Parameter Estimation of Rule-based Models Using Statistical Model Checking

Bing Liu and James R. Faeder
Department of Computational & Systems Biology
School of Medicine, University of Pittsburgh
Pittsburgh, PA 15213, U.S.A.
Email: {liubing,faeder}@pitt.edu

***Abstract*—Rule-based modeling with BioNetGen has been widely used to study the dynamics of complex biochemical systems. Rule-based models can be analyzed by carrying out deterministic or stochastic simulations. However, they are often difficult to calibrate due to many unknown parameters and limited experimental training data. Here we present a generic parameter estimation framework for calibrating rule-based models. The experimental data as well as qualitative properties of the system are encoded as a specification formula in a bounded linear temporal logic. Given a candidate set of parameter values we apply the statistical model checking procedure to evaluate the quality of this candidate set. Based on the outcome, a new set of parameters is chosen using a standard global search strategy. We have tested our method on a p53-induced apoptosis model. The results show that our method scales well and efficiently obtains good parameter estimates.**

***Keywords*—systems biology; rule-based modeling; parameter estimation; statistical model checking; BioNetGen**

I. Introduction

Cellular processes are driven by networks of biochemical reactions. The dynamics of these networks play a crucial role in shaping cellular functions. A better understanding of the dynamics of the particular networks targeted by drugs in cells and tissues is required for contemporary drug discovery and pharmacology [1]. Systems biology approaches have been applied to identify new drug targets and new uses of known drugs, and to understand drug side effects in the context of cellular networks [2]. In particular, computational modeling has been recognized as a key approach to gain mechanistic insights of biological systems [3], [4] and to optimize therapeutic strategies [5].

A variety of mathematical formalisms have been proposed to describe the dynamics of biological systems [6]. Ordinary differential equations (ODEs) are often used to model systems, in which molecular species are abundantly available and form a well-stirred mixture in a fixed volume [7]. When the concentrations of species are low, the variability of reaction processes may influence the system's behavior significantly. Stochastic models (e.g. continuous time Markov chain (CTMC)) are then required to capture probabilistic outcomes of the system [8].

A biochemical reaction can be viewed as a *rule*, which specifies how (and how fast) the states of reactants are modified to generate products. Thus, a network of biochemical reactions can be naturally described using a set of rules over the molecules (e.g. proteins), where each molecule has a number of internal states (e.g. the status of post-translational modifications or bindings with other molecules). Formal languages such as BioNetGen [9] and κ [10] have been proposed to quantitatively specify molecules and rules to model biological systems, and this approach is called *rule-based modeling*. A rule-based model is a generalized and compact representation of conventional models such as ODE and CTMC based models. The rules that express a high degree of modularity avoid the explicit enumeration of all possible molecular species or all the states of a system. They also enable a more natural starting point for model development than making *ad hoc* assumptions to decide the model scope [11]. In this paper, we shall focus on one major rule-based modeling approach namely the BioNetGen [9]. The BioNetGen language (BNGL) has been used to model a variety of biological systems [12], [13]. Software tools have been developed to construct, visualize, simulate, and analyze BNGL models [14], [15], [16], [17].

A BNGL model involves kinetic parameters including rate constants and initial conditions. A major bottleneck for quantitative modeling is that in general, only a few of these parameter values will be available or can be measured experimentally, and the rest must be estimated. Parameter estimation is a basic but difficult step in the model construction process. A standard approach to this problem consists of iteratively searching through the space of parameter values and in each round computing the fitness of current parameter values to experimental data. The data will typically consist of time series measurements for a set of proteins observed at a small number of time points. It will be of limited precision and often averaged over a population of cells.

A BNGL model can be interpreted as an ODE model or a stochastic model. The parameter estimation for ODE models can resort to conventional SBML-compactible tools such as COPASI [18], while the one for stochastic models can be carried out using BioNetFit [19] that supports network-free simulation [17], or tools such as ABC-SysBio [20] that implement Gillespie?s stochastic simulation algorithms (SSA). These tools only make use of quantitative data (e.g. time series measurements) to constrain the parameter search space. A standard modeling workflow involves a model validation step, which requires reserving a portion of data as a test set to avoid the overfitting problem. However, in practice, the available

quantitative measurements are often inadequate considering the number of unknown parameters. Consequently, a model can be *non-identifiable* [6]. To cope with this, we recently proposed a statistical model checking (SMC)-based method for calibrating ODE-based models with prior distributions of initial states [21]. This method allows us to estimate parameters by utilizing not only quantitative measurements but also prior knowledge concerning qualitative properties of the systems. The enriched training and test data sets further constrain the parameter search space and may lead to better parameter estimates.

In this paper, we extend our SMC approach and present a parameter estimation method for generic BNGL models. We define a linear temporal logic BLTL to describe the experimental time series data and also encode prior knowledge about the qualitative behavior of the system. We use a sequential hypothesis test known as SPRT [22] to determine the number of trajectories that are to be generated in each round of the parameter estimation while at the same time lending statistical weight to the score assigned to the current set of parameters values. We then use a online model checker to solve the parameter estimation problem. We recall that the goal is to compute the values of unknown parameters so that the resulting model can reproduce experimental observations [23]. A common approach is to formulate this as a non-linear optimization problem, and solve it by iterating two major steps: (i) "guess" the values of the parameters (ii) evaluate the goodness-of-fit of the guessed values. For step (i), guesses may be generated randomly in the first round but later guesses are guided by the goodness-of-fit results of previous rounds according to the chosen search strategy. We use an evolutionary strategy [24] to implement step (i) and deploy our statistical model checking procedure to implement step (ii). BNGL trajectories can be generated using the ODE solver, SSA, NFsim, or hybrid particle/population (HPP) algorithms [25]. In this paper, we focus on demonstrating our method using SSA trajectories. We tested our method on a large p53-induced apoptosis model [8]. The parameter estimation results demonstrate that our SMC-based method is efficient and useful.

