



Synthesis and infeasibility analysis for stochastic models of biochemical systems using statistical model checking and abstraction refinement

Sumit Kumar Jha^{a,*}, Christopher James Langmead^{b,*}

^a Computer Science Department, University of Central Florida, Orlando, FL 32816, United States

^b Computer Science Department and Lane Center for Computational Biology, Carnegie Mellon University, Pittsburgh, PA 15213, United States

ARTICLE INFO

Keywords:

Systems biology
Stochastic systems
Parameter synthesis
Model infeasibility
Statistical model checking

ABSTRACT

The stochastic dynamics of biochemical reaction networks can be modeled using a number of succinct formalisms all of whose semantics are expressed as Continuous Time Markov Chains (CTMC). While some kinetic parameters for such models can be measured experimentally, most are estimated by either fitting to experimental data or by performing *ad hoc*, and often manual search procedures. We consider an alternative strategy to the problem, and introduce algorithms for automatically synthesizing the set of *all* kinetic parameters such that the model satisfies a given high-level behavioral specification. Our algorithms, which integrate statistical model checking and abstraction refinement, can also report the *infeasibility* of the model if no such combination of parameters exists. Behavioral specifications can be given in any finitely monitorable logic for stochastic systems, including the probabilistic and bounded fragments of linear and metric temporal logics. The correctness of our algorithms is established using a novel combination of arguments based on survey sampling and uniform continuity. We prove that the probability of a measurable set of paths is uniformly and jointly continuous with respect to the kinetic parameters. Under a suitable technical condition, we also show that the unbiased statistical estimator for the probability of a measurable set of paths is monotonic in the parameter space. We apply our algorithms to two benchmark models of biochemical signaling, and demonstrate that they can efficiently find parameter regimes satisfying a given high-level behavioral specification. In particular, we show that our algorithms can synthesize up to 6 parameters, simultaneously, which is more than that reported by any other synthesis algorithm for stochastic systems. Moreover, when parameter estimation is desired, as opposed to synthesis, we show that our approach can scale to even higher dimensional spaces, by identifying the single parameter combination that maximizes the probability of the behavior being true in an 11-dimensional system.

© 2011 Elsevier B.V. All rights reserved.

1. Introduction

The stochastic dynamics of biological systems are governed by complex networks of biochemical reactions. Computational modeling of these reaction networks plays a critical role in the study and engineering of biological systems, where it is used to: provide insights into complex, often counter-intuitive phenomena; evaluate the consequences of assumptions and design choices; generate experimentally-verifiable predictions; and to characterize the system's sensitivity to perturbations. One of the most challenging tasks in modeling is identifying parameters. This paper introduces new

* Corresponding authors. Tel.: +1 412 268 7571; fax: +1 412 268 5576.

E-mail addresses: sumit.jha.cmu@gmail.com (S.K. Jha), cjl@cs.cmu.edu (C.J. Langmead).

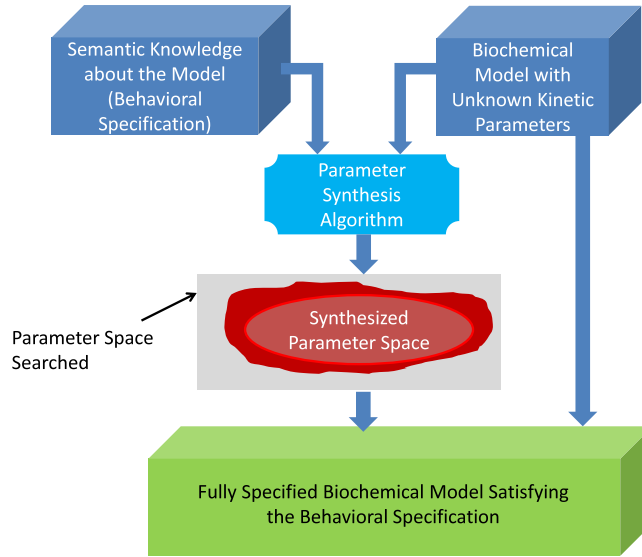
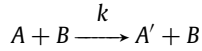


Fig. 1. Parameter Synthesis Problem: Given a biochemical model with unknown kinetic parameters and a behavioral specification about the model, synthesize the kinetic parameters of the biochemical model.

algorithms for synthesizing the parameters for *Continuous Time Markov Chain* (CTMC) models of biochemical reaction networks.

A biochemical reaction network consists of a finite number of distinct molecular species that interact dynamically according to a prescribed set of reaction rules. Each rule describes the production, consumption, or transformation of a subset of species. For example, the rule



describes the transformation of species A into A' . The transformation is mediated by B , but B is otherwise unaffected by the process. The transformation occurs with a forward reaction rate k .

There are a number of formalisms that can be used to model a collection of reaction rules, including Ordinary Differential Equations (ODEs) and CTMCs. Of these, CTMCs are sometimes preferred because they precisely model the stochastic nature of biochemical interactions [25,26,49], whereas ODEs are deterministic. The state transitions in a CTMC model correspond to discrete changes in the number of copies of each species, due to the execution of a reaction rule. The rates of the stochastic state transitions are derived from the kinetic rate constants in the biochemical reactions.

One of the most difficult challenges that arises in the development of reaction network models is identifying rate parameters that are consistent with the empirical behavior of the biochemical system being modeled. Unfortunately, it is often impractical or impossible to measure *every* rate constant specified in a given model, especially those between short-lived, transient species. For this reason, parameters are often estimated by fitting the model to any available data or else by performing a time-consuming and typically *ad hoc* search through the parameter space.

In this paper, we consider an alternative approach to identifying rate parameters—*parameter synthesis* (Fig. 1). Unlike parameter estimation, which returns a single combination of rate constants, parameter synthesis returns a bounded volume in parameter space such that any combination of rate constants inside the volume is guaranteed to produce a model that is consistent with a given *high-level behavioral specification*. These behavioral specifications may refer to experimental observations or more generally to any property the modeler asserts should hold. Synthesis is clearly a more challenging problem than parameter estimation because there may be an uncountably infinite number of unique parameter combinations that must be characterized as either satisfying the specification, or not. Synthesizing parameters for CTMCs is especially challenging because of the stochastic nature of the model.

We introduce three algorithms for studying the parameter space of stochastic biochemical models. The first two algorithms solve the synthesis problem for CTMCs using a combination of statistical model checking and abstraction refinement. Specifically, given: (i) a CTMC, \mathcal{M} , with unknown parameters, (ii) a property specification, ϕ , and (iii) a probability threshold, ρ , our algorithms obtain a bounded region in parameter space, $\tilde{\mathcal{V}}_{\phi,\rho}$, such that for any choice of parameters $\theta \in \tilde{\mathcal{V}}_{\phi,\rho}$ the resulting model, \mathcal{M}_θ , will satisfy ϕ with probability at least ρ . That is, $\mathcal{M}_\theta \models P_{\geq \rho}(\phi)$. Additionally, for any choice of parameters such that $\theta \notin \tilde{\mathcal{V}}_{\phi,\rho}$, $\mathcal{M}_\theta \not\models P_{\geq \eta}(\phi)$ where η is a parameter that controls the quality of the synthesis algorithm. The computed volume is an approximation of the true volume satisfying the specification, \mathcal{V} , but the difference between $\tilde{\mathcal{V}}$ and \mathcal{V} can be made arbitrarily small at the expense of additional computational resources (by making η close to one). The second algorithm is a modified version of the first, and uses monotonicity and abstraction

refinement to accelerate the synthesis. The third algorithm performs a parameter *search* and finds an approximation to the single parameter combination that maximizes the probability that the property holds. All three algorithms can also decide whether a given property is *infeasible*. That is, whether no choice of parameters can satisfy the specification with probability at least ρ .

The behaviors we consider are those that can be specified as formulas whose truth value can be decided (with a given statistical confidence) using a finite number of finite-length trajectories sampled from the model. In particular, our algorithms can perform parameter synthesis on biochemical stochastic models for properties specified in *Probabilistic Adapted Finitely Monitorable Logic* (see Section 4). The specific contributions of this paper are as follows:

- We present an automated technique for the calibration of stochastic biochemical models against high-level behavioral specifications of the underlying biochemical system. The proofs underlying our algorithm use a new statistical model checking algorithm based on survey sampling. To the best of our knowledge, we are the first to use survey sampling based statistical model checking as a proof mechanism for demonstrating interesting properties of stochastic models. We believe that this is an important contribution of the paper.
- We show that the logarithm of the probability of an adapted finitely monitorable formula being true on a model is *uniformly* and *jointly* continuous with respect to the kinetic parameters in the stochastic biochemical model. The correctness results of our algorithms use this property.
- We apply our algorithm to synthesize up to 6 parameters simultaneously for a model of Fibroblast Growth Factor signaling from the literature [30,31]. We note that this is the first time that the synthesis of as many as six parameters for stochastic models against probabilistic temporal logic properties has been reported in the literature. We also demonstrate our ability to show infeasibility of probabilistic adapted finitely monitorable specifications on a biochemical model in a given parameter range. Finally, when parameter estimation is desired, as opposed to synthesis, we show that our approach can scale to even higher dimensional spaces, by identifying the single parameter combination that maximizes the probability of the formula being true in an 11-dimensional system.

The organization of this paper is as follows: We begin with a review of parameter synthesis techniques, contrasting our methods with those from the literature. Sections 3–5 introduce key concepts and terminology used by our algorithms, including CTMCs, property specifications, statistical model checking, and survey sampling. We discuss our theoretical results in Section 6. Our algorithms are presented in 7, and then applied to benchmark models in Section 8. We summarize our results and discuss directions for future work in Section 9.

2. Related work

The algorithms presented in this paper perform parameter synthesis for stochastic models of biochemical reactions using a given high-level behavioral specification. Our approach uses a novel combination of techniques drawn from several domains, including: statistical model checking, abstraction, sensitivity analysis, and sampling theory. In this section, we summarize the relationship of our work to existing methods.

Parameter synthesis is closely related to the problem of parametric verification, and both can be posed as reachability problems. A variety of algorithms have been developed to address these problems using both symbolic (e.g., [3,1,55]) and numerical methods (e.g., [4,19,45,54]) for non-stochastic finite-state, continuous, and hybrid systems. Such techniques differ from those presented in this paper as they cannot be applied to stochastic models, and are restricted to safety properties.

Our methods perform parameter synthesis using a combination of statistical model checking and abstraction refinement for parameter synthesis. A similar combination of techniques was first proposed in [23]. That method relies on an abstraction-refinement approach [37] for model checking of linear hybrid systems which, unfortunately, cannot be easily adapted to stochastic systems.

The literature for stochastic systems is primarily focused on sensitivity analysis (e.g., [28]) or computing a point estimate for the parameters by fitting to observational data. Bayesian approaches to parameter estimation (e.g., [38–40,58]) can be interpreted as a form of parameter synthesis as they compute a probability distribution over parameters. However, unlike our method, none of these methods admit the use of high-level behavioral specifications.

There are a number of existing algorithms that have been developed to perform parameter synthesis and related tasks for biological processes, but these are either restricted to non-stochastic models (e.g., [7,9,11,15,17,18,20,24,50]), or do not use high-level behavioral specifications (e.g., [48]). Approximate parameter synthesis against temporal specifications has also been studied for the general class of CTMCs by using discretized parameter values and uniformization techniques [29]. However, they are mainly interested in probabilistic time-bounded reachability properties, i.e., given a goal state, does the probability of reaching the goal state within a fixed finite time period lie in a given interval? Our algorithms are meant to analyze more complex and interesting properties like those specified in Probabilistic Bounded Linear Temporal Logic. We also note that no approximation guarantees are provided by the discretization scheme used in the paper, while we present theoretical results to exactly compute the approximation error introduced by our algorithms.

Donaldson and Gilbert [16] have suggested a parameter estimation technique that uses model checking against Probabilistic Linear Temporal Logic to perform parameter estimation for both continuous and stochastic models of biological systems. Their approach uses the notion of a distance metric between the behavior of the model and that expected by the specification, and generates point estimates of the parameters. They combine a model checker with a genetic algorithm

that minimizes the distance between its actual behavior and the desired behavior. They use a simulation-based Monte Carlo Model Checker for Probabilistic Linear Temporal Logic with numerical constraints. Being driven by a genetic algorithm, the aforesaid approach is neutral to the stochastic model under consideration. However, if a model can not satisfy a set of behavioral specifications for any parameter value, the genetic algorithm based approach can not conclusively state that there is no parameter value for which the model will demonstrate the expected behavior. This paper focuses on parameter synthesis for CTMC models of biochemical systems, and is able to provide strong approximation guarantees on the results produced by the algorithm.

We note that the notion of robustness in metric temporal logics and the use of the robustness metrics in guiding a suitable search algorithm over the parameter space for nonlinear continuous models were first introduced by [50]. They define a continuous degree of satisfaction of a temporal logic formula with constraints and use this satisfaction measure as a fitness function to explore the kinetic parameter space with state-of-the-art search algorithms. Further, they also identify the problem that point estimates are themselves not adequate and characterize the robustness of the synthesized parameters. While they evaluate their technique on models of the cell cycle and of the MAPK signalling cascade, it could be applied to any executable model including stochastic and continuous models.

In this paper, we extend some of these concepts to parameter synthesis for CTMC models of biochemical systems. Instead of identifying a single parameter value, we seek to identify all possible parameter values that can satisfy a given behavioral specification. This is particularly important if one wants to build a library of reusable biochemical models (similar to IP design cores used by hardware engineers). We will also provide approximation guarantees for the performance of our algorithms and bound the probability that our algorithm produces an incorrect answer.

3. Stochastic models of biochemical systems

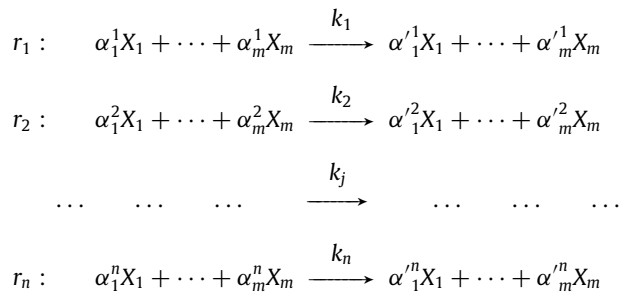
The stochastic process underlying a dynamic biochemical system can be modeled with a Continuous Time Markov Chain [25,26,49]. The key arguments behind the underlying theoretical derivation are that the state of a chemical system changes only through the completion of a chemical reaction, and that the probability of a chemical reaction occurring during a given interval of time is proportional to the probability that the reactants come together with enough energy to overcome the reaction's activation barrier. We now formally define the model on which our algorithms operate.

Definition 1 (*Labeled Continuous Time Markov Chain*). A labeled CTMC is a three tuple (S, R, L) , where

- S is a finite set of states.
- $L : S \rightarrow 2^{AP}$ is a labeling function that labels each state $s \in S$ with a set of atomic propositions from a finite set of atomic propositions AP .
- $R : S \times S \rightarrow \mathbb{R}^+ \cup \{0\}$ is a rate transition matrix. $R(s, s')$ denotes the rate of transition from state s to state s' .

Definition 2 (*Labeled CTMC Path*). A path σ in a labeled Continuous Time Markov Chain is a timed sequence of states $s_0 \xrightarrow{\Delta_0} s_1 \xrightarrow{\Delta_1} s_2 \cdots \xrightarrow{\Delta_{i-1}} s_i$, where Δ_i is the amount of time spent in state s_i .

Our algorithms can be applied to CTMCs of a particular form. Namely, those modeling the dynamics of biochemical reactions. We consider a biochemical reaction system with n biochemical reactions r_1, r_2, \dots, r_n among m biochemical species X_1, X_2, \dots, X_m :



The stoichiometric constants $\alpha_1^1, \dots, \alpha_m^1, \alpha_1'^1, \dots, \alpha_m'^1$ are non-negative integers, and $k_i \in \mathbb{R}^+$ ($1 \leq i \leq n$) are reaction rates. The labeled Continuous Time Markov Chain corresponding to the biochemical system is the three tuple (S, R, L) where:

- $S = \{0, 1, 2, \dots, N\} \times \{0, 1, 2, \dots, N\} \times \cdots \times \{0, 1, 2, \dots, N\}$ (m times). Here, N denotes the maximum number of copies of any biochemical species as the system evolves with time.
- $L(s) = s \equiv (x_1(s), \dots, x_m(s))$ where $x_j(s) \in [0, N]$ denotes the number of entities of biochemical species X_j .
- $R(s, s') = \begin{cases} k_j x_1(s)^{\alpha_1^j} \cdots x_m(s)^{\alpha_m^j} & \text{if } s' \text{ results from executing } r_j \text{ in state } s \\ 0 & \text{otherwise} \end{cases}$

The probability density of a transition from state s to state s' after spending time Δ in s , denoted by $P(s, s', \Delta)$, is $k_j x_1(s)^{\alpha_1^j} \dots x_m(s)^{\alpha_m^j} (e^{-\sum_{i=1}^n \mathbb{I}(r_i) k_i x_1(s)^{\alpha_1^i} \dots x_m(s)^{\alpha_m^i} \Delta_i})$. The indicator function $\mathbb{I}(r_i)$ is 1 if and only if it is possible to execute the reaction r_i in the state s ; otherwise, the indicator function $\mathbb{I}(r_i)$ is zero. The form of the probability density function assumes that the dynamics of the biochemical system is governed by mass-action laws.

