

To appear in:

Towards the Inference of Stochastic Biochemical Network and Parameterized Grammar Models

Guy Yosiphon and Eric Mjolsness

*Donald Bren School of Information and Computer Science
University of California, Irvine, CA 92697-3425*

E-mail: {gyosipho; emj}@uci.edu

UCI ICS TR# 08-07

October 27, 2008

Abstract

Deterministic dynamical systems are used extensively for modeling in biological applications. Unfortunately, such models cannot take into account much of the stochastic behavior that arises in biological systems. Stochastic process models can capture the variance and higher moments that exist in noisy real data. But the use of stochastic process models is limited because of the formidable task of inferring the model parameters' values from observations. In this chapter we discuss a parameter inference scheme for a family of stochastic processes that can be defined by generalized reactions or rewrite rules as they occur within the Stochastic Parameterized Grammars (SPGs) modeling framework.

The chapter is organized as follows. Section 1 provides an overview of the related work on optimization techniques for both deterministic continuous models and stochastic processes. In Section 2 we introduce the SPG modeling framework and derive a sampling scheme for SPG models. The parameter inference problem is defined in Section 3. We describe the general Metropolis-Hastings algorithm in section 3.1, and demonstrate how to apply it to SPG models in 3.2. Section 4 presents the results for optimization in a stochastic model of chemical reactions. Possible extensions and comparison to related algorithms is discussed in Section 5.

1 Introduction

Numerous optimization approaches have been applied to parameter inference in deterministic dynamical systems. Among the first fully automatic parameter estimations for biological network dynamics was [32], which used the Lam-Delosme variant of simulated annealing to fit the 33 parameters, including a 5x5 matrix of connection strengths, in a phenomenological gene regulation network (Artificial neural network or ANN style GRN, [25], Eq. 10) model to image-derived expression domain data in *Drosophila* development. This is still a cutting-edge problem today, 15 years after the original computer experiments. Chapter 1 discusses more modern approaches for optimizing the same systems. A similar algorithm was later parallelized [6]. Genetic algorithms were compared to simulated annealing on these GRN models in [21]. Furthermore, [21] presents the results of applying these optimization methods to variable-structure dynamical systems, specified by a deterministic grammar as defined in [24]. The resulting model maintains a changing number of cells each of which contains a set of chemical kinetic differential equations representing a gene network. Simulated annealing, quasi-Newton methods, and a variety of other deterministic and stochastic optimization methods (including Levenberg-Marquardt, Nelder-Mead, and Hooke-Jeeves) were compared in their applications to more standard chemical kinetic differential equation models in [23]. Lam-Delosme simulated annealing compared favorably with several alternative methods for optimizing signal transduction models in [40].

There has been a recent revival of interest in developing parameter optimization methods. For example [17] applied parallelized differential evolution to the ANN GRN models. [31] applied the adjoint method, a continuous-time method analogous to backpropagation through time in discrete-time neural networks, to efficiently compute gradients for gradient algorithms, and applied them to mass action kinetic models of *Drosophila* wing development. [27] applied recent developments in evolutionary strategies to ANN GRN's. Another major category of research looks to Bayesian inference for a firm statistical foundation in structure and parameter estimation. An essential requirement is a practical prior distribution on structures and/or parameters. [37] demonstrates a GRN inference method for feed-forward transcriptional networks using an approximate but tractable graph prior. Other graph priors are possible, as in [4], [30].

Turning to parameter estimation in stochastic dynamics, [13] developed a time-subdivision approach to learning reaction rates in small reaction networks under an SDE approximation to the usual stochastic dynamics. Chap-

ter 9 provides an overview of the SDE approximation and also includes a recent improvement that tackles the convergence problem of MCMC algorithm for SDE inference. Other parameter inference approximations which are based on the variational formulation of the posterior probability are introduced in Chapter 10 . A statistical mechanics perspective is applied in [7]. Others could be cited *e.g.* from recent workshops [3], [19]. All currently have strong restrictions on their domain of applicability, especially in the size of inferable system.

Also for parameter inference in stochastic dynamical systems, Green [14] has introduced a sampling method for probability distributions which are defined over spaces of varying dimensionality such as the variable-structure systems considered above. The method, known as reversible jump Markov Chain Monte Carlo (MCMC), is based on the Metropolis-Hastings (MH) [15] scheme (which will be described in later section). The algorithm performs "jumps" between models (or states) of different dimensionality. Therefore, it is suitable to handle transitions between reaction paths of different length (hence different number of variables). The reversible jump method requires an application-based design of jump transitions. The jump transitions that are commonly defined are birth, death, split or merge of random variables.