A. Related Work

The BioLab algorithm has been developed to verify the temporal properties of BioNetGen models [26]. Its probabilistic BLTL does not allow to encode experimental data since it can not specify that a certain property will hold exactly at time t from now. Further, it uses an offline model checker which is inefficient comparing to our online model checker when dealing with a large number of trajectories. There have been previous attempts to synthesize parameters for quantitative models using model checking methods [27], [28], [29], [30], [31]. Comparing to our method, these methods lack a systematic search method and do not have an automatic way to incorporate qualitative prior knowledge. Our previous work [21] was focused on ODE systems with a prior distribution of initial concentrations. We required that the vector fields defined by the ODE system will be a C^1 (continuously differentiable) function, in order to defined a probability measure to the set of trajectories that satisfy a given specification formula. Here we extend our method by focusing on stochastic rule-based models. We relax the C^1 function assumption since it has been proven that a probability measure can be assigned to the set of CTMC trajectories that satisfy a given specification formula [26].

B. Organization

The next section introduces rule-based models in BioNetGen language. In Section 3, we discuss our specification logic and the statistical model checking procedure. In the subsequent section we present our parameter estimation method based on statistical model checking. We present our experimental results in Section 5. We conclude with a discussion on future research directions.

II. Rule-based Modeling with BioNetGen

A formal and detailed description of the BioNetGen language can be found at [9]. The basic idea is to use *molecules* to describe the building blocks of a biological system such as proteins, genes, and metabolites respectively. Each *molecule* will have a number of sites with associated internal states that are used to represent the status of post-translational modifications or bindings with other *molecules*. The *rules* describe the interactions among *molecules* including associations, dissociations, modifications to the internal state of a site as well as the production or consumption of molecular species. *Patterns* are used to identify a set of *molecules* that share the same internal states. For instance, a protein S with two phosphorylatable sites x and y, we define a *molecule* $S(x, y)$ to represent S, where the sites x and y can take an internal states from u, p (i.e. unphosphorylated and phosphorylated states), denoted as $S(y \sim u \sim p)$. Thus, $S(x \sim u, y \sim u)$ denotes a species S that both its sites x and y are unphosphorylated. Note that the pattern $S(y \sim u)$ matches both $S(x \sim u, y \sim u)$ and $S(x \sim p, y \sim u)$ so that the *rules* follow the *"don't care, don't write"* convention.

Say we have an enzyme $E(z)$, which can bind to S at y and catalyze the phosphorylation of the site y, and then unbind from S. This can be captured by the following rules:

$$E(z) + S(y\ u) \leftrightarrow E(z!1).S(y\ u!1) \qquad k_1, k_2$$
$$E(z!1).S(y \sim u!1) \rightarrow E(z) + S(y \sim p) \qquad k_3$$

where k_1, k_2 and k_3 are rate constants, and ! denotes the binding.

RuleBender [15] provides graphical interface for users to construct, visualize, and simulate BNGL models. Given the initial copy numbers (or concentrations) of molecules: $E(z) = 200$, $E(y \sim u) = 300$, BioNetGen can simulate the time evolution of the system through ODE integration using the CVODE package, Gillespie's stochastic simulation, network-free simulation using the NFsim package, or the HPP algorithm. More details can be found at [14].

III. Statistical Model Checking

To evaluate the goodness of a candidate set of parameter values, we generate a representative set of trajectories by simulating the BioNetGen model with these parameter values and perform statistical model checking (SMC) to assess to what extent those trajectories reproduce the experimental observations. A trajectory generated by simulating a BioNetGen model is a series of time-dependent states of the form $\sigma = (s_0, t_0), (s_1, t_1), \ldots$, which means that the system jumps to state s_{i+1} after staying in state s_i for t_i. Because experimental data were available at a finite number of time points, we discretize the time domain into a finite set of time points $\mathcal{T} = \{0, 1, \ldots, T\}$, where T is the maximum time point allowed (often determined by the last time point for which experimental data were available). SMC is a scalable formal verification technique for testing whether a dynamical system satisfies a given system property encoded as logic formulas (in our case, the BLTL detailed below) with guaranteed confidence levels [32]. The main concept is that given a property φ, there is a probability $Pr(\varphi)$ that a randomly generated trajectory will have the property φ. Thus, whether the system satisfies the property φ can be statistically verified by: (i) setting up the null hypothesis $\mathrm{H0} : Pr(\varphi) \geq p$ and the alternative hypothesis $\mathrm{H1} : Pr(\varphi) < p$, where p is a confidence level chosen by the user, (ii) repeatedly generate trajectories of sufficient length (determined by φ) and check whether they satisfy φ. Standard sequential hypothesis testing methods can be used to terminate this test after deciding whether either the null hypothesis or the alternative hypothesis holds.