Each state in this discrete state-space model for biochemical systems is labeled with the number of copies of various biochemical species in the biochemical system. No two distinct states have exactly the same number of copies of various biochemical species. The rate of transition from one state s to another state s' is proportional to the number of copies of each biochemical species raised to the stoichiometric coefficient for that species in the reaction r_j that takes the system from state s to state s' . The constant of proportionality is given by the rate constant k_j .

Our model has several notable restrictions. We permit only finite state space models that can be represented as CTMCs. We do not consider stochastic differential equations or models that explicitly model space (e.g., Agent Based Models). The semantics of our CTMC models corresponds to mass action kinetics of biochemical reactions. Further, we note that there is at most one biochemical reaction that can take the system from a given state to another given state in a single transition. Thus, our model does not include parallel chemical reactions.

4. Probabilistic adapted finitely monitorable specifications

In this section, we formally define the notion of *high-level behavioral specifications* that we can use to express the observed expected behavior of biochemical systems. A specification is said to be *adapted* to a stochastic process if the truth of the specification can be determined by observing the stochastic process. We are interested in specifications whose truth value can be decided with certainty by observing a finite prefix of the stochastic process. We call the logical formulae that represent such properties *adapted finitely monitorable specifications*.

Due to the stochastic nature of CTMCs, we generally seek to synthesize parameters so that a desired behavior holds with probability at least ρ . We note that our algorithms can be easily extended to handle intervals of probabilities (i.e., $[\rho_1, \rho_2]$). We call such probabilistic properties *probabilistic adapted finitely monitorable* (PAFM) specifications. We survey two different kinds of PAFM specifications:

- Probabilistic Bounded Linear Temporal Logic
- Probabilistic Bounded Metric Temporal Logic

4.1. Probabilistic bounded temporal logic

A special subclass of PAFM specifications on a stochastic model \mathcal{M} can be expressed as formulas in *Probabilistic Bounded Linear Temporal Logic* (PBLTL). The logic PBLTL is similar to Continuous Stochastic Logic (CSL) [61] but does not permit nested probability operators. We first define the syntax and semantics of *Bounded Linear Temporal Logic* (BLTL) [47,46,22].

For a stochastic model \mathcal{M} , let the set of state variables SV be a finite set of discrete-valued variables. A Boolean predicate over SV is a constraint of the form $x \sim v$, where $x \in SV$, $\sim \in \{\geq, \leq, =\}$, and $v \in \mathbb{R}$. A BLTL property is built on a finite set of Boolean predicates over SV using Boolean connectives and temporal operators. The syntax of the logic is given by the following grammar:

$$\phi ::= x \sim v \mid (\phi_1 \vee \phi_2) \mid (\phi_1 \wedge \phi_2) \mid \neg \phi_1 \mid (\phi_1 \mathbf{U}^t \phi_2),$$

where $\sim \in \{\geq, \leq, =\}$, $x \in SV$, $v \in \mathbb{R}$, and $t \in \mathbb{Q}_{\geq 0}$. We can define additional temporal operators such as $\mathbf{F}^t \psi = \mathbf{True} \mathbf{U}^t \psi$, or $\mathbf{G}^t \psi = \neg \mathbf{F}^t \neg \psi$ in terms of the bounded until \mathbf{U}^t .

We define the semantics of BLTL with respect to the paths of \mathcal{M} . The fact that a path σ satisfies property ϕ is denoted by $\sigma \models \phi$. Let $\sigma = (s_0, \Delta_0), (s_1, \Delta_1), \dots$ be an execution of the model along states s_0, s_1, \dots with durations $\Delta_0, \Delta_1, \dots \in \mathbb{R}$. We denote the path starting at state i by σ^i (in particular, σ^0 denotes the original execution σ). The value of the state variable x in σ at the state i is denoted by $V(\sigma, i, x)$. The semantics of BLTL is defined as follows:

- $\sigma^k \models x \sim v$ if and only if $V(\sigma, k, x) \sim v$;
- $\sigma^k \models \phi_1 \vee \phi_2$ if and only if $\sigma^k \models \phi_1$ or $\sigma^k \models \phi_2$;
- $\sigma^k \models \phi_1 \wedge \phi_2$ if and only if $\sigma^k \models \phi_1$ and $\sigma^k \models \phi_2$;
- $\sigma^k \models \neg \phi_1$ if and only if $\sigma^k \models \phi_1$ does not hold (written $\sigma^k \not\models \phi_1$);
- $\sigma^k \models \phi_1 \mathbf{U}^t \phi_2$ if and only if there exists $i \in \mathbb{N}$ such that: (a) $0 \leq \sum_{0 \leq l < i} \Delta_{k+l} \leq t$; (b) $\sigma^{k+i} \models \phi_2$; and (c) for each $0 \leq j < i$, $\sigma^{k+j} \models \phi_1$.

It is known that finite paths of bounded duration are always sufficient for Model Checking BLTL on traces. We now define Probabilistic Bounded Linear Temporal Logic.

Definition 3. A Probabilistic Bounded LTL (PBLTL) formula is a formula of the form $P_{\geq \rho}(\phi)$, where ϕ is a BLTL formula and $\rho \in [0, 1]$.

We say that \mathcal{M} satisfies PBLTL property $P_{\geq \rho}(\phi)$, denoted by $\mathcal{M} \models P_{\geq \rho}(\phi)$, if and only if the probability that a path of \mathcal{M} that satisfies the BLTL property ϕ is greater than or equal to ρ . The problem is well-defined [64] since one can always assign a unique probability measure to the set of executions of \mathcal{M} that satisfy a formula in BLTL.

4.2. Probabilistic bounded metric temporal logic

Another widely used logic for monitoring continuous (and possibly stochastic) systems is the Metric Temporal Logic [41]. There exist efficient monitoring algorithms for Metric Temporal Logic [57,21]. The logic extended with a probability operator naturally defines yet another subset of PAFM specifications.

Metric Temporal Logic (MTL) can specify both lower and upper bounds on the time bounds associated with the temporal operators. The syntax of the MTL property is given by the following grammar:

$$\phi ::= x \sim v \mid (\phi_1 \vee \phi_2) \mid (\phi_1 \wedge \phi_2) \mid \neg \phi_1 \mid (\phi_1 \mathbf{U}^{[t,t']}\phi_2),$$

where $\sim \in \{\geq, \leq, =\}$, $x \in SV$, $v \in \mathbb{Q}$, and $t \in \mathbb{Q}_{\geq 0}$. We can also define additional temporal operators such as $\mathbf{F}^{[t,t']}\psi = \mathbf{True} \mathbf{U}^{[t,t']}\psi$, or $\mathbf{G}^{[t,t']}\psi = \neg \mathbf{F}^{[t,t']}\neg \psi$ in terms of the bounded until $\mathbf{U}^{[t,t']}$. The semantics of Bounded MTL for a trace σ^k starting at the k^{th} state ($k \in \mathbb{N}$) is similar to that of BLTL except for the temporal operator \mathbf{U} :

- $\sigma^k \models \phi_1 \mathbf{U}^{[t,t']}\phi_2$ if and only if there exists $i \in \mathbb{N}$ such that (a) $t \leq \sum_{0 \leq l < i} \Delta_{k+l} \leq t'$; (b) $\sigma^{k+i} \models \phi_2$; and (c) for each $0 \leq j < i$, $\sigma^{k+j} \models \phi_1$.

We note that Bounded Linear Temporal Logic (PBLTL) is a subset of Metric Temporal Logic obtained by setting $t = t'$. We can now define Probabilistic Bounded Metric Temporal Logic.

Definition 4. A Probabilistic Bounded Metric Temporal Logic formula is a formula of the form $P_{\geq \rho}(\phi)$, where ϕ is a Bounded Metric Temporal Logic formula and $\rho \in [0, 1]$.

We say that \mathcal{M} satisfies Probabilistic Bounded Metric Temporal Logic property $P_{\geq \rho}(\phi)$, denoted by $\mathcal{M} \models P_{\geq \rho}(\phi)$, if and only if the probability that an execution of \mathcal{M} satisfies Bounded Metric Temporal Logic property ϕ is greater than or equal to ρ .

5. Statistical model checking

The model checking problem is to algorithmically decide whether a given model, \mathcal{M} , satisfies a given property, ϕ . When the underlying model is stochastic, the *probabilistic model checking problem* is to determine whether the model satisfies the property with probability greater than or equal to a given threshold $\rho \in (0, 1)$, denoted by $\mathcal{M} \models P_{\geq \rho}(\phi)$. While symbolic, exact methods for solving the probabilistic model checking problem exist (e.g., [5,6,12,42]), these algorithms generally do not scale to very large state spaces, like CTMCs of biochemical reaction networks. An alternative approach to solving the probabilistic model checking problem is to use approximate methods which decide the truth of the formula statistically. Such algorithms are called *statistical model checking algorithms* (e.g., [63,64,51,52,35,43]). These algorithms are approximate in the sense that they may return an incorrect answer, but the probability of doing so can be bounded.

Our parameter synthesis algorithms use statistical model checking algorithms to solve key subproblems. More importantly, we will be using statistical approaches to develop our proofs. Instead of proving properties of an arbitrary (possibly infinite) set of paths, we will use the notion of *unbiased statistical estimators* to develop the correctness arguments of our results. In particular, we invoke *survey sampling* (See Section 5.3). Our survey sampling based approach to statistical model checking is the cornerstone of our proofs as it allows us to reason about sample paths from all the possible parameterizations of a given model, without worrying about changes in the probability measures of the individual paths.

We will show in Section 6 that survey sampling leads to a powerful new proof technique for stochastic systems where one analytically lifts the proof from one stochastic system to another. The two stochastic systems are identical, except for a change of probability measure on the paths. We believe this new proof technique is perhaps one of the most interesting contributions of this paper.

In the rest of the paper, we are only interested in knowing whether a stochastic model satisfies a probabilistic adapted finitely monitorable logic formula with a given confidence. That is, deciding whether $\mathcal{M} \models_q P_{\geq \rho}(\phi)$ or $\mathcal{M} \models_q P_{< \rho}(\phi) \equiv \mathcal{M} \not\models_q P_{\geq \rho}(\phi)$ is true. Here, q is the *confidence probability*; it represents our confidence that the answer of the algorithm is correct. Statistical model checking algorithms generally terminate when q is high enough to satisfy user-specified bounds on making Type I and Type II errors. A Type I error occurs when the algorithm decides that the formula is true, when in fact it is false, and a Type II error occurs when the algorithm decides that the formula is false, when in fact it is true.

There are a variety of statistical model checking algorithms from which to choose, and our parameter synthesis algorithm can be easily updated to use any of them. In what follows, we briefly summarize existing statistical model checking algorithms.

5.1. Frequentist statistical model checking algorithms

Frequentist statistical model checking techniques perform *sequential sampling* and use a stopping criterion based on frequentist (a.k.a classical) statistics. That is, they iteratively sample paths from the model until the evidence gathered is sufficient for some frequentist statistical test to make a decision with the desired confidence. These algorithms can be

divided into those that compute an estimate for the true value of ρ , and those that perform hypothesis testing (i.e., selecting between the hypothesis that the model satisfies the formula, versus the hypothesis that it does not.). We note that the cost of generating each sample path can be very time-consuming in some domains (e.g., [13,53,56]), including modeling biochemical systems. Hence, it is important to sample as few traces from the model as possible.

SPRT based statistical model checking. Younes and Simmons introduced the first algorithms for statistical model checking [61,63,62]. Their work is based on statistical hypothesis testing and uses Wald's *Sequential Probability Ratio Test* (SPRT) [60]. The SPRT decides between the null hypothesis $H'_0 : \mathcal{M} \models P_{=\rho_0}(\phi)$ against the alternate hypothesis $H'_1 : \mathcal{M} \models P_{=\rho_1}(\phi)$. Note that these hypotheses are defined in terms of two distinct probability values, ρ_0 and ρ_1 . Such hypotheses are called *simple*. It can be shown that the SPRT is optimal for *simple hypothesis testing*, in the sense that it minimizes the expected number of samples among all the tests satisfying the same Type I and II errors [59]. Notice that the probabilistic model checking problem is actually a choice between two *composite* hypotheses $H_0 : \mathcal{M} \models P_{\geq \rho}(\phi)$ versus $H_1 : \mathcal{M} \models P_{< \rho}(\phi)$. The SPRT is not optimal for composite hypotheses.

Chernoff bound based statistical estimation. Hérault et al. [32] have used the Chernoff bound on the sum of independent random variables to derive a fixed sample size estimator for the true value of ρ . This approach is based on statistical estimation and hence, it may often need a larger number of samples than the approaches based on hypothesis testing. On the other hand, it can estimate the true value with which a model satisfies a given property and solves a much harder problem than hypothesis testing.

P-value based statistical model checking. Sen et al. [51,52] used the *p-value* for the null hypothesis as a statistic for hypothesis testing. The *p-value* is defined as the probability of obtaining observations at least as extreme as the one that was actually seen, given that the null hypothesis is true. Their tool VeStA has two components: the learning-CTMC module and the model-checking module. The first module implements the algorithm to learn a CTMC model [51]. This module correctly identifies the CTMC model in the limit when it is given samples drawn from a distribution generated by a CTMC. The second module is a model-checker that can perform statistical verification of deployed black-box probabilistic systems. Given a set of executions obtained by Monte Carlo simulation and a property, the algorithm uses *p-value* based statistical hypothesis testing to test whether the samples provide evidence to conclude the satisfaction or violation of a CSL property.

Monte Carlo based statistical model checking. Grosu and Smolka have also suggested a Monte Carlo based approach for verifying formulas in LTL [27]. Their algorithm uses a fixed-size sampling strategy that randomly samples lassos from a Büchi automaton in an on-the-fly fashion. The algorithm terminates if it finds a counterexample. Otherwise, the algorithm provides statistical guarantees on the possible presence of a counterexample in the model. This technique is specially interesting for software and hardware verification as it is applicable to non-stochastic systems too.

5.2. Bayesian statistical model checking

Bayesian statistics are an alternative to frequentist statistics. The key distinction between these two approaches is that Bayesian statistics requires that a *prior distribution* over each variable and parameter be specified, and the resulting tests average over these distributions in order to produce a *posterior distribution*. Averaging over these distributions, whether by sampling or through integration, is generally more expensive, computationally, and so Bayesian methods are generally more costly than those based on frequentist statistics. On the other hand, Bayesian methods have the advantage of smaller expected sample sizes [36,35]. In this section, we summarize two recent algorithms for statistical model checking using Bayesian statistics. The first algorithm uses hypothesis testing, and the second one uses estimation. Details for these algorithms can be found in [36,35,43], respectively.

Hypothesis testing based Bayesian statistical model checking

This approach essentially builds upon the work on using the SPRT for statistical Model Checking. Let H_0 denote the hypothesis that the model satisfies the formula (and H_1 denote the contrary). From Bayesian statistics, we know that the ratio of the posterior odds for hypotheses H_0 and H_1 given a set of observed traces d is:

$$\frac{P(H_0|d)}{P(H_1|d)} = \frac{P(d|H_0)}{P(d|H_1)} \frac{P(H_0)}{P(H_1)}. \quad (1)$$

Also, the Bayes factor \mathcal{B} of sample d and hypotheses H_0 and H_1 is $\mathcal{B} = \frac{P(d|H_0)}{P(d|H_1)}$. The Bayes Factor can be used as a measure of relative confidence in H_0 vs. H_1 . Jeffreys [33] interprets a high value of the Bayes factor as a measure of the evidence in favor of H_0 . We have developed a sequential version of Jeffreys' Bayes Factor test and studied its performance [35].

Estimation based Bayesian statistical model checking

An alternative to hypothesis testing based Bayesian statistical model checking was recently introduced [43]. This algorithm performs Bayesian estimation for the mean and variance of the Bernoulli distribution modeling the probability

that the formula is true. The parameters are estimated according to the following well-known formulas:

$$\hat{\rho} = \frac{k + \alpha}{\alpha + \beta + n}$$

$$\hat{v} = \frac{(\alpha + k)(n - k + \beta)}{(\alpha + n + \beta)^2(\alpha + n + \beta + 1)}$$

where $\hat{\rho}$ and \hat{v} are the estimated mean and variance of a Bernoulli distribution after seeing n sample trajectories, of which k satisfied the formula. The prior distribution over ρ is specified in terms of the Beta distribution, which is the conjugate prior of the Bernoulli distribution. α and β are the shape parameters of the Beta distribution.

Two important distinctions between Frequentist and Bayesian Model Checking approaches should be noted:

- Bayesian Statistical Model Checking allows us to specify the prior probability of a formula being true. It also permits an *objective* information-theoretic interpretation of the lack of any prior knowledge to be incorporated into the model checking algorithm.
- Models in Systems Biology often have parameters that are only specified up to a distribution. Frequentist approaches do not provide a natural framework to sample from these models.

