \mathbf{F}\ (\sim \psi)$. A LTL formula formula ψ is declared to be $true$ iff $\pi_0 \models \psi$.

Probabilistic Computation Tree Logic (PCTL)

PCTL [122] is a probabilistic extension of CTL which is a branching time temporal logic. It is useful for reasoning about properties of stochastic systems such as ‘‘if the gene encoding protein A is knocked out then is there an 85% probability that the concentration of protein B drops?”. Carrying forward the notations for atomic propositions and temporal and propositional operators from the discussion of LTL, we describe the syntax and semantics of PCTL.

Syntax of PCTL

- $\forall\ A_i \in AP$, A_i is a PCTL formula;
- if ψ is a PCTL formula, then so is $\sim \psi$;
- if ψ and ψ' are PCTL formula, then so is $\psi \vee \psi'$;
- if ψ and ψ' are PCTL formula, then so are $\mathbf{O}(\psi)$, $\psi \cup^{\leq t} \psi'$; these are referred to as path formulas.
- if ψ is a PCTL formula, p a real number with $0 \leq p \leq 1$ and $\bowtie \in \{\leq, \geq, >, <\}$, then $[\psi]_{\bowtie p}$ is a PCTL formula.

The other derived operators are defined as in the previous discussion on LTL. The main focus is on the quantity "p" which represents the probability of satisfaction of the property. For instance, the formula $\psi U^{\leq t} \psi'_{\geq p}$ expresses that within the next t time units, with at-least a probability p, ψ' will become true and ψ will be true from now until ψ' become true.

PCTL Formulas in PCTL are interpreted over a DTMC $\mathcal{D}$. $\mathcal{D}, s \models \psi$ means that the formula ψ is true at state s in the DTMC $\mathcal{D}$.

Let us denote by a path π, the set of infinite states $(s_0, s_1....)$ such that $\forall i$, s_i are states of $\mathcal{D}$. Let us denote the set of all infinite paths starting from state s_i as $Path(s_i)$, s_i are states of $\mathcal{D}$.

We will now define a probability measure over the set of paths. We will begin by defining *cylinder sets* which is a measure of the set of paths with a common finite prefix. Let $s_0, s_1...s_k$ be a finite sequence of states, we let $Cylinder(s_0, s_1...s_k) = \{\pi \in Path(s_0) | s_0, s_1...s_k$ is the prefix of $\pi\}$. We define its measure as $Pr_{s_0}(Cylinder(s_0, s_1...s_k))$ $= \prod_{0 \leq i < k} R(s_i, s_{i+1})$. For all other states of the DTMC excluding s_0, $Pr_{s_i}(Cylinder$ $(s_0, s_1...s_k)) = 0$. We will now extend this to the $\sigma-$algebra generated by the cylinder sets. The $\sigma-$algebra consists of all the $Cylinder(s_0, s_1...s_k)$ for the set of states $s_0, s_1, s_2...s_k$, the empty set and is closed under the union and complement.

$\mathcal{D}, \pi \models \psi'$ means that the path formula ψ' is true for the path π in the DTMC $\mathcal{D}$ and $\mathcal{D}, \pi[k] \models \psi'$ means that the path formula ψ' is true for the path starting at state k of path π in the DTMC $\mathcal{D}$.

We define $Pr_s(\psi')$ as the summation of the probability measure of all the cylinder sets of paths $\in Path(s)$ which satisfy the formula ψ', $Pr_s\{\ \pi \in Path(s) | \mathcal{D}, \pi \models \psi'\}$. Let p be a real number with $0 \leq p \leq 1$, $\bowtie$ be a comparison operator such that $\bowtie \in \{\leq, \geq, >, <\}$. The satisfaction relation $\mathcal{D}, s \models$ is defined as follows:

- $\mathcal{D}, s \models true$ for all states;
- If $\psi \in AP$, $\mathcal{D}, s \models \psi$ iff ψ is $true$ at s of the DTMC $\mathcal{D}$;
- $\mathcal{D}, s \models \sim \psi$ iff $\mathcal{D}, s \not\models \psi$;
- $\mathcal{D}, s \models \psi \vee \psi'$ iff $\mathcal{D}, s \models \psi$ and $\mathcal{D}, s \models \psi'$;
- $\mathcal{D}, \pi \models \psi\ U^{\leq t}\ \psi'$ iff there exists an $i \leq t$ such that $\mathcal{D}, \pi[i] \models \psi'$ and $\mathcal{D}, \pi[j] \models \psi$, $\forall\ j : 0 \leq j < i$;
- $\mathcal{D}, s \models [\psi]_{\bowtie p}$ iff $Pr_s(\psi) \bowtie p$.

5.3 Model checking algorithms

Given the model and the property encoded in a specific temporal logic formalism, the task of the model checking algorithm is to systematically traverse the state space of the model to check if the property holds.

Model checking LTL formulas The most common method to verify LTL formulas is using an automata-theoretical approach [125]. Informally, the procedure consists of, first, constructing an automaton of the formula $\sim \psi$ where ψ is the LTL formula we need to verify. Next, we compose the original system model which is being verified with the constructed automaton, this produces a product automaton. Then we attempt to find a path from the start state to the end state (of the original model) in the product automaton using a depth first search. If we can find such a path in the product automaton, we report that the formula ψ does not hold for the model and the path constitutes a violation of the formula ψ, it is reported as the counter example.

Model checking PCTL formula We will now briefly discuss the model checking algorithm for PCTL [126, 122]. The algorithm takes as input a DTMC, and the PCTL formula ψ, and outputs the set of states in the model that satisfy ψ. First, the parse tree for ψ is constructed, each node in this tree is labeled with a sub formula of ψ, the leaves represent the atomic propositions or *true*. we start from the leaves of this tree onto sub formulas of increasing complexity to compute the states of the model which satisfy the sub formula. At the end of the computation, the set of states that satisfy the formula ψ are computed. The rules for determining if a state satisfies a formula have already been discussed in the section on PCTL.

In real life scenarios - especially with stochastic models - the state space of models is large, so it is important to use data structures and algorithms that minimize the computational space and time requirements for model checking. In terms of dealing with such large systems, two main caveats need consideration. The first deals with representing these state spaces efficiently and compactly with in the given memory constraints. Next, is to resort to approximate methods of model checking since performing exact computations - especially in the case of probabilistic systems- may be infeasible or time consuming. We will briefly discuss both these aspects in the following.

Methods for state space reduction include use of sparse matrices and symbolic methods. The idea behind symbolic state-space representation is to exploit the regularity and structure in the models. Examples of symbolic data structures are the Binary Decision Diagram (BDD), Multi-Terminal Binary Decision Diagram (MTBDD). BDDs [127] are data structures which are used to represent Boolean functions efficiently. BDDs are directed, acyclic graph, consisting of intermediate decision nodes and terminal nodes labeled with 0 and 1. Each decision node is labeled with a Boolean variable and has two child nodes representing assignment of 0 or 1 to the children. Massive reduction in state space can be achieved by ordering the variables in a specific order and eliminating identical sub-graphs in the BDD. An MTBDD [128] is a data structure that represents a function mapping of Boolean variables to real numbers i.e it can be seen as a directed acyclic graph containing decision nodes and terminal nodes with real numbers (instead of 0 and 1 in BDDs), this structure is effective to compactly represent matrices with real values especially in probabilistic model checking.

Having decided a suitable state space representation, next, in the context of probabilistic models, it is important to compute the probabilities of the properties (temporal logic formula). This entails solving a system of linear equations. Numerical methods exist for solving them [129]. These methods, although are highly accurate fail to scale to large systems. They fall into the category of exact algorithms. Unfortunately, models considered in domains such as systems biology have a much larger state space than those which can be efficiently verified by numerical methods. In such cases, approximate methods are used. One such method works by employing statistical methods to obtain a reliable estimate of the probability of a property by sampling the underlying stochastic model. These methods fall into the category of statistical model checking. The main advantage offered by these methods is that we can sample the stochastic models without explicitly representing them. All we need is a simulatable version of the model in a high level modeling formalism. The main task is to generate executions of the underlying model which we will refer to as $trace$. Once a $trace$ is generated, we check if the property holds for this trace. When enough traces are generated, we perform statistical analysis of these traces to see if they provide enough evidence to suggest that the truth-hood of the property. It is known that in the asymptotic limit of the number of traces, statistical methods converge to the true probability. However since the number of traces that can be drawn is limited, we use statistical analysis methods to provide guarantees on the confidence of the result and the number of traces needed. Many algorithms have been proposed to solve the statistical model checking problem efficiently [130, 131, 132, 133]. These algorithms are based on whether the system to be verified allows for drawing traces in an unrestricted way (white box systems) or if the number or nature of traces that can be drawn from the model is restricted (black box systems). To assert a probabilistic property, these algorithms either estimate the true probability of the property (statistical estimation methods) or formulate it as a statistical hypothesis testing problem.

5.4 Model checking in computational systems biology

There has been considerable interest in adapting formal methods such as model checking for analyzing models in computational systems biology in the past decade. Main challenges in adapting them include, (1) formulating interesting properties to analyze keeping in mind the complex dynamics of biological systems, (2) dealing with the varied modeling formalisms used to model biological systems, (3) dealing with the large state spaces associated with these models especially in probabilistic settings and (4) overall, to use it as a tool that aids in building, calibrating and analyzing highly consistent and accurate models of biological systems. We will briefly discuss some of these applications in this section.

We will discuss existing literature mainly under two themes. First, we discuss applications which focus on non-probabilistic systems. Next, we discuss those which focus on probabilistic systems. Under each theme, we will differentiate methods which focus on analysis of models (assuming a consistent model has been built) and those where model checking is used for performing tasks such as model calibration.

Among the early works to use model checking to analyze dynamics of biological systems, the tool BIOCHAM[88, 134, 135, 136] provides a framework for modeling and analyzing biological systems, it uses a rule based modeling framework for modeling biological systems. The models consist of a set of system variables, their initial states and a set of condition action rules on the variables. These rules, along with the system variables induce a Kripke structure. Queries which constitute biologically interesting properties and how such properties can be expressed using CTL (Computation tree logic) are discussed. Next, a CTL based symbolic model checking algorithm is used for analyzing several qualitative model and quantitative models. They mainly use existing model checkers (the symbolic model checker NuSMV[137] and constraint based model checker DMC[138]). This was among the first frameworks which provided a proof of concept that model checking can be used for useful analysis of biological pathway models and continues to be maintained and updated[139].

Antoniotti and colleagues [140] describe an automaton based approach to study the temporal evolution of complex biochemical reactions modeled as a set of differential algebraic equations. Their main motivation for the work is use model checking to interpret and automate the reasoning process of simulation traces. They summarize simulation traces to an automaton and use CTL to specify queries, their approach is consolidated into a tool “Sympathetica”. They illustrate the method on a model for purine metabolism. Batt and colleagues [89] describe a validation platform for models built with a class of piecewise-linear (PL) differential equations that permit coarse-grained, qualitative analysis of the network dynamics. The analysis is based solely on sign pattern of the derivatives of system variable. Instead of numerical values for the parameters, the method uses inequality constraints that can be inferred from the experimental literature. They convert the equations into a state transition graphs which are conservative approximations of the dynamics of the underlying PL models, these graphs are amenable for temporal logic based verification. CTL was used for specify the queries. The validation approach was applied to the analysis of the network controlling the nutritional stress response in E.coli.

Monteiro and colleagues [90],[141] focus on the issue of constructing interesting, relevant queries for biological models which is usually not an easy task for non-expert users. The authors propose use of "patterns" which are high level query templates which capture complex biologically relevant queries that can be automatically translated into temporal logic formulas. The queries that were represented as patterns (occurrence pattern, exclusion pattern, consequence pattern, sequence pattern, invariance pattern) and concerned the domain of genetic regulatory networks. They show the use of the method on the analysis of the model of E.coli carbon starvation response. Fisher and colleagues[20] build a discrete, state based mechanistic model of vulval development in *Caenorhabditis elegans*, using the reactive modules framework. The model consists of inductive and lateral signaling pathways involved in vulval development and cross talks between them. Next, they use a model checking framework, consolidated in the tool MOCHA[142] to analyze all possible behaviors of the model. Their analysis was able to predict additional details about the mechanism of lateral signaling and the temporal ordering of events in the pathway crucial for stable cell fate. These predictions were also validated experimentally. Other applications of model checking for analyzing non probabilistic systems can be found in [143], [144], [145], [146].

Next, we discuss some work on using model checking in the context of calibrating (parameter estimation) deterministic models. The idea is to formulate expected system behavior as formulas in the temporal logics and using the model checking procedure to search through the high dimensional parameter search space for parameter which can explain the expected behaviors. In this direction, [97] focus on randomly sampling the set of unknown parameters and accepting the set of parameters if the simulation trace satisfies LTL formulas which specify the desired properties of the system. A similar approach is taken by [100] in the context of hybrid functional Petri-nets (HFPN), where millions of parameter sets are sampled and the associated simulation traces are verified in an on-line fashion. However, both methods lack a principled search method for finding satisfactory parameters; they apply a brute force strategy to search the parameter search space. Typically, the parameter search space is high dimensional, in which case these strategies will need impractically large number of samples and are unscalable. The work reported in [99] also consider parameter estimation on a single simulation trace, however they use a evolutionary strategy based search algorithm to guide the search. Methods such as [147, 98] focus on parameter estimation on multi-affine ODE systems; their method relies on explicitly constructing a symbolic encoding of the dynamics of the pathway models and using symbolic model checking to derive parameters.

Moving on to verification in the context of probabilistic systems, [57] introduce the tool PRISM (Probabilistic symbolic model checker), which is an analysis tool for probabilistic systems. System models are described using the PRISM modeling language, a high-level state-based description language. In this language a system is described as the parallel composition of a set of modules. The prism model description is translated into DTMC, CTMC, or a Markov Decision Process (MDP). Properties are specified using PCTL (for DMTCs) or CSL (for CTMCs). In PRISM it is possible to either determine if a probability satisfies a given bound or obtain the actual value. There is also support for the specification and analysis of properties based on costs and rewards. PRISM uses symbolic approaches to store the state space and numerical computation for quantitative probabilistic model checking. PRISM has been widely used in the context of verifying biological systems. In [58], a model the MAPK (Mitogen Activated Protein Kinase) Cascade is constructed using PRISM, which is then converted into a discrete stochastic model. In the paper a population based approach is used to replicate and validate the dynamics of the pathway as reported in the literature. Next, [59] illustrate the use of PRISM to study FGF (Fibroblast Growth Factor) pathway. Calder and colleagues [61],[124] further illustrate use of PRISM for modeling and analyzing RKIP-inhibited extra-cellular signal Regulated Kinase (ERK) pathway where in the concentration of each protein are modeled as discrete abstract quantities, but time is continuous. The CTMC is constructed for the pathway and continuous stochastic logic (CSL) is used to specify temporal properties. They were mainly interested in the role of RKIP on the behavior of the pathway and focus on verifying properties describing steady state and transient profiles for different reaction rates and activation sequences.

Ballarini and colleagues [60] have used the PRISM model checker to gather quantitative characterization of properties of biological systems exhibiting oscillatory behavior.

They use PRISM to develop a Markovian model of both a transient oscillator, known as the 3-way oscillator as well as of its permanent oscillation variant. Exact probabilistic model checking such as the one used in PRISM has the disadvantage that as the models considered become large, they suffer the state space explosion problem and hence exact model checking takes a lot of time and effort and infeasible in some cases. In such cases approximate methods of model checking are often used. Statistical model checking, is one such method which relies on simulating the underlying (large) probabilistic model using a high level simulatable description of the model, and based on the simulations and subsequent statistical analysis, decides if a property holds for the probabilistic model. The main advantage is that, it is not necessary to explicitly construct and store the whole state space of the probabilistic model. These methods have a low time complexity, require low memory and are tunable in terms of the accuracy of the result needed.

Donaldson and colleagues [91] propose a method that resort to taking a fixed number of simulations of the underlying model. They extend probabilistic LTL with numerical constraints (PLTLc) to formulate properties and employ Monte Carlo simulations to approximate the probability of the PLTLc properties. Monte Carlo approximation samples a finite set of paths through the model's state space (trace), the probability of properties is calculated as the number of traces that satisfy the property by the total number of traces drawn. They also introduce a tool called the Monte Carlo Model Checker for PLTLc properties MC2(PLTLc). They use the formulated method to validate properties of the MAPK signaling pathway. In a subsequent paper [148], they use the approach for the parameter estimation problem, they use the temporal logic specification as the expected result and try to estimate the parameters for which the underlying stochastic model conforms to the specification, a genetic algorithm is used to drive the search. Next, [92] introduce the BIOLAB algorithm for statistical probabilistic model checking of CTMC models of biological processes. This was among the first applications of hypothesis testing based statistical model checking for biological models. The main algorithm they use is that of [130], these methods convert the original probabilistic model checking problem into a hypothesis testing problem. The set of initial states of the system comprise of user-specified set of initial conditions and parameter values. Properties are expressed in probabilistic bounded linear temporal logic. BIOLAB then statistically verifies the property using sequential hypothesis testing on executions sampled from the model. The sequential hypothesis testing is carried out using the Walds sequential probability ratio test. The authors show that they can bound the probability of false-positive and false-negative errors, with regard to the predictions the algorithm makes. They validated their approach using the T Cell receptor pathway model. We will briefly discuss hypothesis testing based algorithms now. Here we specifically focus on the problem of checking properties of the form $Pr_{\geq p}\{\psi\}$ where p is the threshold probability against which we want to compare the real probability p' with.

Younes[149] proposed the single sampling based hypothesis testing algorithm where the number of traces (n) is decided upfront. Given $H0 : p' \geq p$ against $H1 : p' < p$, a constant c is also specified that decides the number of samples that should evaluate to $true$ to accept the hypotheses. if $\sum_{i=1}^{n} x_i > c$ then hypothesis $H0$ is accepted, else $H1$ is accepted. The main challenge is to find the pair $< n, c >$ such that $H1$ is accepted with probability utmost α(Type 1 error) when $H0$ holds , and $H0$ is accepted with probability at most β(Type 2 error) when $H1$ holds. Finding the pair $< n, c >$ is non-trivial and the authors describe an algorithm based on binary search to find the pair $< n, c >$ that obeys the bounds. These methods provide no guarantees about the result, however either the null hypothesis or the alternate hypothesis is accepted with bounds $< \alpha, \beta >$ on the probability of the error.

The number of samples in the previous method can be reduced by taking observations into account as they are made, in this regard Younes [130] formulate the probabilistic model-checking problem as a sequential hypothesis-testing problem. After every sample trace is drawn, a statistical test is carried out, the outcome of the test decides if another sample needs to be drawn on if a decision can be made with the last drawn sample. Hence, these methods adapt to difficulty of the problem. For practical considerations the original hypothesis test is relaxed as with a factor δ which represents the indifference region around the threshold p; as indifference region tends to zero, the ideal case is reached. Now, the original hypothesis testing problem is slightly modified to testing the null hypothesis $H0 : p' \geq p + \delta$ against the alternative hypothesis $H1 : p' < p - \delta$.

Let X_i be a Bernoulli random variable such that ($Pr[X_i = 1] = p'$ and $Pr[X_i = 0] = 1 - p'$). An observation of X_i, represented as x_i states if the specified temporal logic formula is $true$ or $false$, for example in our case x_i will be 1 if the ith sample satisfies ψ and 0 if it does not. A sequential sampling algorithm based on Wald's sequential probability test is used to solve the hypothesis testing problem. After taking n samples (observations) $x_1, x_2, ...x_n$ of the system, calculate

$$f_n = \prod_{i=1}^{n} \frac{Pr[X_i = x_i \mid p' = p - \delta]}{Pr[X_i = x_i | p' = p + \delta]} = \frac{[p-\delta]^{(\sum_{i=1}^{n} x_i)}[1-[p-\delta]]^{(n-\sum_{i=1}^{n} x_i)}}{[p+\delta]^{(\sum_{i=1}^{n} x_i)}[1-[p+\delta]]^{(n-\sum_{i=1}^{n} x_i)}} \tag{5.1}$$

Hypothesis $H0$ is accepted if $f_n \geq A$, and Hypothesis $H1$ is accepted if $f_n \leq B$. The constants A and B are chosen such that it results in a test of strength $< \alpha, \beta >$. In practice to satisfy the strength dictated by $< \alpha, \beta >$, choose $A = \frac{1-\beta}{\alpha}$ and $B = \frac{\beta}{1-\alpha}$. Samples are drawn until a decision can be made.

Younes [131] further discuss a modified SPRT algorithm, owing to the issue that the previous algorithm satisfies the error bounds α, β only when the true probability does not lie in the indifference region. In the modified SPRT algorithm, the error for cases when the true probability lies in the indifference region is bounded by introducing a factor (γ) (which controls the probability of an undecided result) such that:

$$Pr[s \vdash_I \phi | (s \mid\approx_T^\delta \phi) \vee (s \mid\approx_\perp^\delta \phi)] \leq \gamma. \tag{5.2}$$

where $s \vdash_I \phi$ represents that the algorithm returns undecided results for ϕ , $s \mid\approx_T^\delta \phi$ represents that the formula ϕ being true (using the algorithm), and $s \mid\approx_\perp^\delta \phi$ represents that the formula ϕ being false (using the algorithm). The algorithm is modified to using two acceptance sampling tests:

$$H0 : p' \geq p \text{ against } H1 : p' < p - \delta \text{ with } < \alpha, \gamma > \tag{5.3}$$

$$H0^{'} : p' \geq p + \delta \text{ against } H1^{'} : p' < p \text{ with } < \gamma, \beta > \tag{5.4}$$

the algorithm is applied to the 2 hypotheses, and $Pr_{\geq p}\{\psi\}$ is reported as *true* if $H0$ and $H0^{'}$ are accepted and $false$ if $H1$ and $H1^{'}$ are accepted, any other combination the results is reported as *undecided*.

Langmead and colleagues [150] argue that the current Probabilistic model checking algorithms based on hypothesis testing uses classical statistical procedures such as Wald's Sequential Probability Ratio test(SPRT) to answer the decision problem are not efficient in terms of the number of samples needed to determine the solution to problem, hence they suggest hypothesis testing based on Bayesian statistical procedures, which requires fewer samples to be considered and also has the advantage of being able to use prior knowledge(which is known in the biological pathway setting) in the form of a probability distribution. They discuss an algorithm for performing model checking using the approach and apply it to the yeast heterotrimetric G protein cycle pathway model. Their algorithm verifies properties of the model expressed as formulas in probabilistic bounded temporal logic (PBLTL) which is a probabilistic version of bounded LTL. The algorithm basically draws samples and checks if it satisfies the temporal logic specification, the number of samples to decide when to accept a hypothesis is decided based on the Bayes factor (determined from the samples, it depends on the sample and the prior probabilities). They sample until the Bayes factor goes above a particular threshold set by the user and then decide to accept or reject the hypothesis.

Further, [151] reports the application of this method to analyze the HMGB1 signaling pathway. Previous methods on statistical model checking applied to the domain of were $offline$ i.e they simulated the model to generate the whole trace, before applying the model checking procedure on the trace. However it is a wasted effort to simulate and generate the whole trace, which is usually an expensive operation. Instead it may be better to use an *online* approach where we model check the trace as it is generated. In this regard [152, 153] use an online approach to perform statistical model checking. Other applications of probabilistic model checking in systems biology settings can be found in [144, 139, 154] etc.

In summary, application of probabilistic model checking in the domain of computational systems biology are mainly focused and moving towards dealing with models which have a large state spaces. For relatively small systems exact methods can be used for analysis. However when considering larger systems, approximate methods such as statistical model checking algorithms have been used. Hence, the need to develop methods where the large state space arising in stochastic models can be effectively dealt, either by more efficient representations or by focusing on approximate methods which can scale.

Chapter 6

Probabilistic model checking on DBNs

6.1 Introduction

Thus far we have discussed how DBNs arise as succinct representations of high dimensional probabilistic dynamics. We have discussed the problem of inferring probability distribution of the state of variables in DBNs. We have also described the basics of probabilistic model checking.

This chapter focuses on analyzing DBN models using probabilistic model checking. Specifically, our focus is on developing probabilistic model checking algorithms for DBNs based on probabilistic inference. Our idea is to combine DBN inference algorithms with temporal logics for doing probabilistic model checking.

In terms of previous work involving DBNs and model checking, the works reported in [155] and [156] are relevant. These approaches focus on solving the probabilistic inference problem on DBNs using model checking. They convert a DBN to a corresponding Markov chain, which is then represented using symbolic data structures such as MTBDDs. Next, they use standard probabilistic model checking algorithms to solve the DBN inference problem. These techniques are limited in application to restricted classes of DBNs and to relatively small systems since it relies on explicitly constructing and symbolically encoding the underlying Markov chain. We are instead interested in the inverse approach of developing model checking frameworks directly on DBN models, since in our case DBNs are succinct representations of large Markov chains. In this direction, we first discuss our temporal logic framework for DBNs. Next, we discuss the logic in relation with PCTL. We follow it up with a discussion of our model checking algorithm and discuss how we use this approach to verify interesting biological properties in our class of DBNs.

6.2 Bounded Linear time Probabilistic Logic

We use a probabilistic variant of linear time temporal logic (LTL) [93] which we call bounded linear time probabilistic logic (BLTPL). Informally, the atomic propositions are of the form $(X, v) \leq c$ or $(X, v) \geq c$ where X is a finite valued random variable corresponding to a node in the DBN and c is rational number in $[0, 1]$, here c indicates the threshold probability. The assertion $(X, v) \leq c$ says that the probability of the random variable X currently having the value v is less than or equal to c; similarly for the assertion $(X, v) \geq c$. Though probability enters the logic only via atomic propositions it turns out that one can still express many interesting dynamical properties. Probabilistic inference algorithms such as the FF, BK or HFF can be used to *approximately* determine the truth-hood of these atomic propositions.

6.2.1 Syntax

We will follow notations that were introduced in the previous chapters. As discussed before, in our temporal logic, the atomic propositions will be of the form $(i, v)\#r$ with $\# \in \{\leq, \geq\}$ and $r \in [0, 1]$. Here (i, v) stands for the random variable X_i of our DBN taking a value v from its domain. The proposition $(i, v) \geq r$, if asserted at time point t, says that $M_i^t(v) \geq r$; similarly for $(i, v) \leq r$. Given the set of atomic propositions AP, a BLTPL formula is inductively defined as:

- *true*, *false* are BLTPL formulas;
- Every atomic proposition $\in AP$ is a BLTPL formula;
- If φ is a BLTPL formula then $\sim \psi$ is an BLTPL formula;
- If φ, φ' are BLTPL formula then $\varphi \vee \varphi'$ is a BLTPL formula;
- If φ, φ' are BLTPL formula then so are $\mathbf{O}(\varphi)$, $\varphi \; \mathbf{U} \; \varphi'$.

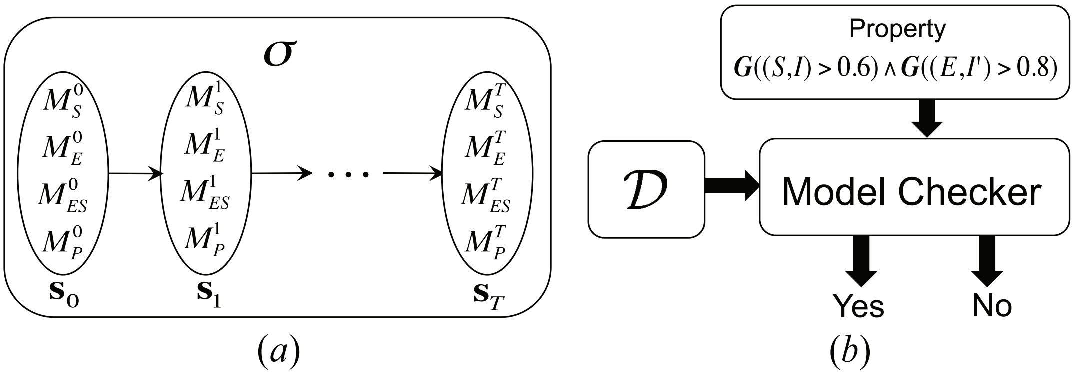


Figure 6.1: (a) The model (sequence of states) defined by the DBN. (b) The model checking procedure.

The derived propositional connectives such as $\wedge, \supset, \equiv$ etc. are defined in the standard fashion. The temporal connectives **F** ("sometime from now") and **G** ("always from now") are defined in the usual way via: $\mathbf{F}(\varphi) = true\ \mathbf{U}\varphi$ and $\mathbf{G}(\varphi) = \sim\ \mathbf{F}(\sim \varphi)$.

6.2.2 Semantics

The formulas are interpreted over the sequence of marginal probability distribution vectors $\sigma = \mathbf{s}_0\mathbf{s}_1 \ldots \mathbf{s}_T$ generated by the DBN $\mathcal{D}$. In other words, for $0 \leq t \leq T$, $\mathbf{s}_t = (M_1^t, M_2^t, \ldots, M_n^t)$. Consequently $\mathbf{s}_t(i) = M_i^t$ for $1 \leq i \leq n$. We also let $\sigma(t) = \mathbf{s}_t$ for $0 \leq t \leq T$. We now define the notion of $\sigma(t) \models \varphi$ (φ holds at t in $\mathcal{D}$) inductively:

- $\sigma(t) \models (i, v) \geq r$ iff $M_i^t(v) \geq r$. Similarly
  $\sigma(t) \models (i, v) \leq r$ iff $M_i^t(v) \leq r$.
- $\sigma(t) \models \sim \varphi$ iff $\sigma(t) \not\models \varphi$
- $\sigma(t) \models \varphi \vee \varphi'$ if either $\sigma(t) \models \varphi$ or if $\sigma(t) \models \varphi'$
- $\sigma(t) \models \boldsymbol{O}\ (\varphi)$ iff $\sigma(t+1) \models \varphi$.
- $\sigma(t) \models \varphi\ \boldsymbol{U}\ \varphi'$ iff there exists $t \leq t' \leq T$ such that $\sigma(t') \models \varphi'$ and for every t'' with $t \leq t'' < t'$, $\sigma(t'') \models \varphi$.