A. Bounded Linear Temporal Logic

Let $\mathcal{S}$ be a finite set of real-valued variables. Our BLTL is defined over a finite set of atomic proposition (AP), which will be of the form $x \# y$, where x and y are arithmetic expressions over real-valued variables in S, and $\# \in \{>, <, =, \geq, \leq\}$. The logic operators in our BLTL consist of $\wedge$ (and), $\vee$ (or), $\neg$ (negation), $\mathbf{O}$ (next), and time bounded $\mathbf{U}$ (until), $\mathbf{G}$ (global), and $\mathbf{F}$ (future). The formulas of BLTL are defined as: (i) every AP as well as the constants true and false are BLTL formulas; (ii) If ψ, ψ' are BLTL formulas then $\neg\psi$ and $\psi \vee \psi'$ are BLTL formulas. (iii) If ψ, ψ' are BLTL formulas and $t \leq T$ is a positive integer then $\mathbf{O}\psi'$ $\psi \mathbf{U}^{\leq t} \psi'$, $\psi \mathbf{U}^{t} \psi'$, $\mathbf{F}^{\leq t}\psi'$ and $\mathbf{G}^{\leq t}\psi'$ are BLTL formulas.

The notion of a trajectory σ satisfying a BLTL-specified property ϕ at time point $t \in \mathcal{T}$ is written as $\sigma, t \models \varphi$. Its semantics is defined as follows:

- $\sigma, t \models AP$ iff AP holds true in state s_t.
- $\neg$ and $\vee$ are interpreted in the usual way.
- $\sigma, t \models \psi \mathbf{U}^{\leq k} \psi'$ iff there exists k' such that $k' \leq k$, $t + k' \leq T$ and $\sigma, t + k' \models \psi'$. Further, $\sigma, t + k'' \models \psi$ for every $0 \leq k'' < k'$.
- $\sigma, t \models \psi \mathbf{U}^{k} \psi'$ iff $t + k \leq T$ and $\sigma, t + k \models \psi'$. Further, $\sigma, t + k' \models \psi$ for every $0 \leq k' < k$.

We define probabilistic BLTL formulas in the form of $Pr_{\geq r}(\psi)$, where $\theta \in (0, 1]$, meaning that the probability that a trajectory σ satisfying φ is at least r. Here the probability measure for ODE systems with prior distribution of initial states is defined in [21], while the one for stochastic systems (CTMCs) is defined in the usual way [26]. The statements we make are in the form of $M \models Pr_{\geq r}(\psi)$, meaning that the probability that the system M satisfies a property ψ is at least r. For example, we express the property "caspase-3 (c) level sustains once it reaches certain threshold" as follows:

$$Pr_{\geq 0.95}(c \leq 1 \mathbf{U}^{\leq 10}(\mathbf{F}^{\leq 56}(c \geq 30 \wedge \mathbf{G}^{\leq 44}(c \leq 30))))$$

B. Statistical Model Checking of Probabilistic BLTL Formulas

With SMC, the verification of such properties can be carried out approximately but with guaranteed confidence levels and error bounds. According to [33], whether $M \models P_{\geq r}\psi$, can be formulated as a sequential hypothesis test between the null hypothesis $\mathrm{H0} : p \geq r + \delta$ and the alternative hypothesis $\mathrm{H1} : p \leq r - \delta$, where p is the probability of M satisfying ψ and δ specifies the indifference region supplied by the user. The *strength* of the test is decided by parameters α and β which bound the Type-I (false positive) and Type-II (false negative) errors respectively. The test proceeds by generating a sequence of sample trajectories $\sigma_1, \sigma_2, \ldots$. One assumes a corresponding sequence of Bernoulli random variables $y_1, y_2 \ldots$, where each y_k is assigned the value 1 if $\sigma_k, 0 \models \psi$. Otherwise y_k is assigned the value 0. For each $m \geq 1$, after drawing m samples, we compute a quantity q_m as:

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

Hypothesis H0 is accepted if $q_m \geq \widehat{A}$, and Hypothesis H1 is accepted if $q_m \leq \widehat{B}$. If neither is the case then another sample is drawn. The constants $\widehat{A}$ and $\widehat{B}$ are chosen such that it results in a test of strength (α, β). In practice, a good approximation is $\widehat{A} = \frac{1-\beta}{\alpha}$ and $\widehat{B} = \frac{\beta}{1-\alpha}$.

C. Tableau-based Online Model Checking

The most computational intensive task in the model checking procedure is simulating the system for each sample. We use a-tableau based online model checking method which combines the process of simulation with model checking. Instead of simulating the system upto T and then apply the (offline) model checking procedure, we simulate the system only until the model checker can make a decision (i.e. returns either sat or unsat). This can often reduce simulation time and the overhead of storing the trajectories.

Specifically, we construct and propagate a finite family of sets $\mathcal{F}$. Each set $F_i \in \mathcal{F}$ contains a finite number of formulas. Let φ, ψ and γ be BLTL formulas. A $\mathsf{literal}$ is defined as an atomic proposition $A \in AP$ or its negation $\neg A$. For the purpose of illustration, let's assume that we convert the given BLTL formulas into a form in which only the atomic propositions can appear in negated form. In other words, we assume our formulas will have the following syntax: (i) Every literal is a formula. (ii) If φ and φ' are formulas so are $\varphi \vee \varphi'$

and $\varphi \wedge \varphi'$, $\mathbf{O}\varphi$, $\mathbf{F}\varphi$, $\mathbf{G}\varphi$, $\varphi \mathbf{U} \varphi'$. Every formula in the original syntax can be expressed as a formula in the above syntax where only the APs are negated.