5.3. Survey sampling based statistical model checking

The common feature of the model checking algorithms in Sections 5.1 and 5.2 is that they sample from the set of unique trajectories defined by a stochastic model with known parameters (or prior distributions over parameters, for the Bayesian methods). These samples are then used to either select between two hypotheses, or estimate the probability that the specification is true. When the parameters (or priors over parameters) of the model are given, then the appropriate sampling strategy is to generate sample trajectories according to the underlying probability distribution implied by the parameters. That is, high-probability trajectories should be sampled with higher frequency than low-probability traces. This biased approach to sampling results in a faster procedure for solving the probabilistic model checking problem.

Our goal is different. We seek to synthesize parameters for a given model. Under these circumstances, *survey sampling* is the appropriate theoretical framework for generating independent and identically distributed (i.i.d.) trajectories from a parameterized family of models. That is, a set of independent samples from a fixed sampling distribution. In contrast to the sampling strategies employed in the previous sections, survey sampling draws samples *uniformly from all unique traces*. That is, the samples are generated without respect to any particular choice of kinetic parameters. The fact that the samples are not affected by the probability space on the paths is crucial to the construction of our proofs and correctness arguments. We note that survey sampling is primarily a mechanism to construct the proofs; the actual computation during the synthesis may use any of the aforementioned statistical model checking algorithms for the sake of efficiency.

Suppose that n samples have been uniformly drawn from the unique traces of the model. Then, for any choice of parameters, θ , we can (retroactively) label each trace σ_i of the model $\mathcal{M}(\theta)$ with a probability value $P_\theta(\sigma_i)$. Each trace is also labeled with an indicator variable $\mathbb{I}(\sigma_i)$ such that the indicator is 1 if the trace satisfies the given adapted finitely monitorable property, and 0 otherwise. Given these, we define the random variable $X_i \equiv P_\theta(\sigma_i)\mathbb{I}(\sigma_i)$. Thus, the sample mean or expected value of X_i (taken over the n samples) is an unbiased estimator¹ of the probability that the model will satisfy the adapted finitely monitorable formula.

$$\bar{X} = E(X) = \frac{1}{n} \sum_{i=1}^n X_i$$

The variance of the sample mean, $\text{Var}(\bar{X})$, can be computed as follows:

$$\text{Var}(\bar{X}) = \frac{1}{n^2} \sum_{i=1}^n \sum_{j=1}^n \text{Cov}(X_i, X_j) = \frac{1}{n^2} \sum_{i=1}^n \sum_{j=1}^n \text{Var}(X_i) = \frac{\sigma^2}{n}$$

The problem with this computation is that it expresses the variance of the mean of observed sample traces in terms of the variance of the *entire* population of unique traces in the system (σ) – the latter quantity is not readily computable. Hence, we would like to express the variance of the sample traces in terms of computable statistics of the observed traces themselves. An unbiased estimator of $\text{Var}(\bar{X})$ is given by:

$$s_{\bar{X}}^2 = \frac{\hat{\sigma}^2}{N} \left(\frac{N - n}{n - 1} \right)$$

where $\hat{\sigma}^2 = \frac{1}{n} \sum_{i=1}^n (X_i - \bar{X})^2$ and N is the total number of samples.

¹ The bias of an estimator is the difference between that estimator's expected value, and the true value of the parameter being estimated. A zero bias estimator is said to be unbiased.

We have shown how the mean and the standard deviation of the sampling distribution of \bar{X} can be computed. The central limit theorem can now be used to show that the sampling distribution can be approximated using a Gaussian distribution. By the central limit theorem,

$$P\left(\frac{\bar{X}_n - \mu}{\sigma\sqrt{n}} \leq z\right) \rightarrow \Phi(z) \quad \text{as } n \rightarrow \infty$$

where Φ is the cumulative distribution function of the standard normal distribution.

To summarize, if we have drawn n samples (and n is large) out of the N possible samples, the distribution of the sample traces is given by the normal distribution $\mathcal{N}(\bar{X}, \frac{N-n}{n(n-1)N} \sum_{i=1}^n (X_i - \bar{X})^2)$. Thus, survey sampling estimates the mean and the variance of the probability of the property being true for the model. We note that the exact form of these expressions or the computation of the sample mean and variance of the sample mean is not a subject of key interest in this paper. The fact that survey sampling can be used for statistical model checking is used to argue the correctness of our algorithms.

6. Problem statement and theorems

Given a parameterized biochemical system $\mathcal{M}(\theta)$ with unknown kinetic parameters θ and a probabilistic adapted finitely monitorable logic formula $Pr_{\geq \rho}(\phi)$, the parameter synthesis problem is to discover a bounded region in parameter space, \mathcal{V} , such that the system \mathcal{M} with parameter values $\theta \equiv (k_1, \dots, k_n) \in \mathcal{V}$, i.e. $\mathcal{M}(\theta)$, satisfies the formula.

A brute-force approach to solving the synthesis problem would involve exhaustively searching the space of all parameter values and using statistical model validation to estimate the probability that each parameter combination results in a model that satisfies the formula. Unfortunately, a brute-force algorithm will not terminate because the search space for the parameter values is uncountably infinite. An alternative approach is to discretize the parameter space and sample from the resulting finite search space. However, two questions remain open:

- I. Can we bound the probability of the formula ϕ being true on the model \mathcal{M} in a *dense set* of parameters by sampling *only finite points* in this dense set?
- II. What is a *good* discretization of the space of parameters?

This paper provides an affirmative answer to the first question. Then we present a methodology to address the second problem by developing a new theoretical characterization for the probability of a formula being true as a function of the reaction rate parameters. We will show that bounded changes in the logarithm of reaction rates make bounded changes in the logarithm of the probability density associated with any *finite path* of non-zero probability measure, and that this change can be made arbitrary small by choosing a sufficiently small change in the reaction rate parameters. To do this we will use the *uniform continuity* of the logarithm of the probability density of a path in a stochastic biochemical model with respect to the logarithm of the reaction rate parameters in a bounded parameter space to prove the correctness of our synthesis algorithms.

6.1. Uniform continuity in the logarithmic parameter space

The change in the logarithm of the probability density associated with any finite path of a biochemical stochastic model can be bounded by a function of the change in the logarithm of the reaction rate (kinetic) parameters. Moreover, this change in the logarithm of the probability density can be made arbitrarily small by choosing a sufficiently small change in the logarithm of the reaction rate parameters.

Theorem 1. If $k_j, k'_j \in (0, M]$ and $|\log k'_j - \log k_j| \leq \delta$, $|k'_j - k_j| \leq M(e^\delta - 1)$.

Proof. Without loss of generality, assume $k_j > k'_j$.

$$\begin{aligned} |\log k'_j - \log k_j| &\leq \delta \\ \Rightarrow \log k_j - \log k'_j &\leq \delta \\ \Rightarrow \log \frac{k_j}{k'_j} &\leq \delta \\ \Rightarrow \frac{k_j}{k'_j} &\leq e^\delta \\ \Rightarrow k_j &\leq k'_j e^\delta \quad (\text{Taking exponential on both sides}) \\ \Rightarrow k_j - k'_j &\leq k'_j (e^\delta - 1) \quad (\text{Subtracting } k'_j \text{ from both sides}) \\ \Rightarrow |k_j - k'_j| &\leq k'_j (e^\delta - 1) \\ \Rightarrow |k_j - k'_j| &\leq M (e^\delta - 1) \quad (\text{Since } k'_j \leq M) \quad \square \end{aligned}$$

Consider a path σ in a biochemical stochastic system $\mathcal{M}(\Theta)$. We recall that Θ denotes the reaction rate parameters. Further, let $P(\sigma)$ denote the probability density associated with the path in $\mathcal{M}(\theta)$ while $P'(\sigma)$ denotes the probability density of the path in $\mathcal{M}(\theta')$. We now show that the difference between the logarithm of $P(\sigma)$ and the logarithm of $P'(\sigma)$ can be made as small as needed by making the difference between θ and θ' sufficiently small.

Theorem 2 (Uniform Continuity of Path Probability Density in Parameter Space). *For every $\epsilon \in \mathcal{R}^+$, there exists $\delta \in \mathcal{R}^+$ such that $|\log P'(\sigma) - \log P(\sigma)| \leq \epsilon$ holds whenever $|\log k'_j - \log k_j| \leq \delta$, for all j ($1 \leq j \leq n$).*

Proof. We present an intuitive sketch of the proof. We know that the probability density of moving from state s_i to state s_{i+1} by executing reaction r_{ji} after time Δ_i is given by

$$P(s_i \xrightarrow{\Delta_i} s_{i+1}) = k_{ji} x_1(s_i)^{\alpha_1^{ji}} \dots x_m(s_i)^{\alpha_m^{ji}} \exp \left(- \sum_{h=1}^n \mathcal{J}(r_h, i) k_h x_1(s_i)^{\alpha_1^h} \dots x_m(s_i)^{\alpha_m^h} \Delta_i \right)$$

Taking logarithms on both sides,

$$\begin{aligned} \log P(s_i \xrightarrow{\Delta_i} s_{i+1}) &= \log (k_{ji} x_1(s_i)^{\alpha_1^{ji}} \dots x_m(s_i)^{\alpha_m^{ji}}) - \sum_{h=1}^n \mathcal{J}(r_h, i) k_h x_1(s_i)^{\alpha_1^h} \dots x_m(s_i)^{\alpha_m^h} \Delta_i \\ &= \log (k_{ji} \gamma_{(i,i+1)}^{ji}) - \sum_{h=1}^n k_h \mathcal{J}(r_h, i) \gamma_{(i,i+1)}^h \Delta_i \end{aligned}$$

Here, $\gamma_{(i,i+1)}^h \stackrel{\text{def}}{=} x_1(s_i)^{\alpha_1^h} \dots x_m(s_i)^{\alpha_m^h}$ is a quantity independent of k_h ($1 \leq h \leq n$).

And so,

$$\begin{aligned} |\log P(s_i \xrightarrow{\Delta_i} s_{i+1}) - \log P'(s_i \xrightarrow{\Delta_i} s_{i+1})| &= |\log k_{ji} - \log k'_{ji}| + \sum_{h=1}^n \mathcal{J}(r_h, i) (k'_h - k_h) \gamma_{(i,i+1)}^h \Delta_i \\ &\leq |\log k_{ji} - \log k'_{ji}| + \gamma_{(i,i+1)}^{\max} \Delta_i \sum_{h=1}^n |k'_h - k_h| \gamma_{(i,i+1)}^{\max} \stackrel{\text{def}}{=} \max_{1 \leq h \leq n} \mathcal{J}(r_h, i) \gamma_{(i,i+1)}^h \Delta_i \end{aligned}$$

Consider the finite path $\sigma \equiv s_0 \xrightarrow{\Delta_0} \dots \xrightarrow{\Delta_{l-1}} s_l$. Let $P(\sigma)$ be the probability density associated with the path in the model $\mathcal{M}(\theta)$ and $P'(\sigma)$ be the probability density associated with the model $\mathcal{M}(\theta')$. We know that $P(\sigma) = P(s_0 \xrightarrow{\Delta_0} s_1) \times \dots \times P(s_{l-1} \xrightarrow{\Delta_{l-1}} s_l)$.

So,

$$\begin{aligned} |\log P(\sigma) - \log P'(\sigma)| &\leq \sum_{i=0}^{l-1} |\log k_{ji} - \log k'_{ji}| + \sum_{i=0}^{l-1} \left(\gamma_{(i,i+1)}^{\max} \Delta_i \sum_{h=1}^n |k'_h - k_h| \right) \dots \text{Triangle Inequality} \\ &\leq l \max_{j_i, i \in [0, l-1]} |\log k_{ji} - \log k'_{ji}| + \left(\sum_{h=1}^n |k'_h - k_h| \right) \sum_{i=0}^{l-1} \left(\gamma_{(i,i+1)}^{\max} \Delta_i \right) \dots \text{Algebraic Manipulation} \\ &\leq l |\log k_j - \log k'_j|^{\max} + \gamma^{\max} \left(\sum_{h=1}^n |k'_h - k_h| \right) \sum_{i=0}^{l-1} \Delta_i \end{aligned}$$

where $\gamma^{\max} \stackrel{\text{def}}{=} \max_{i \in [0, l-1]} \gamma_{(i,i+1)}^{\max}$ and $|\log k_j - \log k'_j|^{\max} \stackrel{\text{def}}{=} \max_{j_i, i \in [0, l-1]} |\log k_{ji} - \log k'_{ji}|$. And so,

$$\begin{aligned} |\log P(\sigma) - \log P'(\sigma)| &\leq l |\log k_j - \log k'_j|^{\max} + \gamma^{\max} \left(\sum_{h=1}^n |k'_h - k_h| \right) \Delta_{\text{total}} \dots \Delta_{\text{total}} \equiv \sum_{i=0}^{l-1} \Delta_i \\ &\leq l \delta + \gamma^{\max} \left(\sum_{h=1}^n M(e^\delta - 1) \right) \Delta_{\text{total}} \dots \text{From Theorem 1} \end{aligned}$$

To show that $|\log P(\sigma) - \log P'(\sigma)| \leq \epsilon$, it is sufficient to show that the following holds:

$$l \delta + \gamma^{\max} \left(\sum_{h=1}^n M(e^\delta - 1) \right) \Delta_{\text{total}} \leq \epsilon$$

From the statement of our theorem, we know that $|\log k_j - \log k'_j|^{max} \leq \delta$. One can verify that the following choice of δ is sufficient to show that $|\log P(\sigma) - \log P'(\sigma)| \leq \epsilon$:

$$\delta = \min \left(\frac{\epsilon}{l(n+1)}, \log \left(\frac{\epsilon}{(n+1) \max(\gamma^{max} M \Delta_{total}, 1)} + 1 \right) \right)$$

$$\stackrel{def}{=} \delta(\epsilon, \mathcal{M}).$$

In the rest of the paper, we will use the notation $\delta(\epsilon, \mathcal{M})$ to denote this value of δ . \square

The uniform continuity arguments we have presented allow us to establish results on a *finite* set of points in a bounded parameter space and then extend the statements of these results to the entire *uncountably infinite* parameter space. A natural follow-up investigation is to characterize the probability of a formula being true as a function on the parameter space. In the following lemma, we define an unbiased statistical estimator of the probability of a formula being true on a model.

Lemma 1 (Survey Sampling based Unbiased Statistical Estimator for the Probability of a Finite Set of Paths). *Given a finite set of independent and identically distributed (i.i.d.) sample paths $\sigma_1, \sigma_2, \dots, \sigma_T$ (of length at most l) drawn uniformly from a (possibly infinite) set of paths \mathcal{T} such that each path is labeled with either 0 or 1 i.e. $L(\sigma_i) \in \{0, 1\}$, an unbiased statistical estimator for the probability of the set of paths with label 1 in \mathcal{T} is given by*

$$\hat{p} \stackrel{def}{=} \frac{\sum_{L(\sigma_t)=1, 1 \leq t \leq T} P(\sigma_t)}{\sum_{t=1}^T P(\sigma_t)}$$

i.e.

$$\hat{p} \stackrel{def}{=} \frac{\sum_{L(\sigma_t)=1, 1 \leq t \leq T} \prod_{i=1}^l k_{j_i}(\sigma_t) \gamma_{(i,i+1)}^{j_i}(\sigma_t) \exp \left(- \sum_{h=1}^n \mathbb{I}(r_h, i, \sigma_t) k_h(\sigma_t) \gamma_{(i,i+1)}^h(\sigma_t) \Delta_i(\sigma_t) \right)}{\sum_{t=1}^T \prod_{i=1}^l \mathbb{I}(r_h, i, \sigma_t) k_{j_i}(\sigma_t) \gamma_{(i,i+1)}^{j_i}(\sigma_t) \exp \left(- \sum_{h=1}^n k_h(\sigma_t) \gamma_{(i,i+1)}^h(\sigma_t) \Delta_i(\sigma_t) \right)}$$

Here, $k_{j_i}(\sigma_t)$, $\gamma_{(i,i+1)}^{j_i}(\sigma_t)$, and $\Delta_i(\sigma_t)$ represent the values of k_{j_i} , $\gamma_{(i,i+1)}^{j_i}$ and Δ_i corresponding to the path σ_t . Also, $\gamma_{(i,i+1)}^h(\sigma_t) \stackrel{def}{=} x_1(s_i(\sigma_t))^{a_1^h} \dots x_m(s_i(\sigma_t))^{a_m^h}$ is a quantity independent of $k_h(\sigma_t)$ ($0 \leq h \leq n$, $1 \leq t \leq T$). The indicator function $\mathbb{I}(r_h, i, \sigma_t)$ indicated whether the reaction r_h was fired at the i^{th} step in the path σ_t .