We say that the DBN $\mathcal{D}$ meets the specification φ and this is denoted as $\mathcal{D} \models \varphi$ iff $\sigma(0) \models \varphi$. The model checking problem is, given $\mathcal{D}$ and φ, to determine whether or not $\mathcal{D} \models \varphi$.

6.3 FF based model checking algorithm

We begin by letting $SF(\varphi)$ denote the set of sub-formulas of φ and define it as follows. Since φ will remain fixed we will write below SF instead of $SF(\varphi)$.

SF is the least set of formulas containing φ such that

- $\sim \varphi' \in SF$ implies $\varphi' \in SF$;
- $\varphi' \vee \varphi'' \in SF$ implies $\varphi', \varphi'' \in SF$;
- $\boldsymbol{O}\varphi' \in SF$ implies $\varphi' \in SF$;
- $\varphi' \boldsymbol{U} \varphi'' \in SF$ implies $\varphi', \varphi'' \in SF$.

The main step is to construct a labeling function st which assigns to each formula $\varphi' \in SF$ a subset of $\{\mathbf{s}_0, \mathbf{s}_1, \ldots, \mathbf{s}_T\}$ denoted $st(\varphi')$. After the labeling process is complete, we declare $\mathcal{D} \models \varphi$ just in case $\mathbf{s}_0 \in st(\varphi)$. Starting with the atomic propositions, the labeling algorithm goes through members of SF in ascending order in terms of their structural complexity. Thus φ' will be treated before $\sim \varphi'$ is treated and both φ' and φ'' will be treated before φ' $\boldsymbol{U}$ φ'' is treated and so on.

Let $\varphi' \in SF(\varphi)$. Then:

- If $\varphi' = A$ then $\mathbf{s}_t \in st(A)$ iff $\sigma(t) \models A$. We run FF to determine this. In other words, $\mathbf{s}_t \in st(A)$ iff $M^t(i, v) \geq r$ where $A = (i, v) \geq r$ and $M^t(i)$ is the marginal distribution of X_i^t computed by FF. Similarly $\mathbf{s}_t \in st(A)$ iff $M_i^t(v) \leq r$ in case $A = (i, v) \leq r$.
- If $\varphi' = \sim \varphi''$ then $\mathbf{s}_t \in st(\varphi')$ iff $\mathbf{s}_t \notin st(\varphi'')$.
- If $\varphi' = \varphi_1 \vee \varphi_2$ then $\mathbf{s}_t \in st(\varphi')$ iff $\mathbf{s}_t \in st(\varphi_1)$ or $\mathbf{s}_t \in st(\varphi_2)$.
- Suppose $\varphi' = \boldsymbol{O}(\varphi'')$. Then $\mathbf{s}_T \notin st(\varphi')$. Further, for $0 \leq t < T$, $\mathbf{s}_t \in st(\varphi')$ iff $\mathbf{s}_{t+1} \in st(\varphi'')$.
- Suppose $\varphi' = \varphi_1 \boldsymbol{U}$ φ_2. Then we decide whether or not $\mathbf{s}_t \in st(\varphi')$ by starting with $t = T$ and then treating decreasing values of t. Firstly $\mathbf{s}_T \in st(\varphi')$ iff $\mathbf{s}_T \in st(\varphi_2)$. Next suppose $t < T$ and we have already decided whether or not $\mathbf{s}_{t'} \in st(\varphi')$ for $t < t' \leq T$. Then $\mathbf{s}_t \in st(\varphi')$ iff $\mathbf{s}_t \in st(\varphi_2)$ or $\mathbf{s}_t \in st(\varphi_1)$ and $\mathbf{s}_{t+1} \in st(\varphi')$.

$\varphi' = \boldsymbol{F}(\varphi'')$ and $\varphi' = \boldsymbol{G}(\varphi'')$ can be handled directly. As in the case of $\boldsymbol{U}$, we start with $t = T$ and consider decreasing values of t:

- Suppose $\varphi' = \boldsymbol{F}(\varphi'')$. Then $\mathbf{s}_T \in st(\varphi')$ iff $\mathbf{s}_T \in st(\varphi'')$. For $t < T$, $\mathbf{s}_t \in st(\varphi')$ iff $\mathbf{s}_t \in st(\varphi'')$ or $\mathbf{s}_{t+1} \in st(\varphi')$.
- Suppose $\varphi' = \boldsymbol{G}(\varphi'')$. Then $\mathbf{s}_T \in st(\varphi')$ iff $\mathbf{s}_T \in st(\varphi'')$. For $t < T$, $\mathbf{s}_t \in st(\varphi')$ iff $\mathbf{s}_t \in st(\varphi'')$ and $\mathbf{s}_{t+1} \in st(\varphi')$.

Due to the fact the model checking procedure just needs to treat one finite sequence as a model, it is particularly simple. Its time complexity is linear in the size of the formula φ whereas in traditional settings it will be exponential in the size of φ.

Figure 6.1 summarizes our model checking procedure. Properties of pathway dynamics are formulated as BLTPL formulas. They are then verified using the above labeling algorithm which will call the FF algorithm when dealing the atomic propositions.

6.3.1 HFF based model checking algorithm

We have outlined our FF based model checking algorithm in the previous subsection. We have previously shown that probabilistic inference based FF can incur significant errors on marginal distributions of biologically important species. In such cases it is important to consider more accurate algorithms such as HFF.

The HFF based model checking procedure is essentially the same as that outlined in the previous subsection, except that in order to evaluate the truth hood of atomic propositions we run HFF with a suitable number of spikes. Therefore, referring back to the previous subsection, $\mathbf{s}_t \in st(A)$ iff $M^t_{HFF}(i, v) \geq r$ where $A = (i, v) \geq r$ and $M^t_{HFF}(i)$ is the marginal distribution of X^t_i computed by HFF. Similarly $\mathbf{s}_t \in st(A)$ iff $M^t_{HFF}(i, v) \leq r$ in case $A = (i, v) \leq r$. All the other steps are exactly same as for FF based analysis.

6.4 Comparing PCTL with BLTPL

PCTL is the most commonly used logic for reasoning about probabilistic models especially discrete time Markov chains. Since DBNs can be seen as factored representations of Markov chains, it is interesting to compare and contrast PCTL with our logic BLTPL.

Our logic BLTPL is interpreted over marginal probability distributions vectors returned by DBN inference algorithms at each time point. The probabilistic assertions are only encoded at the atomic proposition level. The truth value of these probabilistic assertions is assessed by first, computing the marginal probability distributions of the variables involved in the atomic proposition and then comparing it with the threshold specified in the atomic proposition.

PCTL, as discussed in chapter 2, consists of state formulas and path formulas. State formulas represent formulas that are $true$ or $false$ at a specific state of the Markov chain. Path formulas on the other hand are interpreted over specific paths. Formulas with probabilities are state formulas, however they are interpreted over paths that branch out of a particular state.

It has been shown before that the PCTL* (and therefore PCTL which is a subset of PCTL*) is in general *incomparable* with logics that interpret over probability distributions. We refer the reader to [157, 158] for more details. The basic idea is that although PCTL can be used to reason about paths of a probabilistic system, one cannot specifically add constraints to enforce reasoning about specific time points or steps across these paths. Using a similar line of reasoning, our logic BLTPL is incomparable with PCTL.

6.5 Experimental results

Next, we used our model checking procedure to verify interesting properties on the DBNs which arise as approximations of ODE dynamics. The model checking algorithm has been implemented in C++. All the experiments reported here were carried out on a Opteron 2.2 Ghz processor, with 3 GB memory. In what follows we briefly describe each of the pathways for whom we constructed the DBN approximation and verified the corresponding properties. The ODE models of all the pathways in this section were taken from the BioModels database [117].

The EGF-NGF signaling pathway

The details of the model have been described in Chapter 4. The model consists of 32 differential equations and 48 kinetic parameters. 20 of the 48 parameters were singled

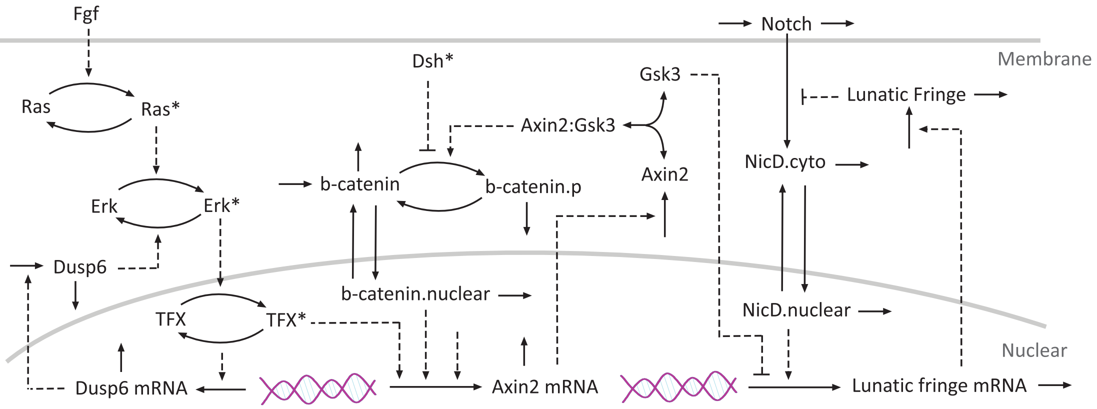


Figure 6.2: Segmentation clock pathway

out to be unknown. The ranges of each variable and unknown parameter were discretized into 5 intervals of equal size. The time step Δt was fixed to be 6 seconds and 3×10^6 trajectories were generated up to 600 seconds to fill up the CPTs associated with the DBN approximation.

The segmentation clock network

During the development of vertebrate embryos, the somites are rhythmically produced to establish the segmentation pattern of the spines. The periodic formation of somites is driven by the oscillatory expression of a large number of genes. The expression of these genes is controlled by an underlying signaling network called the segmentation clock network [159]. The structure of the pathway is shown in figure 6.2. The corresponding ODE model consists of 16 differential equations and 75 kinetic parameters. 39 of the 75 parameters were singled out to be unknown. The ranges of each variable and unknown parameter were discretized into 5 equal-size intervals. The time step Δt was fixed to be 5 minutes while 3×10^6 trajectories were generated up to 500 minutes to fill up the CPTs.

The thrombin-dependent MLC phosphorylation pathway

The endothelial cells form a dynamic barrier between blood and tissues, which plays an important role in various physiological and pathological processes. The barrier function is determined by the contraction of endothelial cells, which is triggered by the MLC phosphorylation and thrombin is an agonist that can induce the MLC phosphorylation through two different signaling cascades [160]. Due to the large size of thrombin-dependent

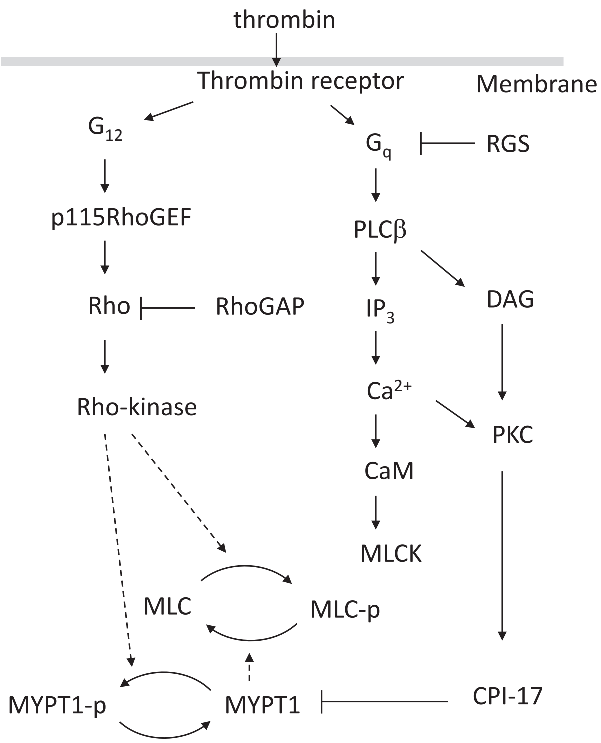


Figure 6.3: The thrombin-dependent MLC phosphorylation pathway

MLC phosphorylation pathway, we only show its major signal transduction events in figure 6.3. This rather large model consists of 105 differential equations, 110 reactions, and 197 kinetic parameters. In constructing the DBN approximation, we singled out 164 of the 197 parameters to be unknown. We discretized the ranges of each variable and unknown parameter into 5 equal-size intervals and fixed the time step Δt to be 2 seconds. To fill up the CPTs, we generated 3×10^6 trajectories up to 200 seconds.

Verification results

For the three case studies we formulated some properties and verified whether they were true or not. For convenience we fixed the values of rate constants and the initial concentrations according to the models taken from the BioModels database[117]. This in turn fixed the truth values of the propositions at time 0.

The EGF-NGF signaling pathway

- It is known that the concentration of EGF and NGF remains constantly high. We formulated this property as the formula:

$$\boldsymbol{G}((EGF, I_4) > 0.9) \wedge \boldsymbol{G}((NGF, I_4) > 0.9)$$

The property was verified to be *true*.

- The profile of activated ERK is expected to reach a peak after which the concentration begins to fall. The corresponding formula was:

  $$(((ERK^*, I_0) > 0.6) \wedge \boldsymbol{F}(((ERK^*, I_3) > 0.6) \wedge \\ \boldsymbol{F}(\boldsymbol{G}((ERK^*, I_2) > 0.6)))$$

  The above query was verified to be *true*.

- We next checked whether the concentration of activated C3G reaches a steady state as experimentally observed. The corresponding formula is:

  $$((C3G^*, I_0) > 0.8) \wedge \boldsymbol{F}(\boldsymbol{G}((C3G^*, I_4) > 0.8))$$

  It was verified to be *true*.

The segmentation clock network

We checked the oscillatory behavior of various species. Following [91], we formulated the property for the oscillatory behavior of Axin as:

$$\boldsymbol{F}(((Axin, I_0) > 0.6) \wedge \boldsymbol{F}(((Axin, I_2) > 0.6) \wedge \boldsymbol{F}(((Axin, I_0) > 0.6) \wedge \boldsymbol{F}(((Axin, I_2) > \\ 0.6) \wedge \boldsymbol{F}((Axin, I_0) > 0.6)))))$$

The property specifies the number of peaks and troughs to be expected in an oscillation cycle within the given time bound of the system. Specifically, it says that initially (with a high probability) the system is at the discretized interval 0 followed by a state some time in future where (with a high probability) the system moves to a higher discretized interval and then falls back to initial levels and so on. This query was verified to be *true*.

The thrombin-dependent MLC phosphorylation pathway

The following are some of the formulas considered for this model:

- The profile of activated Rho starts at a very low level, reaches a high value after which the concentration drops back to the initial level. The corresponding formula was:

$$((Rho^{*}, I_0) > 0.8) \wedge \boldsymbol{F}(((Rho^{*}, I_4) > 0.8) \wedge \\ \boldsymbol{F}((Rho^{*}, I_0) > 0.8)))$$

It was verified to be *true*.

- Rho gets activated and reaches its peak earlier than MLC:

$$((MLC^{*}, I_4) < 0.1)\, \boldsymbol{U}(((Rho^{*}, I_4) > 0.8) \wedge \\ \boldsymbol{O}(\boldsymbol{F}((MLC^{*}, I_4) > 0.7)))$$

This was also verified to be *true*.

- Experimental observations suggest that the concentration of phosphorylated MLC starts at a low level, reaches a high steady state value. The BLTPL formula used to capture this property was:

$$((MLC^{*}, I_0) > 0.7) \wedge \boldsymbol{F}(\boldsymbol{G}((MLC^{*}, I_4) > 0.7))$$

It was verified to be $false$.

- We then formulated a BLTPL formula to describe the behavior where the concentration starts with a low value, reaches a high value (peak) after which it drops back to the initial level.

$$((MLC^{*}, I_0) > 0.7) \wedge \boldsymbol{F}(((MLC^{*}, I_4) > 0.7) \wedge \\ \boldsymbol{F}((MLC^{*}, I_0) > 0.7))$$

This formula evaluated to be *true*. This means the current ODE model is unable to explain the experimental data available for this pathway. Further investigation to identify the missing links of the pathway may be required.

FF is an approximate procedure and hence can incur errors. Finally, to check the accuracy of our FF-based model checking procedure, we used HFF with $32,000$ spikes to infer the marginals for the EGF-NGF and the segmentation clock pathway. All the verification results agreed with FF-based ones except for one formula concerning the profile of activated ERK. This suggests that a good strategy will be to start with a FF-based verification to get an overall picture of the dynamics and then use HFF to improve the accuracy of verification for critical properties.

6.6 Discussion

We have shown in this chapter how algorithms performing DBN inference can be used to for probabilistic model checking on DBNs. We have also formulated a simple probabilistic temporal logic and constructed an approximate but efficient model checking procedure. Though probability enters the picture solely via atomic propositions, one can still formulate many interesting dynamic properties of pathway models. Further, due the fact that there is a *single finite* run, the model checking procedure is particularly simple. Admittedly it is an approximate procedure. The best strategy is to begin with the FF based procedure to get a preliminary feel for the dynamics and in case a biologically crucial property shows up, one can compute its truth value more precisely by using the HFF algorithm.

Chapter 7

Statistical model checking based model calibration

7.1 Introduction

As outlined in the previous chapters, an alternate approach to scalable probabilistic model checking is "statistical model checking". Statistical model checking algorithms work by sampling traces according to the underlying transition probabilities from a stochastic dynamical system model. One then uses statistical tests to ascertain if the drawn samples provide enough evidence to support a probabilistic assertion concerning system satisfying a certain property expressed in temporal logic. In fact, it can be used to verify properties of Markov chains which represent the dynamics induced by the discretization of the value and time domains of the ODEs as described in Chapter 3. The crucial observation that makes this possible is that these large Markov chains need not be explicitly represented. Sampling the initial states and solving the corresponding ODEs, according to the defined discretization scheme, amounts to picking traces from the underlying Markov chain. The model checking procedure no longer depends on the size of the state space of the model. There are different approaches to statistical model checking [161, 130, 131, 18]. Generating traces from ODEs constitutes the most expensive operation. Hence, for repeated analysis tasks many traces have to be generated. In such cases DBNs can act as a much more efficient system to work with, since they provide a succinct representation of the dynamics and can be analyzed efficiently with probabilistic inference algorithms.

Our focus in this chapter is on the applications of statistical model checking for model analysis. Specifically, we propose a novel application of statistical model checking for calibration of biopathway models represented by ODEs. The main considerations w.r.t parameter estimation of ODE models is that the time series data will report the concentration levels of only a few proteins observed at a small number of time points. It will be of limited precision and often averaged over a population of cells. Equally important, the initial concentration levels of the various proteins will also not be available as point values but as interval of values due to cell-to-cell variability. Consequently, when numerically simulating the ODE model, one must resort to Monte Carlo methods to ensure that sufficiently many values from the relevant intervals are being sampled. As a result, parameter estimation will require the generation of a large number of trajectories. Furthermore the number of trajectories generated in each round must be chosen in an ad hoc way. To get around these issues, we use a statistical model checking based approach here.

For the parameter estimation problem we first recall that the goal is to compute the values of unknown parameters so that the resulting model can reproduce the experimental observations and make reliable predictions about behaviors that were not used to fit the parameters. A common approach is to iteratively optimize the agreement between the behavior generated by a parameter set and available experimental data by searching through the space of parameter set values. Typically, the goodness-of-fit of a parameter combination is evaluated by the weighted sum of square error between model prediction and experimental data captured. The two major steps of the optimization algorithm are: (i) "guess" the values of the parameters (ii) evaluate the goodness-of-fit of the guessed values. For step (i), guesses may be generated randomly in the first round but later guesses are guided by the results of previous rounds based on various search strategies. For step (ii), one numerically simulates the ODE system up to the maximum time point for which experimental observations are available. The algorithm is terminated if a sufficiently good fit to data has been achieved or if the computational resources allocated for the task have been exhausted. We propose to use statistical model checking to implement step (ii). We use a mild variant of the probabilistic linear temporal logic PBLTL [18] to formalize both experimental time series data and dynamic trends about pathway behaviors. For the current set of parameter values we evaluate its goodness on the family of trajectories obtained by sampling from the distribution of initial conditions followed by numerical simulations. There is usually substantial cell-to-cell variability in terms of the initial states of different components [17]. Hence it is more appropriate to assume that the initial concentrations of the various species take their values according to a distribution over a set of initial states. Our specifications will state the bounded amounts of errors that can be incurred when matching the simulated behaviors with the data points. In addition we can also include prior knowledge about the qualitative behavior of the pathway such as bi-stability or whether certain time profiles are transient or oscillatory. In addition, the SPRT components of our test [18] -including its statistical nature- also caters for the uncertainties concerning the data. Finally, the SPRT component also determines the number of trajectories that are used to evaluate the goodness of the current set of parameters instead of fixing this number in an arbitrary way. It also in a sense guarantees the statistical strength of the estimation procedure. In this sense our approach deals in a principled manner with the multiple uncertainties surrounding the parameter estimation problem in biological settings.

7.1.1 Related work

There have been some previous attempts to calibrate and analyze pathway models using model checking methods. For instance, the work reported in [97] focuses on randomly sampling the set of unknown parameters and accepting the set of parameters if the simulation trace satisfies a LTL formula that specifies the desired qualitative properties of the system. A similar approach is taken in [100] where a large number of parameter values sets are sampled and the associated simulation traces are verified in an on-line fashion. However, both these studies lack a principled search method and instead rely on a brute force strategy to sample the parameter space. Typically, the parameter search space is high dimensional and hence such strategies would need an impractically large number of samples for realistic pathways. The work reported in [99] considers parameter estimation on a single simulation trace, and incorporates an evolutionary strategy based search algorithm to guide the search, in a deterministic setting. Studies such as [147, 98] carry out parameter estimation on restricted ODEs systems called multi affine systems. Here one first constructs a symbolic representation of the dynamics followed by parameter estimation using symbolic model checking. The large state space of even relatively small pathways and the focus on multi affine ODEs systems severely restrict the applicability of this approach. Probabilistic model checking is used for parameter estimation in [148] where the logic called PLTLc is used to specify properties. A genetic algorithm is used to search for the best set of parameters. A fixed number of samples are generated and the probability of satisfying a property is calculated to be the fraction of the samples which satisfy the property. No attempt is made to validate the quality of the estimated parameters.

Our work is different in the following aspects: We use a statistical model checking framework for parameter estimation. In our specifications we encode both experimental data as point values with confidence intervals and prior qualitative knowledge of the dynamics. We use an on-line model checking algorithm which often terminates before the whole simulation trace is generated and this considerably improves performance. Further our statistical model checking fixes the number of samples to be drawn in a principled way and we can provide statistical guarantees concerning the goodness of a parameter set. Last but not least our method quantitatively factors in the cell-to-cell variability of the initial states as well as the noisiness and limited precision of experimental data.

7.1.2 ODEs based model behaviors

We recall some of the notations developed in the previous sections about ODE systems, there one equation of the form $\frac{dy_i}{dt} = f(\mathbf{y}, \mathbf{r})$ for each molecular species y_i, with f describing the kinetics of the reactions that produce and consume y_i, while $\mathbf{y}$ is the set (vector) of molecular species taking part in these reactions and $\mathbf{r}$ are the rate constants associated with these reactions. The range of values of each variable y_i is assumed to take values in $[v_i^{min}, v_i^{max}]$, v_i^{min} and v_i^{min} non negative rational numbers. Hence the state space of the system will be $\mathbf{V} = [v_1^{min}, v_1^{max}] \times [v_2^{min}, v_2^{max}] \ldots \times [v_n^{min}, v_n^{max}] \subseteq \mathbb{R}_+^n$ where $\mathbb{R}_+$ denotes the set of non-negative reals. Thus $\mathbf{V}$ will be a bounded subset of $\mathbb{R}_+^n$. To capture the cell-to-cell variability and uncertainties regarding the initial states we define for each variable y_i an interval $[v_i^{min:init}, v_i^{max:init}]$ with $v_i^{min} \leq v_i^{min:init} < v_i^{max:init} \leq v_i^{max}$. We set $INIT = [v_1^{min:init}, v_1^{max:init}] \times [v_2^{min:init}, v_2^{max:init}] \ldots \times [v_n^{min:init}, v_n^{max:init}]$. In what follows it will be convenient to represent our system of ODEs in vector form as : $\frac{d\mathbf{y}}{dt} = F(\mathbf{y})$ with $\mathbf{y} = (y_1, y_2, \ldots, y_n)$ and $F(\mathbf{y}(i)) = f_i$.

A function $f : \mathbf{V} \to \mathbb{R}$ is a C^1 function if f', the derivative of f exists at all $\mathbf{v} \in \mathbf{V}$ and is a continuous function. Given the fact that each f_i in our ODEs system is composed out of rational functions we can assume that $f_i \in C^1$ for each i and hence $F : \mathbf{V} \to \mathbf{V}$ is also a C^1 function.

Given $\mathbf{v} \in \mathbf{V}$ the system of ODEs will have a unique solution since $F \in C^1$ [35]. We shall denote this solution by $\mathbf{Y}_\mathbf{v}(t)$. It will satisfy $\mathbf{Y}_\mathbf{v}(0) = \mathbf{v}$ and $\mathbf{Y}'_\mathbf{v}(t) = F(\mathbf{Y}(t))$. We are guaranteed that $\mathbf{Y}(t)$ is a C^0-function (i.e. continuous function) [35]. This fact will be crucial when we later turn to probabilistic verification.

It will be convenient to define the flow $\Phi : \mathbb{R}_+ \times \mathbf{V} \to \mathbf{V}$ of $\mathbf{Y}'_\mathbf{v} = F(\mathbf{Y})$ for arbitrary initial vectors $\mathbf{v}$. Intuitively, $\Phi(t, \mathbf{v})$ is the state reached under the ODEs dynamics if the system starts at $\mathbf{v}$ at time 0. The flow will be the C^0-function given by: $\Phi(t, \mathbf{v}) = \mathbf{Y}_\mathbf{v}(t)$. Thus $\Phi(0, \mathbf{v}) = \mathbf{X}(0) = \mathbf{v}$ and $\partial(\Phi(t, \mathbf{v}))/\partial t = F(\Phi(t, \mathbf{v}))$ for all t [35]. Further, $\Phi(t, \cdot)$ will be bijective and will satisfy $\Phi(t + t', \mathbf{v}) = \Phi(t, \Phi(t', \mathbf{v}))$ for every t, t' in $\mathbb{R}$. In what follows we will often write $\Phi_t(\mathbf{v})$ instead of $\Phi(t, \mathbf{v})$.

In our application the dynamics will be of interest only up to a maximal time point T. Fixing such a T we define a *trajectory* starting from $\mathbf{v} \in \mathbf{V}$ denoted $\sigma_\mathbf{v}$ to be the (continuous) function $\sigma_\mathbf{v} : [0, T] \to \mathbf{V}$ satisfying: $\sigma_\mathbf{v}(t) = \Phi_t(\mathbf{v})$. Then BEH, the behavior of our dynamical system, is the set of trajectories given by: $BEH = \{\sigma_\mathbf{v} \mid \mathbf{v} \in INIT\}$. Our goal is to probabilistically verify the dynamical properties of BEH.

7.2 Statistical model checking of ODEs dynamics

Statistical model checking sampling traces according to the underlying transition probabilities from a stochastic dynamical system model. One then uses statistical tests to ascertain if the drawn samples provide enough evidence to support a probabilistic assertion concerning the system satisfying a certain property. There are different approaches to statistical model checking [161, 130, 131, 18]. In the current work we focus on a sequential hypothesis testing method [130]. However, other approaches can also be easily incorporated into our analysis algorithms. We start with our specification logic.

7.2.1 Bounded linear time temporal logic

Since our trajectories will be of bounded duration it will suffice to use temporal logic known as bounded linear time temporal logic (BLTL). An atomic proposition in our logic will be of the form (i, l, u) with $L_i \leq l < u \leq U_i$. Such a proposition will be interpreted as "the current concentration level of y_i falls in the interval $[l, u]$. We fix a finite set of such atomic propositions $AP = \{A_1, ...A_k\}$. The formulas of BLTL are:

- Every atomic proposition as well as the constants *true*, *false* are BLTL formulas;
- If ψ is a BLTL formula then $\sim \psi$ and $\psi \vee \psi'$ are BLTL formulas.
- If ψ is a BLTL formula then $\mathbf{O}(\psi)$ is a BLTL formula.
- If ψ, ψ' are BLTL formulas and t is a positive integer then $\psi \mathbf{U}^{\leq \mathbf{t}} \psi'$ and $\psi \mathbf{U}^{\mathbf{t}} \psi'$ are BLTL formulas.