The reversible jump method was generalized to a block updating method which was used for inference in stochastic reaction models [5]. The block update scheme modifies multiple variables, which represent the number of reactions (of each type) and the reactions times, in a single MCMC step. A previous work on stochastic epidemic compartment models [10] has also used multiple jump events in each iteration in order to increase the convergence rate of an MCMC sampler. A Simulated Annealing cooling schedule was integrated into the reversible jump method in [1].

[33] have taken an approximate maximum likelihood approach for parameter estimation in stochastic biochemical reactions. The method approximates the number of reactions in each interval and assumes that the total rate does not change inside an interval. A similar approach was taken for simulating chemical reactions by the approximate tau-leap method [11].

An algorithm for inferring grammar-based structure models was introduced in [35]. The grammars, which are based on an L-system formulation [29], describe recursive structures and are used to model images of multicellular bacteria. Their grammars resemble SPGs but are only context free (only a single element on the LHS of rule) and can not model continuous-time processes. The inference procedure is an MCMC method that includes reversible jumps for adding or removing cells and branches from the multicellular structure.

2 Stochastic Parameterized Grammars

Stochastic Parameterized Grammars (SPG's) [26] comprise a formal modeling language based on grammar-like collections of rewrite rules. They have a stochastic process semantics in continuous time, which can be extended to discrete time. The essential idea is that there is a “pool” (unordered set) of fully specified parameter-bearing objects such as {bacterium[x], macrophage[y], redbloodcell[z]} where x , y and z are parameters such as position vectors. A grammar can include rules such as:

$$\{\text{bacterium}[x], \text{macrophage}[y]\} \rightarrow \text{macrophage}[y] \quad \mathbf{with} \quad \rho(\|x - y\|)$$

which specifies the probability per unit time, ρ , that the macrophage ingests and destroys the bacterium as a function of the distance $\|x - y\|$ between their centers. The left hand side (LHS) of the rule is comprised of terms which are matched to a set of input parameterized objects. The terms on the right hand side (RHS) constitute the output objects which are constrained to the parameter matching of the LHS terms. An SPG rule has the following general form:

$$A_1[x_1], A_2[x_2], \dots, A_n[x_n] \rightarrow B_1[y_1], B_2[y_2], \dots, B_m[y_m] \quad \mathbf{with} \quad \rho([x_i], [y_j]) \quad (1)$$

A_i and B_j are the object types and x_i (or y_i) is the uninstantiated parameters' vector of the i th term on the LHS (or RHS). The number of terms on the LHS or RHS is finite and can be zero. The function ρ can be any nonnegative function of the input and output parameters.

The SPG language is a generalization of the probabilistic formulation of chemical reactions. The grammar rules are reaction schemes where each rule term matches any object of the same type, regardless of parameters value. The system's state (or the “pool”) is represented by a vector of copy numbers for every unique parameterized object. Thus, the state space may be infinite or even uncountable when the parameters are continuous. For example, the following rate function is positive over an uncountable parameter space:

$$\text{bacterium}[x] \rightarrow \{\text{bacterium}[y_1], \text{bacterium}[y_2]\} \quad \mathbf{with} \quad \exp(-\alpha((x - y_1)^2 + (x - y_2)^2)) \quad (2)$$

This rule represents a bacterial cell division in which the daughter cells' locations are distributed continuously according to a Gaussian distribution centered around the parent cell's position.

An instantiated grammar rule represents a possible reaction that removes a set of reactants and adds the products. The waiting time for a reaction is exponentially distributed according to the rate w . The exponential rate is defined as the rule's rate function ρ times the number of distinct reactant combinations in a given state, which is denoted by n , *i.e.* $w = n\rho$. The SPG semantics can be summarized as a set of time dependent differential equations over the state probability [36]:

$$\forall a \quad \frac{d}{dt}P(a|t) = \sum_{b \neq a} W_{a,b}P(b|t) + W_{a,a}P(a|t), \quad \text{where } W_{a,a} = - \sum_{b \neq a} W_{b,a}$$

where $P(a, t)$ denotes the probability of state a at time t . The summation is over all possible neighboring states, where $W_{b,a}$ is the rate of a reaction, as defined above, from state b to state a . If $P(t)$ denotes the probability function over the entire state space then the system can be written in an operator form:

$$P'(t) = WP(t)$$

which has the formal solution:

$$P(t) = \exp(tW)P(0) \tag{3}$$

The operator W is the probability rate operator which is composed of the corresponding transition rates between states $W_{a,b}$, and the diagonal entries, $-W_{a,a}$.