Theorem 3 (Uniform Continuity of the Unbiased Estimator). *Let \hat{P} be the unbiased statistical estimator of the probability with which an AFM specification ϕ is true on the model $\mathcal{M}(\theta)$ and \hat{P}' be the unbiased statistical estimator of the probability with which ϕ is true on the model $\mathcal{M}(\theta')$. For every $\epsilon \in \mathbb{R}^+$, there exists $\delta \in \mathbb{R}^+$ such that $|\log \hat{P}' - \log \hat{P}| \leq \epsilon$ holds whenever $|\log k'_j - \log k_j| \leq \delta$, for all j ($1 \leq j \leq n$).*

Proof. Consider a survey sampling based unbiased statistical estimator of the probability of the formula using T samples $\sigma_1, \sigma_2 \dots \sigma_T$. For any $\epsilon/2 \in \mathbb{R}^+$, we know that there exists $\delta_i \in \mathbb{R}^+$ such that $|\log P'(\sigma_i) - \log P(\sigma_i)| \leq \frac{\epsilon}{2}$ holds whenever $|\log k'_j - \log k_j| \leq \delta_i$. Choose δ as the smallest of all δ_i ($1 \leq i \leq T$).

$$\hat{P}' = \frac{\sum_{\sigma \models \phi} P'(\sigma)}{\sum_{\sigma} P'(\sigma)} \quad \text{Statistical Estimator Definition}$$

$$\leq \frac{\sum_{\sigma \models \phi} e^{\frac{\epsilon}{2}} P(\sigma)}{\sum_{\sigma} e^{-\frac{\epsilon}{2}} P(\sigma)} \quad \text{Uniform Continuity of Paths}$$

$$= e^{\epsilon} \frac{\sum_{\sigma \models \phi} P(\sigma)}{\sum_{\sigma} P(\sigma)} \quad \text{Algebraic Manipulation}$$

$$= e^{\epsilon} \hat{P} \quad \text{Statistical Estimator Definition}$$

$$\implies \log \hat{P}' - \log \hat{P} \leq \epsilon \quad \text{Taking log on both sides}$$

Similarly, we can also argue that

$$\begin{aligned}
 \hat{P}' &= \frac{\sum_{\sigma \models \phi} P'(\sigma)}{\sum_{\sigma} P'(\sigma)} && \text{Statistical Estimator Definition} \\
 &\geq \frac{\sum_{\sigma \models \phi} e^{-\frac{\epsilon}{2}} P(\sigma)}{\sum_{\sigma} e^{\frac{\epsilon}{2}} P(\sigma)} && \text{Uniform Continuity of Paths} \\
 &= e^{-\epsilon} \frac{\sum_{\sigma \models \phi} P(\sigma')}{\sum_{\sigma} P(\sigma')} && \text{Algebraic Manipulation} \\
 &= e^{-\epsilon} \hat{P} && \text{Statistical Estimator Definition} \\
 \implies \log \hat{P}' - \log \hat{P} &\geq -\epsilon && \text{Taking log on both sides}
 \end{aligned}$$

Thus, we know that $|\log \hat{P}' - \log \hat{P}| \leq \epsilon$. Hence, the logarithm of the unbiased statistical estimator of the probability of the formula is uniformly continuous in the kinetic parameter space. \square

If one could show that the probability of a formula being true is monotonic as a function on the parameter space, it would be possible to develop abstraction refinement algorithms [14,34] by sampling with varying discretizations of the parameter space. In the following theorem, we show that the probability density of a path is *not* necessarily monotonic in the parameter space and compute the point in the parameter space where the extremum of the probability density of a path is reached.

Theorem 4 (Non-Monotonicity of Path Probability Density in the Parameter Space). *The probability density of a path in a stochastic biochemical model is not necessarily monotonic in the parameter space.*

Proof.

$$\begin{aligned}
 P(\sigma) &= P(s_0 \xrightarrow{\Delta_0} s_1) \times \cdots \times P(s_{l-1} \xrightarrow{\Delta_{l-1}} s_l) \\
 \implies \log P(\sigma) &= \log P(s_0 \xrightarrow{\Delta_0} s_1) + \cdots + \log P(s_{l-1} \xrightarrow{\Delta_{l-1}} s_l) \\
 \implies \frac{1}{P(\sigma)} \frac{dP(\sigma)}{dk_u} &= \frac{d}{dk_u} \sum_{i=0}^{l-1} \left(\log k_{j_i} + \log (\gamma_{(i,i+1)}^{j_i}) - \sum_{h=1}^n \mathbb{I}(r_h, i) k_h \gamma_{(i,i+1)}^h \Delta_i \right) \\
 \implies \frac{dP(\sigma)}{dk_u} &= P(\sigma) \sum_{i=0}^{l-1} \left(\mathbb{I}(u = j_i) \frac{1}{k_u} - \mathbb{I}(r_u, i) \gamma_{(i,i+1)}^u \Delta_i \right)
 \end{aligned}$$

Clearly, $\frac{dP(\sigma)}{dk_u}$ can be either positive or negative depending upon the path in consideration and the value of the kinetic parameters. Continuity ensures that the function $\frac{dP(\sigma)}{dk_u}$ is zero for some value of k_u . \square

Theorem 5 (Extremum of the Probability Density of a Path). *The probability density of a path $\sigma \equiv s_0 \xrightarrow{\Delta_0} s_1 \xrightarrow{\Delta_1} s_2 \cdots \xrightarrow{\Delta_{l-1}} s_l$ in the stochastic model attains a unique extremum at the point $(k_1^{\text{extrema}}, \dots, k_n^{\text{extrema}})$, where*

$$k_u^{\text{extrema}} = \frac{N^{k_u}}{\sum_{i=0}^{l-1} \mathbb{I}(r_u, i) \gamma_{(i,i+1)}^u \Delta_i}$$

and N^{k_u} is the number of times the reaction r_u is executed along the path σ .

Proof.

$$\begin{aligned}
 \log P(\sigma) &= \log P(s_0 \xrightarrow{\Delta_0} s_1) + \cdots + \log P(s_{l-1} \xrightarrow{\Delta_{l-1}} s_l) \\
 &= \sum_{i=1}^{l-1} \log k_{j_i} + \sum_{i=1}^{l-1} \log (\gamma_{(i,i+1)}^{j_i}) - \sum_{i=1}^{l-1} \sum_{h=1}^n \mathbb{I}(r_h, i) k_h \gamma_{(i,i+1)}^h \Delta_i
 \end{aligned}$$

Partially differentiating with respect to each reaction rate parameter and setting the gradient so obtained to 0, we get the desired result. \square

Our negative results on the monotonicity of the probability density of a path with respect to variations in the kinetic parameters make it difficult to argue the monotonicity of the probability of a formula being true on a model. The definition of the unbiased statistical estimator is used to argue that its value remains monotonic in any given positive parameter space under a technical condition. The condition that we need to satisfy is that the estimator should not take the values 0 or 1 anywhere inside the positive parameter space.

Theorem 6 (Absence of Local Extrema of the Unbiased Statistical Estimator). *The unbiased statistical estimator of a non-trivial probability (true with probability neither 0 nor 1) of a measurable set estimated using a finite number of finite length paths in a stochastic biochemical model does not admit a local extrema anywhere in the positive kinetic parameter space.*

Proof. Assume that the unbiased statistical estimator \hat{P} does have a local extrema at $k_u (\neq 0)$, for the sake of contradiction.

$$\hat{P} \stackrel{\text{def}}{=} \frac{\sum_{L(\sigma_t)=1, 1 \leq t \leq T} \prod_{i=1}^l k_{j_i}(\sigma_t) \gamma_{(i,i+1)}^{j_i}(\sigma_t) \exp \left(- \sum_{h=1}^n k_h(\sigma_t) \gamma_{(i,i+1)}^h(\sigma_t) \Delta_i(\sigma_t) \right)}{\sum_{t=1}^T \prod_{i=1}^l k_{j_i}(\sigma_t) \gamma_{(i,i+1)}^{j_i}(\sigma_t) \exp \left(- \sum_{h=1}^n k_h(\sigma_t) \gamma_{(i,i+1)}^h(\sigma_t) \Delta_i(\sigma_t) \right)}$$

$$\stackrel{\text{def}}{=} \frac{P_{\text{one}}(k_u)}{P_{\text{all}}(k_u)} \quad (2)$$

Note that $P_{\text{one}}(k_u)$ represents the sum of the probability densities of all the sampled paths that satisfy the AFM specification ϕ and are labeled 1, and $P_{\text{all}}(k_u)$ simply represents the sum of the probability densities of all the sampled paths. Setting $\frac{\partial \hat{P}}{\partial k_u}$ to 0, we get:

$$P_{\text{all}}(k_u) \frac{\partial P_{\text{one}}(k_u)}{\partial k_u} - P_{\text{one}}(k_u) \frac{\partial P_{\text{all}}(k_u)}{\partial k_u} = 0$$

$$\Rightarrow P_{\text{all}}(k_u) \frac{\partial P_{\text{one}}(k_u)}{\partial k_u} = P_{\text{one}}(k_u) \frac{\partial P_{\text{all}}(k_u)}{\partial k_u} \quad \text{Algebraic Manipulation}$$

$$\Rightarrow \frac{1}{P_{\text{one}}(k_u)} \frac{\partial P_{\text{one}}(k_u)}{\partial k_u} = \frac{1}{P_{\text{all}}(k_u)} \frac{\partial P_{\text{all}}(k_u)}{\partial k_u} \quad \text{Algebraic Manipulation} \quad (3)$$

$$\Rightarrow \frac{1}{\frac{P_{\text{one}}(k_u)}{P_{\text{all}}(k_u)}} \frac{\partial P_{\text{one}}(k_u)}{\partial k_u} = \frac{\partial P_{\text{all}}(k_u)}{\partial k_u} \quad \text{Algebraic Manipulation}$$

$$\Rightarrow \frac{1}{\hat{P}(k_u)} \frac{\partial P_{\text{one}}(k_u)}{\partial k_u} = \frac{\partial P_{\text{all}}(k_u)}{\partial k_u} \quad \text{Definition of } \hat{P}(k_u) \quad (4)$$

Now, let $P_{\text{zero}}(k_u)$ represent the sum of the probability densities of all the sampled paths that do not satisfy the AFM specification ϕ .

$$P_{\text{all}}(k_u) = P_{\text{one}}(k_u) + P_{\text{zero}}(k_u) \quad \text{By Definition}$$

$$\Rightarrow \frac{\partial P_{\text{all}}(k_u)}{\partial k_u} = \frac{\partial P_{\text{one}}(k_u)}{\partial k_u} + \frac{\partial P_{\text{zero}}(k_u)}{\partial k_u} \quad \text{Differentiating both sides}$$

$$\Rightarrow \frac{1}{P_{\text{all}}(k_u)} \frac{\partial P_{\text{all}}(k_u)}{\partial k_u} = \frac{\frac{\partial P_{\text{one}}(k_u)}{\partial k_u} + \frac{\partial P_{\text{zero}}(k_u)}{\partial k_u}}{P_{\text{one}}(k_u) + P_{\text{zero}}(k_u)} \quad \text{Dividing both sides by } P_{\text{all}}(k_u) \quad (5)$$

Also,

$$\frac{\frac{\partial P_{\text{one}}(k_u)}{\partial k_u}}{P_{\text{one}}} = \frac{\frac{\partial P_{\text{one}}(k_u)}{\partial k_u} + \frac{\partial P_{\text{zero}}(k_u)}{\partial k_u}}{P_{\text{one}} + P_{\text{zero}}} \quad \text{From (3) and (5)}$$

$$\Rightarrow \frac{\frac{\partial P_{\text{one}}(k_u)}{\partial k_u}}{P_{\text{one}}} = \frac{\frac{\partial P_{\text{zero}}(k_u)}{\partial k_u}}{P_{\text{zero}}} \quad \text{Algebraic Manipulation} \quad (6)$$

$$\Rightarrow \frac{\frac{\partial P_{\text{one}}(k_u)}{\partial k_u}}{P_{\text{one}} / (P_{\text{one}} + P_{\text{zero}})} = \frac{\frac{\partial P_{\text{zero}}(k_u)}{\partial k_u}}{P_{\text{zero}} / (P_{\text{one}} + P_{\text{zero}})} \quad \text{Dividing both sides}$$

$$\Leftrightarrow \frac{1}{\hat{P}(k_u)} \frac{\partial P_{\text{one}}(k_u)}{\partial k_u} = \frac{1}{1 - \hat{P}(k_u)} \frac{\partial P_{\text{zero}}(k_u)}{\partial k_u} \quad \text{Estimator Definition} \quad (7)$$

Hence,

$$\frac{1}{\hat{P}(k_u)} \frac{\partial P_{\text{one}}(k_u)}{\partial k_u} = \frac{1}{1 - \hat{P}(k_u)} \frac{\partial P_{\text{zero}}(k_u)}{\partial k_u} = \frac{\partial P_{\text{all}}(k_u)}{\partial k_u} \quad \text{From Eqs. (4) and (7)}$$

Now, given a finite set of paths S and the sum of probability density P_S of these paths,

$$\begin{aligned}
 P_S &= \sum_{\sigma \in S} P(\sigma) \quad (\text{By Definition}) \\
 \Rightarrow \frac{\delta P_S}{\delta k_u} &= \frac{\delta}{\delta k_u} \sum_{\sigma \in S} P(\sigma) \\
 \Rightarrow \frac{\delta P_S}{\delta k_u} &= \sum_{\sigma \in S} \frac{\delta}{\delta k_u} (P(\sigma)) \quad (\text{Derivative of Finite Sums}) \\
 \Rightarrow \frac{\delta P_S}{\delta k_u} &= \sum_{\sigma \in S} \left(P(\sigma) \sum_{i=0}^{l(\sigma)-1} \left(\mathbb{I}(u = j_i(\sigma)) \frac{1}{k_u} - \mathbb{I}(r_u, i) \gamma_{(i,i+1)}^u \Delta_i(\sigma) \right) \right) \\
 \Rightarrow \frac{\delta P_S}{\delta k_u} &= \sum_{\sigma \in S} P(\sigma) \left(\frac{N_u(\sigma)}{k_u} - \sum_{i=0}^{l(\sigma)-1} (\mathbb{I}(r_u, i) \gamma_{(i,i+1)}^u \Delta_i(\sigma)) \right)
 \end{aligned}$$

Now,

$$\frac{\partial P_{\text{one}}(k_u)}{\partial k_u} = \frac{P_{\text{one}}(k_u)}{P_{\text{zero}}(k_u)} \frac{\partial P_{\text{zero}}(k_u)}{\partial k_u} \quad \text{From Eq. (6)} \quad (8)$$

$$\Rightarrow \sum_{\sigma \in S, \sigma \models \phi} P(\sigma) \left(\frac{N_u(\sigma)}{k_u} - \sum_{i=0}^{l(\sigma)-1} \mathbb{I}(r_u, i) \gamma_{(i,i+1)}^u \Delta_i(\sigma) \right) = \sum_{\sigma \in S, \sigma \models \phi} P(\sigma) \left(\frac{1}{P_{\text{zero}}(k_u)} \frac{\partial P_{\text{zero}}(k_u)}{\partial k_u} \right) \quad (9)$$

Consider a more accurate statistical estimator with one more samples s . Without loss of generality, assume that the sample satisfies the formula ϕ . Let k'_u be the point at which the new estimator reached an extrema.

$$\sum_{\sigma \in S \cup s, \sigma \models \phi} P(\sigma) \left(\frac{N_u(\sigma)}{k'_u} - \sum_{i=0}^{l(\sigma)-1} \mathbb{I}(r_u, i) \gamma_{(i,i+1)}^u \Delta_i(\sigma) \right) = \sum_{\sigma \in S \cup s, \sigma \models \phi} P(\sigma) \left(\frac{1}{P_{\text{zero}}(k'_u)} \frac{\partial P_{\text{zero}}(k'_u)}{\partial k_u} \right) \quad (10)$$

Thus, subtracting Eq. (8) from Eq. (10), we get

$$\begin{aligned}
 &\sum_{\sigma \in S, \sigma \models \phi} \left(P(\sigma) \frac{N_u(\sigma)}{k'_u} - P(\sigma) \frac{N_u(\sigma)}{k_u} \right) + P(s) \left(\frac{N_u(s)}{k'_u} - \sum_{i=0}^{l(s)-1} (\mathbb{I}(r_u, i) \gamma_{(i,i+1)}^u \Delta_i(s)) \right) \\
 &= \sum_{\sigma \in S, \sigma \models \phi} \left(P(\sigma) \frac{1}{P_{\text{zero}}(k'_u)} \frac{\partial P_{\text{zero}}(k'_u)}{\partial k_u} - P(\sigma) \frac{1}{P_{\text{zero}}(k_u)} \frac{\partial P_{\text{zero}}(k_u)}{\partial k_u} \right) + P(s) \left(\frac{1}{P_{\text{zero}}(k'_u)} \frac{\partial P_{\text{zero}}(k'_u)}{\partial k_u} \right)
 \end{aligned}$$

Since, $\frac{1}{k_u}$, P_{zero} , $P(\sigma)$, $\frac{\partial P_{\text{zero}}}{\partial k_u}$ are continuous functions of k_u in the positive parameter space, the following holds true as the number of samples used by the statistical estimator increases and if the unbiased statistical estimator converges i.e. $k'_u \rightarrow k_u$:

$$\begin{aligned}
 P(s) \left(\frac{N_u(s)}{k_u} - \sum_{i=0}^{l(s)-1} (\mathbb{I}(r_u, i) \gamma_{(i,i+1)}^u \Delta_i(s)) \right) &= P(s) \left(\frac{1}{P_{\text{zero}}(k_u)} \frac{\partial P_{\text{zero}}(k_u)}{\partial k_u} \right) \\
 \Rightarrow k_u &= \frac{N_u(s)}{\left(\frac{1}{P_{\text{zero}}(k_u)} \frac{\partial P_{\text{zero}}(k_u)}{\partial k_u} + \sum_{i=0}^{l(s)-1} \mathbb{I}(r_u, i) \gamma_{(i,i+1)}^u \Delta_i(s) \right)}
 \end{aligned}$$

The location of the extrema is a function of the new sample we chose unless $\frac{1}{P_{\text{zero}}(k_u)} \frac{\partial P_{\text{zero}}(k_u)}{\partial k_u} \rightarrow \infty$. If $P_{\text{zero}}(k_u) \neq 0$, then the only possibility is $\frac{\partial P_{\text{zero}}(k_u)}{\partial k_u} \rightarrow \infty$ i.e. $\sum_{\sigma \in S} P(\sigma) \left(\frac{N_u(\sigma)}{k_u} - \sum_{i=0}^{l(\sigma)-1} (\mathbb{I}(r_u, i) \gamma_{(i,i+1)}^u \Delta_i(\sigma)) \right) \rightarrow \infty$. The latter is not possible as $k_u \neq 0$ and the other terms are bounded for any simulation.