The derived propositional operators such as $\wedge$, $\supset$, $\equiv$ and the temporal operators $\mathbf{G}^{\leq \mathbf{t}}$, $\mathbf{F}^{\leq \mathbf{t}}$, $\mathbf{F}^{\mathbf{t}}$ are defined in the usual way. We have mildly strengthened PBLTL so that we can say that exactly at time t from now a certain property will hold. As we show in the next section, this will enable us to encode experimental data in the logical specification when solving the parameter estimation problem.

We will interpret the formulas of our logic at the finite set of time points $\mathcal{T} = \{0, 1, \ldots, T\}$. We do so since experimental data will be available only at a finite number discrete time points. We assume T has been chosen such that it exceeds the last time for which experimental data is available. Secondly, high dimensional ODEs systems will not admit a closed form solution and hence trajectories will have to be generated through numerical simulations and hence will have values defined only at a bounded number discrete time points. Hence it suffices to work with a sufficiently large but finite and discrete time domain $\mathcal{T}$. We assume that the unit of time interval has been chosen appropriately and it includes all the relevant time points such as those mentioned in the formula. Further, we have assumed here only for convenience that the time points are spaced evenly. The semantics of the logic is defined in terms of the relation $\sigma, k \models \varphi$ where σ is a trajectory in BEH and $t \in \mathcal{T}$.

Hence, we will define the semantics via the relation $\mathbf{v}, t \models \varphi$ for $\mathbf{v} \in INIT$, with the understanding that $\mathbf{v}$ stands for the trajectory $\sigma_{\mathbf{v}}$.

- $\sigma, t \models (i, l, u)$ iff $l \leq \sigma(t)(i) \leq u$ where $\sigma(t)(i)$ is the ith component of the n-dimensional vector $\sigma(t) \in V$.
- $\sigma, t \models \sim \psi$ iff $\sigma, t \not\models \psi$.

- $\sigma, t \models \psi \vee \psi'$ iff $\sigma, t \models \psi$ or $\sigma, t \models \psi'$.
- $\sigma, t \models \mathbf{O}(\psi)$ iff $\sigma, t+1 \models \psi$, $t < T$.
- $\sigma, t \models \psi \mathbf{U}^{\leq \mathbf{k}} \psi'$ iff there exists k' such that $k' \leq k$, $t + k' \in \mathcal{T}$ and $\sigma, t + k' \models \psi'$. Further $\sigma, t + k'' \models \psi$ for every $0 \leq k'' < k'$.
- $\sigma, t \models \psi \mathbf{U}^{\mathbf{k}} \psi'$ iff $\sigma, t + k \models \psi'$. Further $\sigma, t + k' \models \psi$ for every $0 \leq k' < k$.

As usual, we define $models(\psi) = \{\sigma | \sigma, 0 \models \psi,\, \sigma \in BEH\}$.

Probabilistic BLTL

Next we wish to make statements of the form $P_{>0.9}(\psi)$ where the intended meaning is that the "fraction" of trajectories in BEH that fall in $models(\psi)$ exceeds 0.9. To assign precise meaning such a statement we need to define a probability measure over sets of trajectories. Note however that $\sigma \in BEH$ is completely determined by $\sigma(0)$, the (vector) value it assumes at $t = 0$. Hence we will identify BEH with $INIT$, the set of initial states. To make this explicit we define $Models(\psi) \subseteq INIT$ as: $\mathbf{v} \in Models(\psi)$ iff $\sigma \in models(\psi)$ and $\sigma(0) = \mathbf{v}$.

To assign a probability to $Models(\psi)$ we construct a probability measure over the standard σ-algebra generated by the open intervals contained in $INIT$. To make this more precise, recall that $INIT = \prod_{i=1}^{n}[v_i^{min:init}, v_i^{max:init}]$. Then $\mathcal{B}(INIT)$ -written for convenience as just $\mathcal{B}$ below- is the smallest subset of 2^{INIT} satisfying:

- Suppose $v_i^{min:init} \leq l_i < u_i \leq v_i^{max:init}$ for each i. Then $\prod_{i=1}^{n}(l_i, u_i) \in \mathcal{B}$
- $INIT \in \mathcal{B}$
- If $B \in \mathcal{B}$ then $\overline{B} = INIT - B \in \mathcal{B}$.
- If $\{B_1, B_2, \ldots B_k \ldots\}$ is a countable family of sets in $\mathcal{B}$ then $\bigcup_i B_i \in \mathcal{B}$.

The probability measure we define over $\mathcal{B}$ will be based on the assumption that each initial state in $INIT$ is equally likely to be assumed by the system. This so called uniform distribution assumption is made when there is no prior knowledge about which initial states are more likely to be assumed by the pathway under study. However, when such information is available it can be incorporated into our method in a straightforward fashion. Here we make this assumption only for technical convenience. Now suppose

$\prod_{i=1}^{n}(l_i, u_i) \in \mathcal{B}$. We define $P(\prod_{i=1}^{n}(l_i, u_i)) = \prod_{i}^{n} \frac{u_i - l_i}{v^{max:init} - v^{min:init}}$. It is a standard fact that P extends in a unique way to the probability measure $P : \mathcal{B} \rightarrow [0, 1]$ such that $P(INIT) = 1$ and $P(\emptyset) = 0$. Our goal now is to show that $Models(\psi) \in \mathcal{B}$ for every formula ψ. This will then ensure that $P(\models (\psi)$ is well-defined.

Let ψ be a formula and $t \in \mathcal{T}$. Then $\|\psi\|_t \subseteq INIT$ is defined inductively as follows.

- $\|(i, l, u)\|_t = \{\mathbf{v} \mid \sigma_{\mathbf{v}}, t \models (i, l, u)\}$. Recall that $\sigma_{\mathbf{v}}$ is the trajectory in BEH with $\sigma_{\mathbf{v}}(0) = \mathbf{v}$.

- $\| \sim \psi\|_t = INIT - \|\psi\|_t$

- $\|\psi \vee \psi'\|_t = \|\psi\|_t \cup \|\psi'\|_t$

- $\|\psi \mathbf{U}^{\leq \mathbf{k}} \psi'\|_{\mathbf{t}} = \bigcup_{\mathbf{k'} \leq \mathbf{k}, \mathbf{t} + \mathbf{k'} \leq \mathbf{T}} (\|\psi'\|_{\mathbf{t}+\mathbf{k'}} \cap (\bigcap_{\mathbf{0} \leq \mathbf{k''} < \mathbf{k'}} \|\psi\|_{\mathbf{t}+\mathbf{k''}}))$

- $\|\psi \mathbf{U}^{\mathbf{k}} \psi'\|_{\mathbf{t}} = (\|\psi'\|_{\mathbf{t}+\mathbf{k}} \cap (\bigcap_{\mathbf{0} \leq \mathbf{k'} < \mathbf{k}} \|\psi\|_{\mathbf{t}+\mathbf{k'}})$

We now recall that due to the assumption that each f_i is a C^1 function, the flow derived from the solution to the ODEs is guaranteed to be a continuous function. Consequently $\Phi_t : \mathbf{V} \rightarrow \mathbf{V}$ is also a continuous function for every $t \in [0, T]$. This in turn implies Φ_t is in fact a *measurable* function in the sense if $B \in \mathcal{B}$ then $\Phi_t^{-1}(B) = \{\mathbf{v} \mid \Phi_t(\mathbf{v}) \in B\}$ is a member of $\mathcal{B}$. This fact will play a crucial role in establishing the following result.

Theorem 7.2.1. *Let ψ be a formula and $t \in \mathcal{T}$. Then the following statements hold.*

1. $\|\psi\|_t \in \mathcal{B}$.

2. $Models(\psi) = \|\psi\|_0$.

3. $Models(\psi) \in \mathcal{B}$.

Proof. To prove the first part by structural induction, we note that $\{\mathbf{v} | l \leq vv(i) \leq u\} = \prod_{j=1}^{n}(l_j, u_j)$ where $l_j = L_j$ and $u_j = U_j$ if $j \neq i$ and $l_j = l$ and $u_j = u$ if $j = i$ and hence $B \in \mathcal{B}$ where for convenience we set $B = \{\mathbf{v} | l \leq vv(i) \leq u\}$. From the definitions it follows that $\mathbf{v'} \in \|(i, l, u)\|_t$ iff $\sigma_{\mathbf{v'}}, t \models (i, l, u)$ iff $l \leq \Phi_t(\mathbf{v'}) \leq u$ iff $\Phi_t(\mathbf{v'}) \in B$. This shows that $\|(i, l, u)\|_t = \Phi_t^{-1}(B)$ and since Φ_t is measurable we are assured that $\Phi_t^{-1}(B) \in \mathcal{B}$.

Next we note that $\|\psi\|_t, \|\psi'\|_t \in \mathcal{B}$ then $\| \sim \psi\|_t \in \mathcal{B}$ and $\|\psi \vee \psi'\|_t \in \mathcal{B}$ since $\mathcal{B}$ is closed under complementation and (countable) union. Similarly from $\|\psi\|_t, \|\psi'\|_t \in \mathcal{B}$ we can conclude that $\|\psi \mathbf{U}^{\leq \mathbf{k}} \psi'\|_{\mathbf{t}}, \|\psi \mathbf{U}^{\mathbf{k}} \psi'\|_{\mathbf{t}} \in \mathcal{B}$ since $\mathcal{B}$ is closed under countable intersections as well. The remaining two parts of the result follow from the definitions. □

We can now define the formulas of PBLTL as:

- $P_{\geq r}\psi$ and $P_{\leq r'}\psi$ are PBLTL formula provided $r \in [0, 1)$, $r' \in (0, 1]$ and ψ is a BLTL formula.
- If φ and φ' are PBLTL formulas then so are $\sim \varphi$ and $\varphi \vee \varphi'$.

We shall say that $\mathcal{S}$, the system of ODEs meets the specification $P_{\geq r}\psi$ -and this denoted $\mathcal{S} \models P_{\geq r}\psi$ - iff $P(Models(\psi)) \geq r$ while $\mathcal{S} \models P_{\leq r'}\psi$ iff $P(Models(\psi)) \leq r'$. The clauses for negation and disjunction are defined in the obvious way. Our goal now is to construct a statistical model checking procedure based on sequential hypothesis testing to verify PBLTL specifications.

7.2.2 Statistical model checking of PBLTL formulas

According to [130], whether $\mathcal{S} \models P_{\geq r}\psi$ can be formulated as a sequential hypothesis test between the null hypothesis H0 : $p \geq r + \delta$ against the alternative hypothesis H1 : $p < r - \delta$ where $p = P(Models(\psi))$. Here, δ is the indifference region supplied by the user. The *strength* of the test is decided by parameters α and β which represent the Type-1 and Type-2 errors respectively. Thus the verification is carried out approximately but with guaranteed confidence levels and error bounds.

The test proceeds by generating a sequence of sample trajectories $\sigma_1, \sigma_2, \ldots$ by randomly sampling an initial state from $INIT$ and assume a corresponding sequence of Bernoulli random variables $Z_1, Z_2 \ldots$ where each Z_k takes the value 1 with probability p and the value 0 with probability $1 - p$. For each trajectory σ_k we check if $\sigma_k, 0 \models \psi$ (and therefore if $\sigma_k(0) \in Models(\psi)$. After drawing m samples we compute a quantity f_m as:

$$f_m = \frac{[r-\delta]^{(\sum_{i=1}^{m} y_i)}[1-[r-\delta]]^{(m-\sum_{i=1}^{m} y_i)}}{[r+\delta]^{(\sum_{i=1}^{m} y_i)}[1-[r+\delta]]^{(m-\sum_{i=1}^{m} y_i)}} \tag{7.1}$$

Hypothesis H0 is accepted if $f_m \geq \widehat{A}$, and Hypothesis H1 is accepted if $f_m \leq \widehat{B}$. If neither is the case then another sample is drawn. The constants $\widehat{A}$ and $\widehat{B}$ are so chosen such that it results in a test of strength $\langle \alpha, \beta \rangle$. In practice, a good approximation turns out to be $\widehat{A} = \frac{1-\beta}{\alpha}$ and $\widehat{B} = \frac{\beta}{1-\alpha}$.

Online model checking to verify properties specified in PBLTL

Given a PBLTL formula of the form $P_{\geq r}(\psi)$, where ψ is an BLTL formula and $r \in [0, 1]$, we use the statistical model checking algorithms outlined before to check if the formula holds for the system with the thresholds specified using r. The most resource intensive task in the model checking procedure is simulating the ODEs. Typically, to verify a simulation trace, one generates the whole ODE simulation trace (for the time frame of interest) before applying the model checking procedure on it (off-line approaches). Instead one can combine simulation and model checking together i.e simulate the ODE system only until the model checker can make a decision. This approach – known as online method– has the advantage of saving computational resources and the over head of storing the trajectories before applying model checking.

We use a tableau based online model checking procedure. Online approaches have the advantage of conserving CPU, memory resources and have a lower amortized time complexity. The method relies on constructing and propagating a finite family of sets $\mathcal{F}$. Each set $F_i \in \mathcal{F}$ contains a finite number of formulas. Let φ, ψ and γ be BLTL formulas. A literal is defined as an atomic proposition $A \in AP$ or its negation $\sim A$. For the purpose of illustration, let us assume that we convert the given BLTL formulas into a form in which only the atomic propositions can appear in negated form (in Negative Normal Form). Any formula can be converted to this form in a straight forward procedure.

For a family of sets $D = \{D_1, D_2, ..., D_j\}$, where each D_i is a set of formulas, we first define the $\bigotimes$ operation. Suppose $D1$ and $D2$ are two such families. Then $D1 \bigotimes D2 = \{Y1 \cup Y2 | Y2 \in D1, Y2 \in D2\}$.

For a formula φ, we define the family of closure sets $cl(\varphi)$ by structural induction on φ using:

- If φ is a truth constant or a literal then $cl(\varphi)$ =$\{\{\varphi\}\}$.
- If $\varphi = \psi \vee \gamma$ then $cl(\varphi) = cl(\psi) \cup cl(\gamma)$.
- If $\varphi = \psi \wedge \gamma$ then $cl(\varphi) = cl(\psi) \bigotimes cl(\gamma)$.
- If $\varphi = \mathbf{O}\psi$ then $cl(\varphi) = \{\{\mathbf{O}\psi\}\}$.

- If $\varphi = \mathbf{F}\psi$ then $cl(\varphi) = cl(\psi) \cup cl(\mathbf{OF}\psi)$.

- If $\varphi = \mathbf{G}\psi$ then $cl(\varphi) = cl(\psi) \bigotimes cl(\mathbf{OG}\psi)$.

- If $\varphi = \psi\mathbf{U}^{\leq \mathbf{k}}\gamma$ then $cl(\varphi) = cl(\gamma) \cup (cl(\psi) \bigotimes cl(\boldsymbol{O}(\psi\mathbf{U}^{\leq \mathbf{k}}\gamma)))$.

If we have a set of formulas $W = \{\varphi_1, \varphi_2, \ldots, \varphi_n\}$, then the closure $cl(W)$ can be written as $cl(W) = cl(\varphi_1) \bigotimes cl(\varphi_2) \ldots \bigotimes cl(\varphi_n)$. We can also extend the notion of closure to families of sets of formulas such as $\mathcal{F} = \{W_1, W_2, \ldots, W_k\}$, and say that the closure set of $\mathcal{F}$ is $cl(\mathcal{F}) = cl(W_1) \cup cl(W_2) \ldots cl(W_k)$. We call the set of formulas W a *leaf set* iff $cl(W) = W$. Further, a set W is *inconsistent* iff (i) for an atomic proposition A, $A \in W$ and $\sim A \in W$ or (ii) for some formula φ, both $\mathbf{O}\varphi \in W$ and $\mathbf{O} \sim \varphi \in W$.

Proposition: The following assertions hold.

- W is a leaf set iff each formula in W is a literal or a $\mathbf{O}$ formula.

- $cl(\varphi)$ is a leaf family for each φ.

- $cl(W)$ is a leaf family for every finite set of formulas W.

- $cl(\mathcal{F})$ is a leaf family for every family of formula sets $\mathcal{F}$.

Suppose the current system state is s_t. If W is a leaf set then W is *dead* at time t iff W is inconsistent or $\sigma, t \not\models \ell$ for some literal $\ell \in W$. Consequently, a family of leaf sets $\mathcal{F}$ is dead iff $\forall W_i \in \mathcal{F}$: W_i is dead. Furthermore, $\mathcal{F}$ is terminal iff $\exists W_i \in \mathcal{F} : W_i$ is not dead and $next(W_i) = \emptyset$, where $next(W_i) = \{\psi | \mathbf{O}\psi \in W_i\}$.

Now assume we are given a formula φ and want to check if the system trajectory satisfies φ. We propagate a family of sets and start with $\mathcal{F}^0 = cl(\varphi)$. Inductively, assume that we are given the family of sets $\mathcal{F}^t$ for $t < T$. If $\mathcal{F}^t$ is dead, then we set $\mathcal{F}^{t+1} = false$, and if $\mathcal{F}^t$ is terminal then we set $\mathcal{F}^{t+1} = true$. Otherwise, $\mathcal{F}^t$ is neither dead nor terminal. In this case we know that $\exists W_1, W_2, ..., W_k \in \mathcal{F}^t, k \geq 1$ which are not dead. Since these sets are not dead, we know that $\forall i, 1 \leq i \leq k : next(W_i) \neq \emptyset$. We can then build the family of sets for time $t + 1$ as $\mathcal{F}^{t+1} = cl(next(W_1)) \cup cl(next(W_2)) \ldots \cup cl(next(W_k))$. The process terminates at time $t < T$ if $\forall W \in \mathcal{F}^t$ is $false$ and returns $s(0) \not\models \varphi$ or if $\exists W \in \mathcal{F}^t$ which is $true$, and returns $\sigma, 0 \models \varphi$. Furthermore, if $t = T$, if $\mathcal{F}^t$ is a terminal leaf family at time point T, the process terminates and returns that $\sigma, t \models \varphi$. Otherwise it returns $\sigma, 0 \not\models \varphi$.

Specifying dynamics using PBLTL

In this section we describe how our knowledge about the dynamics of the systems can be encoded as a BLTL formula.

First we consider the case when we have experimental time course data. Let $O \subseteq \{x_1, x_2, \ldots, x_n\}$ be the set of variables for which experimental data is available and which has been fixed as training data to be used for parameter estimation. Assume $\mathcal{T}_i = \{\tau^i_1, \tau^i_2, \ldots, \tau^i_{T_i}\}$ are the time points at which the concentration level of x_i has been measured and reported as $[\ell^i_t, u^i_t]$ for each $t \in \mathcal{T}_i$. Here the interval $[\ell^i_t, u^i_t]$ is so chosen that it reflects the noisiness, the limited precision and the cell-population-based nature of the experimental data. For each $t \in \mathcal{T}_i$ we define the formula $\psi^t_i = \mathbf{F^t(i, \ell^i_t, u^i_t)}$. Then $\psi^i_{exp} = \bigwedge_{t \in \mathcal{T}_i} \psi^t_i$. We then set $\psi_{exp} = \bigwedge_{i \in O} \psi^i_{exp}$. In case the species x_i has been measured under multiple experimental conditions, then the above encoding scheme is extended in the obvious way.

Often qualitative dynamic trends will be available – typically from the literature – for some of the molecular species in the pathway. For instance, we may know that a species shows transient activation in which its level rises in the early time points and later falls back to initial levels. Similarly, a species may be known to show oscillatory behavior with certain characteristics. Such information can be described as BLTL formulas that we term to be *trend* formulas. We let ψ_{qlty} to be the conjunction of all the trend formulas.

Finally we fix the PBLTL formula $P_{\geq r}(\psi_{exp} \wedge \psi_{qlty})$, where r will capture the confidence level with which we wish to assess the goodness of the fit of the current set of parameters to experimental data and qualitative trends. We also fix an indifference region δ and the strength of the test (α, β). The constants r, δ, α and β are to be fixed by the user. In our application it will be useful to exploit the fact that both ψ_{exp} and ψ_{qlty} are conjunctions and hence can be evaluated separately. As shown in [130, 162], one can choose the strength of each of these tests to be $(\frac{\alpha}{J}, \beta)$, where J is the total number of conjuncts in the specification. This will ensure that the overall strength of the test is (α, β). Further, the results for the individual statistical tests can be used to compute the objective function associated with the global search strategy, as detailed below.

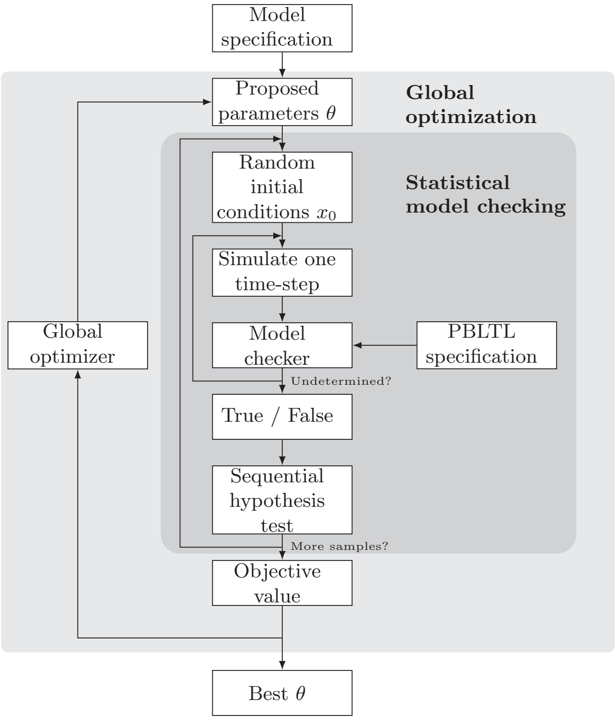


Figure 7.1: Statistical model checking based parameter estimation

7.2.3 Parameter estimation using statistical model checking

Let $\theta = \{c_1, c_2, \ldots, c_K\}$ be the set of unknown rate constants whose values we wish to estimate. The outer loop of our parameter estimation procedure will run as follows. We shall assume for convenience that the search strategy uses a single set of parameter values (one for each unknown rate constant) in each round. Figure 7.1 illustrates the process.

(i) Fix θ_0, which assigns a value to each unknown rate constant. This represents the initial guess. Set $\ell = 0$.

(ii) With θ_ℓ as the current set of rate constant values, run the statistical model checking procedure to verify the individual conjuncts of $\psi_{exp} \wedge \psi_{qlty}$ with the chosen strengths.

(iii) Based on the answers returned by these tests compute $Obj(\theta_\ell)$, where Obj is the objective function.

(iv) Check if the value of the objective function is sufficiently high or ℓ has reached a predetermined bound.

(v) If yes, return θ_ℓ as the estimated value.

(vi) Else fix a new set of rate constant values $\theta_{\ell+1}$ as dictated by the search strategy. Increment ℓ to $\ell+1$ and return to step (ii).

The objective function is formed as follows. Let θ be an assignment of values to the unknown rate constants. Let J^i_{exp} $(= T_i)$ be the number of conjuncts in ψ^i_{exp} and J_{qlty} the number of conjuncts in ψ_{qlty}. Let $J^{i,+}_{exp}(\theta)$ be the number of formulas of the form ψ^t_i (a conjunct in ψ^i_{exp}) such that the statistical test for $P_{\geq r}(\psi^t_i)$ accepts the null hypothesis (that is, $P_{\geq r}(\psi^t_i)$ holds) with the strength $(\frac{\alpha}{J}, \beta)$, where $J = J_{qlty} + \sum_{i \in O} J^i_{exp}$. Similarly, let $J^+_{qlty}(\theta)$ be the number of conjuncts in ψ_{qlty} of the form $\psi_{\ell,qlty}$ that pass the statistical test $P_{\geq r}(\psi_{\ell,qlty})$ with the strength $(\frac{\alpha}{J}, \beta)$. Then $Obj(\theta)$ is computed via:

$$Obj(\theta) = J^+_{qlty}(\theta) + \sum_{i \in O} \frac{J^{i,+}_{exp}(\theta)}{J^i_{exp}} \tag{7.2}$$

Thus the goodness to fit of θ is measured by how well it agrees with the qualitative properties as well as the number of experimental data points with which there is acceptable agreement. To avoid over-training the model, we do not insist that every qualitative property and every data point must fit well with the dynamics predicted by θ. It is possible to introduce additional terms to the objective function in order to speed up convergence in practice. We add the term $\frac{\sum_{k=1}^{(J_{qlty}+\sum_{i \in O} J^i_{exp})} \frac{n^+_k}{n_k}}{J_{qlty}+\sum_{i \in O} J^i_{exp}}$ to our original objective value. Here n^+_k, n_k denote the number of sample trajectories evaluating to *true* and the total number of samples needed to verify the k^{th} PBLTL formula.

The search strategy deployed in step (vi) above will use the values $Obj(\theta_\ell)$ to traverse the space of candidate parameter vectors. The search method can be *local* or *global*. Local methods such as the Levenberg-Marquardt algorithm [64] have the advantage of converging fast, but can get stuck in local minima. Global methods such as Genetic Algorithms (GA) [163], and Stochastic Ranking Evolutionary Strategy (SRES) [69] – although computationally more intensive – are much better at avoiding local minima and *in principle* monotonically improve the estimates in proportion to the computational effort.

In practice, global methods usually maintain a *set* of parameter value vectors in each round. Each round is called a *generation* and the current set of parameter value vectors is called a *population*. Here, for the sake of convenience, we have explained the basic structure of the algorithm by pretending that each population is a singleton. We use the

SRES strategy in our work since it is known to perform well in the context of pathway models [70]. The particular choice of search algorithm, however, is orthogonal to our proposed method.

7.3 Results

We discuss the application of our method using four case studies. The first model is the repressilator pathway where we show that the model can be trained to reproduce oscillations with specified properties. Next, we consider the EGF-NGF pathway, where only quantitative experimental data was used to calibrate the model. Next, we look at the segmentation clock pathway, where we use a combination of both dynamic trend based properties and experimental data to calibrate and analyze the model.

The key parameters used for the statistical model checking algorithm were $\alpha = 0.05$, $\beta = 0.05$, $\delta = 0.05$ for all the experiments. All experiments reported here were carried out on a PC with a 3.4Ghz i7 processor with 8GB RAM. The framework is implemented in MATLAB and C++. ODE systems are numerically solved using the SUNDIALS CVODE package [164], which is integrated into our framework using wrappers from [165, 166]. The code has been optimized to take advantage of the multi-core architecture of modern hardware, the experiments results shown here have been run on 8 threads.

For the pathways reported in this section, we considered global optimization with stochastic ranking evolutionary search (SRES). Additional details of the case studies are described in the appendix.

7.3.1 The repressilator pathway

The repressilator is a synthetic gene network originally introduced by [167]. The network consists of three genes linked in an inhibitory cycle. The ODE model of the pathway, consists of 3 mRNA transcripts and 3 associated protein products. m_1, m_2, m_3 represent mRNA transcripts of 3 genes and p_1, p_2, p_3 are the protein products for each mRNA respectively.

$$\begin{aligned}\frac{\mathrm{d}m_i}{\mathrm{d}t} &= -\gamma m_i + \frac{\alpha}{1 + kp_j^n} \\ \frac{\mathrm{d}p_i}{\mathrm{d}t} &= \beta(m_i - p_i)\end{aligned}$$

Parameter	range	Parameter estimate(SRES)
$\alpha 1$	$[0, 100]$	80.0087
$\alpha 2$	$[0, 100]$	92.04954
$\alpha 3$	$[0, 100]$	56.14092
$\gamma 1$	$[0, 200]$	168.5096
$\gamma 2$	$[0, 200]$	176.3156
$\gamma 3$	$[0, 200]$	140.9322
$k1$	$[0, 16]$	6.883317
$k2$	$[0, 16]$	7.521114
$k3$	$[0, 16]$	12.6742

Table 7.1: Repressilator pathway: Unknown parameters with range and parameter estimation results

Depending on the values of the parameters $\alpha, \beta, \gamma, k, n$, the protein products show sustained oscillations.

Parameter estimation We assumed 9 parameters corresponding to the parameters α, γ and k for each of the mRNA transcripts to be unknown. By specifying the properties of the oscillations (see Table 7.2), we attempt to recover 9 unknown parameters. It is interesting to note that specification of the dynamics can be made without access to experimental data, based only on qualitative prior knowledge. All the properties were required to hold with a high probability, the threshold probability chosen to be 0.9 (we have also run experiments for different values of threshold probability; these results are reported in the appendix). We fixed the range of the unknown parameters as shown in Table 7.1. The initial states of all the species were assumed to be uniformly distributed in a range 10% around the nominal initial concentration. For instance, the first property in table 7.2 says that the initial concentration of p_1 is less than 0.1, between $4 - 10$ time points the level of p_1 reaches a high value between 1.3 and 1.5 and overall the profile of p_1 shows an oscillation pattern with at-least 3 troughs and 3 crests whose values are given in the formula. Similarly for other species of the pathway. Next, we ran global optimization based on SRES, with population size 100 for 50 generations. After completing the optimization, all specified properties were met. Figure 7.2 shows the time course profiles of all 6 species sampled according to their assumed initial concentrations, using the obtained parameter estimate. The parameter estimation procedure took 54.96 seconds. The time profile of the three protein species fits the dynamic trends encoded as PBLTL formulas.