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

- If φ is true or a literal then $cl(\varphi) = \{\{\varphi\}\}$.
- If $\varphi = \psi \vee \gamma$ then $cl(\varphi) = cl(\psi) \cup cl(\gamma)$.
- If $\varphi = \psi \wedge \gamma$ then $cl(\varphi) = cl(\psi) \times cl(\gamma)$.
- If $\varphi = \boldsymbol{O}\psi$ then $cl(\varphi) = \{\{\boldsymbol{O}\psi\}\}$.
- If $\varphi = \boldsymbol{F}\psi$ then $cl(\varphi) = cl(\psi) \cup cl(\boldsymbol{OF}\psi)$.
- If $\varphi = \boldsymbol{G}\psi$ then $cl(\varphi) = cl(\psi) \times cl(\boldsymbol{OG}\psi)$.
- If $\varphi = \psi \boldsymbol{U} \gamma$ then $cl(\varphi) = cl(\gamma) \cup (cl(\psi) \times cl(\boldsymbol{O}(\psi \boldsymbol{U} \gamma))$.

If we have a set of formulas $Y = \{\varphi_1, \varphi_2, \ldots, \varphi_n\}$, then the closure $cl(Y)$ can be written as $cl(Y) = cl(\varphi_1) \times cl(\varphi_2) \ldots \times cl(\varphi_n)$. We can also extend the notion of closure to families of sets of formulas such as $\mathcal{F} = \{Y_1, Y_2, \ldots, Y_k\}$, and say that the closure set of $\mathcal{F}$ is $cl(\mathcal{F}) = cl(Y_1) \cup cl(Y_2) \ldots cl(Y_k)$

We call the set of formulas Y a *leaf set* iff $cl(Y) = Y$. Further, a set Y is *inconsistent* iff (i) for an AP p, $p \in Y$ and $\neg p \in Y$ or (ii) for some formula φ, both $\mathbf{O}\varphi \in Y$ and $\mathbf{O}\neg\varphi \in Y$. The following assertions hold.

- Y is a leaf set iff each formula in Y is a literal or a $\mathbf{O}$ formula.
- $cl(\varphi)$ is a leaf family for each φ.
- $cl(Y)$ is a leaf family for every finite set of formulas Y.
- $cl(\mathcal{F})$ is a leaf family for every family of formula sets $\mathcal{F}$.

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

Now assume we are given a formula φ and want to check in an online manner if the system trajectory satisfies φ. We propagate a family of sets and start with $\mathcal{F}^0 = cl(\varphi)$. Inductively, assume that we are given the family of sets $\mathcal{F}^t$ for $t < T$. If $\mathcal{F}^t$ is dead, then we set $\mathcal{F}^{t+1} =$ false, and if $\mathcal{F}^t$ is terminal then we set $\mathcal{F}^{t+1} =$ true. Otherwise, $\mathcal{F}^t$ is neither dead nor terminal. In this case we know that $\exists Y_1, Y_2, ..., Y_k \in \mathcal{F}^t, k \geq 1$ which are not dead. Since these sets are not dead, we know that $\forall i, 1 \leq i \leq k : next(Y_i) \neq \emptyset$. We can then build the family of sets for time $t+1$ as $\mathcal{F}^{t+1} = cl(next(Y_1)) \cup cl(next(Y_2)) \ldots \cup cl(next(Y_k))$.

The process terminates at time $t < T$ if $\forall Y \in \mathcal{F}^t$ is false and returns $s(0) \not\models \varphi$ or if $\exists Y \in \mathcal{F}^t$ which is true, and returns $s(0) \models \varphi$. Furthermore, if $t = T$, if $\mathcal{F}^t$ is a terminal leaf family at $s(T)$, the process terminates and returns that $s(0) \models \varphi$. Otherwise it returns $s(0) \not\models \varphi$.

IV. SMC-based Parameter Estimation

A. Knowledge Encoding

We first describe how *quantitative* training data can be encoded as a BLTL formula. Let $O \subseteq \{x_1, x_2, \ldots, x_n\}$ be the set of variables for which experimental measurements are available as the training data. Assume $\mathcal{T}_i = \{\tau_1^i, \tau_2^i, \ldots, \tau_{T_i}^i\}$ are the time points at which the concentration level of x_i has been measured and reported as $[\ell_t^i, u_t^i]$ for each $t \in \mathcal{T}_i$. Here the interval $[\ell_t^i, u_t^i]$ is so chosen that it reflects the noisiness (i.e. error bars), the limited precision and the cell-population-based nature of the experimental data.

For each $t \in \mathcal{T}_i$ we define the formula $\psi_i^t = \mathbf{F}^t(\ell_t^i \leq x_i i \wedge x_i \leq u_t^i)$. Then $\psi_{exp}^i = \bigwedge_{t \in \mathcal{T}_i} \psi_i^t$. We then set $\psi_{exp} = \bigwedge_{i \in O} \psi_{exp}^i$. In case the species x_i has been measured under multiple experimental conditions, then the above encoding scheme is extended in the obvious way.

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

Finally we fix the probabilistic BLTL formula $P_{\geq r}(\psi_{exp} \wedge \psi_{qlty})$, where r will capture the confidence level with which we wish to assess the goodness of the fit of the current set of parameters to experimental data and qualitative trends. We also fix an indifference region δ and the strength of the test (α, β). The constants r, δ, α and β are to be fixed by the user. In our application it will be useful to exploit the fact that both ψ_{exp} and ψ_{qlty} are conjunctions and hence can be evaluated separately. As shown in [33], [34], one can choose the strength of each of these tests to be $(\frac{\alpha}{J}, \beta)$, where J is the total number of conjuncts in the specification. This will ensure that the overall strength of the test is (α, β). Further, the results for the individual statistical tests can be used to compute the objective function associated with the global search strategy, as detailed below.