Thus, there is no extrema in the positive parameter space. \square

In this section, we have shown several important results about the probability of a formula being true on a stochastic biochemical model:

- (i) The logarithm of the probability density of a path in a stochastic biochemical model is *uniformly* and *jointly* continuous in the logarithmic kinetic parameter space. Our proof is constructive and hence, suggests a natural sampling based algorithm which we present in Section 7.1. We are aware that a non-constructive proof would be simpler but not algorithmically useful.

- (ii) The probability density of a path is not necessarily monotonic in the parameter space. Thus, the natural mechanism of using monotonicity of paths to argue the monotonicity of the statistical estimator of the probability of a model satisfying a formula is not available.
- (iii) An indirect proof using *survey sampling* based unbiased statistical estimators establishes that the unbiased statistical estimator is indeed monotonic in the positive parameter space under certain technical conditions. This provides the opportunity for constructing efficient synthesis and search algorithms. We present these algorithms in Sections 7.2 and 7.3 respectively.

7. Parameter synthesis algorithms

We have characterized the parameter space of an *adapted finitely monitorable* formula being true on a stochastic biochemical model in the previous section. Now, we use our understanding of the parameter space to suggest efficient algorithms for parameter synthesis of stochastic biochemical models against high-level behavioral specifications.

7.1. Parameter synthesis using uniform continuity

We have shown that the probability density of a path does not change arbitrarily as we change the reaction rate parameters of a stochastic biochemical system. This result will now enable us to prove results on the *dense* parameter space (with uncountably many parameter values) by sampling *only finitely many parameter values* in the parameter space. **Algorithm 1** takes five inputs:

- (i) Stochastic Biochemical Model \mathcal{M} with unknown kinetic parameters Θ ,
- (ii) A high-level behavioral specification about the system specified in a probabilistic adapted finitely monitorable logic $Pr_{\geq \rho}(\phi)$,
- (iii) The space in which the possible values θ of reaction parameters are to be searched: $\theta \in [\Theta_1^{\min}, \Theta_1^{\max}] \times \dots \times [\Theta_{n_\Theta}^{\min}, \Theta_{n_\Theta}^{\max}]$
- (iv) An error tolerance η such that $\sqrt{\rho} < \eta < 1$: A number close to 1 which specifies the acceptable error in the synthesis of parameters. All points within the synthesized parameter set will satisfy the adapted finitely monitorable property with probability at least ρ , and those outside the set satisfy the specification with probability no greater than $\frac{\rho}{\eta^2}$.
- (v) A confidence value, q , which will be passed to the statistical model validation algorithm that is called as a subroutine.

The algorithm initializes the set of satisfying parameters to the empty set. It then uses the error tolerance η to compute ϵ , the required resolution of the discretization of the logarithm of the probability space. Next, the algorithm discretizes the logarithmic parameter space. Note that the notation $[\log \Theta_1^{\min}, \log \Theta_1^{\max}]_\delta$ is used to represent the set $\{\log \theta \mid \log \Theta_1^{\min} \leq \log \theta \leq \log \Theta_1^{\max}, \text{ and } \log \theta = z \cdot \delta \text{ for some } z \in \mathbb{Z}\}$. For each discrete box in the logarithmic parameter space, the algorithm samples a point and tests whether it satisfies the property with probability at least $\frac{\rho}{\eta}$. If so, the algorithm adds the exponential of this point (and the discrete hyperbox of size δ around it) to the set of synthesized parameters. We use the notation $\mathcal{B}_\delta((c_1, \dots, c_n))$ to represent the hyperbox of size δ around the center point (c_1, \dots, c_n) i.e. the set $\{(x_1, \dots, x_n) \mid \max_{1 \leq i \leq n} |c_i - x_i| \leq \delta\}$. The algorithm terminates after each discrete box has been examined. If no parameter combination produces a model that satisfies the formula, the algorithm reports that the model is *infeasible* with respect to the given high-level behavioral specification. Knowledge about the infeasibility of a model is of practical importance, because it indicates that the model itself has structural flaws (eg., missing biochemical pathways) which need to be addressed before parameter synthesis can be attempted. This is significant because manual *ad hoc* search procedures can never prove the infeasibility of the model with respect to the behavioral specification, and the designer is left to wonder if the model is actually infeasible or she has just not found the right parameters for the model yet.

Theorem 7. If θ is a point in the synthesized parameter set $\tilde{\mathcal{V}}_{\phi, \rho}$ returned by **Algorithm 1**, $\mathcal{M}(\theta) \models_q Pr_{\geq \rho}(\phi)$.

Proof. Suppose θ is a point in $\tilde{\mathcal{V}}_{\phi, \rho}$. Then,

- (i) By the construction of the set $\tilde{\mathcal{V}}_{\phi, \rho}$ by the algorithm, there exists a θ_{\log} such that $\theta \in \bigcup_{b \in \mathcal{B}_\delta(\theta_{\log})} \{\exp(b)\}$ and $\mathcal{M}(\exp(\theta_{\log})) \models_q Pr_{\geq \frac{\rho}{\eta}}(\phi)$.
- (ii) Since $\theta \in \bigcup_{b \in \mathcal{B}_\delta(\theta_{\log})} \{\exp(b)\}$ i.e. θ lies in the δ -neighbourhood of θ_{\log} , $|\theta - \theta_{\log}| < \delta$.
- (iii) By uniform continuity, we know that $\mathcal{M}(\theta)$ satisfies $Pr_{\geq (\frac{\rho}{\eta} e^\epsilon)}(\phi)$. Note that $e^\epsilon < 1$ by construction.
- (iv) By our choice of η , we know that $e^\epsilon = \eta$.

Hence, $\mathcal{M}(\theta)$ satisfies $Pr_{\geq \rho}(\phi)$ up to the confidence probability q . \square

Theorem 8. If θ is not a point in the synthesized parameter set $\tilde{\mathcal{V}}_{\phi, \rho}$ returned by **Algorithm 1**, $\mathcal{M}(\theta)$ satisfies $Pr_{\leq \frac{\rho}{\eta^2}}(\phi)$.

Proof. Suppose θ is not a point in $\tilde{\mathcal{V}}_{\phi, \rho}$. Then,

- (i) By the construction of the set $\tilde{\mathcal{V}}_{\phi, \rho}$ in the algorithm, there does not exist any point θ_{\log} such that $\theta \in \bigcup_{b \in \mathcal{B}_\delta(\theta_{\log})} \{\exp(b)\}$ and $\mathcal{M}(\exp(\theta_{\log})) \models_q Pr_{\geq \frac{\rho}{\eta}}(\phi)$.

Algorithm 1 Parameter Synthesis using Statistical model validation

Require: Parameterized Biochemical Model $\mathcal{M}(\Theta)$,
 Probabilistic Adapted Finitely Monitorable Formula $Pr_{\geq \rho}(\phi)$,
 Parameter Space $\theta \in [\Theta_1^{\min}, \Theta_1^{\max}] \times \dots \times [\Theta_{n_\Theta}^{\min}, \Theta_{n_\Theta}^{\max}]$,
 Error Tolerance in PAFM Specification $\sqrt{\rho} < \eta < 1$,
 Confidence Probability q .

Ensure: Set $\tilde{\mathcal{V}}_{\phi, \rho}$ of parameter values such that
 (i) $\forall \theta \in \tilde{\mathcal{V}}_{\phi, \rho}, \mathcal{M}(\theta) \models_q Pr_{\geq \rho}(\phi)$, and
 (ii) $\forall \theta \notin \tilde{\mathcal{V}}_{\phi, \rho}, \mathcal{M}(\theta) \models_q Pr_{\leq \frac{\rho}{\eta^2}}(\phi)$.

// Initialize set of parameter values to the empty set.

$\tilde{\mathcal{V}}_{\phi, \rho} = \{\}$

// Compute ϵ and δ from η .

$\epsilon = \lfloor \log \eta \rfloor$

$\delta = \delta(\epsilon/2, \mathcal{M})$

// Search the discretized parameter space.

for all $\theta_{\log} \in [\log \Theta_1^{\min}, \log \Theta_1^{\max}]_\delta \times \dots \times [\log \Theta_{n_\Theta}^{\min}, \log \Theta_{n_\Theta}^{\max}]_\delta$ **do**

// If a parameter value satisfies the PAFM formula with probability $\frac{\rho}{\eta}$

if $\mathcal{M}(\exp(\theta_{\log})) \models_q Pr_{\geq \frac{\rho}{\eta}}(\phi)$ **then**

// Add the δ -ball around this parameter value to the set S .

$\tilde{\mathcal{V}}_{\phi, \rho} = \tilde{\mathcal{V}}_{\phi, \rho} \cup \left\{ \bigcup_{b \in \mathcal{B}_\delta(\theta_{\log})} \{\exp(b)\} \right\} \quad // B_\delta(x) \stackrel{\text{def}}{=} \{y \mid |x - y| \leq \delta\}$

end if

end for

if $\tilde{\mathcal{V}}_{\phi, \rho} == \{\}$ **then**

Print “Model is infeasible”.

end if

- (ii) But the algorithm must have sampled a point θ'_{\log} such that $\theta \in \bigcup_{b \in \mathcal{B}_\delta(\theta'_{\log})} \{\exp(b)\}$.
- (iii) Since the algorithm did not add this point to S , it must be the case that $\mathcal{M}(\exp(\theta'_{\log})) \models_q Pr_{< \frac{\rho}{\eta}}(\phi)$.
- (iv) Since $\theta \in \bigcup_{b \in \mathcal{B}_\delta(\theta'_{\log})} \{\exp(b)\}$ i.e. θ lies in the δ -neighbourhood of θ'_{\log} , $|\theta - \theta'_{\log}| < \delta$.
- (v) By uniform continuity, we know that $\mathcal{M}(\theta)$ satisfies $Pr_{\leq (\frac{\rho}{\eta e^\epsilon})}(\phi)$.
- (vi) By our choice of η , we know that $e^\epsilon = \eta$.

Hence, $\mathcal{M}(\theta)$ satisfies $Pr_{\leq \frac{\rho}{\eta^2}}(\phi)$ up to the confidence probability q . \square

Theorem 9. The number of discrete parameter values sampled by the algorithm is polylogarithmic in the error tolerance of the PAFM specification η .

Proof. Given the error tolerance η , the discretization ϵ chosen by the algorithm in the logarithmic probability space is logarithmic in the error tolerance η . The discretization δ of the logarithmic parameter space is $C\epsilon$, where C is a factor independent of ϵ .

From the algorithm, we know that the number of parameter values to be sampled is $\left(\max_{1 \leq i \leq n_\Theta} n_\Theta \log \frac{\Theta_i^{\max}}{\Theta_i^{\min}} \frac{1}{\delta} \right)^{n_\Theta}$.

Rewriting, the number of sampled values is $\left(\max_{1 \leq i \leq n_\Theta} \frac{n_\Theta}{\delta^{n_\Theta}} \log \frac{\Theta_i^{\max}}{\Theta_i^{\min}} \right)$, which is the same as

$$\left(\max_{1 \leq i \leq n_\Theta} \frac{n_\Theta}{(C \lfloor \log \eta \rfloor)^{n_\Theta}} \log \frac{\Theta_i^{\max}}{\Theta_i^{\min}} \right). \quad \square$$

7.2. Faster parameter synthesis using abstraction refinement

We have also shown that the unbiased statistical estimator for the probability of a measurable set of paths satisfying a formula does not admit any local extrema as we change the reaction rate parameters, unless the probability of a formula being true on the model is either unity or zero somewhere in the logarithmic parameter space being explored. We can therefore modify the previous algorithm to perform a hierarchical search through the parameter space to accelerate

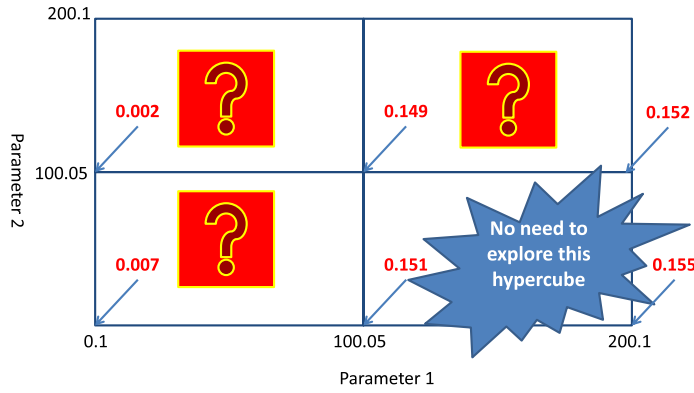


Fig. 2. Central idea behind the abstraction refinement algorithm.

synthesis. We refer to this hierarchical decomposition of the parameter space as *abstraction-refinement*. We are not the first to use hierarchical abstractions and binary search refinements. Similar ideas have been pursued in the context of nonlinear hybrid systems [34] and stochastic systems [29]. However, our monotonicity results (see Section 6) considerably simplify our algorithm.

The central idea behind abstraction refinement is illustrated in Fig. 2. Suppose we want to check that a certain formula is true with probability *no more* than 0.14. If all the corners of a hyperbox satisfy the formula, or if all the corners do not satisfy the formula, then our monotonicity results (see Section 6) allow one to stop further analysis of this hyperbox. In the figure, the lower right corner box clearly does not satisfy the PAFM specification and need not be further analyzed at all.

The abstraction refinement based algorithm takes the same parameters as our earlier algorithm. However, it assumes that the probability of the formula does not vanish or reach unity anywhere in the parameter space being analyzed.

Given the monotonicity of the probability in each of the parameters, we can construct underapproximate and overapproximate abstractions in a bounded parameter space. An underapproximate (overapproximate) abstraction is the parameterized model which satisfies an AFM specification with the minimum (maximum) probability in a given parameter space.

Definition 5 (Underapproximate Abstraction). Given parameters Θ and a bounded parameter space $S = [\Theta_1^{\min}, \Theta_1^{\max}] \times \dots \times [\Theta_{n_\Theta}^{\min}, \Theta_{n_\Theta}^{\max}]$ over which the probability of the AFM specification being true is monotonic in each of the parameters, the variable m_i is assigned the value 1 if the probability of a formula being true is monotonically increasing in the parameter Θ_i ; it is zero if the probability is monotonically decreasing.

The underapproximate abstraction of the set of models $\mathcal{M}(\theta)$ in the bounded parameter space $\theta \in [\Theta_1^{\min}, \Theta_1^{\max}] \times \dots \times [\Theta_{n_\Theta}^{\min}, \Theta_{n_\Theta}^{\max}]$ is given by the model $\mathcal{M}(\theta_{\min})$, where $\theta_{\min} = ((1 - m_1)(\Theta_1^{\max} - \Theta_1^{\min}) + \Theta_1^{\min}, \dots, (1 - m_{n_\Theta})(\Theta_{n_\Theta}^{\max} - \Theta_{n_\Theta}^{\min}) + \Theta_{n_\Theta}^{\min})$. We also denote the underapproximate model $\mathcal{M}(\theta_{\min})$ by $\underline{\mathcal{M}}(S)$.

Definition 6 (Overapproximate Abstraction). Given parameters Θ and a bounded parameter space $[\Theta_1^{\min}, \Theta_1^{\max}] \times \dots \times [\Theta_{n_\Theta}^{\min}, \Theta_{n_\Theta}^{\max}]$ over which the probability of the AFM specification being true is monotonic in each of the parameters, the variable m_i is assigned the value 1 if the probability of a formula being true is monotonically increasing in the parameter Θ_i ; it is zero if the probability is monotonically decreasing.

The overapproximate abstraction of the set of models $\mathcal{M}(\theta)$ in the bounded parameter space $\theta \in [\Theta_1^{\min}, \Theta_1^{\max}] \times \dots \times [\Theta_{n_\Theta}^{\min}, \Theta_{n_\Theta}^{\max}]$ is given by the model $\mathcal{M}(\theta_{\min})$, where $\theta_{\min} = (m_1(\Theta_1^{\max} - \Theta_1^{\min}) + \Theta_1^{\min}, \dots, m_{n_\Theta}(\Theta_{n_\Theta}^{\max} - \Theta_{n_\Theta}^{\min}) + \Theta_{n_\Theta}^{\min})$. We also denote the overapproximate model $\mathcal{M}(\theta_{\min})$ by $\overline{\mathcal{M}}(S)$.