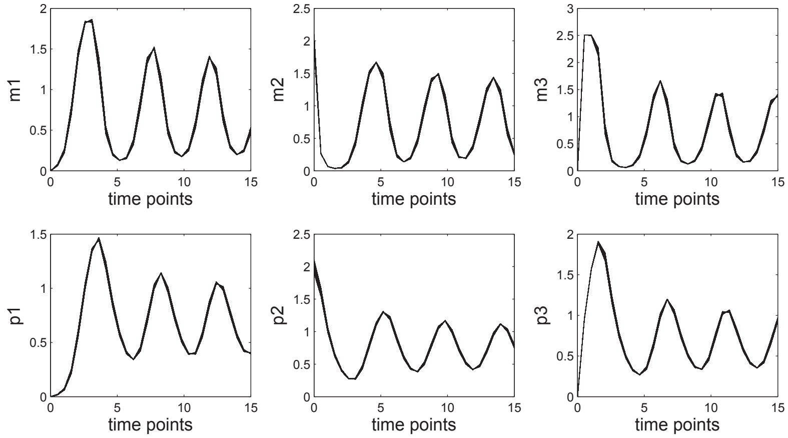


Figure 7.2: Time profile of all the species in the repressilator pathway based on the best parameters returned by SRES based parameter estimation

Species name	Property
p_1	$[p_1 \leq 0.1] \wedge \sim F^{\leq 4}([1.3 \leq p_1 \leq 1.5] \wedge F^{\leq 10}[1.3 \leq p_1 \leq 1.5] \wedge F([1.3 \leq p_1 \leq 1.5] \wedge F([0.3 \leq p_1 \leq 0.4] \wedge F([1.05 \leq p_1 \leq 1.15] \wedge F([0.35 \leq p_1 \leq 0.45] \wedge F([1 \leq p_1 \leq 1.1] \wedge F([0.35 \leq p_1 \leq 0.45]))))))$
p_2	$[1.9 \leq p_2 \leq 2.1] \wedge F^{\leq 10}[0.2 \leq p_2 \leq 0.3] \wedge F([0.2 \leq p_2 \leq 0.3] \wedge F([1.15 \leq p_2 \leq 1.25] \wedge F([0.3 \leq p_2 \leq 0.4] \wedge F([1.0 \leq p_2 \leq 1.1] \wedge F([0.35 \leq p_2 \leq 0.45] \wedge F([0.95 \leq p_2 \leq 1.05]))))))$
p_3	$[p_3 \leq 0.2] \wedge F^{\leq 10}[1.55 \leq p_3 \leq 1.7] \wedge F([1.55 \leq p_3 \leq 1.7] \wedge F([0.275 \leq p_3 \leq 0.375] \wedge F([1 \leq p_3 \leq 1.2] \wedge F([0.35 \leq p_3 \leq 0.45] \wedge F([1 \leq p_3 \leq 1.2] \wedge F([0.35 \leq p_3 \leq 0.45]))))))$

Table 7.2: Repressilator pathway: Properties

7.3.2 The EGF-NGF signaling pathway

We refer to the EGF-NGF pathway and the corresponding ODE model that was discussed in Chapter 4. Figure 4.2 depicts the corresponding signaling pathway. The ODE model consists of 32 differential equations and 48 associated rate parameters.

Parameter estimation Details of the parameters and the range of unknown parameters can be found in the table 7.3. In order to test the performance of the statistical model checking based parameter estimation method, 20 of the 48 parameters were designated to be unknown. We synthesized experimental time series data for 9 species { bounded EGFR, bounded NGFR, active Sos, active C3G, active Akt, active p90RSK, active Erk, active Mek, active PI3K }, measured at { 2, 5, 10, 15, 20, 25, 30, 40, 50 } minutes. This data was synthesized using prior knowledge about initial conditions and parameters. The threshold probability was chosen to be 0.8. To mimic western blot data which is cell population based, we averaged 10^4 random trajectories generated by sampling initial concentration levels, then we added observation noise with standard deviation 5% to the

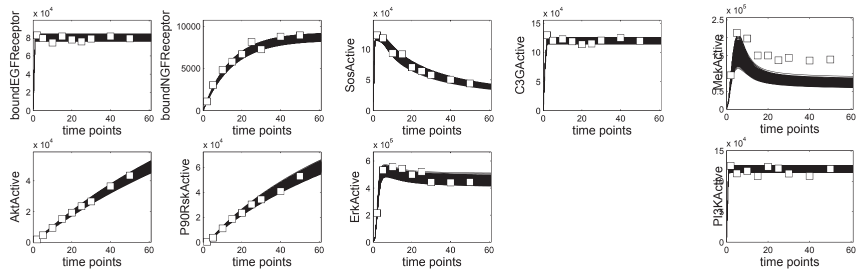
Number	Parameter	range	Parameter estimate(SRES)
1	$k1$	$[0, 0.000218503]$	0.00009690973
2	$k2$	$[0, 0.121008]$	0.01155505
3	$k3$	$[0, 0.00000138209]$	0.0000001352723
4	$k4$	$[0, 0.0723811]$	0.008147492
5	$k11$	$[0, 323.44]$	49.13858
6	$k12$	$[0, 359543]$	327526.7
7	$k15$	$[0, 8.84096]$	2.201634
8	$k17$	$[0, 1857.59]$	77.01694
9	$k23$	$[0, 98.5367]$	13.62002
10	$k27$	$[0, 0.213697]$	0.1621894
11	$k28$	$[0, 7635230]$	6283265
12	$k29$	$[0, 106.737]$	12.21933
13	$k33$	$[0, 0.566279]$	0.4359513
14	$k34$	$[0, 6539510]$	5865839
15	$k37$	$[0, 1469.12]$	385.3151
16	$k38$	$[0, 128762]$	28287.23
17	$k39$	$[0, 14.0145]$	1.857971
18	$k40$	$[0, 109656]$	40.02646
19	$k43$	$[0, 22.0995]$	4.905653
20	$k44$	$[0, 10254600]$	3744344

Table 7.3: EGF-NGF pathway: Unknown parameters with range

simulated values. We used the data of 7 of these species for training the parameters and reserved the rest for testing the quality of the estimated parameter values. The data points were converted into logic formulas and used to guide parameter estimation. The initial states of all the species were assumed to be uniformly distributed in a range of 10% with respect to the assumed initial concentration. Error tolerance for the experimental data was chosen to be 10% around the experimental data value. Parameter estimation was done using the following setting : population size 200 for 150 generations. The time taken by SRES based search was $\sim$ 2.9 hours. Figure 7.3(a) shows the fit to training data for simulated time profiles with the best parameters returned by the SRES based procedure, figure 7.3(b) shows the fit to test data which was not used for training the parameters.

7.3.3 The segmentation clock network

We refer to the segmentation clock pathway and the corresponding ODE model that was discussed in Chapter 6. Figure 6.2 depicts the corresponding signaling pathway. The ODE model consists of 16 differential equations and 75 kinetic rate parameters.



(a) (b)

Figure 7.3: Time profile of (a)training and (b)test data for the corresponding species in the EGF-NGF pathway based on the best parameters returned by SRES based approach

Parameter estimation We follow the case study presented in [2]. For the experiments, we assumed 39 of the 75 parameter values as unknown. The initial states of all the species were assumed to be uniformly distributed in a range of 10% around the nominal initial concentration. We use a combination of dynamic trends and quantitative experimental data in this case study. Specifically, we synthesized population based experimental time series data for Axin 2 mRNA measured at 14 time points up to 200 minutes using the method described in the previous example. For 5 other species (Notch protein, nuclear NicD, Lunatic fringe mRNA, active ERK and Dusp6 mRNA), we encoded the dynamic trends as properties in our logic. We assumed that the dynamic trend of 2 species (cytosolic NicD and Dusp6 protein) were also available, this was used as the test data. Table 7.4 and table 7.5 depict these properties encoded in our logic. The threshold probability was chosen to be 0.8. Details of the parameters and the corresponding range can be found in table 7.6. Parameter estimation was done with population size 200 for 300 generations. The time taken by SRES based search was $\sim$ 2.36 hours. Figure 7.4(a) shows the simulation profile of the 6 proteins with the estimated parameters. Figure 7.4(b) shows that dynamic trends of simulated time profiles fit the test set. The estimated parameters fit the trend and quantitative experimental data well.

7.4 Discussion

In this chapter, we have proposed a statistical model checking based approach for the parameter estimation of biopathway models. We used a slightly modified version of PBLTL to encode both quantitative experimental data and qualitative dynamic trends of pathway dynamics as logical formulas. Assuming a uniform distribution over a set of

Species name	Property
$Notch\ protein$	$(([0.45 \leq Notch\ protein \leq 0.55] \wedge F^{\leq 3}([\ Notch\ protein \leq 0.05])) \wedge (F([\ Notch\ protein \leq 0.05] \wedge F([0.10 \leq Notch\ protein \leq 0.15] \wedge F([\ Notch\ protein \leq 0.05] \wedge F([0.10 \leq Notch\ protein \leq 0.15]))))))$
$nuclear\ NicD$	$(([\ nuclear\ NicD \leq 0.012]) \wedge (F([0.07 \leq nuclear\ NicD \leq 0.08] \wedge F([\ nuclear\ NicD \leq 0.012] \wedge F([0.07 \leq nuclear\ NicD \leq 0.08] \wedge F([\ nuclear\ NicD \leq 0.012]))))))$
$Lunatic\ fringe\ mRNA$	$(([\ Lunatic\ fringe\ mRNA \leq 0.4]) \wedge (F([\ Lunatic\ fringe\ mRNA \geq 2.2] \wedge F([\ Lunatic\ fringe\ mRNA \leq 0.4] \wedge F([\ Lunatic\ fringe\ mRNA \geq 2.2] \wedge F([\ Lunatic\ fringe\ mRNA \leq 0.4]))))))$
$active\ ERK$	$([\ active\ ERK \leq 0.27] \wedge F^{\leq 3}([1.9\ le\ active\ ERK \leq 2.2])) \wedge (F([1.9\ le\ active\ ERK \leq 2.2] \wedge F([\ active\ ERK \leq 0.27] \wedge F([1.9\ le\ active\ ERK \leq 2.2] \wedge F([\ active\ ERK \leq 0.27]))))))$
$Dusp6\ mRNA$	$([\ Dusp6\ mRNA \leq 1]) \wedge (F([\ Dusp6\ mRNA \geq 5.5] \wedge F([\ Dusp6\ mRNA \leq 1] \wedge F([\ Dusp6\ mRNA \geq 5.5] \wedge F([\ Dusp6\ mRNA \leq 1])))))$

Table 7.4: Segmentation pathway: Properties used for training, additional constraints were added to limit the number of crests and troughs

Species name	Property
$Dusp6\ protein$	$(([\ Dusp6\ protein \leq 0.5]) \wedge (F([9 \leq Dusp6\ protein \leq 11] \wedge F([\ Dusp6\ protein \leq 0.5] \wedge F([9 \leq Dusp6\ protein \leq 11] \wedge F([\ Dusp6\ protein \leq 0.5]))))))$
$cytosolic\ nicD$	$(([\ cytosolic\ nicD \leq 0.5]) \wedge (F([\ cytosolic\ nicD \geq 1.0] \wedge F([\ cytosolic\ nicD \leq 1.0] \wedge F([\ cytosolic\ nicD \geq 1.0] \wedge F([\ cytosolic\ nicD \leq 1.0]))))))$

Table 7.5: Segmentation pathway:Test properties

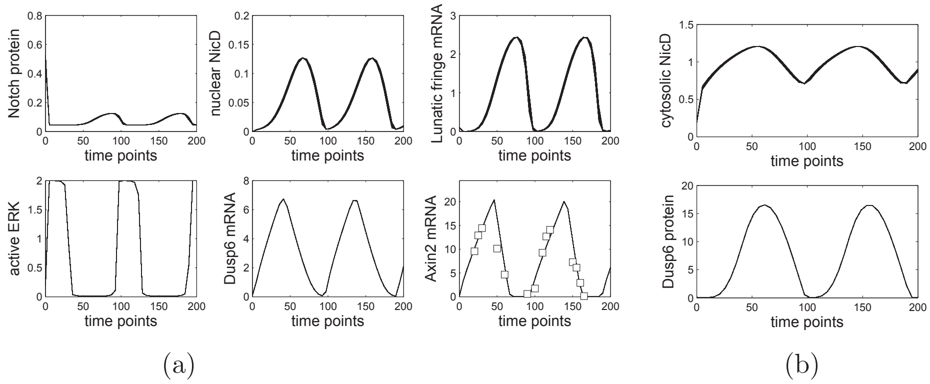


Figure 7.4: Time profile of (a)training and (b)test data for the corresponding species in the segmentation clock pathway based on the best parameters returned by SRES based approach

Number	Parameter	range	Parameter estimate(SRES)
$k1$	KdN	$[0, 2.8]$	1.854774
$k2$	vsN	$[0, 0.46]$	0.2254612
$k3$	vdN	$[0, 5.64]$	3.016938
$k4$	$kt1$	$[0, 0.2]$	0.1066553
$k5$	$kt2$	$[0, 0.2]$	0.1228041
$k6$	$KdNan$	$[0, 0.002]$	0.001628184
$k7$	$VdNan$	$[0, 0.2]$	0.1067782
$k8$	$KdMF$	$[0, 1.536]$	1.395118
$k9$	$KIG1$	$[0, 5]$	1.969339
$k10$	vsF	$[0, 6]$	2.358976
$k11$	vmF	$[0, 3.84]$	3.098625
$k12$	KdF	$[0, 0.74]$	0.2501358
$k13$	vdF	$[0, 0.78]$	0.6268464
$k14$	ksF	$[0, 0.6]$	0.2876905
$k15$	$kd2$	$[0, 14.124]$	4.661996
$k16$	vMB	$[0, 3.28]$	1.432242
$k17$	KaB	$[0, 1.4]$	1.187312
$k18$	$vMXa$	$[0, 1]$	0.9953178
$k19$	$ksAx$	$[0, 0.04]$	0.03657672
$k20$	$vdAx$	$[0, 1.2]$	0.05869855
$k21$	$KdAx$	$[0, 1.26]$	0.5040457
$k22$	$kt3$	$[0, 1.4]$	0.08752867
$k23$	$kt4$	$[0, 3]$	2.460013
$k24$	$ksDusp$	$[0, 1]$	0.6604028
$k25$	$vdDusp$	$[0, 4]$	2.230291
$k26$	$KdDusp$	$[0, 1]$	0.03116861
$k27$	$kcDusp$	$[0, 2.7]$	2.352255
$k28$	$KaFgf$	$[0, 1]$	0.03527007
$k29$	$KaRas$	$[0, 0.206]$	0.1144681
$k30$	$KdRas$	$[0, 0.2]$	0.1080222
$k31$	$KaMDusp$	$[0, 1]$	0.6799779
$k32$	$KdMDusp$	$[0, 1]$	0.9590261
$k33$	$VMsMDusp$	$[0, 1.8]$	1.344481
$k34$	$VMdMDusp$	$[0, 1]$	0.7772506
$k35$	$VMaRas$	$[0, 9.936]$	8.065443
$k36$	$VMdRas$	$[0, 0.82]$	0.3543762
$k37$	$VMaErk$	$[0, 6.6]$	6.375076
$k38$	$VMaX$	$[0, 3.2]$	3.097873
$k39$	$VMdX$	$[0, 1]$	0.537238

Table 7.6: Segmentation Clock pathway: Unknown parameters with range

Pathway	Number of parameters	Search algorithm setting	SRES	Avg sample size per test	(min,max) samples
Repressilator	9	$Gen: 50$ $Pop: 100$	54.94sec	12.96	$(3, 439)$
Clock	39	$Gen: 300$ $Pop: 200$	2.36hrs	45	$(6, 1484)$
EGF-NGF	20	$Gen: 150$ $Pop: 200$	2.9 hrs	150.11	$(37, 1831)$

Table 7.7: Summary of parameter estimation tasks

initial states we show how the probability of the property being met by the behavior of the model can be assessed using a statistical model checking procedure. By combining this method with a global search strategy, we arrive at a parameter estimation procedure.

We demonstrated the applicability of our method with the help of 3 ODE based biopathway models: the repressilator pathway, the EGF-NGF pathway and the segmentation clock network. Our method successfully obtained good parameter estimates using noisy cell-population data and qualitative knowledge. The results show that our method scales well and can cope with large biological networks.

Chapter 8

Toll like receptor modeling

The previous chapter discussed our statistical model checking framework for parameter estimation of ODE models. This chapter focuses on the application and scalability of our approach to the study of Toll like receptor (TLR) pathways which are crucial players in immune response. Specifically, we built a new ODE model for the TLR3 and TLR7 pathways. We investigate possible crosstalk mechanisms which lead to synergistic immune response when these receptors are triggered in a certain order and a specific time interval. This study has been conducted in collaboration with biologists from the Department of Biological Science, National University of Singapore. The pathway is considerably large; we estimated 100 unknown rate constants using our framework. Here, we use a combination of both dynamic trend based properties and experimental data to calibrate and analyze the model. The results show that our framework is scalable to large systems. More importantly, we were able to gain crucial insights about the most plausible crosstalk mechanism which could lead to the observed synergy effect.

8.1 Biological context

Toll like receptors (TLRs) [168, 169, 170, 171] are a class of receptor molecules that play a crucial role in innate immune response. They act as the first line of defense against attack by external agents such as viruses and bacteria. These receptors are members of a broader family of pattern recognition receptors (PRRs). They recognize specific pathogen-associated molecular patterns (PAMPs) on the external agents and through a series of signaling events, trigger immune response manifested through production of interferons(IFNs) and inflammatory cytokines.

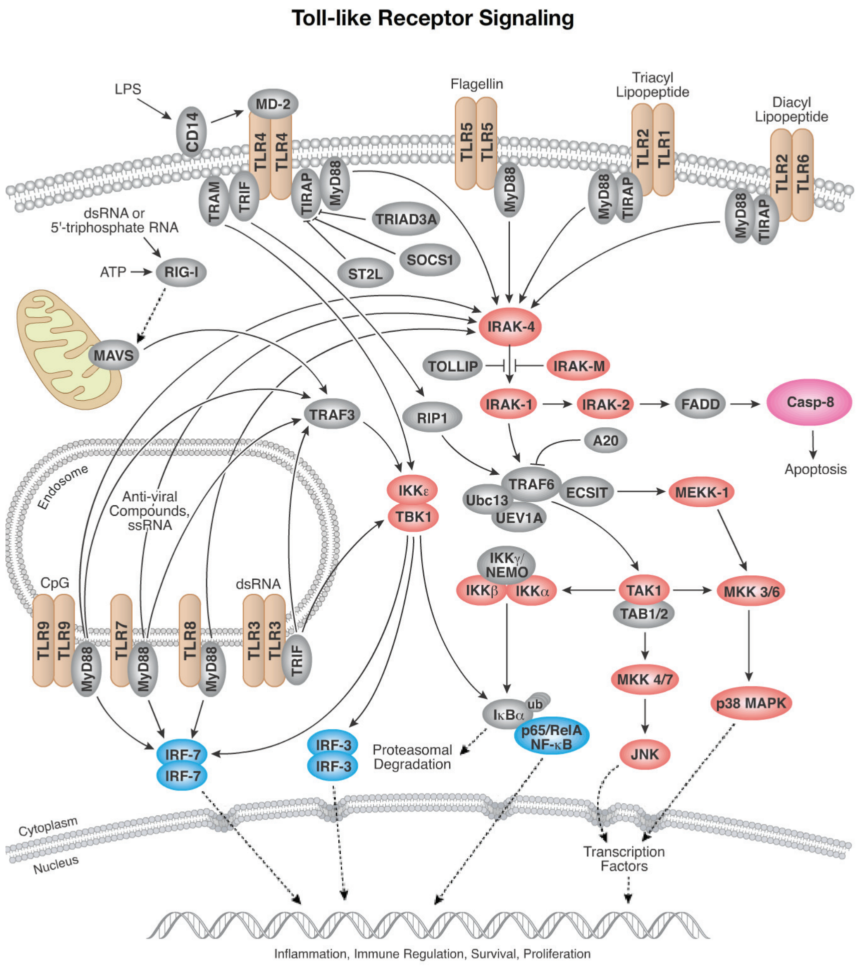


Figure 8.1: Overview of TLR pathway. Taken from *http : //www.cellsignal.com*

There are 13 TLRs characterized in mammals. All the TLR receptors are structurally conserved, and are mainly divided into 2 groups based on their cellular localization and the PAMPs they recognize. TLR-1,2,4,5,6,11 are expressed mainly on cell surface and recognize microbial membrane components such as lipids, proteins etc. TLR-3,7,8,9 are expressed in the intracellular vesicles such as endoplasmic reticulum, endosomes, lysosomes and endo-lysosomes; they mainly recognize microbial nucleic acids. Figure 8.1 provides an overview of TLR signaling pathways. Our interest is mainly on TLR3 and TLR7 receptors, the signaling cascades they trigger, the immune response they lead to and the crosstalk mechanisms they are involved in. Now, we will describe the signaling cascades triggered by the TLR3 and TLR7 pathways.

TLR3 pathway TLR3 recognizes double stranded Ribo-Nucleic acid (dsRNA) derived from viruses or virus-infected cells and synthetic analogues of dsRNA such as polyinosinic-polycytidylic acid(poly(I:C)).

TLR3 transduces the signal mainly via the adaptor protein TIR domain containing adapter-inducing interferon-β (TRIF) dependent pathway. The signal culminates in the activation of IRF3 and NF-kB which subsequently leads antiviral immune response, characterized the production of interferons and cytokines.

Specifically, TRIF forms a multi protein signaling complex along with TRAF6, TRADD, FADD and RIP1 for the activation of TAK1 complex as shown in figure 8.3. Activated TAK1 complex, in turn activates the IKK complex (NEMO:IKKb:IKKa).

Usually NF-kB is associated with IkBa in the cytoplasm, here, IkBa sequesters with the transcription factor NF-kB which renders NF-kB inactive. Activated IKK complex, phosphorylates IkBa (that is sequestered to NF-kB), this leads to the dissociation and nuclear translocation of NF-kB. NF-kB then induces the transcription and translation of inflammatory cytokines. TAK1 complex simultaneously activates the MAPKs Erk, p38 and JNK by inducing the phosphorylation of MAPK kinases, which in turn activates the AP-1 transcription factor which then induces the transcription of inflammatory cytokines.

More significantly, the TRIF-dependent pathway leads to IRF3 activation and subsequent type-1-IFN production. TRIF, along with TRAF3 recruits a signaling complex involving TBK1 and IKKi (IKKe), which catalyze the phosphorylation of IRF3 and induce its nuclear translocation. Phosphorylated IRF3 in the nucleus, is a transcription factor, then induces the transcription and subsequent translation of Type-1-IFNs.

In summary, TLR3 induces antiviral immune response by promoting production of type 1 IFNs predominantly and cytokines to a lesser extent. The main signaling intermediaries in this pathway are IRF3, NF-kB and AP-1. IRF3 leads to the production of Type 1 IFNs while NF-kB and AP-1 lead to production of inflammatory cytokines. Details of the pathway can be found in figure 8.3.

TLR7 pathway TLR7 on the other hand, recognizes single stranded RNA (ssRNA) from ssRNA viruses and imidazoquinoline derivatives such as imiquimod and resiquimod (R-848) in endolysosomes. TLR7s are highly expressed in plasmacytoid dendrite cell (pDCs), although their expression can be observed in macrophages too. In fact, in our

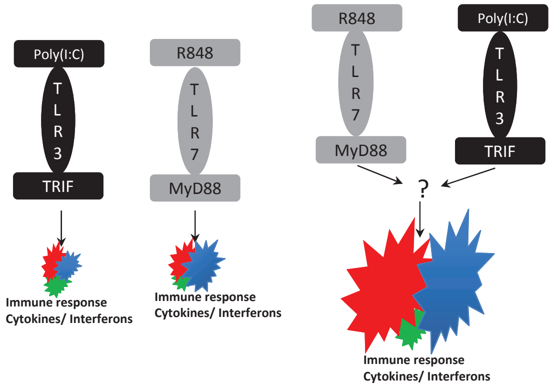


Figure 8.2: TLR3, TLR7 synergy

case, we are interested in their effect on macrophages. They predominantly activate NF-kB and IRF7 via MyD88 dependent pathway to induce the production of inflammatory cytokines and type I IFNs respectively. Details of the TLR7 pathway can be found in figure 8.3.

TLR7 initiates its response cascade by first activating MyD88 which in turn recruits and activates IL-1 receptor associated kinases, IRAK4, IRAK1, IRAK2 and IRAK-M. Activated IRAK complex then interacts with TRAF6. These proteins then activate TAB2 and TAB3, the regulatory components of the kinase TAK1 complex, to activate TAK1. Activated TAK1 complex, in turn activates the IKK complex (NEMO:IKKb:IKKa). Usually NF-kB is associated with IkBa in the cytoplasm which renders NF-kB inactive. Activated IKK complex, phosphorylates IkBa (that is sequestered to NF-kB), this leads to the dissociation and nuclear translocation of NF-kB. NF-kB, which is a transcription factor then induces the transcription and translation of inflammatory cytokines. TAK1 complex simultaneously activates the MAPKs Erk, p38 and JNK by inducing the phosphorylation of MAPK kinases, which in turn activate various transcription factors, including AP-1. These transcription factors then induce the transcription of inflammatory cytokines. These form the predominant signaling events of the TLR7 signaling cascade.

To a lesser extent, the TLR7 cascade activates the transcription factor IRF7, which is usually constitutively expressed in the nucleus and is in inactive form. IRF7 binds to forms a multi protein signaling complex with IRAK4, TRAF6, TRAF3, IRAK1. This leads to the phosphorylation of IRF7, which then dissociates from the complex and translocates into the nucleus. Here it plays a role in the transcription of genes for type I IFNs.

Synergistic crosstalk between TLR3 and TLR7 pathways There have been several studies which show the cooperation between different TLR pathways [172, 173, 172, 174]. In this study, we are interested in the possible crosstalk between the TLR3 and TLR7 pathways which leads to synergistic immune response (see figure 8.2). Experimental data suggests that when mouse bone marrow derived macrophage(BMDM) cells are stimulated with either R848 or Poly(I:C) separately they elicit normal immune response. However, when the system is stimulated combinatorially with a particular ordering of these ligands, with a particular time interval between the stimulation, the immune response is synergistically increased. Specifically, the -Poly(I:C)-8 hour interval- R848-stimulation is shown to have maximum synergy effect. Our goal is to investigate specific crosstalk mechanisms between these two pathways which can help explain the synergy.

The following hypotheses were formulated in collaboration with biologists and through literature. Details of the associated crosstalk mechanisms are shown in red in figure 8.3.

H1: TLR3 activation leads to activation of IRF3 to its phosphorylated form. Next, the phosphorylated IRF3 or one of its downstream activated molecules, which we refer to as FactorX, bind to NF-kB and activated AP-1 to form an enhanceosome complex inside the nucleus. This enhanceosome in turn enhances the transcription of IL6 and IL12 mRNA in a synergistic manner[175, 176].

H2: TLR3 activation leads to production of type I IFNs. Type I IFNs can further bind to the cell surface and trigger a second series of signaling cascades which leads to, first, activation of the PI3K-Akt cascade that in turn activate the NEMO-IKKb-IKKa complex. The activated complex helps in the breakdown of NF-kB complex, leading to the release of NF-kB which further activates IL6 and IL12 mRNA production[177].

H3: TLR3 activation leads to production of Type I IFNs. Type I IFNs further bind to the cell surface and trigger a second series of signaling cascades which leads to, first, activation of the Tyk2-Jak1 complex and then the Stat1-Stat2 proteins. These activated protein complexes further activate a protein, which we refer to as Factor Y. Factor Y bind to NF-kB and activated AP-1 to form an enhanceosome complex inside the nucleus, this enhanceosome in turn enhances the transcription of IL6 and IL12 mRNA in a synergistic manner[176, 177].

8.2 Construction of the ODE model

A schematic representation of combined pathway is shown in figure 8.3. The edges and the key components of the pathway were chosen based on existing literature about TLR signaling and specific mechanisms we were interested in investigating. The initial ODE model was implemented using the tool COPASI[178]. It consists of 84 species (including delay variables), 103 reactions and 127 kinetic rate constants. Out of these 127 kinetic rate constants, 27 rate constants which correspond to NF-kB pathway were adapted from [179], the remaining parameters were assumed to be unknown and estimated. Biological processes such as protein degradation, association, transport, delay, translation etc. are modeled using mass action kinetics. Activation of proteins is modeled with Michaelis-Menten kinetics. Transcription is modeled using the formalism outlined in [180], this formalism allows for modeling synergistic activation and deactivation explicitly.

In terms of previous work on using computational systems biology approaches to study TLR pathways, Oda and Kitano[181] present a comprehensive map of the TLR signaling network. They build a statistic representation of the network using existing literature. This representation, although is useful to understand the links between the different players involved in the pathway, is not useful for studying the kinetic aspects of the system. There are models which study the dynamics of TLR3 and TLR4 signaling based on ODEs in [182, 183, 184, 185]; but these models are either very crude or are incomplete and have too few pathway players. This limits their use for a systematic study of these pathways. There is a however rich literature on modeling the NF-kB pathway [186, 187, 188, 179] which constitutes one of the core components of the TLR pathway.

8.3 Parameter estimation

As discussed before, we implemented our initial model in the tool COPASI, which offers a good user interface for initial model construction. The details of unknown parameters can be found in table 8.1 and table 8.2.