B. Parameter Estimation Algorithm

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

1) Fix θ_0, which assigns a value to each unknown rate constant. This represents the initial guess. Set $\ell = 0$.
2) With θ_ℓ as the current set of rate constant values, run the statistical model checking procedure to verify the individual conjuncts of $\psi_{exp} \wedge \psi_{qlty}$ with the chosen strengths.
3) Based on the answers returned by these tests compute $F(\theta_\ell)$, where F is the objective function.
4) Check if the value of the objective function is sufficiently high or ℓ has reached a predetermined bound.
5) If yes, return θ_ℓ as the estimated value.

6) Else fix a new set of rate constant values $\theta_{\ell+1}$ as dictated by the search strategy. Increment ℓ to $\ell+1$ and return to step (2).

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

$$F(\theta) = J^+_{qlty}(\theta) + \sum_{i \in O} \frac{J^{i,+}_{exp}}{J^i_{exp}} \tag{2}$$

Thus the goodness to fit of θ is measured by how well it agrees with the qualitative properties as well as the number of experimental data points with which there is acceptable agreement. To avoid over-training the model, we do not insist that every qualitative property and every data point must fit well with the dynamics predicted by θ. It is possible to introduce additional terms to the objective function in order to speed up convergence in practice. We discuss one such method in the supplementary material [35].

The search strategy deployed in step (6) above will use the values $F(\theta_\ell)$ to traverse the space of candidate parameter vectors. The search method can be *local* or *global*. Local methods such as the Levenberg-Marquardt algorithm [36] have the advantage of converging fast, but can get stuck in local minima. Global methods such as Genetic Algorithms (GA) [37], and Stochastic Ranking Evolutionary Strategy (SRES) [24] – although computationally more intensive – are much better at avoiding local minima and *in principle* monotonically improve the estimates in proportion to the computational effort.

In practice, global methods usually maintain a *set* of parameter value vectors in each round. Each round is called a *generation* and the current set of parameter value vectors is called a *population*. Here, for the sake of convenience, we have explained the basic structure of the algorithm by pretending that each population is a singleton. We use the SRES strategy in our work since it is known to perform well in the context of pathway models [23]. The particular choice of search algorithm, however, is orthogonal to our proposed method.

V. Results

We have tested our method on a p53-induced apoptosis model [8]. The tumor suppressor protein p53 is an important mediator of cell response to genotoxic stress. The model has been used to evaluate pharmacological strategies for controlling ionizing radiation (IR)-induced cell death in order to mitigate radiation damage and alleviating the side effects of anti-cancer radiotherapy manifested in surrounding tissue morbidity. We constructed a rule-based model using BioNetGen based on the reaction network shown in Figure 1. Our model consists of 86 rules and 160 parameters. We used the nominal parameter values to generate synthetic experimental data by simulating the model using SSA. To mimic a population of single cell measurements, we generated 100 trajectories and compute their means and standard deviations at 5 time points for 4 species to assemble the training data set. We also incorporated the following qualitative knowledge into the training data.

Property 1: Mdm2 reaches its peak after p53.

$$\begin{aligned}
&Pr_{\geq 0.9}(p53^{(N)} \leq 0.01nM \wedge Mdm2^{(N)} \leq 0.01nM \\
&\mathbf{F}^{\leq 100h}(p53^{(N)} \geq 1nM \wedge Mdm2^{(N)} \leq 0.01nM \wedge \\
&\mathbf{F}^{\leq 100h}(p53^{(N)} \geq 3nM \wedge Mdm2^{(N)} \leq 0.4nM \wedge \\
&\mathbf{F}^{\leq 100h}(p53^{(N)} \leq 4nM \wedge Mdm2^{(N)} \leq 0.4nM))))
\end{aligned}$$

The above property specifies that the level of nuclear p53 reaches a peak value between 3 and 4 nM before the level of nuclear Mdm2 reaching a peak value around 0.4 nM.

Property 2: Sustained caspase-3 once its level reaches certain threshold.

$$Pr_{\geq 0.9}(\mathbf{F}^{\leq 56h}(C3 \geq 0.3nM \wedge \mathbf{G}^{\leq 44h}(C3 \leq 0.3nM)))$$

The above property specifies that after caspase-3 concentration reaches 0.3nM, it will sustain for at least 44 h and triggers downstream apoptotic cascade.

Property 3: p53 pulses induce oscillatory behaviors of target genes.

$$\begin{aligned}
&Pr_{\geq 0.9}(\mathbf{F}^{\leq 56h}(Bax \geq 0.06nM \wedge PUMA \geq 0.06nM \wedge \\
&(\mathbf{F}^{\leq 56h}(Bax \leq 0.04nM \wedge PUMA \leq 0.04nM \wedge \\
&(\mathbf{F}^{\leq 56h}(Bax \geq 0.06nM \wedge PUMA \geq 0.06nM \wedge \\
&(\mathbf{F}^{\leq 56h}(Bax \leq 0.04nM \wedge PUMA \leq 0.04nM))))
\end{aligned}$$

The above property specifies the oscillatory behavior of Bax and PUMA.