The algorithm first constructs two empty sets $\tilde{\mathcal{V}}_{\phi, \rho}$ and $\tilde{\mathcal{V}}_{\phi, \rho}^c$ containing the space of parameters that do and do not satisfy the specification, respectively. We also compute a coarse-grained discretization of the parameter space by dividing each parameter value into two parts. For each hyperbox formed, if the model satisfies the probabilistic adapted finitely monitorable logic formula with probability more than ρ for the underapproximate model $\underline{\mathcal{M}}$ in the parameter space defined by the hyperbox, then we add this hyperbox to the set $\tilde{\mathcal{V}}_{\phi, \rho}$ of parameter values satisfying the specification. The monotonicity of the parameter values (see Theorem 6) provide the technical justification for doing so. On the contrary, if the model satisfies the formula with probability less than ρ for the overapproximate model $\overline{\mathcal{M}}$ in the parameter space defined by the hyperbox, then we add this hyperbox to the set $\tilde{\mathcal{V}}_{\phi, \rho}^c$ of parameter values not satisfying the specification.

If a hyperbox is neither in the set of parameter values satisfying or not satisfying the PAFM specification, and is larger than the minimal threshold size dictated by the error tolerance of the PAFM specification η , we refine the hyperbox by splitting each parameter value into two parts and we continue with the algorithm. If the hyperbox has become smaller than the threshold size dictated by the error tolerance η , we stop analyzing this hyperbox any further.

Theorem 10. If θ is a point in the set of unknown hyperboxes U in Algorithm 2 with error tolerance η , then $\mathcal{M}(\theta) \models_q \Pr_{< \frac{\rho}{\eta}}(\phi)$, and $\mathcal{M}(\theta) \models_q \Pr_{> \rho\eta}(\phi)$.

Algorithm 2 Faster Parameter Synthesis using Abstraction Refinement

Require: Parameterized Biochemical Model $\mathcal{M}(\theta)$, Probabilistic Adapted Finitely Monitorable Formula $Pr_{\geq \rho}(\phi)$, Parameter Space $\theta \in [\Theta_1^{\min}, \Theta_1^{\max}] \times \dots \times [\Theta_{n_\theta}^{\min}, \Theta_{n_\theta}^{\max}]$, Error Tolerance in PAFM Formula $\sqrt{\rho} < \eta < 1$, Confidence Probability q .

Ensure: Set $\tilde{\mathcal{V}}_{\phi, \rho}$ of parameter values such that

- (i) $\forall \theta \in \tilde{\mathcal{V}}_{\phi, \rho}, \mathcal{M}(\theta) \models_q Pr_{\geq \rho}(\phi)$, and
- (ii) $\forall \theta \in \tilde{\mathcal{V}}_{\phi, \rho}^c, \mathcal{M}(\theta) \models_q Pr_{< \rho}(\phi)$.

$\tilde{\mathcal{V}}_{\phi, \rho} = \{\}, \tilde{\mathcal{V}}_{\phi, \rho}^c = \{\}$ {Initialize satisfying (unsatisfying) parameter values}

$\epsilon = \lfloor \log \eta \rfloor$ {Compute discretization constant ϵ from error tolerance η }

for all $i = 1$ to n_c **do**

$$\delta_i = \frac{C_i^{\max} - C_i^{\min}}{2}$$

end for

$U = [\log C_1^{\min}, \log C_1^{\max}]_{\delta_1} \times \dots \times [\log C_{n_c}^{\min}, \log C_{n_c}^{\max}]_{\delta_{n_c}}$ {Search discretized space}

for all $\theta_{\log} \in U$ **do**

$\{H(x)$ is a hyperbox with each side of length δ_i around $x\}$

if $\mathcal{M}(H(\theta_{\log})) \models_q Pr_{\geq \rho}(\phi)$ **then**

$\tilde{\mathcal{V}}_{\phi, \rho} = \tilde{\mathcal{V}}_{\phi, \rho} \cup \{\exp(H(\theta_{\log}))\}$ {Parameter value satisfies the spec. with probability at least ρ } {Add hyperbox around this parameter value to S .}

end if

if $\mathcal{M}(H(\theta_{\log})) \models_q Pr_{< \rho}(\phi)$ **then**

$\tilde{\mathcal{V}}_{\phi, \rho}^c = \tilde{\mathcal{V}}_{\phi, \rho}^c \cup \{\exp(H(\theta_{\log}))\}$ {Parameter value satisfies the spec. with probability less than ρ .} {Add hyperbox around this parameter value to S^c .}

end if

$U = U \setminus (\tilde{\mathcal{V}}_{\phi, \rho} \cup \tilde{\mathcal{V}}_{\phi, \rho}^c)$ {Update the unknown part of the parameter space.}

for all $i = 1$ to n_c **do**

$$\delta_i = \frac{\delta_i}{2}$$

end for

if $\delta_i < \delta(\epsilon/2, M)$ **then**

Report these hyperboxes as unknown. {Hyperbox Size is too small}

Break;

end if

$U = [\log C_1^{\min}, \log C_1^{\max}]_{\delta_1} \times \dots \times [\log C_{n_c}^{\min}, \log C_{n_c}^{\max}]_{\delta_{n_c}}$ {Further discretize the unknown part of the parameter space.}

end for

if $S == \{\}$ **then**

Print “Model is infeasible”.

end if

Proof. We will show that $\mathcal{M}(\theta) \models_q Pr_{< \frac{\rho}{\eta}}(\phi)$, and $\mathcal{M}(\theta) \models_q Pr_{> \rho\eta}(\phi)$ if $\theta \in U$.

- (i) Suppose θ satisfies $\mathcal{M}(\theta) \models_q Pr_{\geq \frac{\rho}{\eta}}(\phi)$. Consider the hyperbox $H(\theta)$ of size $\delta(\epsilon, M)$ around this point. By uniform continuity and our choice of discretization, every point c in the hyperbox (in particular, the corners of the hyperbox) satisfy $\mathcal{M}(c) \models_q Pr_{\geq \frac{\rho}{\eta}e^\epsilon}(\phi)$. But, by construction, we know that $e^\epsilon = \eta$. Hence, the corners satisfy the formula $Pr_{\geq \rho}(\phi)$ with confidence q . But, in that case, θ would be in $\tilde{\mathcal{V}}_{\phi, \rho}$ and not in U .
- (ii) Suppose that θ satisfies $\mathcal{M}(\theta) \models_q Pr_{\leq \rho\eta}(\phi)$. Consider the hyperbox $H(\theta)$ of size $\delta(\epsilon, M)$ around this point. By uniform continuity, every point c in the hyperbox (in particular, the corners of the hyperbox) satisfy $\mathcal{M}(c) \models_q Pr_{\leq \frac{\rho\eta}{e^\epsilon}}(\phi)$. But, by construction, we know that $e^\epsilon = \eta$. But, in that case, $\mathcal{M}(c) \models_q Pr_{\leq \rho}(\phi)$ and θ would be in $\tilde{\mathcal{V}}_{\phi, \rho}^c$ and not in U . \square

The theorem points out that the quality of the answer obtained by our abstraction refinement algorithm depends on the error tolerance parameter η . As η approaches one, the size of the set U with unknown parameter values becomes smaller.

7.3. Parameter search and model infeasibility using gradient descent

The previous two algorithms solve the parameter synthesis problem. In this section, we consider a slightly different problem, and find the parameter combination that maximizes the probability that the given formula will hold on the model, or reports that the model and the PAFM specification are mutually infeasible over the given parameter space. The algorithm takes the same inputs as the parameter synthesis algorithms. It begins by computing the smallest step size, δ , that will guarantee that the ratio of the probabilities associated with any two points inside any hyperbox of length δ will not exceed η . Then, the algorithm samples a random point in the parameter space and computes the gradient of the probability at this

point in the parameter space using the equations in [Theorem 4](#). The algorithm then moves by a step of δ in the direction of the gradient. If the algorithm crosses the parameter space to be searched, it stops. The algorithm checks if the last point sampled in the parameter space satisfies the adapted finitely monitorable property with probability at least ρ . If so, it reports this point in the parameter space as the best parameter that can be synthesized. Otherwise, it declares that the parameter space does not contain any parameter values that enable the model to η -robustly satisfy the PAFM specification i.e. there exists no parameter value θ such that $\mathcal{M}(\theta) \models_q Pr_{\geq \frac{\rho}{\eta}}(\phi)$.

Algorithm 3 Synthesis and Infeasibility Analysis using Gradient Descent

Require: Parameterized Biochemical Model $\mathcal{M}(\Theta)$,
 Probabilistic Adapted Finitely Monitorable Formula $Pr_{\geq \rho}(\phi)$,
 Parameter Space $\theta \in [\Theta_1^{\min}, \Theta_1^{\max}] \times \dots \times [\Theta_{n_\Theta}^{\min}, \Theta_{n_\Theta}^{\max}]$,
 Error Tolerance in PAFM Formula $\sqrt{\rho} < \eta < 1$,
 Confidence Probability q ,

Ensure: (i) A point θ_0 of parameter values such that $\mathcal{M}(\theta_0) \models_q Pr_{\geq \rho}(\phi)$, or
 (ii) Show that for all θ in the parameter space $\mathcal{M}(\theta) \not\models_q Pr_{\geq \frac{\rho}{\eta}}(\phi)$

{Initialize parameter value to a random point}
 $\theta_0 = \text{RandomPoint}([\Theta_1^{\min}, \Theta_1^{\max}] \times \dots \times [\Theta_{n_\Theta}^{\min}, \Theta_{n_\Theta}^{\max}])$
 {Compute ϵ and δ from η }
 $\epsilon = \lfloor \log \eta \rfloor$
 $\delta = \delta(\epsilon/2, \mathcal{M})$
 {Search the discretized parameter space}
while $\theta_0 \in [\log \Theta_1^{\min}, \log \Theta_1^{\max}]_\delta \times \dots \times [\log \Theta_{n_\Theta}^{\min}, \log \Theta_{n_\Theta}^{\max}]_\delta$ **do**

$\text{Gradient}(\theta_0) = \frac{\delta P}{\delta \Theta}$, where $\mathcal{M}(\exp(\theta_0)) \models_q Pr_{=P}(\phi)$

$\theta_0 = \theta_0 + \frac{\text{Gradient}(\theta_0)}{|\text{Gradient}(\theta_0)|} \delta$

end while

{If the parameter value satisfies the PAFM formula with probability ρ }
if $\mathcal{M}(\exp(\theta_0)) \models_q Pr_{\geq \rho}(\phi)$ **then**
 Print parameter $\exp(\theta_0)$. {Report this parameter value.}
 STOP.
end if

Print “Model is infeasible”.

Theorem 11. If the algorithm reports that the model is infeasible, then there does not exist any parameter value θ in the specified parameter space such that $\mathcal{M}(\theta) \models_q Pr_{\geq \frac{\rho}{\eta}}(\phi)$.

Proof. Suppose the algorithm reports that the model is infeasible and there is actually a point θ such that $\mathcal{M}(\theta) \models_q Pr_{\geq \frac{\rho}{\eta}}(\phi)$. Consider the $\delta(\epsilon, \mathcal{M})$ -neighborhood of θ . If our algorithm sampled a point θ' in this neighborhood, it would have found that $\mathcal{M}(\theta') \models_q Pr_{\geq \rho}(\phi)$ (from [Theorem 2](#)) and stopped. Hence, our algorithm must not have sampled a point in the $\delta(\epsilon, \mathcal{M})$ -neighborhood of θ . As the probability of a formula being true is monotonic over the parameter space, this is only possible if our algorithm sampled a parameter value with a higher probability that the probability associated with $\delta(\epsilon, \mathcal{M})$ -neighborhood of θ . Thus, the algorithm could not have reported that the model is infeasible, contradicting our assertion. \square

Lemma 2. If the algorithm reports that θ as the best synthesized parameter value with probability ρ_{syn} and θ_{max} is the actual parameter value that satisfies the specification with highest probability ρ_{max} , then $\rho_{\text{syn}} \geq \eta \rho_{\text{max}}$.

Proof. Suppose the algorithm reports ρ_{syn} such that $\rho_{\text{syn}} < \eta \rho_{\text{max}}$. But, θ is in the $\delta(\epsilon, \mathcal{M})$ -neighborhood of θ_{max} ; otherwise, the algorithm would not have stopped. By invoking uniform continuity (from [Theorem 2](#)), we have a proof by contradiction. \square

8. Experimental results

We analyzed stochastic models [30,31] of the Fibroblast Growth Factor and Cell Cycle biochemical signalling pathways against known behavioral specifications. We used a distributed computing cluster comprised of 20 nodes to run our experiments — each node had two Intel(R) Xeon(TM) processors with a clock speed of 3.06GHz and an on-chip cache size of 512 KB. Fibroblast growth factors (FGF) are a family of molecules involved in embryonic development, healing of wounds,

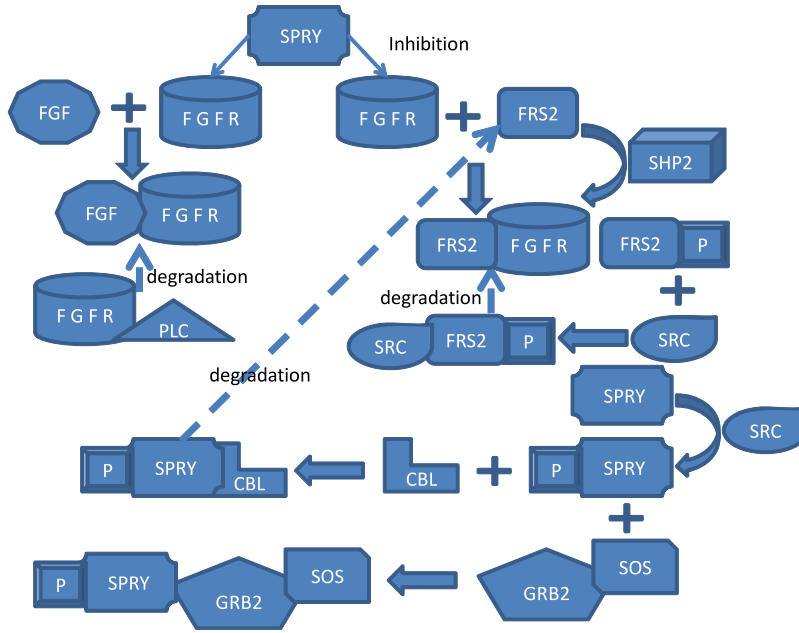


Fig. 3. Cartoon representation of the fibroblast growth factor receptor pathway.

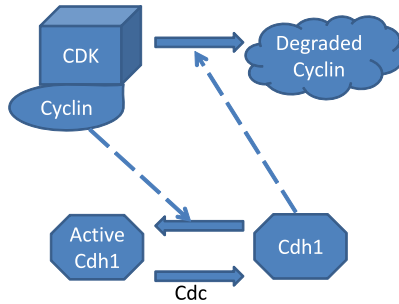


Fig. 4. Cartoon representation of the Cell Cycle Pathway. (Dashed arrows indicate catalysis/promotion of a reaction by a substrate.)

and the development of new blood vessels (angiogenesis). As FGFs control the growth and differentiation of cells and are involved in angiogenesis, perturbations of this pathway are relevant at various stages of the development of cancer.

The FGF model (Fig. 3) comprises 10 base species: (i) the FGF molecule; (ii) the FGF receptor (FGFR); (iii) a FGFR-specific substrate (FRS2); (iv) the phosphatase Shp2; (v-vi) the kinases PLC and Src; (vii) the inhibitor Spry; (viii) the ubiquitin ligase Cbl; (ix) the adaptor protein Grb; and (x) the exchange factor Sos. These base species can bind to form additional species (FGF-FGFR, FGFR-FRS2, Shp2-FRS2, Src-FRS2, Grb-FRS2, PLC-FGFR, Spry-Src, Spry-Cbl, Spry-Grb, Grb-Sos), or degrade. The phosphatase and kinases cause state changes in the species (i.e., dephosphorylation and phosphorylation, respectively). The binding of Grb to Sos is an important event in the MAPK/ERK pathway, which regulates translation and transcription in the cell. The FGF model can be downloaded from <http://www.prismmodelchecker.org/casestudies/examples/fgf.sm.php>.

We also studied a model of cell cycle control (Fig. 4) that consists of five species: (i) cyclin; (ii) cyclin-dependent protein kinases (CDKs); (iii) cyclin-dependent kinase inhibitor (CKI); (iv) the phosphatase Cdc; and (v) the tumor suppressor gene Cdh1. CDKs are activated by binding to cyclins, and they control DNA synthesis and chromosome condensation during the initial phases of cell division. CDK activity is regulated through multiple mechanisms including cyclin, CKIs, and phosphorylation. We studied the absence of bound cyclin in our properties and the influence of kinetic parameters on the binding of cyclin. The cell cycle model can be downloaded from <http://www.prismmodelchecker.org/casestudies/examples/cyclin.sm>.

8.1. Parameter synthesis using uniform continuity

We first performed experiments using the simplest of our algorithms discussed in Section 7.1. For the Fibroblast Growth Factor model, we perform parameter synthesis on a high level behavioral specification expressed in Probabilistic Bounded Metric Temporal Logic that was studied in [30,31]. The behavior concerns the probability that molecule Grb2 is bound to molecule FRS2 (denoted by $FRS2_GRB$), and that FRS2 is not degraded at the time instant T .