Time course data was available for activated ERK, activated p38, phosphorylated JNK, phosphorylated IkBa, IL6mRNA and IL12mRNA. The different experimental conditions for which we had time course experimental data and those that were used

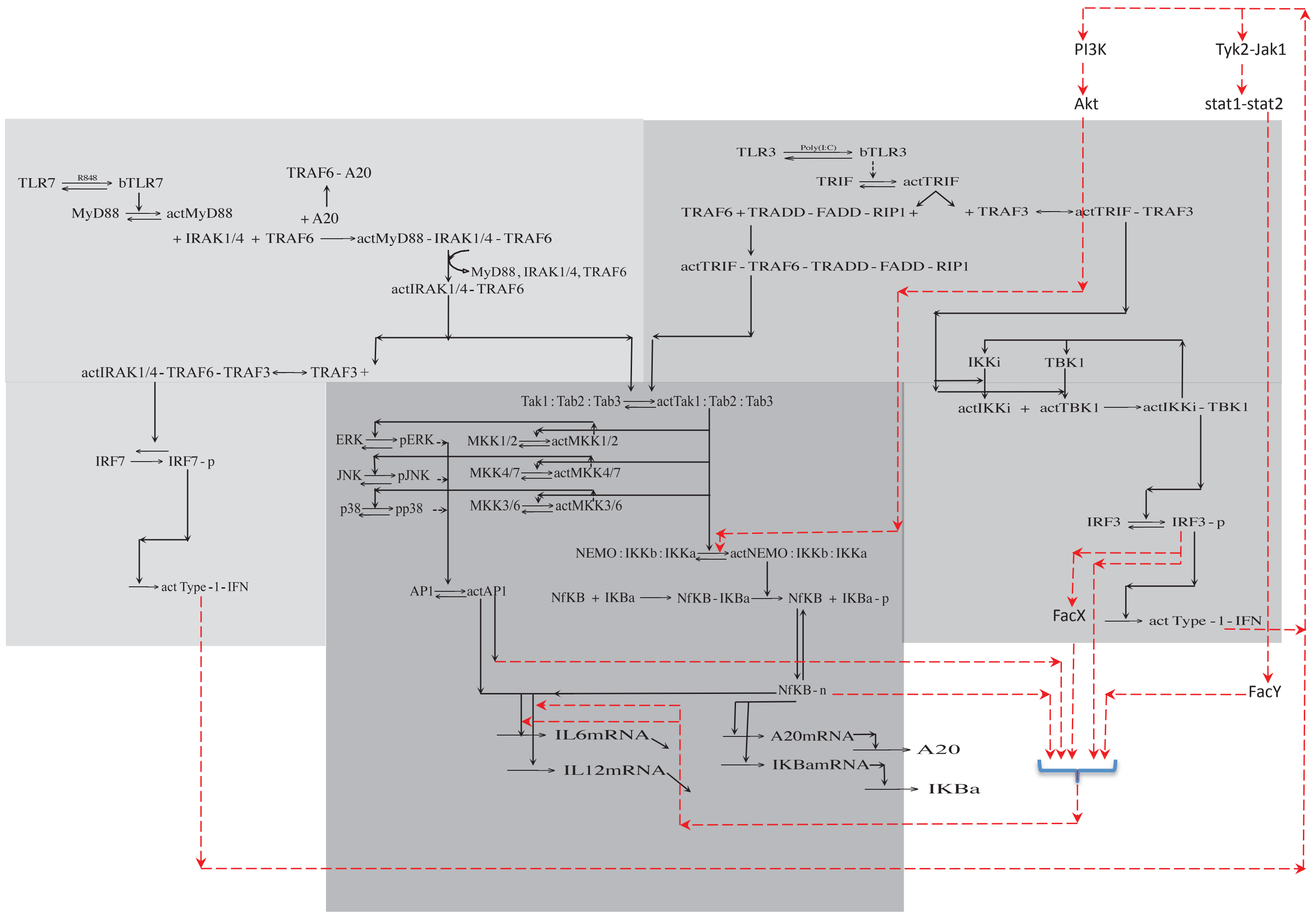


Figure 8.3: The reaction network graph of the mathematical model of TLR pathway. The red dotted lines indicate the proposed crosstalk mechanisms. The kinetic equations of individual reactions can be found in the appendix.

in the current study were 1) TLR3 stimulation (I) 2) TLR7 Stimulation (R) 3) TLR7 and TLR3 stimulation at the same time (IR) 4) TLR3 stimulation initially followed by a 8 hour interval, after which TLR7 pathway is stimulated (I08R) 5) TLR3 stimulation initially followed by a 24 hour interval, after which TLR7 pathway is stimulated(I24R). The time frame of the model was 48 hours (2880 minutes).

We used the statistical model checking framework discussed in the previous chapter for parameter estimation. For activated ERK, activated p38, phosphorylated JNK and phosphorylated IkBa, we converted the time course data for different time points into formulas in our logic. For IL6mRNA, IL12mRNA, for all the experimental repeats, we encoded the experimental data into dynamic trends in our logic. Table 8.3 depicts these properties encoded in our logic. The time course data corresponding to the I24R experiment was reserved as test data to evaluate the quality of our parameter estimates, the data of all other experiments was used to calibrate the model.

The threshold probability was fixed to be 0.8, initial concentrations were allowed to vary 5% around their nominal values . Parameter estimation was done with a population size 100 for 500 generations.

Figures 8.4 , 8.8, 8.5 , 8.6 and 8.7 show the fit to data for the simulation profiles using the best predicted parameter values from our SRES based search method for the activated ERK, activated p38, phosphorylated JNK, phosphorylated IkBa, IL6mRNA and IL12mRNA species. Figure 8.9 shows the fit to test data. The model predictions fit the training experimental data well for most of the cases. In some cases, for instance, the simulation profiles of activated ERK and activated p38 in case of TLR3 stimulation (I) were unable to reproduce the trends of the data well. This is likely due to the simplifications assumed by our model. For instance, the species that have not been included in the model may affect the fitting results. However during our analysis we found that the particular wing of the pathway contributed less to the synergy effect that we intended to investigate. Hence, we went ahead with the current model for further analysis. To understand the dependence of the immune response with respect to the time duration between the TLR3 and TLR7 stimulation, we simulated the model with increasing time duration between the I and R stimulation. Figure 8.10 shows that the immune response (IL6mRNA and IL12mRNA) steadily increases until about the 8 hour interval mark, after which the immune response starts to drop (for clarity, we only plot

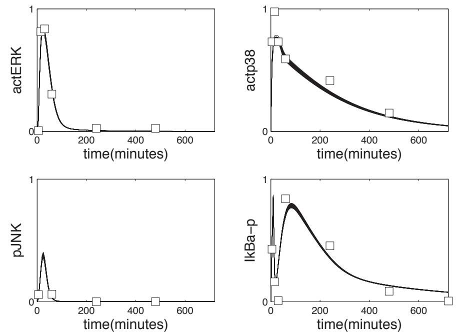


Figure 8.4: TLR pathway- parameter estimation results, training data - (R) stimulation (normalized concentration vs time(minutes))

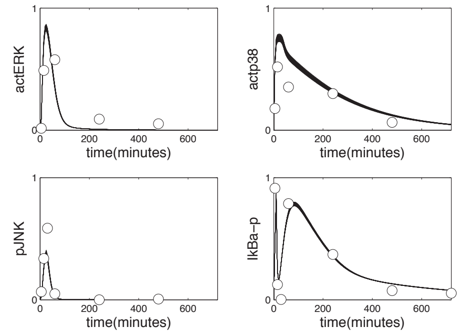


Figure 8.5: TLR pathway- parameter estimation results, training data - (IR)stimulation (normalized concentration vs time(minutes))

one simulation trace from the assume initial value intervals).

We started with the model with all the 3 hypothesized crosstalk mechanisms. To understand which among them was the most crucial, we knock out each mechanism one at a time to see the observed effect on the system. Figure 8.11(a) depicts the case when all the three crosstalk mechanisms are included in the model. Next, we shut down the reactions leading to H1, keeping reactions involved in H2 and H3 intact. The results can be found in figure 8.11(b). It can be observed that this crosstalk only affects the IR stimulation, i.e when this crosstalk is knocked out the synergy observed during IR stimulation is affected. There is no significant effect on the levels of IL6mRNA or IL12mRNA when there is a time gap between $Poly(I : C)$ and $R848$ stimulations.

Next, we knocked out reactions involved in H2, keeping reactions involved in H1 and H3 intact. The results can be found in figure 8.11(c). It is observed that this crosstalk has negligible effect on the observed synergy effect. Finally, we knocked out reactions corresponding to H3, keeping reactions involved in H1 and H2 intact. The results can be found in figure 8.11(d). The results show that this crosstalk has the maximal effect on

Parameter	range	Parameter estimate(SRES)
$k0$	$[0, 10]$	4.313
$k1$	$[0, 125]$	62.0718
$k2$	$[0, 160]$	79.7282
$k3$	$[0, 40]$	18.3319
$k4$	$[0, 0.5]$	0.40115
$k5$	$[0, 1]$	0.55527
$k6$	$[0, 1]$	0.45502
$k7$	$[0, 20]$	10.2145
$k8$	$[0, 0.5]$	0.11481
$k9$	$[0, 100]$	49.5518
$k10$	$[0, 1000]$	93.6679
$k11$	$[0, 100]$	11.5939
$k12$	$[0, 1000]$	83.0183
$k13$	$[0, 5]$	0.85
$k17$	$[0, 0.5]$	0.065976
$k18$	$[0, 0.5]$	0.49757
$k19$	$[0, 1]$	0.85
$k21$	$[0, 1]$	0.4798
$k22$	$[0, 1]$	0.3374
$k23$	$[0, 0.5]$	0.092955
$k24$	$[0, 0.5]$	0.032515
$k25$	$[0, 0.5]$	0.2846
$k26$	$[0, 0.5]$	0.000000019
$k27$	$[0, 0.5]$	0.44633
$k28$	$[0, 0.5]$	0.05
$k29$	$[0, 100]$	53.21
$k30$	$[0, 100]$	64.0901
$k31$	$[0, 1]$	0.85
$k32$	$[0, 0.5]$	0.12827
$k33$	$[0, 0.5]$	0.12776
$k35$	$[0, 1]$	0.024
$k37$	$[0, 1]$	0.39288
$k56$	$[0, 10000]$	9952
$k57$	$[0, 1]$	0.66636
$k58$	$[0, 1]$	0.93819
$k59$	$[0, 0.5]$	0.019657
$k60$	$[0, 1]$	1
$k61$	$[0, 0.5]$	0.0061633
$k62$	$[0, 1]$	0.88438
$k63$	$[0, 1]$	1
$k64$	$[0, 1]$	0.9936
$k65$	$[0, 1]$	0.81826
$k66$	$[0, 1]$	0.24018
$k67$	$[0, 1]$	0.040904
$k68$	$[0, 0.5]$	0.018947
$k69$	$[0, 0.5]$	0.012172
$k70$	$[0, 1]$	1
$k71$	$[0, 0.5]$	0.18546
$k72$	$[0, 1]$	0.54758
$k73$	$[0, 0.5]$	0.0029922
$k74$	$[0, 0.5]$	0.0033346

Table 8.1: TLR pathway: Unknown parameters with range

Parameter	range	Parameter estimate(SRES)
$k75$	$[0, 0.5]$	0.055068
$k76$	$[0, 0.5]$	0.0045339
$k77$	$[0, 0.5]$	0.0045044
$k78$	$[0, 0.5]$	0.4142
$k79$	$[0, 0.5]$	0.34747
$k80$	$[0, 10]$	4.178
$k81$	$[0, 0.5]$	0.011097
$k82$	$[0, 10000]$	7712.52
$k83$	$[0, 10]$	2.3537
$k84$	$[0, 20000]$	19991.5
$k85$	$[0, 0.5]$	0.11164
$k86$	$[0, 0.1]$	0.0000045
$k87$	$[0, 0.5]$	0.49955
$k88$	$[0, 0.5]$	0.000004564
$k89$	$[0, 0.5]$	0.00007886
$k90$	$[0, 0.5]$	0.0018139
$k91$	$[0, 0.5]$	0.37241
$k92$	$[0, 0.5]$	0.040712
$k93$	$[0, 0.5]$	0.016907
$k94$	$[0, 0.5]$	0.038708
$k95$	$[0, 0.5]$	0.013188
$k96$	$[0, 0.5]$	0.15234
$k97$	$[0, 10]$	4.9829
$k98$	$[0, 0.5]$	0.00008793
$k99$	$[0, 0.5]$	0.0010413
$k100$	$[0, 0.5]$	0.0000345
$k101$	$[0, 1]$	0.83475
$k102$	$[0, 0.5]$	0.16222
$k103$	$[0, 1]$	0.9542
$k104$	$[0, 1]$	0.62167
$k105$	$[0, 1]$	0.000079701
$k106$	$[0, 10]$	8.8227
$k107$	$[0, 1.5]$	0.10893
$k108$	$[0, 10]$	9.7043
$k109$	$[0, 0.5]$	0.00012893
$k110$	$[0, 100]$	83.7732
$k111$	$[0, 0.5]$	0.18255
$k112$	$[0, 100]$	18.5703
$k113$	$[0, 10]$	4.3817
$k114$	$[0, 100]$	55.1036
$k115$	$[0, 1]$	1
$k116$	$[0, 0.5]$	0.27158
$k117$	$[0, 0.5]$	0.42091
$k121$	$[0, 0.5]$	0.12151
$k122$	$[0, 0.5]$	0.0079654
$k123$	$[0, 0.5]$	0.0069245
$k124$	$[0, 0.5]$	0.0081743
$k125$	$[0, 1]$	0.83038
$k126$	$[0, 0.5]$	0.0024459

Table 8.2: TLR pathway: Unknown parameters with range

Experiment	Species name	Property
R	$IL6mRNA$	$(\neg(F^{\leq 90}([IL6mRNA \geq 0.014]))) \wedge (F^{\leq 98}([0.014 \leq IL6mRNA \leq 0.06])) \wedge F(([IL6mRNA \geq 0.014]) \wedge F([IL6mRNA \leq 0.005]))$
R	$IL12mRNA$	$((\neg(F^{\leq 60}([IL12mRNA \geq .10]))) \wedge (F^{\leq 98}([0.1 \leq IL12mRNA \leq 0.15]))) \wedge F(([IL12mRNA \geq .10]) \wedge F([IL12mRNA \leq 0.004]))$
I	$IL6mRNA$	$G([IL6mRNA \leq 0.005])$
I	$IL12mRNA$	$G([IL12mRNA \leq 0.02])$
IR	$IL6mRNA$	$((\neg(F^{\leq 75}([IL6mRNA \geq 0.14]))) \wedge (F^{\leq 100}([0.14 \leq IL6mRNA \leq 0.165]))) \wedge F(([IL6mRNA \geq 0.14]) \wedge F([IL6mRNA \leq 0.05]))$
IR	$IL12mRNA$	$((\neg(F^{\leq 90}([IL12mRNA \geq .43]))) \wedge (F^{\leq 98}([0.35 \leq IL12mRNA \leq 0.6]))) \wedge F(([IL12mRNA \geq .43]) \wedge F([IL12mRNA \leq 0.05]))$
$I08R$	$IL6mRNA$	$((\neg(F^{\leq 120}([IL6mRNA \geq 0.4]))) \wedge (F^{\leq 195}([0.4 \leq IL6mRNA \leq 0.6]))) \wedge F(([[IL6mRNA \geq 0.4]]) \wedge F([IL6mRNA \leq 0.05]))$
$I08R$	$IL12mRNA$	$((\neg(F^{\leq 120}([IL12mRNA \geq 4.3]))) \wedge (F^{\leq 195}([4.3 \leq IL12mRNA \leq 6]))) \wedge F(([IL12mRNA \geq 4.3]) \wedge F([IL12mRNA \leq 0.5]))$

Table 8.3: TLR pathway: Properties of IL6mRNA and IL12mRNA, the total time frame of the system (2880 minutes) was divided into 576 time points each separated by 5 minutes

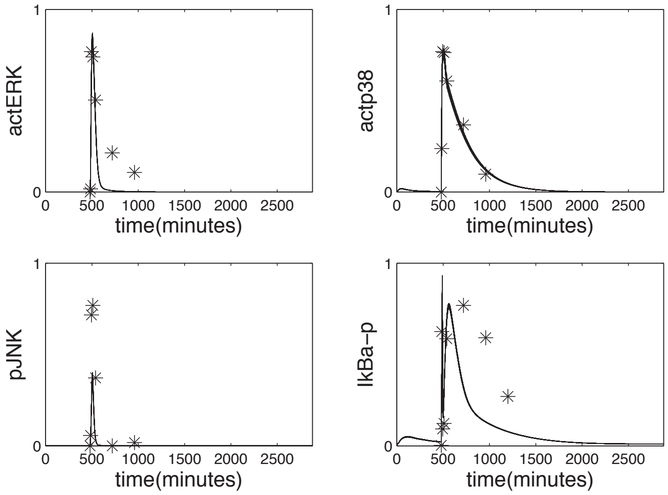


Figure 8.6: TLR pathway- parameter estimation results, training data - (I08R)stimulation (normalized concentration vs time(minutes))

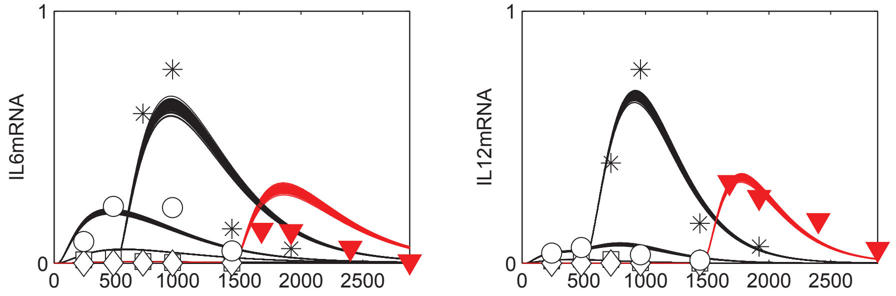


Figure 8.7: TLR pathway, parameter estimation results, training data - IL6mRNA and IL12mRNA profiles (normalized concentration vs time(minutes))

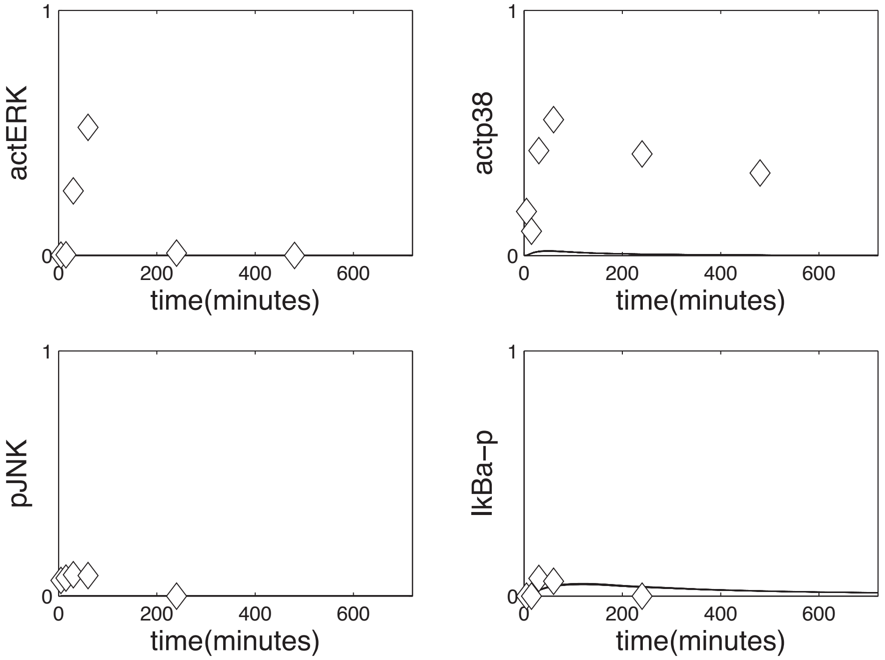


Figure 8.8: TLR pathway- parameter estimation results, training data - (I) stimulation (normalized concentration vs time(minutes))

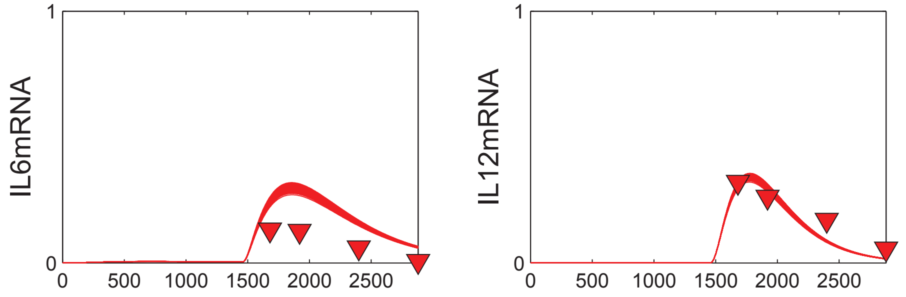


Figure 8.9: TLR pathway- parameter estimation results, test data - (I24R) stimulation (normalized concentration vs time(minutes))

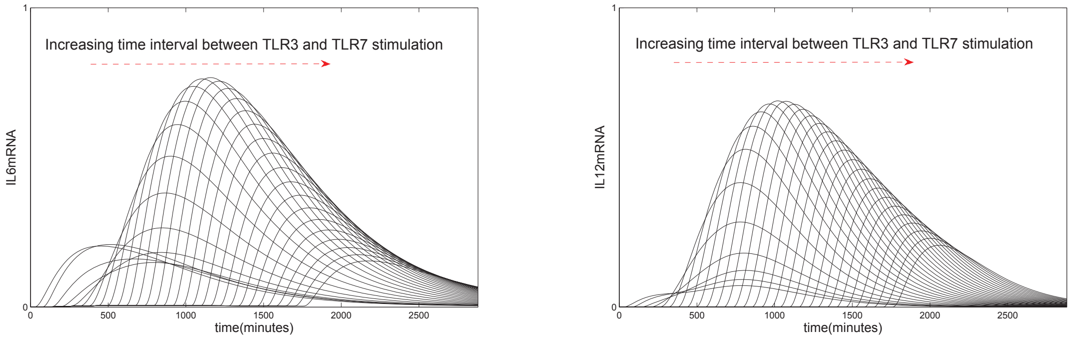


Figure 8.10: Model prediction for concentrations profiles of IL6mRNA and IL12mRNA with increasing time interval between I and R stimulation (normalized concentration vs time(minutes))

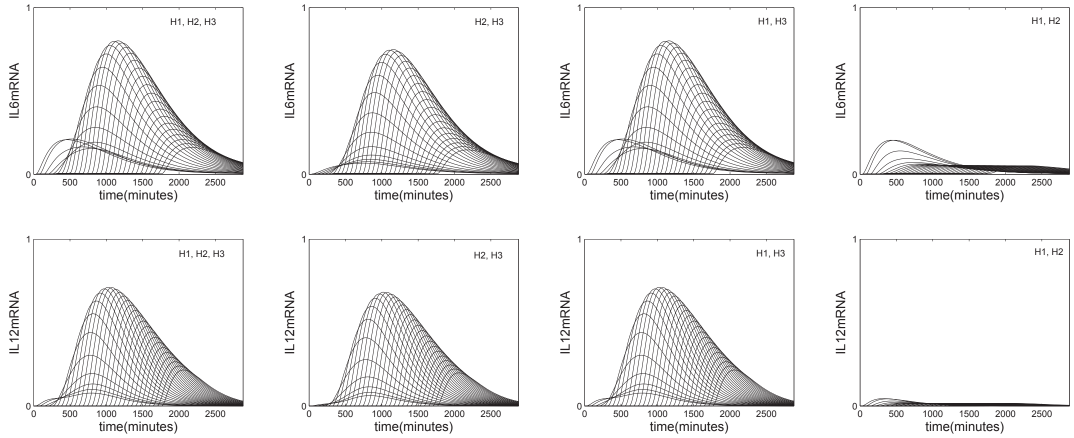


Figure 8.11: Effect of different crosstalk mechanisms on synergy (normalized concentration vs time(minutes))

the synergy.

8.4 Discussion

We have constructed an integrated ODE model for the TLR3 and TLR7 pathways to investigate synergistic crosstalk mechanisms between the two pathways. We estimated unknown parameters using our statistical model checking framework. Next, we performed knock-out experiments to find the most important crosstalk mechanism leading to synergy. Our initial analysis suggests that the crosstalk mediated by Type-1-IFN and subsequent release of factors which affect the transcription of IL6 and IL12 is the most promising candidate. We are currently working with biologists to see if these findings can be experimentally validated.

In the future we plan to investigate other crosstalk mechanism namely, TLR7 pathway results in activation of IRF7, and this phosphorylated IRF7 causing the activation of IRF3 which triggers its response in the usual way (discussed early in the section). This link may be especially important when considering stimulation in the other order namely stimulation of TLR7 stimulation followed by stimulation of TLR3. We have not considered this aspect in the current study.

Chapter 9

Conclusion

The focus of this thesis was on developing and application of scalable approximate probabilistic model checking algorithms for analysis of dynamics of stochastic models of biopathways.

First, we developed a probabilistic model checking framework for analyzing DBNs which can arise as succinct representations of Markov chains. Specifically, we have proposed a new temporal logic called BLTPL, tailored for analysis of DBN models. Probabilities are encoded in BLTPL at the level of atomic propositions. BLTPL formulas are interpreted over a linear sequence of marginal probability vectors. Interesting properties concerning the dynamics of biopathways can be formulated using BLTPL.

Model checking on DBNs is based on using probabilistic inference for computing the marginal probability distributions of variables. Atomic propositions of BLTPL are evaluated against these marginal probability distributions. However, it is well known that exact probabilistic inference on DBNs is infeasible for large DBNs such as those used in our setting.

Approximate methods for probabilistic inference of DBNs, such as FF and BK, rely on computing and propagating probability distributions approximately. These algorithms can make considerable errors, as evident in our case studies. To get around this, we proposed an improved probabilistic inference method for DBNs called HFF, which, in addition to maintaining and propagating belief states in a factored form, also maintains a small number of full dimensional state vectors called spikes and their probabilities at each time slice. By tuning the number of spikes, one can gain accuracy at the cost of increased but polynomial (quadratic) computational cost. We have provided an error analysis for

HFF as well as FF which shows that HFF is more accurate. We have demonstrated the efficiency of HFF with the help of relatively large DBNs arising from the EGF-NGF pathway and the Epo mediated ERK signaling pathway. In all cases, we found that the errors suffered by FF and BK (with singleton clusters) were high for the marginal distributions of some biologically significant species. The errors incurred by HFF were always lower and they reduced monotonically when the number of spikes was increased.

We proposed an approximate but efficient probabilistic model checking framework for DBNs based on FF and HFF algorithms. Our approach is generic and can be used for analyzing DBNs that arise in other settings.

Next, we focused on statistical model checking algorithms. We proposed a statistical model checking based approach for parameter estimation of biopathway models. We used a slight variant of temporal logic PBLTL to encode both quantitative experimental data and qualitative properties of pathway dynamics as logical formulas. We assume a uniform distribution over a set of initial states and show how the probability of the property being met by the behavior of the model can be assessed using a statistical model checking procedure. By combining this method with a global search strategy, we arrive at a parameter estimation procedure. We have demonstrated the applicability of our method with the help previously published ODE based models of the represillator pathway, the EGF-NGF pathway, the segmentation clock network pathway. Our method successfully obtains good parameter estimates using noisy cell-population data and qualitative knowledge. The results show that our method scales well and can cope with large biological networks. The procedures we developed are generic and have the potential to be applied to other stochastic models of biopathways [189].

We then applied our developed framework to build and analyze a new ODE model for the TLR3 and the TLR7 pathway based on existing literature in collaboration with biologists. We were specifically interested in investigating the observed synergy in immune response when these two pathways were triggered in a certain order and with a certain time interval. First, we hypothesized 3 crosstalk mechanisms that could explain the synergy and modeled them into our pathway. We then trained the model using our statistical model checking framework to explain the available experimental data. Once we had trained the model, we performed knock out experiments to find the most important crosstalk mechanism. Our initial analysis suggests that the crosstalk mediated by Type-1-IFN and subsequent release of factors which affect the transcription of IL6 and IL12 is the most promising candidate.

9.1 Future work

There are many interesting future lines of work that can be considered. First, for the HFF algorithm, we maintain spikes which are full dimensional state vectors. These spikes are propagated with minimal error to reduce overall errors. We recognize that it may not be necessary to maintain spikes which are full dimensional state vectors. Instead, like BK, it may be interesting to maintain spikes over cluster of variables which do not considerably affect each other. This would reduce the overall overhead of maintaining and propagating full dimensional probability vectors. Finding the right way to cluster variables would still be a concern.

Our logic, BLTPL is simple in the sense that the probabilistic assertions are encoded at the atomic propositions level. Although, we are able to express many interesting biological properties with this logic, a challenging future work will be to consider more sophisticated forms of the logic which are more expressive.

Applying model checking for analyzing probabilistic graphical models such as dynamic Bayesian networks is still at infancy, we envision that these approaches will be an active area of research in the near future. Another direction of future research is that our logic and procedure is currently used for reasoning about DBNs in a bounded time setting. We assume that the time frame of interest is bounded; this is so since in our application we know the time frame of interest. It will be interesting to enhance these methodologies for unbounded time horizons which arise in more general settings.