We fixed a subset of 10 rate constants to be unknown, and run our parameter estimation procedure. The experiments were carried out on a machine with two Intel Xeon E5-2650 2.00GHz processors and 32GB RAM. The parameters used for the statistical model checking algorithm were $r = 0.9$, $\alpha = 0.05$, $\beta = 0.05$, $\delta = 0.05$. Parameter estimation was done with a population size of 50 per generation and for 50 generations. The time taken by SRES-based search was 4.2 hours. Figure 2 shows the fit to training data for simulated time profiles with the best parameters found. Figure 3 shows simulated time profiles fit the specified qualitative properties.

VI. Conclusion

We have presented a statistical model checking based framework for estimating unknown parameters of rule-based models. Our method can utilize both quantitative experimental data and qualitative properties of system dynamics as training and test data. We perform online model checking to evaluate the

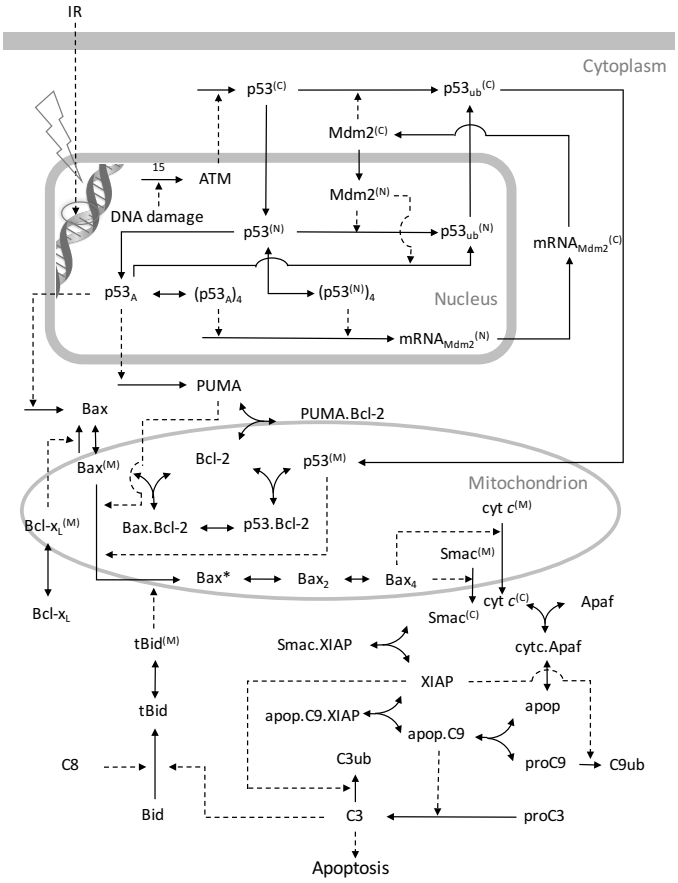

Fig. 1. Reaction network.

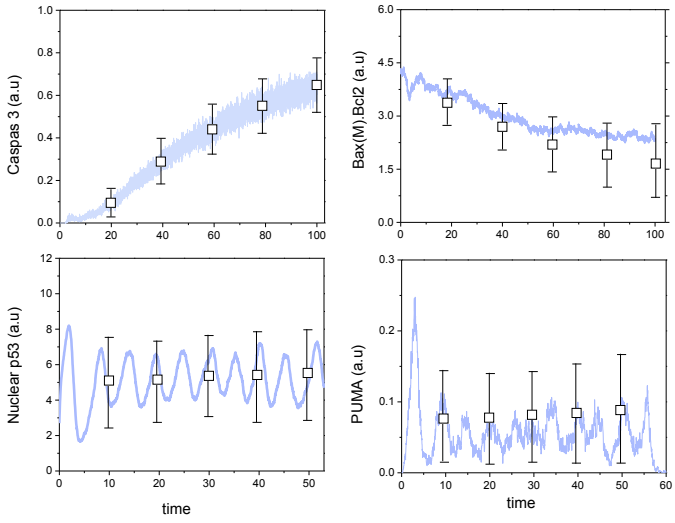

Fig. 2. Simulation results vs. quantitative training data.

goodness of parameter values and use an evolutionary strategy to explore the parameter space. We have tested our method on a stochastic p53 pathway model. The preliminary results look promising. Currently, we focused on testing our method by performing Gillespie's stochastic simulations. We plan to

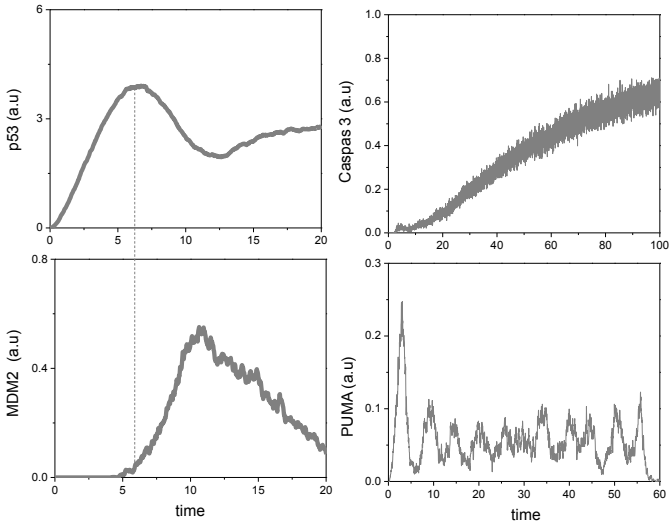

Fig. 3. Simulation results reproduce qualitative properties.

carry out more case studies using the network-free simulator. We also plan to develop a GPU-enabled implementation to further improve the performance [38], [39], [40].