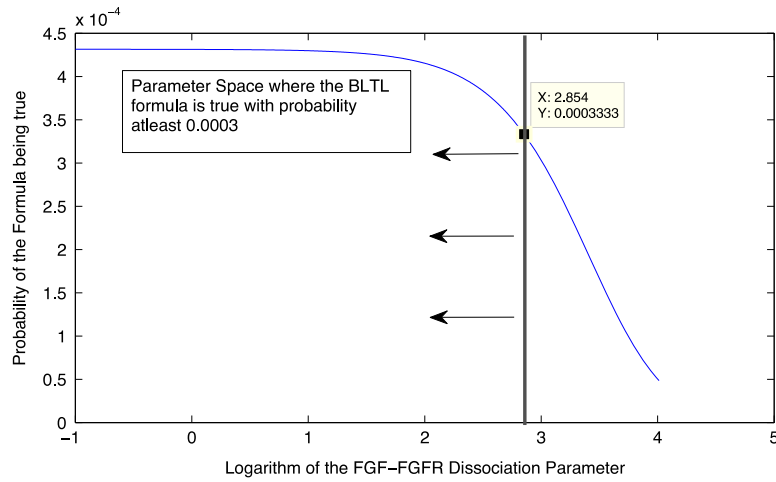


Fig. 5. Synthesized 1-D parameter space. (The parameter space to the left of the arrows satisfies the formula $Pr_{\geq 3.0 \times 10^{-4}} [\text{True } U^{[1,1]}(FRS2_GRB > 0 \ \& \ degFRS2 = 0)]$.)

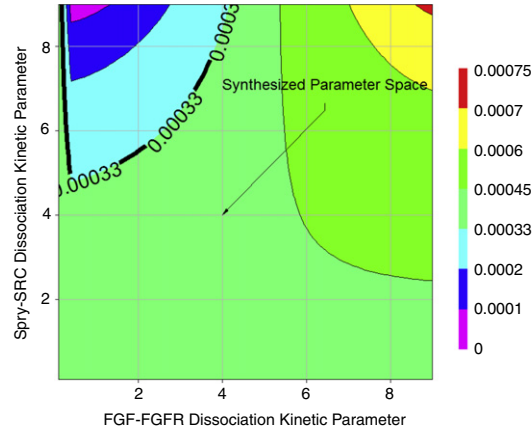


Fig. 6. Synthesized 2-D parameter space. (The synthesized parameter space lying to the right of the contour 0.00033 satisfies $Pr_{\geq 3.0 \times 10^{-4}} [\text{True } U^{[1,1]}(FRS2_GRB > 0 \ \& \ degFRS2 = 0)]$.)

In Fig. 5, we studied the influence of varying the FGF-FGFR dissociation parameter on the probability of the formula being true. We show the results of using our algorithms on the following formula:

$$Pr_{\geq 3.0 \times 10^{-4}} [\text{True } U^{[1,1]}(FRS2_GRB > 0 \ \& \ degFRS2 = 0)]$$

Our synthesized parameter space is correct with probability 0.99 (i.e., $q = 0.99$ in Algorithm 1). We are able to demonstrate that the PBMTL formula is true whenever the *logarithm* of the FGF-FGFR dissociation parameter lies between -1 s^{-1} and 2.854 s^{-1} .

We considered the problem of synthesizing two parameters simultaneously – the Spry-SRC and FGF-FGFR dissociation rates, using the PBLTL formula stated above. The results of our experiments are plotted in Fig. 6. The region of the plot to the right of the contour line denoting probability 0.00033 contains they only parameter values that enable the model to satisfy the PBLTL formula. Note that the reason the value is 0.00033, and not 0.0003, is because we chose 0.91 as our choice of η in Algorithm 1. Hence, the parameter space we can confidently state to satisfy the PAFM formula based on our discretized sampling algorithm is the one which satisfies the formula with probability at least $\frac{0.0003}{0.91} = 0.00033$.

We also analyzed the following PBMTL property:

$$Pr_{\geq 0.2} [\text{True } U^{[60,60]}(FRS2_GRB > 0 \ \& \ degFRS2 = 0)]$$

Note that the two differences between this and the previous property are the time and probability bounds. The results obtained by our analysis are shown in Fig. 7. The algorithm reports that the combination of parameters to the right of the contour labeled 0.22 satisfy the formula.

Next, we analyzed the Cell Cycle model using uniform continuity arguments against the following Probabilistic Bounded Linear Temporal Logic property:

$$Pr_{\geq 0.4} [\text{True } U^{60} \text{cyclin_bound} = 0]$$

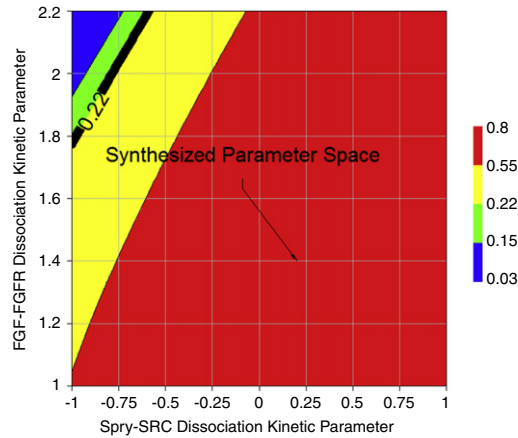


Fig. 7. Synthesized parameter space for fibroblast growth factor model. (The synthesized parameter space lying to the right of the contour 0.22 satisfies the formula $Pr_{\geq 0.2} [\text{True } \mathbf{U}^{[60,60]}(FRS2_GRB > 0 \ \& \ degFRS2 = 0)]$.)

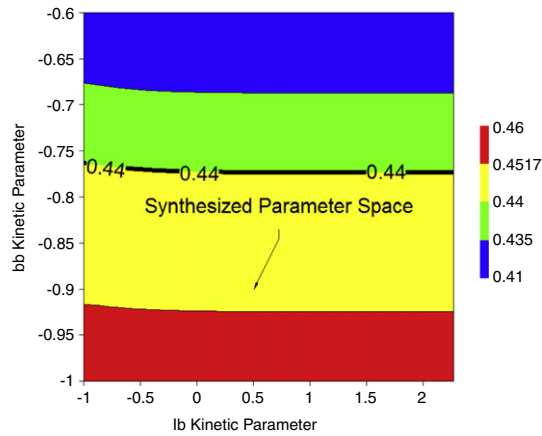


Fig. 8. Synthesized parameter space for the cell cycle model. (The synthesized parameter space lying below the contour 0.44 satisfies $Pr_{\geq 0.4} [\text{True } \mathbf{U}^{60}(\text{cyclin_bound} = 0)]$.)

The results of our analysis are presented in Fig. 8. The algorithm reports that the combination of parameters below the contour labeled 0.22 satisfy the formula.

Algorithm 1 is computationally quite expensive; it required about three days on a forty processor cluster to run our experiments for a two dimensional system. The search for algorithms with low computational requirements yields the abstraction refinement algorithm that we present in the next section. The abstraction refinement based synthesis algorithms are more scalable and we discuss a benchmark example with six dimensions.

8.2. Parameter synthesis using abstraction refinement

In this section, we report the results of using our abstraction refinement based algorithm (see Algorithm 2). We studied the performance of building abstractions using the FGF signal transduction pathway model (see Fig. 3) with as many as six parameters. We first asked our algorithm to synthesize the parameter where following statement is true:

$$Pr_{\geq 0.8} [\text{True } \mathbf{U}^{[60,60]}(FRS2_GRB > 0 \ \& \ degFRS2 = 0)]$$

Our implementation took 30.7 min and confirmed that it is infeasible for the model to satisfy the formula in the parameter range being searched. Naturally, we cannot visualize the surface in a 6-dimensional space, as we could for the 1 and 2 dimensional cases. We note that our uniform continuity based algorithms would not be able to answer this question within a reasonable amount of time. For example, assuming a discretization of 0.1 in the logarithmic parameter space, it would take over 500 years for the proof to be completed. Thus, monotonicity is a really important property for tackling problems in high dimensions.

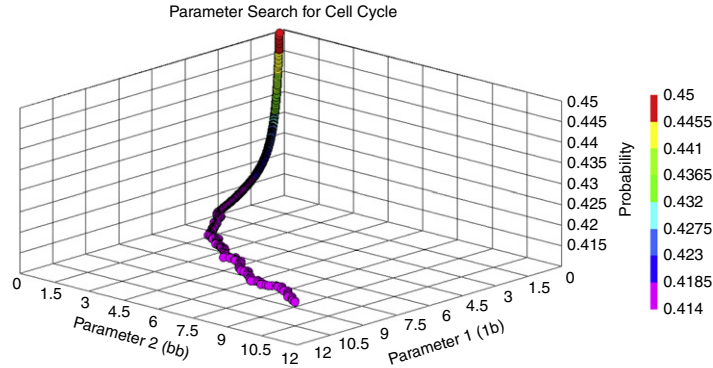
In Table 1, we used the following property for synthesizing the parameter space:

$$Pr_{\geq 0.15} [\text{True } \mathbf{U}^{[1,1]}(FRS2_GRB > 0 \ \& \ degFRS2 = 0)]$$

Table 1

Abstraction Refinement: Parameter Synthesis for probability above 0.15. (A * next to a value means that it was estimated by extrapolating from the run-times for problems with a smaller number of hyperboxes. For example, the 13.9 days estimate for 262,144 hyperboxes was obtained by extrapolating from the 9.8 h it took for 4096 hyperboxes.

#Hyperboxes (exhaustive search)	Time (exhaustive search)	#Hyperboxes (Algorithm 2)	Time (Algorithm 2)
64	27.46 min	64	27.46 min
4,096	9.8 h	640	1.55 h
262,144	13.9 days*	15,744	20.1 h
16,777,216	1.7 years*	<1,007,616*	<38 days*

**Fig. 9.** Parameter search in action for the cell cycle model.

Note that the number of hyperboxes to be explored and the time taken for the exhaustive search columns in Table 1 provide a lower bound for the time taken by Algorithm 1. Here, we just studied the performance of guided refinement on various abstractions with 64, 4096 and 262,144 hyperboxes. In each of these cases, a naive algorithm would need to analyze all of these hyperboxes by refining them into even smaller hyperboxes. We found that an analysis of the 64 hyperboxes formed at the first step showed that only 10 of them needed to be refined and further analyzed. While the analysis of 4096 boxes takes 9.8 h, the analysis of the 10 well-chosen boxes (by exploring 640 hyperboxes) takes about one and a half hours. The savings become much more impressive as we refine the size of the hyperbox and increase the number of hyperboxes. If we were to analyze all of the 262,144 boxes, we *estimate* (via extrapolation) that it would take over 13.9 days. On the other hand, if we use abstraction refinement based on the monotonicity argument, we only need to analyze at most 15,744 hyperboxes and that takes about 20 h. We further estimate that the exhaustive analysis of all 16,777,216 smaller hyperboxes would take about 1.7 years, but that the refinement algorithm would take no more than 38 days.

When abstractions can be built over a space of models (as opposed to over parameters), abstraction refinement is often an efficient technique to employ. This *may* lead to considerable savings in many cases. On the other hand, one can cleverly construct cases where abstraction refinement may have to work as hard as the original analysis algorithm. However, this is not possible in our case, because we are performing abstractions over the parameters. The monotonicity of the parameter space ensures that abstraction refinement will always run faster in synthesizing kinetic parameters of stochastic biochemical systems.

8.3. Parameter estimation using gradient descent

Gradient descent based algorithms for synthesizing single parameter values are truly scalable to high dimensions as they do not need to do a search (exhaustive or otherwise) of the entire probability space. We applied our gradient descent algorithm to the problem of synthesizing the lb and bb kinetic parameters for the Cell Cycle model. We wanted to find a parameter value that satisfies the following Probabilistic Bounded Metric Temporal Logic specification:

$$Pr_{\geq 0.45} [\text{True } U^{[60,60]} \text{cyclin_bound} = 0]$$

The results of our algorithm are plotted in Fig. 9. The algorithm suggests the parameter tuple (0.349, 0.124) as a parameter value that satisfies the specification. The algorithm took only 21 min to produce this result. We further ran our algorithm to find the maximal value in the parameter space between 0.01 and 200 for both the parameters. Our algorithm reported that the maximum value of the probability is 0.539 and lies at the point (0.0101, 0.0101) which is the closest point we sampled to one of the corners of the given parameter space.

Our cell cycle model has 11 kinetic parameters that can be varied. We considered the parameter synthesis problem involving all of the eleven parameters with the aim of satisfying the following formula:

$$Pr_{\leq 0.4} [\text{True } U^{[60,60]} \text{cyclin_bound} = 0]$$

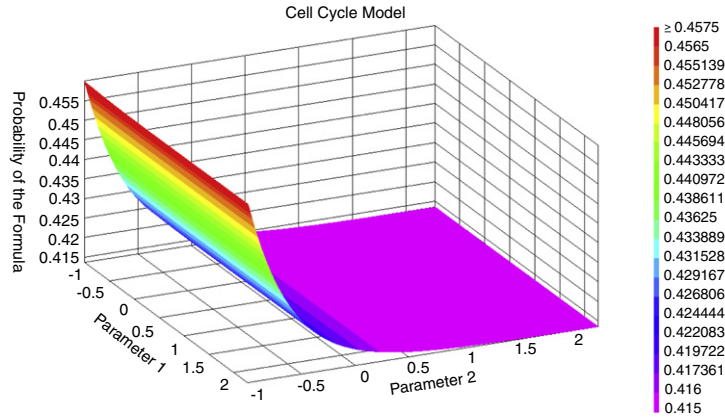


Fig. 10. Probability of the formula $[\text{True } U^{60}_{\text{cyclin_bound}} = 0]$ for the Cell Cycle model. Note that the variation along parameter 1(lb) is much smaller than variation along parameter 2(bb).

We restricted the search space for the kinetic parameters to between 0.001 and 1. Our binary search based implementation of Algorithm 3 took 3.4 h to report the parameter value (0.112512, 0.5005, 0.5005, 0.5005, 0.5005, 0.5005, 0.165856, 0.5005, 0.5005, 0.5005) that satisfies the formula with probability 0.3986.

We then searched for a parameter that satisfies the following formula:

$$Pr_{\geq 0.99} [\text{True } U^{[60,60]}_{\text{cyclin_bound}} = 0]$$

Our implementation took 52.3 min to produce the following parameter values that satisfy the property with probability exceeding 0.9902: (0.5005, 0.5005, 0.5005, 0.5005, 0.5005, 0.326276, 0.5005, 0.5005, 0.5005, 0.5005, 0.5005). We also asked our algorithm to find the maximum value of the parameter possible and we obtained 0.9999 as the answer within the probability space we were searching.

9. Conclusion and future work

We have introduced new algorithms to synthesize kinetic parameter values that enable a biochemical model to satisfy a *Probabilistic Adapted Finitely Monitorable Logic* specification. The specification captures the biological knowledge that is known about the biochemical system being modeled. We applied our algorithms to two benchmark models from the literature and demonstrated that our abstraction-refinement based algorithm is capable of synthesizing six parameters simultaneously. To the best of our knowledge, this is the largest number of parameters that have been synthesized at once for stochastic models against a given formula. Moreover, our gradient search algorithm is capable of finding the single parameter combination that maximizes the probability of the formula being true over 11 parameters simultaneously. We are also developing a tool to synthesize parameters of stochastic biochemical models from behavioral descriptions.

Our algorithms can also demonstrate the infeasibility of a model with respect to a high level behavioral specification in a given parameter space. This is a very useful debugging tool for biochemical modeling, and ensures that the modeler does not waste her time searching for a non-existent parameter combinations. The gradient search algorithm is capable of carrying out infeasibility analysis in very high dimensional parameter spaces.

The most important contribution of the paper is a new technique for constructing proofs about stochastic biochemical models based on survey sampling. Parameterized biochemical models give rise to families of CTMCs, each of which has a different probability measure over the set of possible paths. The key point, however, is that the set of possible paths is the same. Thus, survey sampling provides a natural framework to develop analytic arguments about biochemical models. We believe that this technique for developing proofs may be important in areas beyond model checking and parameter synthesis. Particularly, the formal sensitivity analysis of τ -leaping methods [10,49,2] may be susceptible to an analysis along these lines. We are currently investigating this direction along with our collaborators.

Several interesting directions for future work remain open. Our results and constructive proofs can be extended to study the impact of variation in different parameters on the probability of an adapted finitely monitorable formula being true on a model. Such an algorithmic sensitivity analysis of the probability of a formula being true with respect to the various parameters can then be used as a *preprocessing step* to guide parameter search algorithms in very high dimensional parameter spaces. Consider Fig. 10 which shows the variation in probability as we vary the lb and bb parameters of the cell cycle model. It is clear that variation along one of the parameters impacts the model much more than along the other dimension. Such a preliminary sensitivity analysis could be used to pick a small number of parameters out of a large number of unknown parameters before parameter synthesis is attempted on the smaller parameter set. We also note that the proof of Lemma 1 provides a framework of computing sensitivity using statistical sampling and can be developed into an efficient sensitivity analysis algorithm for stochastic biochemical models.