Statistical model checking has been an active area of research recently, since it offers a scalable, model-size independent alternative for probabilistic model checking. Our work on using statistical model checking for parameter calibration can be further applied to other stochastic modeling formalisms such as those arising as CTMCs, stochastic differential equations [189, 150] etc. It will be interesting to adapt our procedure for performing sensitivity analysis tasks.

Another direction of work is the use of GPU for both these lines of work, primarily by taking advantage of the potential of parallelism offered by both these approaches.

Currently, we do not exploit the use of GPU for performing our DBN based model checking framework. Specifically FF and HFF can be implemented on GPUs. In this context, the sum-of-product algorithm implementation presented in [190] promises to offer helpful pointers. Similarly, the statistical model checking framework has a massive amount of inherent parallelism which can be exploited, the SPRT test can be parallelized by considering group-sequential sampling where one performs statistical tests after drawing a group of samples rather than a single sample as it is done currently. In this connection, works such as [191], [192] promise to offer helpful pointers.

Finally, there are a number of extensions possible on our work with Toll like receptors, we intend to use our predictions from the model to formulate and analyze more crosstalk mechanisms and biological hypotheses. In general, there is no established computational model for the TLR3 and TLR7 pathway. Hence, our model can be used as a crucial starting point for future modeling efforts of the TLR system. It is well known that the immune system is highly coordinated. Models for other components of immune systems such as the complement system[1], T-cell activation[193] etc., exist. It will be interesting to integrate these models together to gain a holistic view of the immune system.

Chapter 10

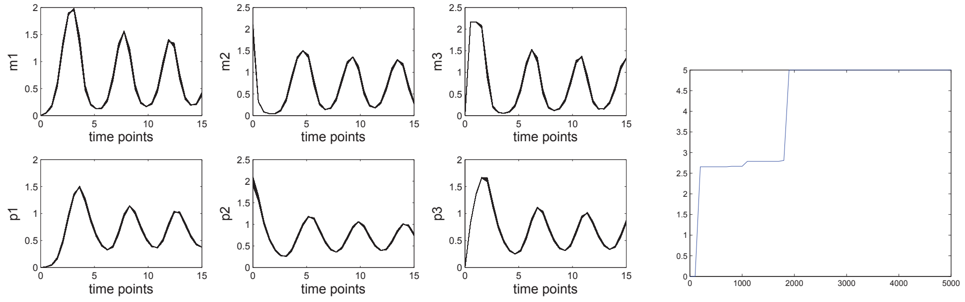
Appendix

10.1 Statistical model checking

This section presents additional details about our case studies in the statistical model checking chapter.

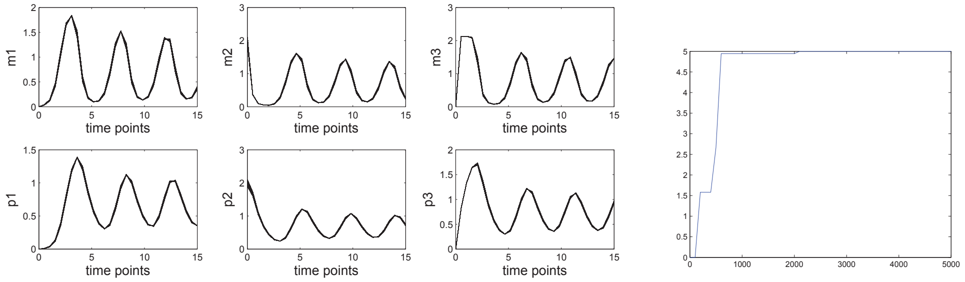
If there is a limit on the number of samples that can be drawn to evaluate the test, [194] discuss computing the p-value of the hypotheses to make a decision on the truth hood. This method is adapted from statistical model checking of black box systems[133]. This is useful in our case too since, we can limit the samples for the evaluation of each parameter combination.

We evaluated this strategy for our case studies since we have to repeatedly perform the test for every combination of parameters picked by the search algorithm. In some cases the number of samples that may be needed to be drawn can be high, in such cases it is practical to have a limit on maximum number of samples that can be drawn to evaluate the test. We set this sample limit to 100, i.e. once the test consumes 100 samples, we uses a p-value based approach to decide the truth-hood of the formula. A comparison of this heuristic with the original statistical test is presented for all the case studies described in the main text.



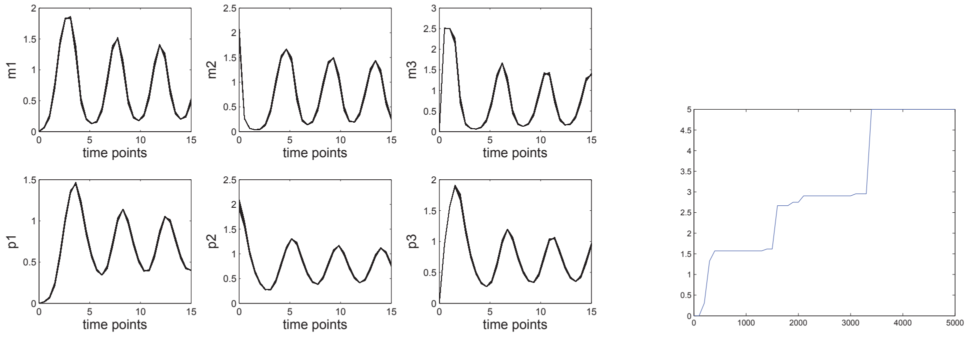
(a) (b)

Figure 10.1: (a)Time profile of all the species in the repressilator pathway based on the best parameters returned by SRES based parameter estimation,(b) objective value vs number of generations, r=0.8



(a) (b)

Figure 10.2: (a)Time profile of all the species in the repressilator pathway based on the best parameters using the p-value based, SRES search,(b) objective value vs number of generations, r=0.8

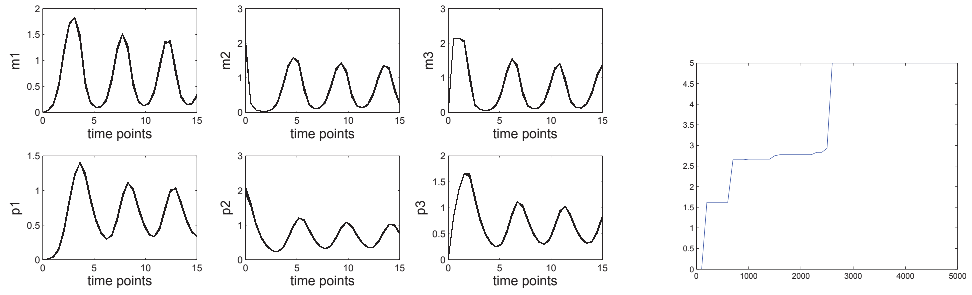


(a) (b)

Figure 10.3: (a)Time profile of all the species in the repressilator pathway based on the best parameters returned by SRES based parameter estimation,(b) objective value vs number of generations, r=0.9

Table 10.1: Repressilator pathway: Unknown parameters with range : SRES

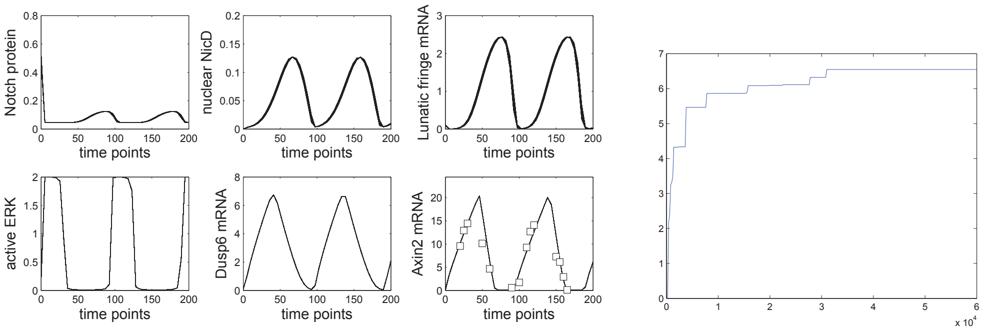
$Parameter$	$range$	$SRES, r = 0.8$	$SRES(p - value), r = 0.8$	$(SRES), r = 0.9$	$SRES(p - value), r = 0.9$
$\alpha1$	$[0, 100]$	81.21886	71.4383	80.0087	86.15479
$\alpha2$	$[0, 100]$	51.95532	69.58357	92.04954	68.90892
$\alpha3$	$[0, 100]$	75.57755	72.6164	56.14092	73.12696
$\gamma1$	$[0, 200]$	189.7099	152.5638	168.5096	178.5928
$\gamma2$	$[0, 200]$	88.04731	139.5069	176.3156	130.0404
$\gamma3$	$[0, 200]$	163.9563	154.6911	140.9322	156.9079
$k1$	$[0, 16]$	10.86995	11.94785	6.883317	11.73143
$k2$	$[0, 16]$	8.125588	8.763583	7.521114	12.15338
$k3$	$[0, 16]$	11.99097	10.42376	12.6742	14.44549



(a)

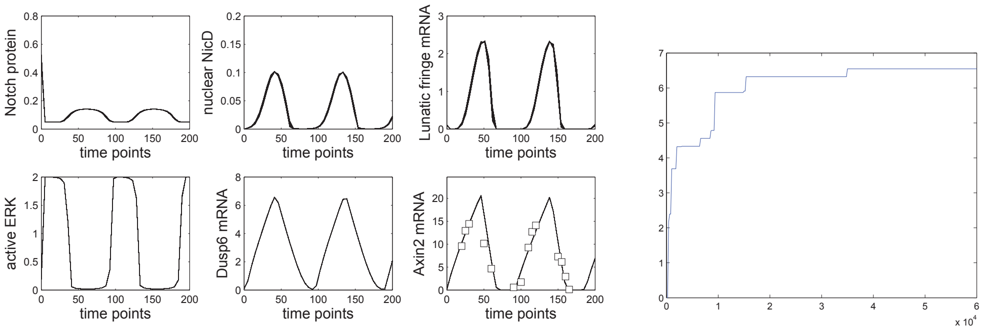
(b)

Figure 10.4: (a)Time profile of all the species in the repressilator pathway based on the best parameters using the p-value based, SRES search,(b) objective value vs number of generations, r=0.9



(a) (b)

Figure 10.5: Segmentation clock (a)Parameter estimation results - training and test data - SRES algorithm (b) objective value vs number of generations, r=0.8

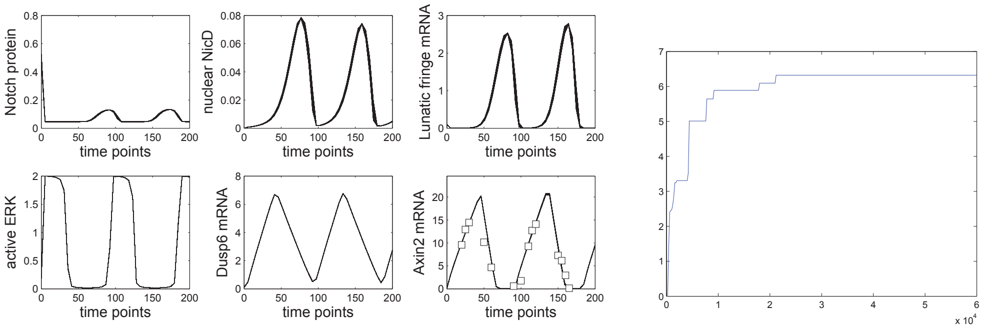


(a) (b)

Figure 10.6: Segmentation clock (a)Parameter estimation results - training and test data - SRES algorithm - p-value (b) objective value vs number of generations, r=0.8

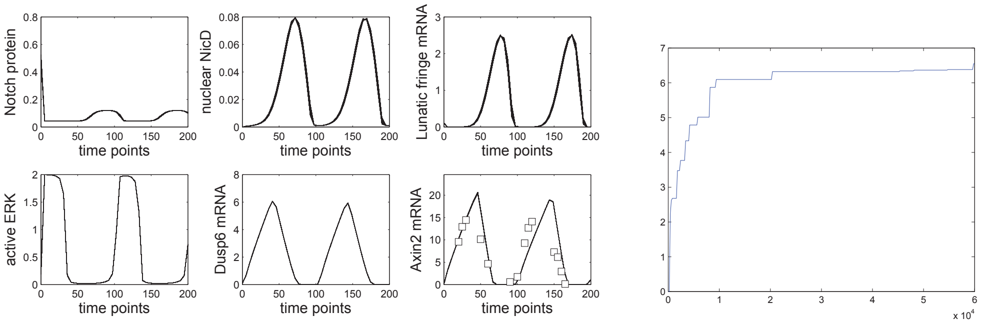
Table 10.2: Segmentation Clock pathway: Unknown parameters with range : SRES

ID	$Parameter$	$range$	$SRES, r = 0.8$	$SRES(p - value), r = 0.8$	$(SRES), r = 0.9$	$SRES(p - value), r = 0.9$
$k1$	KdN	$[0, 2.8]$	1.854774	2.217943	2.315076	2.69582
$k2$	vsN	$[0, 0.46]$	0.2254612	0.2586278	0.2352315	0.2306074
$k3$	vdN	$[0, 5.64]$	3.016938	4.191141	4.595336	4.409697
$k4$	$kt1$	$[0, 0.2]$	0.1066553	0.1094273	0.07170109	0.08448644
$k5$	$kt2$	$[0, 0.2]$	0.1228041	0.1841493	0.194562	0.03265237
$k6$	$KdNan$	$[0, 0.002]$	0.001628184	0.0005853016	0.0005344087	0.0007947555
$k7$	$VdNan$	$[0, 0.2]$	0.1067782	0.0914824	0.09176074	0.115546
$k8$	$KdMF$	$[0, 1.536]$	1.395118	0.8349247	1.130922	1.501019
$k9$	$KIG1$	$[0, 5]$	1.969339	1.870566	4.074387	3.100746
$k10$	vsF	$[0, 6]$	2.358976	2.90498	3.354143	5.584448
$k11$	vmF	$[0, 3.84]$	3.098625	2.905488	3.231351	3.670347
$k12$	KdF	$[0, 0.74]$	0.2501358	0.6605053	0.2122703	0.3421939
$k13$	vdF	$[0, 0.78]$	0.6268464	0.6216776	0.7059511	0.7366934
$k14$	ksF	$[0, 0.6]$	0.2876905	0.4768662	0.4896845	0.3595641
$k15$	$kd2$	$[0, 14.124]$	4.661996	3.49936	2.54389	4.613024
$k16$	vMB	$[0, 3.28]$	1.432242	0.1834212	0.46476	0.3118348
$k17$	KaB	$[0, 1.4]$	1.187312	0.9453801	1.314423	1.33507
$k18$	$vMXa$	$[0, 1]$	0.9953178	0.989499	0.9818487	0.9960803
$k19$	$ksAx$	$[0, 0.04]$	0.03657672	0.03321188	0.01616315	0.003290748
$k20$	$vdAx$	$[0, 1.2]$	0.05869855	0.2735278	0.09342579	0.5846336
$k21$	$KdAx$	$[0, 1.26]$	0.5040457	0.947641	0.7892819	0.869053
$k22$	$kt3$	$[0, 1.4]$	0.08752867	0.9508061	0.6430629	0.1705873
$k23$	$kt4$	$[0, 3]$	2.460013	2.635853	2.711086	2.202319
$k24$	$ksDusp$	$[0, 1]$	0.6604028	0.8951006	0.9289015	0.4887567
$k25$	$vdDusp$	$[0, 4]$	2.230291	2.920256	2.269688	2.257857
$k26$	$KdDusp$	$[0, 1]$	0.03116861	0.1623344	0.6197283	0.6940275
$k27$	$kcDusp$	$[0, 2.7]$	2.352255	0.8794429	0.5670736	1.910287
$k28$	$KaFgf$	$[0, 1]$	0.03527007	0.73803	0.3965763	0.2437455
$k29$	$KaRas$	$[0, 0.206]$	0.1144681	0.1505811	0.08747173	0.1371592
$k30$	$KdRas$	$[0, 0.2]$	0.1080222	0.1814883	0.1394507	0.1714378
$k31$	$KaMDusp$	$[0, 1]$	0.6799779	0.9006577	0.5618566	0.5469411
$k32$	$KdMDusp$	$[0, 1]$	0.9590261	0.7786136	0.3420334	0.2816923
$k33$	$VMsMDusp$	$[0, 1.8]$	1.344481	1.401655	1.352437	1.144567
$k34$	$VMdMDusp$	$[0, 1]$	0.7772506	0.7288679	0.8075002	0.4979048
$k35$	$VMaRas$	$[0, 9.936]$	8.065443	8.570167	8.82782	6.304438
$k36$	$VMdRas$	$[0, 0.82]$	0.3543762	0.7806555	0.4856174	0.4095424
$k37$	$VMaErk$	$[0, 6.6]$	6.375076	5.869864	4.52774	5.053729
$k38$	$VMaX$	$[0, 3.2]$	3.097873	2.386614	2.121978	1.573657
$k39$	$VMdX$	$[0, 1]$	0.537238	0.8771659	0.6829796	0.7114252



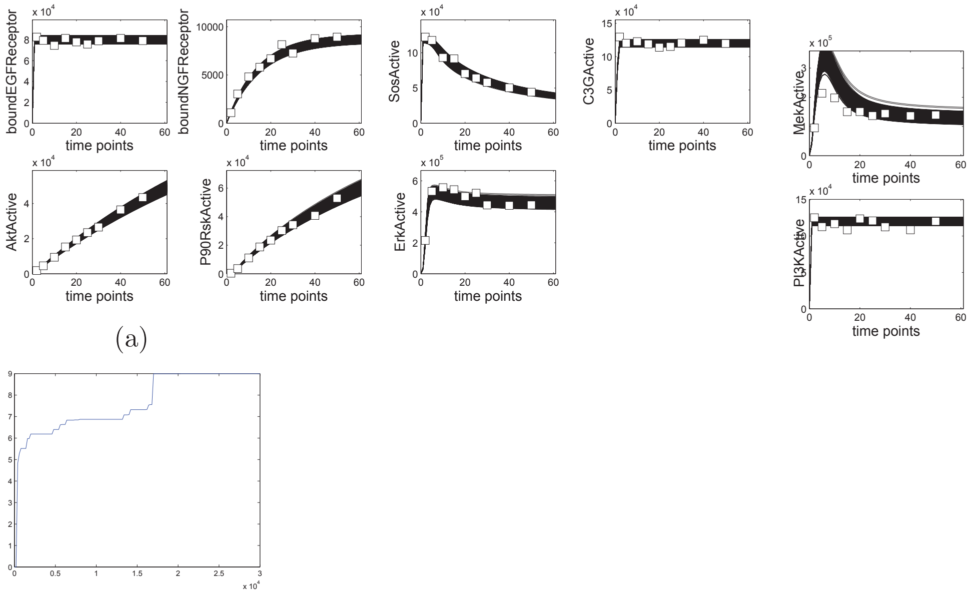
(a) (b)

Figure 10.7: Segmentation clock (a)Parameter estimation results - training and test data - SRES algorithm (b) objective value vs number of generations, r=0.9



(a) (b)

Figure 10.8: Segmentation clock (a)Parameter estimation results - training and test data - SRES algorithm - p-value(b) objective value vs number of generations, r=0.9



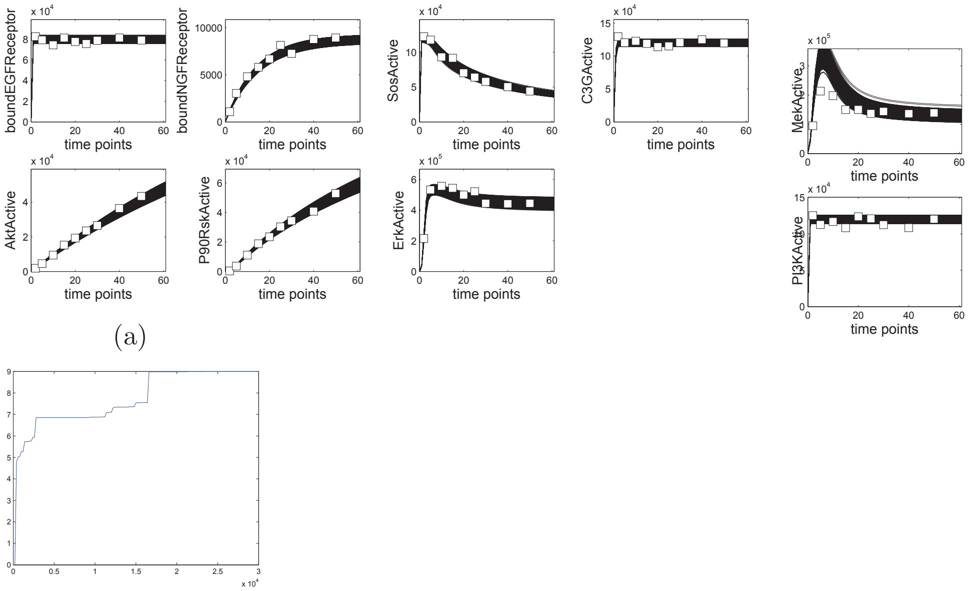
(a)

(b)

Figure 10.9: EGF-NGF pathway (a)Parameter estimation results - training and test data - SRES algorithm (b) objective value vs number of generations, r=0.8

Table 10.3: EGF-NGF pathway: Unknown parameters with range : SRES

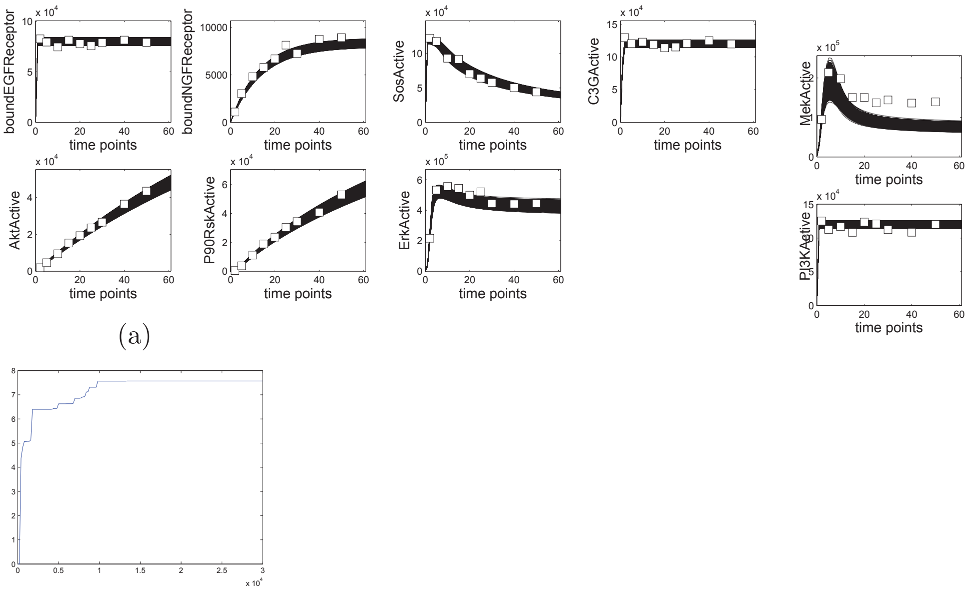
ID	$Parameter$	$range$	$SRES, r = 0.8$	$SRES(p - value), r = 0.8$	$(SRES), r = 0.9$	$SRES(p - value), r = 0.9$
1	$k1$	$[0, 0.000218503]$	$9.691E - 05$	$8.963E - 05$	$7.507E - 05$	$1.474E - 04$
2	$k2$	$[0, 0.121008]$	$1.156E - 02$	$6.262E - 02$	$5.840E - 02$	$9.109E - 03$
3	$k3$	$[0, 0.00000138209]$	$1.353E - 07$	$1.366E - 07$	$1.381E - 07$	$1.372E - 07$
4	$k4$	$[0, 0.0723811]$	$8.148E - 03$	$7.956E - 03$	$1.167E - 02$	$9.021E - 03$
5	$k11$	$[0, 323.44]$	$4.914E + 01$	$1.659E + 02$	$3.469E + 01$	$2.530E + 02$
6	$k12$	$[0, 359543]$	$3.275E + 05$	$3.072E + 05$	$3.220E + 05$	$1.113E + 05$
7	$k15$	$[0, 8.84096]$	$2.202E + 00$	$1.298E + 00$	$2.962E + 00$	$8.484E - 01$
8	$k17$	$[0, 1857.59]$	$7.702E + 01$	$5.948E + 01$	$1.152E + 02$	$4.744E + 01$
9	$k23$	$[0, 98.5367]$	$1.362E + 01$	$6.885E + 00$	$1.071E + 01$	$5.036E + 00$
10	$k27$	$[0, 0.213697]$	$1.622E - 01$	$1.578E - 01$	$1.092E - 01$	$1.928E - 01$
11	$k28$	$[0, 7635230]$	$6.283E + 06$	$6.305E + 06$	$4.411E + 06$	$7.609E + 06$
12	$k29$	$[0, 106.737]$	$1.222E + 01$	$2.716E + 01$	$2.662E + 01$	$8.661E + 01$
13	$k33$	$[0, 0.566279]$	$4.360E - 01$	$4.463E - 01$	$4.652E - 01$	$4.354E - 01$
14	$k34$	$[0, 6539510]$	$5.866E + 06$	$6.200E + 06$	$6.420E + 06$	$6.238E + 06$
15	$k37$	$[0, 1469.12]$	$3.853E + 02$	$4.447E + 02$	$7.847E + 02$	$8.062E + 02$
16	$k38$	$[0, 128762]$	$2.829E + 04$	$1.584E + 04$	$1.228E + 05$	$2.708E + 04$
17	$k39$	$[0, 14.0145]$	$1.858E + 00$	$9.790E + 00$	$5.759E + 00$	$1.033E + 01$
18	$k40$	$[0, 109656]$	$4.003E + 01$	$2.394E + 02$	$1.909E + 02$	$2.197E + 02$
19	$k43$	$[0, 22.0995]$	$4.906E + 00$	$1.458E + 01$	$9.742E + 00$	$1.899E + 01$
20	$k44$	$[0, 10254600]$	$3.744E + 06$	$5.994E + 06$	$4.885E + 06$	$6.776E + 06$



(a)

(b)

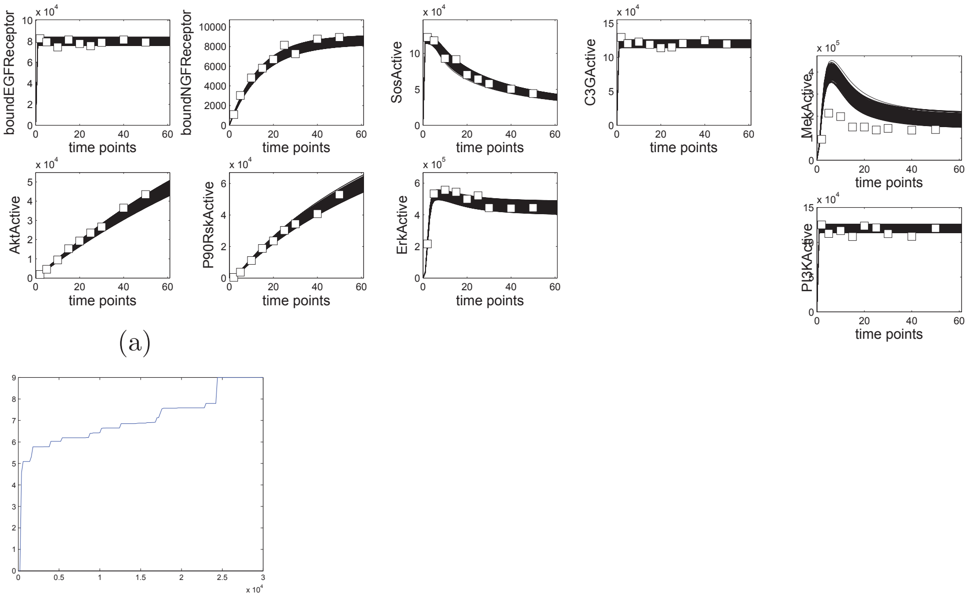
Figure 10.10: EGF-NGF pathway (a)Parameter estimation results - training and test data - SRES algorithm - p-value (b) objective value vs number of generations, r=0.8



(a)

(b)

Figure 10.11: EGF-NGF pathway (a)Parameter estimation results - training and test data - SRES algorithm (b) objective value vs number of generations, r=0.9



(a)

(b)

Figure 10.12: EGF-NGF pathway (a)Parameter estimation results - training and test data - SRES algorithm - p-value(b) objective value vs number of generations, r=0.9

Pathway	number of parameters	search algorithm setting	SRES	Avg sample size per test	(min,max) samples	SRES-pvalue	Avg sample size per test	(min,max) samples
Repressilator	9	Gen : 50 Pop : 100	54.94sec	12.96	$(3, 439)$	46.64 sec	10.35	$(3, 100)$
Clock	39	Gen : 300 Pop : 200	2.36hrs	45	$(6, 1484)$	2.1 hrs	41.53	$(6, 100)$
EGF-NGF	20	Gen : 150 Pop : 200	2.9 hrs	150.11	$(37, 1831)$	1.7 hrs	83.3	$(37, 100)$