Acknowledgment

This was supported by the NIH award P41-GM103712.

References

[1] P. K. Sorger and B. Schoeberl, "An expanding role for cell biologists in drug discovery and pharmacology," *Molecular Biology of the Cell*, vol. 23, no. 21, pp. 4162–4164, 2012.

[2] S. I. Berger and R. Iyengar, "Role of systems pharmacology in understanding drug adverse events," *Wiley Interdisciplinary Reviews: Systems Biology and Medicine*, vol. 3, no. 2, pp. 129–135, 2011.

[3] B. Liu, J. Zhang, P. Y. Tan, D. Hsu, A. M. Blom, B. Leong, S. Sethi, B. Ho, J. L. Ding, and P. S. Thiagarajan, "A computational and experimental study of the regulatory mechanisms of the complement system," *PLoS Computational Biology*, vol. 7, no. 1, 2011.

[4] B. Liu, Q. Liu, S. Palaniappan, I. Bahar, P. S. Thiagarajan, and J. L. Ding, "Innate Immune Memory and Homeostasis May Be Conferred Through TLR3-TLR7 Pathway Crosstalk," *Science Signaling*, vol. 9, no. 436, p. ra70, 2016.

[5] K. Leder, K. Pitter, Q. Laplant, D. Hambardzumyan, B. D. Ross, T. A. Chan, E. C. Holland, and F. Michor, "Mathematical modeling of pdgf-driven glioblastoma reveals optimized radiation dosing schedules," *Cell*, vol. 156, no. 3, pp. 603–616, 2014.

[6] B. Liu and P. S. Thiagarajan, "Modeling and Analysis of Biopathways Dynamics," *Journal of Bioinformatics and Computational Biology*, vol. 10, no. 4, p. 1231001, 2012.

[7] B. B. Aldridge, J. M. Burke, D. a. Lauffenburger, and P. K. Sorger, "Physicochemical modelling of cell signalling pathways." *Nature Cell Biology*, vol. 8, no. 11, pp. 1195–203, 2006.

[8] B. Liu, D. Bhatt, Z. N. Oltvai, J. S. Greenberger, and I. Bahar, "Significance of p53 dynamics in regulating apoptosis in response to ionizing radiation, and polypharmacological strategies." *Scientific Reports*, vol. 4, p. 6245, 2014.

[9] J. R. Faeder, M. L. Blinov, and W. S. Hlavacek, "Rule-based modeling of biochemical systems with BioNetGen," *Methods in Molecular Biology*, vol. 500, no. 1, pp. 113–167, 2009.

[10] V. Danos, J. Feret, W. Fontana, and J. Krivine, "Abstract interpretation of cellular signalling networks," in *VMCAI'08*, vol. 4905 LNCS, 2008, pp. 83–97.

[11] L. A. Chylek, L. A. Harris, J. R. Faeder, and W. S. Hlavacek, "Modeling for (physical) biologists: an introduction to the rule-based approach," *Physical Biology*, vol. 12, no. 4, p. 45007, 2015.

[12] L. A. Chylek, L. A. Harris, C. S. Tung, J. R. Faeder, C. F. Lopez, and W. S. Hlavacek, "Rule-based modeling: A computational approach for studying biomolecular site dynamics in cell signaling systems," pp. 13–36, 2014.

[13] Q. Wang, N. Miskov-Zivanov, B. Liu, J. R. Faeder, M. Lotze, and E. M. Clarke, *Formal Modeling and Analysis of Pancreatic Cancer Microenvironment*. Cham: Springer International Publishing, 2016, pp. 289–305. [Online]. Available: http://dx.doi.org/10.1007/978-3-319-45177-0_18

[14] L. A. Harris, J. S. Hogg, J.-J. Tapia, J. A. P. Sekar, S. Gupta, I. Korsunsky, A. Arora, D. Barua, R. P. Sheehan, and J. R. Faeder, "BioNetGen 2.2: Advances in Rule-Based Modeling." *Bioinformatics*, 2016.

[15] W. Xu, A. M. Smith, J. R. Faeder, and G. E. Marai, "RuleBender: A visual interface for rule-based modeling," *Bioinformatics*, vol. 27, no. 12, pp. 1721–1722, 2011.

[16] J. E. Wenskovitch, L. A. Harris, J.-J. Tapia, J. R. Faeder, and G. E. Marai, "MOSBIE: a tool for comparison and analysis of rule-based biochemical models." *BMC bioinformatics*, vol. 15, no. 1, p. 316, 2014.

[17] M. W. Sneddon, J. R. Faeder, and T. Emonet, "Efficient modeling, simulation and coarse-graining of biological complexity with NFsim," *Nature Methods*, vol. 8, no. 2, pp. 177–183, 2011.

[18] S. Hoops, S. Sahle, R. Gauges, C. Lee, J. Pahle, N. Simus, M. Singhal, L. Xu, P. Mendes, and U. Kummer, "COPASI - a COmplex PAthway SImulator," *Bioinformatics*, vol. 22, no. 24, pp. 3067–3074, 2006.