Another interesting direction for future work is to develop a monitoring framework for stochastic biochemical systems. It is unrealistic to assume that biochemists will ever translate their knowledge into fragments of temporal logics. On the other hand, MATLAB and even high level languages like C are now standard in many undergraduate programs across the world. It is important to construct a suitable framework for developing monitors in these languages that can be used directly by biochemists. While some work has been done on automatically mining formal knowledge bases or ontologies from biochemical literature, the use of these ontologies to validate models is limited at best. It would be interesting to bridge the gap and provide formal methods based tools to verify models against existing ontologies.

Finally, we note that existing stochastic biochemical models are often hard-wired with a “best-guess” value of the parameters that makes the model “work”. Unfortunately, when different models are combined in a modular fashion (e.g., combining models of different pathways), those models that “work” in isolation might not work together as components of the larger system. These inconsistencies will come into focus as the science of Systems Biology matures, and we begin putting models built by different scientists together. An interesting question that we are investigating is the *re-synthesis* of parameter regimes when putting models together as components.

References

- [1] R. Alur, C. Courcoubetis, N. Halbwachs, T.A. Henzinger, P.-H. Ho, X. Nicollin, A. Olivero, J. Sifakis, S. Yovine, The algorithmic analysis of hybrid systems, *Theoret. Comput. Sci.* 138 (1) (1995) 3–34.
- [2] D.F. Anderson, A. Ganguly, T.G. Kurtz, Error analysis of tau-leap simulation methods, 2009. URL: <http://arxiv.org/abs/0909.4790>.
- [3] A. Annichini, E. Asarin, A. Bouajjani, Symbolic techniques for parametric reasoning about counter and clock systems, in: E.A. Emerson, A.P. Sistla (Eds.), CAV, in: *Lecture Notes in Computer Science*, vol. 1855, Springer, 2000, pp. 419–434.
- [4] E. Asarin, T. Dang, A. Girard, Hybridization methods for the analysis of nonlinear systems, *Acta Inform.* 43 (7) (2007) 451–476.
- [5] C. Baier, E.M. Clarke, V. Hartonas-Garmhausen, M.Z. Kwiatkowska, M. Ryan, Symbolic model checking for probabilistic processes, in: P. Degano, R. Gorrieri, A. Marchetti-Spaccamela (Eds.), ICALP, in: *Lecture Notes in Computer Science*, vol. 1256, Springer, 1997, pp. 430–440.
- [6] C. Baier, B.R. Haverkort, H. Hermanns, J.-P. Katoen, Model-checking algorithms for continuous-time Markov chains, *IEEE Trans. Softw. Eng.* 29 (6) (2003) 524–541.
- [7] G. Batt, B. Yordanov, R. Weiss, C. Belta, Robustness analysis and tuning of synthetic gene networks, *Bioinformatics* 23 (18) (2007) 2415–2422. URL: <http://bioinformatics.oxfordjournals.org/cgi/content/abstract/23/18/2415>.
- [8] A. Bemporad, A. Bicchi, G.C. Buttazzo (Eds.), Hybrid systems: computation and control, in: 10th International Workshop, HSCC 2007, Pisa, Italy, April 3–5, 2007, Proceedings, in: *Lecture Notes in Computer Science*, vol. 4416, Springer, 2007.
- [9] L. Calzone, F. Fages, S. Soliman, Biocham: an environment for modeling biological systems and formalizing experimental knowledge, *Bioinformatics* 22 (14) (2006) 1805–1807.
- [10] Y. Cao, D.T. Gillespie, L.R. Petzold, Adaptive explicit-implicit tau-leaping method with automatic tau selection, *J. Chem. Phys.* 126 (22) (2007) 224101. URL: <http://link.aip.org/link/JCP/126/224101/1>.
- [11] N. Chabrier, F. Fages, Symbolic model checking of biochemical networks, in: C. Priami (Ed.), CMSB, in: *Lecture Notes in Computer Science*, vol. 2602, Springer, 2003, pp. 149–162.
- [12] F. Ciesinski, C. Baier, Liqueur: a tool for qualitative and quantitative linear time analysis of reactive systems, in: QEST, IEEE Computer Society, 2006, pp. 131–132.
- [13] E.M. Clarke, J.R. Faeder, C.J. Langmead, L.A. Harris, S.K. Jha, A. Legay, Statistical model checking in biolab: Applications to the automated analysis of t-cell receptor signaling pathway, in: M. Heiner, A.M. Uhrmacher (Eds.), CMSB, in: *Lecture Notes in Computer Science*, vol. 5307, Springer, 2008, pp. 231–250.
- [14] E.M. Clarke, O. Grumberg, S. Jha, Y. Lu, H. Veith, Counterexample-guided abstraction refinement, in: E.A. Emerson, A.P. Sistla (Eds.), CAV, in: *Lecture Notes in Computer Science*, vol. 1855, Springer, 2000, pp. 154–169. URL: <http://dblp.uni-trier.de/db/conf/cav/cav2000.html#ClarkeGJV00>.
- [15] B.C. Daniels, Y.-J. Chen, J.P. Sethna, R.N. Gutenkunst, C.R. Myers, Sloppiness, robustness, and evolvability in systems biology, 2008. URL: <http://arxiv.org/abs/0805.2628>.
- [16] R. Donaldson, D. Gilbert, A model checking approach to the parameter estimation of biochemical pathways, in: CMSB'08: Proceedings of the 6th International Conference on Computational Methods in Systems Biology, Springer-Verlag, Berlin, Heidelberg, 2008, pp. 269–287.
- [17] A. Donzé, G. Clermont, C.J. Langmead, Parameter synthesis in nonlinear dynamical systems: application to systems biology, in: S. Batzoglou (Ed.), RECOMB, in: *Lecture Notes in Computer Science*, vol. 5541, Springer, 2009, pp. 155–169.
- [18] A. Donzé, G. Clermont, C.J. Langmead, Parameter synthesis in nonlinear dynamical systems: application to systems biology, *J. Comput. Biol.* 17 (3) (2010) 325–336.
- [19] A. Donzé, O. Maler, Systematic simulation using sensitivity analysis, in: [8], 2007, pp. 174–189.
- [20] S. Drulhe, G. Ferrari-Trecate, H. de Jong, A. Viari, Reconstruction of switching thresholds in piecewise-affine models of genetic regulatory networks, in: J.P. Hespanha, A. Tiwari (Eds.), HSCC, in: *Lecture Notes in Computer Science*, vol. 3927, Springer, 2006, pp. 184–199.
- [21] D. Drusinsky, M. Tak Shing, Monitoring temporal logic specifications combined with time series constraints, *J. UCS* 9 (11) (2003) 1261–1276.
- [22] B. Finkbeiner, H. Sipma, Checking finite traces using alternating automata, *Electronic Notes Theor. Comput. Sci.* 55 (2) (2001).
- [23] G. Frehse, S.K. Jha, B.H. Krogh, A counterexample-guided approach to parameter synthesis for linear hybrid automata, in: M. Egerstedt, B. Mishra (Eds.), HSCC, in: *Lecture Notes in Computer Science*, vol. 4981, Springer, 2008, pp. 187–200.
- [24] R. Ghosh, A. Tiwari, C. Tomlin, Automated symbolic reachability analysis; with application to delta-notch signaling automata, in: [44], 2003, pp. 233–248, 233–248.
- [25] D.T. Gillespie, Exact stochastic simulation of coupled chemical reactions, *J. Phys. Chem.* 81 (25) (1977) 2340–2361.
- [26] D.T. Gillespie, Stochastic simulation of chemical kinetics, *Annual Rev. Phys. Chem.* 58 (1) (2007) 35–55. doi:10.1146/annurev.physchem.58.032806.104637.
- [27] R. Grosu, S.A. Smolka, Monte carlo model checking, in: N. Halbwachs, L.D. Zuck (Eds.), TACAS, in: *Lecture Notes in Computer Science*, vol. 3440, Springer, 2005, pp. 271–286.
- [28] R. Gunawan, Y. Cao, L. Petzold, F.J. Doyle, Sensitivity analysis of discrete stochastic systems, *Biophys. J.* 88 (4) (2005) 2530–2540. doi:10.1529/biophysj.104.053405.
- [29] T. Han, J.P. Katoen, A. Mereacre, Approximate parameter synthesis for probabilistic time-bounded reachability, in: Proceedings of the IEEE Real-Time Systems Symposium, RTSS 2008, Barcelona, Spain, IEEE Computer Society Press, Los Alamitos, 2008, pp. 173–182.
- [30] J. Heath, M. Kwiatkowska, G. Norman, D. Parker, O. Tymchyshyn, Probabilistic model checking of complex biological pathways, in: C. Priami (Ed.), Proc. Computational Methods in Systems Biology, CMSB'06, in: *Lecture Notes in Bioinformatics*, vol. 4210, Springer Verlag, 2006, pp. 32–47.
- [31] J. Heath, M. Kwiatkowska, G. Norman, D. Parker, O. Tymchyshyn, Probabilistic model checking of complex biological pathways, *Theoret. Comput. Sci.* 319 (3) (2008) 239–257.
- [32] T. Héault, R. Lassaigne, F. Magniette, S. Peyronnet, Approximate probabilistic model checking, in: Proc. 5th International Conference on Verification, Model Checking and Abstract Interpretation, VMCAI'04, in: LNCS, vol. 2937, Springer, 2004.

- [33] H. Jeffreys, *Theory of probability* by Harold Jeffreys, 3rd edition, Clarendon Press, Oxford, 1961.
- [34] S. Jha, B.A. Brady, S.A. Seshia, Symbolic reachability analysis of lazy linear hybrid automata, in: J.-F. Raskin, P.S. Thiagarajan (Eds.), *FORMATS*, in: *Lecture Notes in Computer Science*, vol. 4763, Springer, 2007, pp. 241–256. URL: <http://dblp.uni-trier.de/db/conf/formats/formats2007.html#JhaBS07>.
- [35] S.K. Jha, E.M. Clarke, C.J. Langmead, A. Legay, A. Platzer, P. Zuliani, A Bayesian approach to model checking biological systems, Tech. Rep. CMU-CS-09-110, Computer Science Department, Carnegie Mellon University, 2009.
- [36] S.K. Jha, E.M. Clarke, C.J. Langmead, A. Legay, A. Platzer, P. Zuliani, A Bayesian approach to model checking biological systems, in: P. Degano, R. Gorrieri (Eds.), *CMSB*, in: *Lecture Notes in Computer Science*, vol. 5688, Springer, 2009, pp. 218–234.
- [37] S.K. Jha, B.H. Krogh, J.E. Weimer, E.M. Clarke, Reachability for linear hybrid automata using iterative relaxation abstraction, in: [8], 2007, pp. 287–300.
- [38] S. Julier, J. Uhlmann, A new extension of the kalman filter to nonlinear systems, in: *Int. Symp. Aerospace/Defense Sensing, Simul. and Controls*, Orlando, FL, 1997. URL: <http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.5.2891>.
- [39] R.E. Kalman, A new approach to linear filtering and prediction problems, *Trans. ASME* (1960) 35–45.
- [40] J.H. Kotecha, P.M. Djuric, Gaussian particle filtering, in: *Proceedings of the 11th IEEE Signal Processing Workshop on Statistical Signal Processing*, 2001, pp. 429–432. URL: [doi:10.1109/SSP.2001.955314](http://doi.org/10.1109/SSP.2001.955314).
- [41] R. Koymans, Specifying real-time properties with metric temporal logic, *Real-Time Syst.* 2 (4) (1990) 255–299.
- [42] M.Z. Kwiatkowska, G. Norman, D. Parker, Prism 2.0: A tool for probabilistic model checking, in: *QEST*, IEEE Computer Society, 2004, pp. 322–323.
- [43] C. Langmead, Generalized Queries and Bayesian Statistical Model Checking in Dynamic Bayesian Networks: Application to Personalized Medicine, in: *Proc. of the 8th International Conference on Computational Systems Bioinformatics (CSB)*, 2009, pp. 201–212.
- [44] O. Maler, A. Pnueli (Eds.), Hybrid systems: computation and control, in: *6th International Workshop, HSCC 2003 Prague, Czech Republic, April 3–5, 2003, Proceedings*, in: *Lecture Notes in Computer Science*, vol. 2623, Springer, 2003.
- [45] I. Mitchell, C. Tomlin, Level set methods for computation in hybrid systems, in: N.A. Lynch, B.H. Krogh (Eds.), *HSCC*, in: *Lecture Notes in Computer Science*, vol. 1790, Springer, 2000, pp. 310–323.
- [46] S.S. Owicki, L. Lamport, Proving liveness properties of concurrent programs, *ACM Trans. Program. Lang. Syst.* 4 (3) (1982) 455–495.
- [47] A. Pnueli, The temporal logic of programs, in: *FOCS*, IEEE, 1977, pp. 46–57.
- [48] M. Quach, N. Brunel, F. d'Alché Buc, Estimating parameters and hidden variables in non-linear state-space models based on odes for biological networks inference, *Bioinformatics* 23 (23) (2007) 3209–3216.
- [49] M. Rathinam, L.R. Petzold, Y. Cao, D.T. Gillespie, Stiffness in stochastic chemically reacting systems: The implicit tau-leaping method, *J. Chem. Phys.* 119 (24) (2003) 12784–12794. doi:10.1063/1.1627296.
- [50] A. Rizk, G. Batt, F. Fages, S. Soliman, On a continuous degree of satisfaction of temporal logic formulae with applications to systems biology, in: *IN: CMSB'08: Proceedings of the 6th International Conference on Computational Methods in Systems Biology*, Springer-Verlag, Berlin, Heidelberg, 2008, pp. 251–268.
- [51] K. Sen, M. Viswanathan, G. Agha, Statistical model checking of black-box probabilistic systems, in: R. Alur, D. Peled (Eds.), *CAV*, in: *Lecture Notes in Computer Science*, vol. 3114, Springer, 2004, pp. 202–215.
- [52] K. Sen, M. Viswanathan, G. Agha, On statistical model checking of stochastic systems, in: K. Etessami, S.K. Rajamani (Eds.), *CAV*, in: *Lecture Notes in Computer Science*, vol. 3576, Springer, 2005, pp. 266–280.
- [53] D.E. Shaw, M.M. Deneroff, R.O. Dror, J.S. Kuskin, R.H. Larson, J.K. Salmon, C. Young, B. Batson, K.J. Bowers, J.C. Chao, M.P. Eastwood, J. Gagliardo, J.P. Grossman, C.R. Ho, D.J. Ierardi, I. Kolossváry, J.L. Klepeis, T. Layman, C. McLeavey, M.A. Moraes, R. Mueller, E.C. Priest, Y. Shan, J. Spengler, M. Theobald, B. Towles, S.C. Wang, Anton, a special-purpose machine for molecular dynamics simulation, in: *ISCA'07: Proceedings of the 34th Annual International Symposium on Computer Architecture*, ACM, New York, NY, USA, 2007, pp. 1–12.
- [54] O. Stursberg, B.H. Krogh, Efficient representation and computation of reachable sets for hybrid systems, in: [44], 2003, pp. 482–497.
- [55] T.A. Henzinger, B. Horowitz, R. Majumdar, H. Wong-Toi, Beyond hytech: hybrid systems analysis using interval numerical methods, in: *HSCC*, in: *Lecture Notes in Computer Science*, Springer, 2000, pp. 130–144.
- [56] M. Taiji, T. Narumi, Y. Ohno, N. Futatsugi, A. Suenaga, N. Takada, A. Konagaya, Protein explorer: A petaflops special-purpose computer system for molecular dynamics simulations, in: *SC'03: Proceedings of the 2003 ACM/IEEE Conference on Supercomputing*, IEEE Computer Society, Washington, DC, USA, 2003, p. 15.
- [57] P. Thati, G. Rosu, Monitoring algorithms for metric temporal logic specifications, *Electronic Notes Theor. Comput. Sci.* 113 (2005) 145–162.
- [58] R. van der Merwe, A. Doucet, N. de Freitas, E.A. Wan, The unscented particle filter, in: T.K. Leen, T.G. Dietterich, V. Tresp (Eds.), *NIPS*, MIT Press, 2000, pp. 584–590.
- [59] A. Wald, Sequential tests of statistical hypotheses, *Ann. Math. Statist.* 16 (2) (1945) 117–186. doi:10.1214/2Faoms%2F1177731118.
- [60] A. Wald, *Sequential Analysis*, John Wiley and Son, New York, 1947.
- [61] H.L.S. Younes, Verification and planning for stochastic processes with asynchronous events. Ph.D. Thesis, Pittsburgh, PA, USA, chair-Reid G. Simmons, 2004.
- [62] H.L.S. Younes, M.Z. Kwiatkowska, G. Norman, D. Parker, Numerical vs. statistical probabilistic model checking, *STTT* 8 (3) (2006) 216–228.
- [63] H.L.S. Younes, R.G. Simmons, Probabilistic verification of discrete event systems using acceptance sampling, in: E. Brinksma, K.G. Larsen (Eds.), *CAV*, in: *Lecture Notes in Computer Science*, vol. 2404, Springer, 2002, pp. 223–235.
- [64] H.L.S. Younes, R.G. Simmons, Statistical probabilistic model checking with a focus on time-bounded properties, *Inf. Comput.* 204 (9) (2006) 1368–1409.