Table 10.4: Summary of parameter estimation tasks

10.2 TLR3-TLR7 : the ODE model

$\frac{d(x0)}{dt}$ = -(1*k0*x2*x0) ;

$\frac{d(x1)}{dt}$ = +(1*k0*x2*x0) -(1*k1*x1) ;

$\frac{d(x2)}{dt}$ = -(1*k0*x2*x0) +(1*k0*x2*x0) ;

$\frac{d(x3)}{dt}$ = +(1*k2*x1*x4) -(1*k3*x3*x5*x6) ;

$\frac{d(x4)}{dt}$ = -(1*k2*x1*x4) +(1*k6*x7) ;

$\frac{d(x5)}{dt}$ = -(1*k3*x3*x5*x6) +(1*k18*x10) ;

$\frac{d(x6)}{dt}$ = -(1*k3*x3*x5*x6) -(1*(k25*x6*x8-k26*x9)) +(1*k18*x10) -(1*k60*x42*x44*x6) +(1*k70*x45) ;

$\frac{d(x7)}{dt}$ = +(1*k3*x3*x5*x6) -(1*k6*x7) ;

$\frac{d(x8)}{dt}$ = -(1*(k25*x6*x8-k26*x9)) + (1*k34*x26) - (1*k36*x8) - (1*k53*x22*x8) + (1*k53*x22*x8) ;

$\frac{d(x9)}{dt}$ = +(1*(k25*x6*x8-k26*x9)) ;

$\frac{d(x10)}{dt}$ = +(1*k6*x7) -(1*k7*x10*x59) +(1*k7*x10*x59) -(1*(k8*x10*x11-k27*x33)) - (1*k18*x10) ;

$\frac{d(x11)}{dt}$ = -(1*(k8*x10*x11-k27*x33)) -(1*(k58*x42*x11-k59*x43)) +(1*k69*x43) ;

$\frac{d(x12)}{dt}$ = +(1*k23*x16) -((k9*x61*x12)/(x12+k82));

$\frac{d(x13)}{dt}$ = +(1*k24*x17) -((k10*x61*x13)/(x13+k83));

$\frac{d(x14)}{dt}$ = -(1*k11*x14*x16) +(1*k17*x15) ;

$\frac{d(x15)}{dt}$ = +(1*k11*x14*x16) -(k19*x15*x20) +(k19*x15*x20) -(1*k17*x15) ;

$\frac{d(x16)}{dt}$ = -(1*k11*x14*x16) + (1*k11*x14*x16) - (1*k23*x16) + (1*((k9*x61*x12)/(x12+k82)));

$\frac{d(x17)}{dt}$ = -(1*k12*x19*x17) + (1*k12*x19*x17) - (1*k24*x17) + (1*(k10*x61*x13)/(x13+k83));

$\frac{d(x18)}{dt}$ = +(1*k12*x19*x17) -(k13*x18*x20) +(k13*x18*x20) -(1*k22*x18) ;

$\frac{d(x19)}{dt}$ = -(1*k12*x19*x17) +(1*k22*x18) ;

$\frac{d(x20)}{dt}$ = -(k19*x15*x20) -(k13*x18*x20) +(k28*x78) -(k31*x31*x20) ;

$\frac{d(x21)}{dt}$ = -(1*k80*x21*x61) +(k20) -(1*k49*x21) -(1*k103*x74*x21) ;

$\frac{d(x22)}{dt}$ = -(1*k14*x22*x24) +(1*k54*x35) -(1*k39*x22*x23) +(1*k40*x36) -(1*k41*x22) -(1*k50*x22) -(1*k53*x22*x8) +(1*k81*x60) +(1*k103*x74*x21) ;

$\frac{d(x23)}{dt}$ = -(1*k16*x25*x23) +(1*k47*x27) -(1*k48*x23) -(1*k39*x22*x23) -(k44*x23-k45*x75) ;

$\frac{d(x24)}{dt}$ = -(1*k14*x22*x24) +(1*k16*x25*x23) -(1*k38*x24) +(k43*x76) ;

Identifier	Name	Initial concentration
x0	TLR7	0.5311
x1	bTLR7	0
x2	R848	0
x3	actMyD88	0
x4	MyD88	0.87387
x5	IRAK1-4	0.12871
x6	TRAF6	1
x7	actMyD88-IRAK1-4-TRAF6	0
x8	A20	0.0048
x9	TRAF6-A20	0
x10	actIRAK1-4-TRAF6	0
x11	TRAF3	0.2
x12	Mkk1-2	0.99308
x13	Mkk3-6	0.00017715
x14	ERK	0.075538
x15	actERK	0
x16	actMkk1-2	0
x17	actMkk3-6	0
x18	actp38	0
x19	p38	0.45797
x20	AP-1	0.2
x21	NEMO:IKK-b:IKK-a	0.2
x22	NEMO:IKK-b-p:IKK-a	0
x23	IkBa	0.0025
x24	IkBa-Nfkb	0.0592
x25	Nfkb	0.003
x26	A20mRNA	0
x27	IkBamRNA	0
x28	IRF7	0.2
x29	JNK	0.31951
x30	actMkk4-7	0
x31	pJNK	0
x32	Mkk4-7	0.0064145
x33	actIRAK1-4-TRAF6-TRAF3	0
x34	IkBa-p	0
x35	NEMO:IKK-b-p:IKK-a-Nfkb-IkBa	0
x36	NEMO:IKK-b-p:IKK-a-IkBa	0
x37	inactiveIKK	0
x38	Poly(I:C)	0
x39	TLR3	0.21814
x40	bTLR3	0
x41	TRIF	0.2
x42	actTRIF	0
x43	actTRIF-TRAF3	0
x44	TRADD-FADD-RIP1	0.2
x45	actTRIF-TRADD-FADD-RIP1-TRAF6	0

Table 10.5: TLR3-TLR7 Pathway. List of species

Identifier	Name	Initial concentration
x46	Ikki	0.2
x47	actIkki	0
x48	TBK1	0.17637
x49	actTBK1	0
x50	actTBK1Ikki	0
x51	IRF3	0.18486
x52	A	0
x53	A2	0
x54	A3	0
x55	B2	0
x56	B3	0
x57	IL6mRNA	0
x58	IL12mRNA	0
x59	Tak1-Tab2-Tab3	0.99997
x60	NEMO:IKK-b:IKK-actTak1-Tab2-Tab3	0
x61	actTak1-Tab2-Tab3	0
x62	B	0
x63	C	0
x64	C2	0
x65	C3	0
x66	Type1-IFN	0
x67	Tyk2-Jak1	0.26036
x68	actTyk2-Jak1	0
x69	Stat1-Stat2	0.5845
x70	actStat1-Stat2	0
x71	PI3k	0.6152
x72	activatedPI3k	0
x73	Akt	0.24483
x74	activatedAkt	0
x75	IkBa-n	0.0034
x76	IkBa-n-Nfkb-n	0.0001
x77	Nfkb-n	0.0023
x78	actAP-1	0
x79	nucactStat1-Stat2	0
x80	factX	0
x81	facY	0
x82	IRF3-p	0
x83	IRF7-p	0

Table 10.6: TLR3-TLR7 pathway. List of species

Parameter	Value
$k14$	60
$k15$	0.15
$k16$	30
$k20$	0.0015
$k21$	0.4798
$k22$	0.3374
$k34$	30
$k38$	0.0012
$k39$	12
$k40$	6
$k41$	0.09
$k42$	30
$k43$	0.6
$k44$	0.085
$k45$	0.04
$k46$	0.024
$k47$	30
$k48$	0.006
$k49$	0.0075
$k50$	0.0075
$k51$	0.0075
$k52$	0.00003
$k53$	6
$k54$	6
$k55$	0.00003

Table 10.7: TLR3-TLR7 Pathway. List of known parameters

$$\frac{d(x25)}{dt} = -(1*k16*x25*x23) -(k15*x25) +(1*k38*x24) +(1*k54*x35) ;$$

$$\frac{d(x26)}{dt} = +(k52*x77) -(1*k34*x26) +(1*k34*x26) -(1*k35*x26) ;$$

$$\frac{d(x27)}{dt} = +(k55*x77) -(1*k47*x27) +(1*k47*x27) -(1*k46*x27) ;$$

$$\frac{d(x28)}{dt} = -(k4*x33*x28) +(k5*x83) ;$$

$$\frac{d(x29)}{dt} = -(1*k29*x29*x30) +(1*k32*x31) ;$$

$$\frac{d(x30)}{dt} = -(1*k29*x29*x30) +(1*k29*x29*x30) - (1*k33*x30) + (1*(k30*x61*x32)/(x32+k84));$$

$$\frac{d(x31)}{dt} = +(1*k29*x29*x30) -(k31*x31*x20) +(k31*x31*x20) -(1*k32*x31) ;$$

$$\frac{d(x32)}{dt} = +(1*k33*x30) -(1*(k30*x61*x32)/(x32+k84));$$

$$\frac{d(x33)}{dt} = +(1*(k8*x10*x11-k27*x33)) -(k4*x33*x28) +(k4*x33*x28) ;$$

$$\frac{d(x34)}{dt} = -(1*k37*x34) +(1*k38*x24) +(1*k54*x35) +(1*k40*x36) ;$$

$$\frac{d(x35)}{dt} = +(1*k14*x22*x24) -(1*k54*x35) ;$$

$$\frac{d(x36)}{dt} = +(1*k39*x22*x23) -(1*k40*x36) ;$$

$$\frac{d(x37)}{dt} = +(1*k41*x22) -(1*k51*x37) +(1*k53*x22*x8) ;$$

$$\frac{d(x38)}{dt} = -(1*k56*x38*x39) +(1*k56*x38*x39) ;$$

$$\frac{d(x39)}{dt} = -(1*k56*x38*x39) ;$$

$$\frac{d(x40)}{dt} = +(1*k56*x38*x39) -(1*k57*x40*x41) +(1*k57*x40*x41) -(1*k67*x40) ;$$

$$\frac{d(x41)}{dt} = -(1*k57*x40*x41) +(1*k69*x43) +(1*k70*x45) ;$$

$$\frac{d(x42)}{dt} = +(1*k57*x40*x41) -(1*(k58*x42*x11-k59*x43)) -(1*k60*x42*x44*x6) ;$$

$$\frac{d(x43)}{dt} = +(1*(k58*x42*x11-k59*x43)) -(1*k69*x43) ;$$

$$\frac{d(x44)}{dt} = -(1*k60*x42*x44*x6) +(1*k70*x45) ;$$

$$\frac{d(x45)}{dt} = +(1*k60*x42*x44*x6) -(1*k61*x45*x59) +(1*k61*x45*x59) -(1*k70*x45) ;$$

$$\frac{d(x46)}{dt} = -(1*k62*x43*x46) +(1*k66*x50) ;$$

$$\frac{d(x47)}{dt} = +(1*k62*x43*x46) -(1*k64*x49*x47) ;$$

$$\frac{d(x48)}{dt} = -(1*k63*x43*x48) +(1*k66*x50) ;$$

$$\frac{d(x49)}{dt} = +(1*k63*x43*x48) -(1*k64*x49*x47) ;$$

$$\frac{d(x50)}{dt} = +(1*k64*x49*x47) -(k65*x50*x51) +(k65*x50*x51) -(1*k66*x50) ;$$

$$\frac{d(x51)}{dt} = -(k65*x50*x51) +(k68*x82) ;$$

$$\frac{d(x52)}{dt} = -(1*k71*x52) - (1*k72*x52) + k113*(1-((1/(1+k105*x78))* (1/(1+k106*x78*x77*x81^1.5))*$$

$$(1/(1+k107*x77))* (1/(1+k108*x77*x80*x78)))) ;$$

$$\frac{d(x53)}{dt} = +(1*k72*x52) -(1*k73*x53) ;$$

$$\frac{d(x54)}{dt} = +(1*k73*x53) -(1*k74*x54) ;$$

$$\frac{d(x55)}{dt} = +(1*k75*x62) -(1*k76*x55) ;$$

$\frac{d(x56)}{dt}$ = +(1*k76*x55) -(1*k77*x56) ;

$\frac{d(x57)}{dt}$ = +(1*k74*x54) -(1*k78*x57) ;

$\frac{d(x58)}{dt}$ = +(1*k77*x56) -(1*k79*x58) ;

$\frac{d(x59)}{dt}$ = -(1*k7*x10*x59) -(1*k61*x45*x59) +(1*k21*x61) ;

$\frac{d(x60)}{dt}$ = +(1*k80*x21*x61) -(1*k81*x60) ;

$\frac{d(x61)}{dt}$ = +(1*k7*x10*x59) - (1*k80*x21*x61) + (1*k61*x45*x59) - (1*k21*x61) + (1*k81*x60) ;

$\frac{d(x62)}{dt}$ = -(1*k75*x62) + k114*(1-((1/(1+k109*x77)) * (1/(1+k110*x78*x77*$x81^1$.5)) * (1/(1+k111*x78)) *(1/(1+k112*x77*x82*x78))));

$\frac{d(x63)}{dt}$ = +(k85*(0.0001+0.9999*(1-(1/(1+k87*x82))*(1/(1+k88*x78*x82*x77))))) -(1*k121*x63) -(1*k125*x63) +(k86*(0.0001+0.9999(1-(1/(1+k89*x83)))));

$\frac{d(x64)}{dt}$ = +(1*k121*x63) -(1*k122*x64) ;

$\frac{d(x65)}{dt}$ = +(1*k122*x64) -(1*k123*x65) ;

$\frac{d(x66)}{dt}$ = +(1*k123*x65) -(1*k124*x66) -(1*k90*x66*x67) +(1*k90*x66*x67) + (k98*(0.0001+0.9999(1-(1/(1+k99*x79))))) - (1*k100*x66*x71) + (1*k100*x66*x71) ;

$\frac{d(x67)}{dt}$ = -(1*k90*x66*x67) +(1*k92*x68) ;

$\frac{d(x68)}{dt}$ = +(1*k90*x66*x67) -(1*k91*x68*x69) +(1*k91*x68*x69) -(1*k92*x68) ;

$\frac{d(x69)}{dt}$ = -(1*k91*x68*x69) +(1*k93*x70) ;

$\frac{d(x70)}{dt}$ = +(1*k91*x68*x69) -(1*k93*x70) -(k94*x70-k95*x79) ;

$\frac{d(x71)}{dt}$ = -(1*k100*x66*x71) +(1*k102*x72) ;

$\frac{d(x72)}{dt}$ = +(1*k100*x66*x71) -(1*k101*x72*x73) +(1*k101*x72*x73) -(1*k102*x72) ;

$\frac{d(x73)}{dt}$ = -(1*k101*x72*x73) +(1*k104*x74) ;

$\frac{d(x74)}{dt}$ = +(1*k101*x72*x73) -(1*k103*x74*x21) +(1*k103*x74*x21) -(1*k104*x74) ;

$\frac{d(x75)}{dt}$ = -(0.2*k42*x77*x75) +(k44*x23-k45*x75) ;

$\frac{d(x76)}{dt}$ = +(0.2*k42*x77*x75) -(k43*x76) ;

$\frac{d(x77)}{dt}$ = -(k52*x77) +(k52*x77) - (k55*x77) + (k55*x77) +(k15*x25) - (0.2*k42*x77*x75);

$\frac{d(x78)}{dt}$ = +(k19*x15*x20) +(k13*x18*x20) -(k28*x78) +(k31*x31*x20) ;

$\frac{d(x79)}{dt}$ = +(k94*x70-k95*x79) ;

$\frac{d(x80)}{dt}$ = +(0.2*k115*x82) -(0.2*k116*x80) ;

$\frac{d(x81)}{dt}$ = +(0.2*(k96*(0.0001+0.9999(1-(1/(1+k97*x79)))))) -(0.2*k126*x81) ;

$\frac{d(x82)}{dt}$ = +(k65*x50*x51) -(k68*x82) -(0.2*k115*x82) +(0.2*k115*x82) ;

$\frac{d(x83)}{dt}$ = +(k4*x33*x28) -(k5*x83) ;

Bibliography

[1] Bing Liu, Jing Zhang, Pei Yi Tan, David Hsu, Anna M. Blom, Benjamin Leong, Sunil Sethi, Bow Ho, Jeak Ling Ding, and P. S. Thiagarajan. A computational and experimental study of the regulatory mechanisms of the complement system. *PLoS Computational Biology*, 7(1):e1001059, 2011.

[2] B. Liu, D. Hsu, and PS Thiagarajan. Probabilistic approximations of odes based bio-pathway dynamics. *Theoretical Computer Science*, 412(21):2188–2206, 2011.

[3] P.L. Ståhl and J. Lundeberg. Toward the single-hour high-quality genome. *Annual Review of Biochemistry*, 81:359–378, 2012.

[4] M. Ashburner, C.A. Ball, J.A. Blake, D. Botstein, H. Butler, J.M. Cherry, A.P. Davis, K. Dolinski, S.S. Dwight, J.T. Eppig, et al. Gene ontology: tool for the unification of biology. *Nature genetics*, 25(1):25, 2000.

[5] I. Schomburg, A. Chang, and D. Schomburg. Brenda, enzyme data and metabolic information. *Nucleic acids research*, 30(1):47–49, 2002.

[6] J.L. Sussman, D. Lin, J. Jiang, N.O. Manning, J. Prilusky, O. Ritter, and EE Abola. Protein data bank : database of three-dimensional structural information of biological macromolecules. *Acta Crystallographica Section D: Biological Crystallography*, 54(6):1078–1084, 1998.

[7] A. Bairoch and B. Boeckmann. The SWISS-PROT protein sequence data bank. *Nucleic Acids Research*, 20(suppl):2019, 1992.

[8] R. Apweiler, A. Bairoch, C.H. Wu, W.C. Barker, B. Boeckmann, S. Ferro, E. Gasteiger, H. Huang, R. Lopez, M. Magrane, et al. Uniprot: the universal protein knowledgebase. *Nucleic acids research*, 32(suppl 1):D115–D119, 2004.

[9] V. Matys, E. Fricke, R. Geffers, E. Goessling, M. Haubrock, R. Hehl, K. Hornischer, D. Karas, A.E. Kel, O.V. Kel-Margoulis, et al. Transfac®: transcriptional regulation, from patterns to profiles. *Nucleic acids research*, 31(1):374–378, 2003.

[10] P.K. Kreeger and D.A. Lauffenburger. Cancer systems biology: a network modeling perspective. *Carcinogenesis*, 31(1):2–8, 2010.

[11] H. Kitano. Systems biology: a brief overview. *Science*, 295(5560):1662–1664, 2002.

[12] Bing Liu, David Hsu, and P. S. Thiagarajan. Probabilistic Approximations of ODEs based Bio-Pathway Dynamics. *Theor. Comput. Sci.*, 412:2188–2206, 2011.

[13] Edmund M. Clarke, Orna Grumberg, and Doron A. Peled. *Model Checking*. MIT Press, 1999.

[14] Daphne Koller and Nir Friedman. *Probabilistic Graphical Models - Principles and Techniques*. MIT Press, 2009.

[15] Xavier Boyen and Daphne Koller. Tractable Inference for Complex Stochastic Processes. In *Proc. 14th Int. Conf. Uncertainty in Artificial Intelligence (UAI '98)*, pages 33–42, 1998.

[16] Kevin P. Murphy and Yair Weiss. The Factored Frontier Algorithm for Approximate Inference in DBNs. In *Proc. 17th Int. Conf. Uncertainty in Artificial Intelligence (UAI '01)*, pages 378–385, 2001.

[17] S.L. Spencer, S. Gaudet, J.G. Albeck, J.M. Burke, and P.K. Sorger. Non-genetic origins of cell-to-cell variability in trail-induced apoptosis. *Nature*, 459(7245):428–432, 2009.

[18] Sumit K. Jha, Edmund M. Clarke, Christopher J. Langmead, Axel Legay, AndrÃl' Platzer, and Paolo Zuliani. A bayesian approach to model checking biological systems. In *Proceedings of the 7th International Conference on Computational Methods in Systems Biology*, CMSB '09, pages 218–234, Berlin, Heidelberg, 2009. Springer-Verlag.

[19] E. Simao, E. Remy, D. Thieffry, and C. Chaouiya. Qualitative modelling of regulated metabolic pathways: application to the tryptophan biosynthesis in e. coli. *Bioinformatics*, 21(suppl 2):ii190–ii196, 2005.

[20] J. Fisher, N. Piterman, A. Hajnal, and T.A. Henzinger. Predictive modeling of signaling crosstalk during c. elegans vulval development. *PLoS Computational Biology*, 3(5):e92, 2007.

[21] M.A. Schaub, T.A. Henzinger, and J. Fisher. Qualitative networks: a symbolic approach to analyze biological signaling networks. *BMC systems biology*, 1(1):4, 2007.

[22] F. Hua, S. Hautaniemi, R. Yokoo, and D.A. Lauffenburger. Integrated mechanistic and data-driven modelling for multivariate analysis of signalling pathways. *Journal of The Royal Society Interface*, 3(9):515–526, 2006.

[23] K.A. Janes and M.B. Yaffe. Data-driven modelling of signal-transduction networks. *Nature Reviews Molecular Cell Biology*, 7(11):820–828, 2006.

[24] T. Immonen, R. Gibson, T. Leitner, M.A. Miller, E.J. Arts, E. Somersalo, and D. Calvetti. A hybrid stochastic-deterministic computational model accurately describes spatial dynamics and virus diffusion in hiv-1 growth competition assay. *Journal of Theoretical Biology*, 2012.

[25] Bree B. Aldridge, John M. Burke, Douglas A. Lauffenburger, and Peter K. Sorger. Physicochemical modelling of cell signalling pathways. *Nature Cell Biology*, 8:1195–1203, 2006.

[26] A.W. Leung. *Systems of Nonlinear Partial Differential Equations: Applications to Biology and Engineering*. Mathematics and Its Applications. Kluwer Academic Publishers, 1989.

[27] L Raeymaekers. Dynamics of Boolean networks controlled by biologically meaningful functions. *Journal of theoretical biology*, 218(3):331–341, oct 2002. PMID: 12381434.

[28] Hiroshi Matsuno, Yukiko Tanaka, Hitoshi Aoshima, Atsushi Doi, Mika Matsui, and Satoru Miyano. Biopathways representation and simulation on hybrid functional Petri net. *In Silico Biol*, 3(3):389–404, 2003.

[29] Derek Ruths, Melissa Muller, Jen Te Tseng, Luay Nakhleh, and Prahlad T. Ram. The signaling Petri net-based simulator: a non-parametric strategy for characteriz-

ing the dynamics of cell-specific signaling networks. *PLoS Computational Biology*, 4(2):1–15, 2008.

[30] Vincent Danos, Jérôme Feret, Walter Fontana, Russell Harmer, and Jean Krivine. Rule-based modelling of cellular signalling. In *CONCUR*, pages 17–41, 2007.

[31] Marta Z. Kwiatkowska, Gethin Norman, and David Parker. PRISM: Probabilistic Symbolic Model Checker. In *Proc. 12th Int. Conf. Modelling Techniques and Tools for Computer Performance Evaluation (TOOLS '02)*, pages 200–204, 2002.

[32] Federica Ciocchetta, Andrea Degasperi, Jane Hillston, and Muffy Calder. Some Investigations Concerning the CTMC and the ODE Model Derived From Bio-PEPA. *Electr. Notes Theor. Comput. Sci.*, 229(1):145–163, 2009.

[33] H.M. Sauro. *Enzyme Kinetics for Systems Biology*. Future Skill Software, 2011.

[34] E. Klipp, R. Herwig, A. Kowald, C. Wierling, and H. Lehrach. *Systems Biology in Practice*. Wiley-VCH Verlag GmbH & Co. KGaA, Weinheim, FRG, 2005.

[35] Morris W. Hirsch, Stephen Smale, and Robert L. Devaney. *Differential equations, dynamical systems and an introduction to chaos*. Elsevier, 2 edition, 2004.

[36] K.E. Atkinson. *An introduction to numerical analysis*. Wiley, 1989.

[37] J. D. Lambert. *Numerical Methods for Ordinary Differential Systems*. New York: Wiley, 1992.

[38] LR Petzold and AC Hindmarsh. Lsoda. *Computing and Mathematics Research Division, I-316 Lawrence Livermore National Laboratory, Livermore, CA*, 94550, 1997.

[39] Scott D. Cohen and Alan C. Hindmarsh. Cvode, a stiff/nonstiff ode solver in c. *Comput. Phys.*, 10(2):138–143, March 1996.

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

[41] K. Ahnert and M. Mulansky. Odeint-solving ordinary differential equations in c++. *arXiv preprint arXiv:1110.3397*, 2011.

[42] Gregory Batt, Calin Belta, and Ron Weiss. Temporal logic analysis of gene networks under parameter uncertainty. *IEEE Trans Circuits Syst I / Automat. Control (Special Issue on Systems Biology)*, 53:215–229, 2008.

[43] Hidde de Jong. Modeling and simulation of genetic regulatory systems: a literature review. *J Comput Biol*, 9(1):67–103, 2002.

[44] L. Glass and S.A. Kauffman. Co-operative components, spatial localization and oscillatory cellular dynamics. *Journal of theoretical biology*, 34(2):219–237, 1972.

[45] L. Glass and S.A. Kauffman. The logical analysis of continuous, non-linear biochemical control networks. *Journal of Theoretical Biology*, 39(1):103–129, 1973.

[46] R.B. Trelease, R.A. Henderson, and J.B. Park. A qualitative process system for modeling nf-κb and ap-1 gene regulation in immune cell biology research. *Artificial Intelligence in Medicine*, 17(3):303–321, 1999.

[47] T. Akutsu, S. Miyano, and S. Kuhara. Algorithms for inferring qualitative models of biological networks. In *Pacific Symposium on Biocomputing*, volume 5, pages 290–301, 2000.

[48] K.R. Heidtke and S. Schulze-Kremer. Design and implementation of a qualitative simulation model of lambda phage infection. *Bioinformatics*, 14(1):81–91, 1998.

[49] H.H. McAdams and A. Arkin. Stochastic mechanisms in gene expression. *Proceedings of the National Academy of Sciences*, 94(3):814–819, 1997.

[50] M.B. Elowitz, A.J. Levine, E.D. Siggia, and P.S. Swain. Stochastic gene expression in a single cell. *Science Signalling*, 297(5584):1183, 2002.

[51] D.T. Gillespie. Exact stochastic simulation of coupled chemical reactions. *The journal of physical chemistry*, 81(25):2340–2361, 1977.

[52] M.A. Gibson and J. Bruck. Efficient exact stochastic simulation of chemical systems with many species and many channels. *The journal of physical chemistry A*, 104(9):1876–1889, 2000.

[53] Y. Cao, H. Li, and L. Petzold. Efficient formulation of the stochastic simulation algorithm for chemically reacting systems. *The journal of chemical physics*, 121(9):4059–4067, 2004.

[54] H. Resat, H.S. Wiley, and D.A. Dixon. Probability-weighted dynamic monte carlo method for reaction kinetics simulations. *The Journal of Physical Chemistry B*, 105(44):11026–11034, 2001.

[55] D.T. Gillespie. Approximate accelerated stochastic simulation of chemically reacting systems. *The Journal of Chemical Physics*, 115(4):1716–1733, 2001.

[56] Federica Ciocchetta, Adam Duguid, Stephen Gilmore, Maria Luisa Guerriero, and Jane Hillston. The Bio-PEPA tool suite. In *QEST '09*, pages 309–310, Washington, DC, USA, 2009. IEEE Computer Society.

[57] Marta Kwiatkowska, Gethin Norman, and David Parker. PRISM: probabilistic symbolic model checker. In *Computer Performance Evaluation: Modelling Techniques and Tools*, pages 113–140. 2002.

[58] Marta Kwiatkowska, Gethin Norman, and David Parker. Using probabilistic model checking in systems biology. *SIGMETRICS Perform. Eval. Rev.*, 35(4):14–21, 2008.

[59] John Heath, Marta Kwiatkowska, Gethin Norman, David Parker, and Oksana Tymchyshyn. Probabilistic model checking of complex biological pathways. *Theor. Comput. Sci.*, 391(3):239–257, 2008.

[60] Paolo Ballarini, Radu Mardare, and Ivan Mura. Analysing biochemical oscillation through probabilistic model checking. *Electron. Notes Theor. Comput. Sci.*, 229(1):3–19, 2009.

[61] Muffy Calder, Vladislav Vyshemirsky, David Gilbert, and Richard Orton. Analysis of signalling pathways using continuous time markov chains. In *Transactions on Computational Systems Biology VI*, pages 44–67. 2006.

[62] G. Joshi-Tope, M. Gillespie, I. Vastrik, P. D'Eustachio, E. Schmidt, B. de Bono, B. Jassal, GR Gopinath, GR Wu, L. Matthews, et al. Reactome: a knowledgebase of biological pathways. *Nucleic acids research*, 33(suppl 1):D428–D432, 2005.

[63] H. Ogata, S. Goto, K. Sato, W. Fujibuchi, H. Bono, and M. Kanehisa. Kegg: Kyoto encyclopedia of genes and genomes. *Nucleic acids research*, 27(1):29–34, 1999.

[64] K. Levenberg. A method for the solution of certain nonlinear problems in least squares. *Quart. Appl. Math.*, 1994:164–168, 2.

[65] D.W. Marquardt. An algorithm for least squares estimation of nonlinear parameters. *SIAM Journal*, 11:431–441, 1963.