[19] B. R. Thomas, L. A. Chylek, J. Colvin, S. Sirimulla, A. H. A. Clayton, W. S. Hlavacek, and R. G. Posner, "BioNetFit: A fitting tool compatible with BioNetGen, NFsim and distributed computing environments," *Bioinformatics*, vol. 32, no. 5, pp. 798–800, 2015.

[20] J. Liepe, C. Barnes, E. Cule, K. Erguler, P. Kirk, T. Toni, and M. P. H. Stumpf, "ABC-SysBio-approximate bayesian computation in python with GPU support," *Bioinformatics*, vol. 26, no. 14, pp. 1797–1799, 2010.

[21] S. K. Palaniappan, B. M. Gyori, B. Liu, D. Hsu, and P. S. Thiagarajan, "Statistical model checking based calibration and analysis of bio-pathway models," in *CMSB'13*, vol. 8130 LNBI, 2013, pp. 120–134.

[22] R. Simmons and H. Younes, "Probabilistic Verification of Discrete Event Systems Using Acceptance Sampling," *Computer Aided Verification 2002*, pp. 223–235, 2002.

[23] C. G. Moles, P. Mendes, and J. R. Banga, "Parameter estimation in biochemical pathways: A comparison of global optimization methods," *Genome Res.*, vol. 13, no. 11, pp. 2467 –2474, 2003.

[24] T. Runarsson and X. Yao, "Stochastic ranking for constrained evolutionary optimization," *IEEE T. Evolut. Comput.*, vol. 4, pp. 284–294, 2000.

[25] J. S. Hogg, L. A. Harris, L. J. Stover, N. S. Nair, and J. R. Faeder, "Exact Hybrid Particle/Population Simulation of Rule-Based Models of Biochemical Systems," *PLoS Computational Biology*, vol. 10, no. 4, 2014.

[26] E. M. Clarke, J. R. Faeder, C. J. Langmead, L. A. Harris, S. K. Jha, and A. Legay, "Statistical Model Checking in BioLab: Applications to the Automated Analysis of T-Cell Receptor Signaling Pathway," in *Computational Methods in Systems Biology: 6th International Conference CMSB 2008, Rostock, Germany, October 12-15, 2008. Proceedings*, 2008, pp. 231–250.

[27] L. Calzone, N. Chabrier-Rivier, F. Fages, and S. Soliman, "Machine learning biochemical networks from temporal logic properties," *T. Comput. Syst. Biol. VI*, pp. 68–94, 2006.

[28] A. Saito, M. Nagasaki, A. Doi, K. Ueno, and S. Miyano, "Cell fate simulation model of gustatory neurons with microRNAs double-negative feedback loop by hybrid functional Petri net with extension," *Genome Informatics*, vol. 17, no. 1, pp. 100–111, 2006.

[29] A. Rizk, G. Batt, F. Fages, and S. Soliman, "A general computational method for robustness analysis with applications to synthetic gene networks," *Bioinformatics*, vol. 25, no. 12, pp. i169–i178, 2009.

[30] B. Liu, S. Kong, S. Gao, P. Zuliani, and E. M. Clarke, *Parameter Synthesis for Cardiac Cell Hybrid Models Using δ-Decisions*. Cham: Springer International Publishing, 2014, pp. 99–113. [Online]. Available: http://dx.doi.org/10.1007/978-3-319-12982-2_8

[31] ——, "Towards personalized prostate cancer therapy using delta-reachability analysis," in *Proceedings of the 18th International Conference on Hybrid Systems: Computation and Control*, ser. HSCC '15. New York, NY, USA: ACM, 2015, pp. 227–232. [Online]. Available: http://doi.acm.org/10.1145/2728606.2728634

[32] H. L. S. Younes, M. Kwiatkowska, G. Norman, and D. Parker, "Numerical vs. statistical probabilistic model checking," in *International Journal on Software Tools for Technology Transfer*, vol. 8, no. 3, 2006, pp. 216–228.

[33] H. L. S. Younes and R. G. Simmons, "Probabilistic verification of discrete event systems using acceptance sampling," in *CAV*. Springer Berlin / Heidelberg, 2002, pp. 223–235.

[34] ——, "Statistical probabilistic model checking with a focus on time-bounded properties," *Inform. Comput.*, vol. 204, pp. 1368–1409, 2006.

[35] Supplementary information and source code, http://www.comp.nus.edu.sg/~rpsysbio/SMC/.

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

[37] D. Goldberg, *Genetic algorithms in search, optimization, and machine learning*. Addison-Wesley, 1989.

[38] R. Ramanathan, Y. Zhang, J. Zhou, B. M. Gyori, W.-F. Wong, and P. S. Thiagarajan, *Parallelized Parameter Estimation of Biological Pathway Models*. Cham: Springer International Publishing, 2015, pp. 37–57. [Online]. Available: http://dx.doi.org/10.1007/978-3-319-26916-0_3

[39] B. Liu, A. Hagiescu, S. K. Palaniappan, B. Chattopadhyay, Z. Cui, W. F. Wong, and P. S. Thiagarajan, "Approximate probabilistic analysis of biopathway dynamics," *Bioinformatics*, vol. 28, no. 11, pp. 1508–1516, 2012.

[40] A. Hagiescu, B. Liu, R. Ramanathan, S. K. Palaniappan, Z. Cui, B. Chattopadhyay, P. S. Thiagarajan, and W.-F. Wong, "GPU code generation for ODE-based applications with phased shared-data access patterns," *ACM Transactions on Architecture and Code Optimization*, vol. 10, no. 4, pp. 1–19, 2013.