[66] D.B. Fogel, L.J. Fogel, and J.W. Atmar. Meta-evolutionary programming. In *25th Asiloma Conference on Signals, Systems and Computers.*, pages 540–545, Asilomar, 1992. IEEE Computer Society,.

[67] R. Hooke and T. A. Jeeves. "Direct search" solution of numerical and statistical problems. *Journal of the Association for Computing Machinery*, 8:212–229, 1961.

[68] T. Back, D.B. Fogel, and Z. Michalewicz. *Handbook of evolutionary computation.* Oxford University Press, 1997.

[69] T. Runarsson and X. Yao. Stochastic ranking for constrained evolutionary optimization. *IEEE Transactions on Evolutionary Computation*, 4:284–294, 2000.

[70] Carmen G. Moles, Pedro Mendes, and Julio R. Banga. Parameter estimation in biochemical pathways: A comparison of global optimization methods. *Genome Research*, 13(11):2467 –2474, 2003.

[71] Geoffrey Koh, Huey Fern Carol Teong, Marie-Veronique Clement, David Hsu, and P. S. Thiagarajan. A decompositional approach to parameter estimation in pathway modeling: a case study of the Akt and MAPK pathways and their crosstalk. volume 22, pages e271–e280, 2006.

[72] Geoffrey Koh, Lisa Tucker-Kellogg, David Hsu, and P. S. Thiagarajan. Composing globally consistent pathway parameter estimates through belief propagation. In *Proceedings of the 7th international workshop on Algorithms in Bioinformatics*, WABI '07, pages 420–430, Berlin, Heidelberg, 2007. Springer-Verlag.

[73] M. Morohashi, A.E. Winn, M.T. Borisuk, H. Bolouri, J. Doyle, and H. Kitano. Robustness as a measure of plausibility in models of biochemical networks. *Journal of theoretical biology*, 216(1):19–30, 2002.

[74] D. Battogtokh and J.J. Tyson. Bifurcation analysis of a model of the budding yeast cell cycle. *arXiv preprint q-bio/0404006*, 2004.

[75] J. Lu, H.W. Engl, P. Schuster, et al. Inverse bifurcation analysis: application to simple gene systems. *Algorithms Mol. Biol*, 1(11), 2006.

[76] George Von Dassow, Eli Meir, Edwin M Munro, and Garrett M Odell. The segment polarity network is a robust developmental module. *Nature*, 406(6792):188–192, 2000.

[77] Maria Rodriguez-Fernandez, Pedro Mendes, Julio R Banga, et al. A hybrid approach for efficient and robust parameter estimation in biochemical pathways. *Biosystems*, 83(2):248–265, 2006.

[78] Marta Cascante, Laszlo G Boros, Begoña Comin-Anduix, Pedro de Atauri, Josep J Centelles, Paul W-N Lee, et al. Metabolic control analysis in drug discovery and disease. *Nature Biotechnology*, 20(3):243–249, 2002.

[79] B. Schoeberl, C. Eichler-Jonsson, E.D. Gilles, and G. Muller. Computational modeling of the dynamics of the map kinase cascade activated by surface and internalized egf receptors. *Nature biotechnology*, 20(4):370–375, 2002.

[80] H.X. Zhang, W.P. Dempsey, and J. Goutsias. Probabilistic sensitivity analysis of biochemical reaction systems. *Journal of Chemical Physics*, 131(9):Art–No, 2009.

[81] A. Saltelli, M. Ratto, T. Andres, F. Campolongo, J. Cariboni, D. Gatelli, M. Saisana, and S. Tarantola. *Global sensitivity analysis: the primer*. Wiley-Interscience, 2008.

[82] M. Rodriguez-Fernandez, J.R. Banga, and F.J. Doyle III. Novel global sensitivity analysis methodology accounting for the crucial role of the distribution of input parameters: application to systems biology models. *International Journal of Robust and Nonlinear Control*, 2012.

[83] Kwang Hyun Cho, Sung Young Shin, Walter Kolch, and Olaf Wolkenhauer. Experimental design in systems biology, based on parameter sensitivity analysis using a Monte Carlo method: A case study for the TNFα-mediated NF-κB signal transduction pathway. *Simulation*, 79(12):726–739, 2003.

[84] Zhike Zi, Kwang Hyun Cho, Myong Hee Sung, Xuefeng Xia, Jiashun Zheng, and Zhirong Sun. In silico identification of the key components and steps in IFN-γ induced JAK-STAT signaling pathway. *FEBS Letters*, 579(5):1101–1108, 2005.

[85] Maria Rodriguez-Fernandez and Julio R. Banga. Global sensitivity analysis of a biochemical pathway model. In *IWPACBB*, pages 233–242, 2008.

[86] Zhike Zi, Yanan Zheng, Ann E Rundell, and Edda Klipp. SBML-SAT: a systems biology markup language (SBML) based sensitivity analysis tool. *BMC Bioinformatics*, 9:342, aug 2008.

[87] J.Y. Choi, J.W. Harvey, and M.H. Conklin. Use of multi-parameter sensitivity analysis to determine relative importance of factors influencing natural attenuation of mining contaminants. *US Geological Survey Toxic Substances Hydrology Program: Contamination from hard-rock mining*, 1:185, 1999.

[88] Nathalie Chabrier and François Fages. Symbolic model checking of biochemical networks. In *Proceedings of the First International Workshop on Computational Methods in Systems Biology*, pages 149–162, London, UK, UK, 2003. Springer-Verlag.

[89] Gregory Batt, Delphine Ropers, Hidde de Jong, Johannes Geiselmann, Radu Mateescu, Michel Page, and Dominique Schneider. Validation of qualitative models of genetic regulatory networks by model checking: analysis of the nutritional stress response in *Escherichia coli*. *Bioinformatics*, 21(suppl 1):i19 –i28, 2005.

[90] Pedro T. Monteiro, Delphine Ropers, Radu Mateescu, Ana T. Freitas, and Hidde de Jong. Temporal logic patterns for querying dynamic models of cellular interaction networks. *Bioinformatics*, 24(16):i227–233, aug 2008.

[91] Robin Donaldson and David Gilbert. A Monte Carlo model checker for probabilistic LTL with numerical constraints. Technical report, University of Glasgow, Department of Computing Science, 2008.

[92] Edmund Clarke, James Faeder, Christopher Langmead, Leonard Harris, Sumit Jha, and Axel Legay. Statistical model checking in BioLab: applications to the

automated analysis of T-Cell receptor signaling pathway. In *Computational Methods in Systems Biology*, pages 231–250. 2008.

[93] Amir Pnueli. The temporal logic of programs. In *FOCS'77*, pages 46–57, 1977.

[94] Edmund M. Clarke and E. Allen Emerson. Design and synthesis of synchronization skeletons using Branching-Time temporal logic. In *Logic of Programs, Workshop*, pages 52–71, London, UK, UK, 1982. Springer-Verlag.

[95] Jean-Pierre Queille and Joseph Sifakis. Specification and verification of concurrent systems in CESAR. In *Proceedings of the 5th Colloquium on International Symposium on Programming*, pages 337–351, London, UK, 1982. Springer-Verlag.

[96] William S Hlavacek. How to deal with large models. *Molecular Systems Biology*, 5:240, 2009.

[97] L. Calzone, N. Chabrier-Rivier, F. Fages, and S. Soliman. Machine learning biochemical networks from temporal logic properties. *Transactions on Computational Systems Biology VI*, pages 68–94, 2006.

[98] Jiri Barnat, Lubos Brim, Adam Krejci, Adam Streck, David Safranek, Martin Vejnar, and Tomas Vejpustek. On parameter synthesis by parallel model checking. *IEEE/ACM Trans. Comput. Biol. Bioinformatics*, 9(3):693–705, may 2012.

[99] Aurélien Rizk, Gregory Batt, François Fages, and Sylvain Soliman. On a continuous degree of satisfaction of temporal logic formulae with applications to systems biology. In *Proceedings of the 6th International Conference on Computational Methods in Systems Biology*, CMSB '08, pages 251–268, Berlin, Heidelberg, 2008. Springer-Verlag.

[100] Chen Li, Masao Nagasaki, Chuan Hock Koh, and Satoru Miyano. Online model checking approach based parameter estimation to a neuronal fate decision simulation model in Caenorhabditis elegans with hybrid functional Petri net with extension. *Molecular Biosystems*, 7(5):1576–92, 2011.

[101] S.M. Ross. Stochastic processes. 1996, 2001.

[102] Darren J. Wilkinson. Bayesian methods in bioinformatics and computational systems biology. *Briefings in Bioinformatics*, 8(2):109–116, mar 2007.

[103] N Friedman, M Linial, I Nachman, and D Pe'er. Using Bayesian networks to analyze expression data. *Journal of computational biology: a journal of computational molecular cell biology*, 7(3-4):601–620, 2000. PMID: 11108481.

[104] N.M. Oliver, B. Rosario, and A.P. Pentland. A Bayesian computer vision system for modeling human interactions. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 22(8):831 –843, aug 2000.

[105] Daphne Koller and Brian Milch. Multi-agent influence diagrams for representing and solving games. *Games and Economic Behavior*, 45(1):181–221, oct 2003.

[106] Robert Fung and Brendan Del Favero. Applying Bayesian networks to information retrieval. *Commun. ACM*, 38(3):42–ff., mar 1995.

[107] Pedro F. Felzenszwalb and Daniel P. Huttenlocher. Efficient Belief Propagation for Early Vision. *Int. J Comput. Vision*, 70:41–54, 2006.

[108] Robert J. Mceliece, David J. C. Mackay, and Jung fu Cheng. Turbo Decoding as an Instance of Pearl's "Belief Propagation" Algorithm. *IEEE J. Sel. Area. Comm.*, 16:140–152, 1998.

[109] N. Friedman. Inferring Cellular Networks Using Probabilistic Graphical Models. *Science*, 303:799–805, 2004.

[110] B.-E. Perrin, L. Ralaivola, A. Mazurie, S. Bottani, J. Mallet, and F. d'Alche-Buc. Gene networks inference using dynamic bayesian networks. *Bioinformatics*, 19(Suppl 2):ii138–ii148, oct 2003.

[111] Bing Liu, P. S. Thiagarajan, and David Hsu. Probabilistic Approximations of Signaling Pathway Dynamics. In *Proc. 7th Int. Conf. Computational Methods in Systems Biology (CMSB '09)*, pages 251–265, 2009.

[112] Liu Bing. *Probabilistic Approximation and Analysis Techniques for Bio-Pathway Models*. PhD thesis, National University of Singapore, 2010.

[113] Kevin Patrick Murphy. *Dynamic Bayesian Networks: Representation, Inference and Learning*. PhD thesis, University of California, Berkely, 2002.

[114] Sucheendra K. Palaniappan, S. Akshay, Blaise Genest, and P. S. Thiagarajan. A Hybrid Factored Frontier Algorithm for Dynamic Bayesian Network Models of Biopathways. In *Proc. 9th Int. Conf. on Computational Methods in Systems Biology (CMSB '11)*, pages 35–44, 2011.

[115] P. Bremaud. *Markov Chains: Gibbs Fields, Monte Carlo Simulation and Queues*. Springer, 2010.

[116] Kevin S. Brown, Colin C. Hill, Guillermo A. Calero, C R Myers, K H Lee, and Richard A. Cerione. The Statistical Mechanics of Complex Signaling Networks : Nerve Growth Factor Signaling. *Phys. Biol.*, 1:184–195, 2004.

[117] N. Le Novere, B. Bornstein, A. Broicher, M. Courtot, M. Donizelli, H. Dharuri, L. Li, H. Sauro, M. Schilstra, B. Shapiro, J.L. Snoep, and M. Hucka. BioModels Database: A free, centralized database of curated, published, quantitative kinetic models of biochemical and cellular systems. *Nucleic Acids Res*, 34:D689–D691, 2006.

[118] Marcel Schilling, Thomas Maiwald, Stefan Hengl, Dominic Winter, Clemens Kreutz, Walter Kolch, Wolf D Lehmann, Jens Timmer, and Ursula Klingmüller. Theoretical and Experimental Analysis Links Isoform-specific ERK Signalling to Cell Fate Decisions. *Mol. Syst. Biol.*, 5, 2009.

[119] Kevin P. Murphy. Bayes Net Toolbox for Matlab, 2012. http://bnt.googlecode.com.

[120] Boris N. Kholodenko. Untangling the Signalling Wires. *Nat. Cell Biol.*, 9:247–249, 2007.

[121] M.C. Browne, E.M. Clarke, and O. Grümberg. Characterizing finite kripke structures in propositional temporal logic. *Theoretical Computer Science*, 59(1):115–131, 1988.

[122] Hans Hansson and Bengt Jonsson. A logic for reasoning about time and reliability. *Formal Aspects of Computing*, 6:512–535, 1994.

[123] Christel Baier, Boudewijn Haverkort, Holger Hermanns, and Joost-Pieter Katoen. Model-Checking algorithms for Continuous-Time Markov Chains. *IEEE Trans. Softw. Eng.*, 29(6):524–541, Jun 2003.

[124] Muffy Calder, Vladislav Vyshemirsky, David Gilbert, and Richard Orton. Analysis of signalling pathways using the PRISM model checker. *Proceedings of Computational Methods in Systems Biology (CMSB 2005)*, pages 179—190, 2005.

[125] M.Y. Vardi and P. Wolper. An automata-theoretic approach to automatic program verification. In *Proceedings of the First Symposium on Logic in Computer Science*. IEEE Computer Society, 1986.

[126] C. Courcoubetis and M. Yannakakis. Verifying temporal properties of finite-state probabilistic programs. In *Foundations of Computer Science, 1988., 29th Annual Symposium on*, pages 338 –345, oct 1988.

[127] Kenneth L. McMillan. *Symbolic Model Checking*. Kluwer Academic Publishers, Norwell, MA, USA, 1993.

[128] E. M. Clarke, K. L. McMillan, X Zhao, M. Fujita, and J. Yang. Spectral transforms for large boolean functions with applications to technology mapping. In *Proceedings of the 30th international Design Automation Conference*, DAC '93, pages 54–60, New York, NY, USA, 1993. ACM.

[129] Håkan L. S Younes, Marta Kwiatkowska, Gethin Norman, and David Parker. Numerical vs. statistical probabilistic model checking. *International Journal on Software Tools for Technology Transfer.*, 8:216–228, June 2006.

[130] Håkan L. S. Younes and Reid G. Simmons. Probabilistic verification of discrete event systems using acceptance sampling. In *Proceedings of the 14th International Conference on Computer Aided Verification*, pages 223–235, London, UK, 2002. Springer-Verlag.

[131] Håkan L. S. Younes. Error control for probabilistic model checking. In E. Emerson and Kedar Namjoshi, editors, *Verification, Model Checking, and Abstract Interpretation*, volume 3855 of *Lecture Notes in Computer Science*, pages 142–156. Springer Berlin / Heidelberg, 2006.

[132] Koushik Sen, Mahesh Viswanathan, and Gul Agha. Statistical model checking of black-box probabilistic systems. In *In 16th conference on Computer Aided Verification (CAVŠ04), volume 3114 of LNCS*, pages 202–215. Springer, 2004.

[133] Håkan L. S. Younes. Probabilistic verification for ”black-box” systems. In Kousha Etessami and Sriram K. Rajamani, editors, *Computer Aided Verification*, volume 3576 of *Lecture Notes in Computer Science*, pages 253–265. Springer Berlin / Heidelberg, 2005.

[134] Nathalie Chabrier-Rivier, Marc Chiaverini, Vincent Danos, FranÃğois Fages, and Vincent SchÃd'chter. Modeling and querying biomolecular interaction networks. *Theor. Comput. Sci.*, 325(1):25–44, 2004.

[135] Nathalie Chabrier-Rivier, François Fages, and Sylvain Soliman. The biochemical abstract machine BIOCHAM. In *Computational Methods in Systems Biology*, pages 172–191. 2005.

[136] L. Calzone, F. Fages, and S. Soliman. Biocham: an environment for modeling biological systems and formalizing experimental knowledge. *Bioinformatics*, 22(14):1805–1807, 2006.

[137] A. Cimatti, E. Clarke, F. Giunchiglia, and M. Roveri. NUSMV: a new symbolic model checker. *International Journal on Software Tools for Technology Transfer*, 2:2000, 2000.

[138] Henny B. Sipma, Tomas E. Uribe, and Zohar Manna. Deductive model checking. *Form. Methods Syst. Des.*, 15(1):49–74, July 1999.

[139] E. De Maria, F. Fages, and S. Soliman. On coupling models using model-checking: Effects of irinotecan injections on the mammalian cell cycle. In *Computational Methods in Systems Biology*, pages 142–157. Springer, 2009.

[140] Marco Antoniotti, Alberto Policriti, Nadia Ugel, and Bud Mishra. Model building and model checking for biochemical processes. *Cell Biochemistry and Biophysics*, 38(3):271–286, 2003.

[141] Pedro T. Monteiro, Delphine Ropers, Radu Mateescu, Ana T. Freitas, and Hidde de Jong. Temporal logic patterns for querying qualitative models of genetic regulatory networks. In *Proceeding of the 2008 conference on ECAI 2008: 18th European Conference on Artificial Intelligence*, pages 229–233. IOS Press, 2008.

[142] R. Alur, T. Henzinger, F. Mang, S. Qadeer, S. Rajamani, and S. Tasiran. Mocha: Modularity in model checking. In *Computer Aided Verification*, pages 521–525. Springer, 1998.

[143] G. Bernot, J.P. Comet, A. Richard, and J. Guespin. Application of formal methods to biological regulatory networks: extending Thomas asynchronous logical approach with temporal logic. *Journal of theoretical biology*, 229(3):339–347, 2004.

[144] J. Barnat, L. Brim, I. Černá, S. Dražan, and D. Šafránek. Parallel model checking large-scale genetic regulatory networks with divine. *Electronic Notes in Theoretical Computer Science*, 194(3):35–50, 2008.

[145] C. Li, M. Nagasaki, K. Ueno, and S. Miyano. Simulation-based model checking approach to cell fate specification during caenorhabditis elegans vulval development by hybrid functional petri net with extension. *BMC systems biology*, 3(1):42, 2009.

[146] J. Barnat, L. Brim, D. Safranek, and M. Vejnar. Parameter scanning by parallel model checking with applications in systems biology. In *Parallel and Distributed Methods in Verification, 2010 Ninth International Workshop on, and High Performance Computational Systems Biology, Second International Workshop on*, pages 95–104. IEEE, 2010.

[147] Gregory Batt, Michel Page, Irene Cantone, Gregor Goessler, Pedro Monteiro, and Hidde de Jong. Efficient parameter search for qualitative models of regulatory networks using symbolic model checking. *Bioinformatics*, 26(18):i603 –i610, 2010.

[148] Robin Donaldson and David Gilbert. A model checking approach to the parameter estimation of biochemical pathways. In *Computational Methods in Systems Biology*, pages 269–287. 2008.

[149] Håkan L. S Younes. *Verification and Planning for Stochastic Processes with Asynchronous Events*. PhD thesis, Carnegie Mellon University, 2005.

[150] Edmund M Clarke, Christopher James Langmead, Axel Legay, Andre Platzer, and Paolo Zuliani. Statistical model checking for complex stochastic models in systems biology. 2009.

[151] Haijun Gong, Paolo Zuliani amd Anvesh Komuravelli, James R Faede, and Edmund M Clarke. Analysis and verification of the HMGB1 signaling pathway. *BMC Bioinform.*, 11(Suppl 7)(S10):1–13, 2010.

[152] Paolo Ballarini, Michele Forlin, Tommaso Mazza, and Davide Prandi. Efficient parallel statistical model checking of biochemical networks. *arXiv:0912.2551*, dec 2009. EPTCS 14, 2009, pp. 47-61.

[153] Chuan Hock Koh, Masao Nagasaki, Ayumu Saito, Chen Li, Limsoon Wong, and Satoru Miyano. MIRACH: Efficient model checker for quantitative biological pathway models. *Bioinformatics*, 27(5):734 –735, 2011.

[154] P. Ballarini, T. Mazza, A. Palmisano, and A. Csikasz-Nagy. Studying irreversible transitions in a model of cell cycle regulation. *Electronic Notes in Theoretical Computer Science*, 232:39–53, 2009.

[155] C. J Langmead, S. Jha, and E. M Clarke. Temporal-logics as query languages for dynamic bayesian networks: Application to d. melanogaster embryo development. *Technical Report*, 2006.

[156] Christopher James Langmead. Generalized queries and bayesian statistical model checking in dynamic bayesian networks: Application to personalized medicine. In *Proc. 8th Ann. Intnl Conf. on Comput. Sys. Bioinf. (CSB*, pages 201–212, 2009.

[157] D. Beauquier, A. Rabinovich, and A. Slissenko. A logic of probability with decidable model-checking. In *Computer Science Logic*, pages 371–402. Springer, 2002.

[158] V.A. Korthikanti, M. Viswanathan, G. Agha, and Y.M. Kwon. Reasoning about mdps as transformers of probability distributions. In *Quantitative Evaluation of Systems (QEST), 2010 Seventh International Conference on the*, pages 199–208. IEEE, 2010.

[159] Albert Goldbeter and Olivier Pourquieb. Modeling the segmentation clock as a network of coupled oscillations in the notch, wnt and fgf signaling pathways. *Journal of Theoretical Biology*, 252:574–585, 2008.

[160] Akio Maedo, Yuichi Ozaki, Sudhir Sivakumaran, Tetsuro Akiyama, Hidetoshi Urakubo, Ayako Usami, Miharu Sato, Kozo Kaibuchi, and Shinya Kuroda. Ca^{2+}-

independent phospholipase A2-dependent sustained Rho-kinase activation exhibits all-or-none response. *Genes Cells*, 11:1071–1083, 2006.

[161] Thomas Herault, Richard Lassaigne, Frederic Magniette, and Sylvain Peyronnet. Approximate probabilistic model checking. In Bernhard Steffen and Giorgio Levi, editors, *Verification, Model Checking, and Abstract Interpretation*, volume 2937 of *Lecture Notes in Computer Science*, pages 307–329. Springer Berlin / Heidelberg, 2003.

[162] Håkan L. S Younes and Reid G Simmons. Statistical probabilistic model checking with a focus on time-bounded properties. *Information and Computation*, 204:1368–1409, 2006.

[163] D.E. Goldberg. *Genetic algorithms in search, optimization, and machine learning*. Addison-Wesley, 1989.

[164] A.C. Hindmarsh, P.N. Brown, K.E. Grant, S.L. Lee, R. Serban, D.E. Shumaker, and C.S. Woodward. Sundials: Suite of nonlinear and differential/algebraic equation solvers. *ACM Transactions on Mathematical Software (TOMS)*, 31(3):363–396, 2005.

[165] J. Vanlier, CA Tiemann, PAJ Hilbers, and NAW van Riel. An integrated strategy for prediction uncertainty analysis. *Bioinformatics*, 28(8):1130–1135, 2012.

[166] N. van Riel. Speeding up simulations of ode models in matlab using cvode and mex files. 2012.

[167] M.B. Elowitz and S. Leibler. A synthetic oscillatory network of transcriptional regulators. *Nature*, 403(6767):335–338, 2000.

[168] Osamu Takeuchi and Shizuo Akira. Pattern recognition receptors and inflammation. *Cell*, 140(6):805–820, mar 2010.

[169] Taro Kawai and Shizuo Akira. The role of pattern-recognition receptors in innate immunity: update on toll-like receptors. *Nature immunology*, 11(5):373–384, may 2010. PMID: 20404851.

[170] Taro Kawai and Shizuo Akira. Innate immune recognition of viral infection. *Nature Immunology*, 7(2):131–137, feb 2006.

[171] Shizuo Akira, Satoshi Uematsu, and Osamu Takeuchi. Pathogen recognition and innate immunity. *Cell*, 124(4):783–801, feb 2006. PMID: 16497588.

[172] Tobias Warger, Philipp Osterloh, Gerd Rechtsteiner, Melanie Fassbender, Valeska Heib, Beate Schmid, Edgar Schmitt, Hansjorg Schild, and Markus P Radsak. Synergistic activation of dendritic cells by combined toll-like receptor ligation induces superior CTL responses in vivo. *Blood*, 108(2):544–550, jul 2006. PMID: 16537810.

[173] Qing Zhu, Colt Egelston, Aravindhan Vivekanandhan, Satoshi Uematsu, Shizuo Akira, Dennis M. Klinman, Igor M. Belyakov, and Jay A. Berzofsky. Toll-like receptor ligands synergize through distinct dendritic cell pathways to induce t cell responses: Implications for vaccines. *Proceedings of the National Academy of Sciences*, 105(42):16260–16265, oct 2008.

[174] Giorgio Trinchieri and Alan Sher. Cooperation of toll-like receptor signals in innate immune defence. *Nature reviews. Immunology*, 7(3):179–190, mar 2007. PMID: 17318230.

[175] Andrea Oeckinghaus, Matthew S Hayden, and Sankar Ghosh. Crosstalk in NF-kB signaling pathways. *Nature Immunology*, 12(8):695–708, jul 2011.

[176] K. Honda and T. Taniguchi. Irfs: master regulators of signalling by toll-like receptors and cytosolic pattern-recognition receptors. *Nature Reviews Immunology*, 6(9):644–658, 2006.

[177] L.C. Platanias. Mechanisms of type-i-and type-ii-interferon-mediated signalling. *Nature Reviews Immunology*, 5(5):375–386, 2005.

[178] Stefan Hoops, Sven Sahle, Ralph Gauges, Christine Lee, Jurgen Pahle, Natalia Simus, Mudita Singhal, Liang Xu, Pedro Mendes, and Ursula Kummer. COPASI - a COmplex PAthway SImulator. *Bioinformatics*, 22(24):3067–3074, 2006.

[179] Tomasz Lipniacki, Pawel Paszek, A R Allan R Brasier, Bruce Luxon, and Marek Kimmel. Mathematical model of NF-kappaB regulatory module. *Journal of theoretical biology*, 228(2):195–215, may 2004. PMID: 15094015.

[180] S. Kuttykrishnan, J. Sabina, L.L. Langton, M. Johnston, and M.R. Brent. A quantitative model of glucose signaling in yeast reveals an incoherent feed forward loop leading to a specific, transient pulse of transcription. *Proceedings of the National Academy of Sciences*, 107(38):16743–16748, 2010.

[181] Kanae Oda and Hiroaki Kitano. A comprehensive map of the toll-like receptor signaling network. *Molecular systems biology*, 2:2006.0015, 2006. PMID: 16738560.

[182] Mohamed Helmy, Jin Gohda, Jun-ichiro Inoue, Masaru Tomita, Masa Tsuchiya, and Kumar Selvarajoo. Predicting novel features of Toll-Like receptor 3 signaling in macrophages. *PLoS ONE*, 4(3):e4661, mar 2009.

[183] K. Selvarajoo. Decoding the signaling mechanism of toll-like receptor 4 pathways in wild type and knockouts. *E-Cell System–Basic Concepts and Applications*, 2007.

[184] Markus W Covert, Thomas H Leung, Jahlionais E Gaston, and David Baltimore. Achieving stability of lipopolysaccharide-induced NF-kappaB activation. *Science (New York, N.Y.)*, 309(5742):1854–1857, sep 2005. PMID: 16166516.

[185] Jayalakshmi Krishnan, Kumar Selvarajoo, Masa Tsuchiya, Gwang Lee, and Sang-dun Choi. Toll-like receptor signal transduction. *Experimental and molecular medicine*, 39(4):421–438.

[186] Alexander Hoffmann, Andre Levchenko, Martin L. Scott, and David Baltimore. The IkB-NF-kB signaling module: Temporal control and selective gene activation. *Science*, 298(5596):1241–1245, nov 2002.

[187] Taro Kawai and Shizuo Akira. Signaling to NF-kappaB by toll-like receptors. *Trends in molecular medicine*, 13(11):460–469, nov 2007. PMID: 18029230.

[188] Geoffrey Koh and Dong-Yup Lee. Mathematical modeling and sensitivity analysis of the integrated TNFα -mediated apoptotic pathway for identifying key regulators. *Comput. Biol. Med.*, 41(7):512–528, jul 2011.

[189] D.J. Wilkinson. *Stochastic modelling for systems biology.* CRC Press, 2011.

[190] M. Silberstein, A. Schuster, D. Geiger, A. Patney, and J.D. Owens. Efficient computation of sum-products on gpus through software-managed cache. In *Proceedings*

of the 22nd annual international conference on Supercomputing, pages 309–318. ACM, 2008.

[191] Bing Liu, Andrei Hagiescu, Sucheendra K. Palaniappan, Bipasa Chattopadhyay, Zheng Cui, Weng-Fai Wong, and P. S. Thiagarajan. Approximate probabilistic analysis of biopathway dynamics. *Bioinformatics*, 28(11):1508–1516, jun 2012.

[192] J. Barnat, L. Brim, M. Ceska, and T. Lamr. Cuda accelerated ltl model checking. In *Parallel and Distributed Systems (ICPADS), 2009 15th International Conference on*, pages 34–41. IEEE, 2009.

[193] R.V. Culshaw and S. Ruan. A delay-differential equation model of HIV infection of CD4+ T-cells. *Mathematical Biosciences*, 165(1):27–39, 2000.

[194] Chuan H. Koh, Sucheendra K. Palaniappan, P. S. Thiagarajan, and Limsoon Wong. Improved statistical model checking methods for pathway analysis. *BMC Bioinformatics*, 13(Suppl 17):S15, dec 2012.