It is impossible to derive a general algebraic solution of the probability function (Equation 3) or its moments since the time-evolution operator W could encode computationally sophisticated dynamics. One generally applicable approach is to use sampling techniques. In order to create samples from this distribution, the Time-Ordered Product Expansion (TOPE) is used. The TOPE is a valuable tool for studying such stochastic processes in physics [22] using vector notation:

$$\begin{aligned} P(t) &= \exp(tW) \cdot P(0) = \exp(t(W_0 + W_1)) \cdot P(0) \\ &= \sum_{n=0}^{\infty} \int_0^t dt_1 \int_{t_1}^t dt_2 \cdots \int_{t_{n-1}}^t dt_n \exp((t - t_n)W_0)W_1 \times \\ &\quad \exp((t_n - t_{n-1})W_0) \cdots W_1 \exp(t_1W_0) \cdot P(0) \end{aligned} \tag{4}$$

Here W_0 is a solvable or easily computable part of W , so the exponentials $\exp(tW_0)$ can be computed or sampled more easily than $\exp(tW)$. An obvious choice is to take W_0 to be the diagonal part of W , in which case we can derive Gillespie's well-known Stochastic Simulation Algorithm (SSA) for simulating chemical reaction networks Gillespie [12]. The SSA algorithm generates a multi-reaction path by jumping in time and sampling in each iteration the next waiting time and reaction event. Correspondingly, the TOPE with a diagonal $W_0 \equiv D$ interprets the probability of arbitrary state a at time t as the sum of probabilities over all possible multi-reaction paths (which result in state a at time t). To see this correspondence in more detail, we argue as follows.

A recursive form of the TOPE is that if τ_q is the time interval between events q and $q + 1$, then immediately after event k ,

$$\begin{aligned} P(a, [\tau_q | 0 \leq q \leq k-1], k | c, t) = \\ \sum_b \int_0^t d\tau P(a, \tau_{k-1}, 1 | b, \tau) P(b, [\tau_q | 0 \leq q \leq k-2], k-1 | c, t-\tau), \end{aligned}$$

an expression closely related to the Chapman-Kolmogorov equation [34], where $W_1 = \hat{W} \equiv W - D$ and where

$$\begin{aligned} P(a, [\tau_q | 0 \leq q \leq k], k | b, t) = \\ \left\{ \exp(-\tau_k D) \left[\prod_{q=k-1, k-2, \dots, 0} \hat{W} \exp(-\tau_q D) \right] \delta \left(t - \sum_{q=0}^k \tau_q \right) \right\}_{ab} \end{aligned}$$

and the immediately post-event probabilities are

$$P(a, [\tau_q | 0 \leq q \leq k-1], k | b, t) = P(a, [\tau_q | 0 \leq q \leq k-1, \tau_k = 0], k | b, t) .$$

An event-oriented simulation requires the use of Bayes' rule to convert from $P(a, [\tau_q | 0 \leq q \leq k-1], k | c, t)$ to $P(a, [\tau_q | 0 \leq q \leq k-1], t | c, k)$, and this in turn requires a distribution on simulation time, t . To this end, suppose there is a small constant probability per unit time, $\epsilon = 1/T_{\text{long}}$, that a simulation of the stochastic process defined by W comes to an end. This termination process will provide an exponential prior distribution on the time variable t . Taking the limit in which T_{long} is much longer than the time t of interest to us, so that $\epsilon t = t/T_{\text{long}} \rightarrow 0$, the parameter T_{long} drops out of the calculation and the Bayes-transformed simulation statistics are unaffected by the probability of pending termination.

Carrying out the indicated calculations and integrating out the random time intervals τ_q , we find that probability of a multi-reaction path under both TOPE and SSA can be expressed as:

$$P(\cdot|k) \approx \mathcal{W} \circ P(\cdot|k-1) \quad (5)$$

where

$$\mathcal{W}(a, t'|b, t) = \hat{W}_{ab} \exp(- (t' - t) D_{bb}) \mathbf{1}(t' \geq t).$$

and the “ \circ ” inner product operation combines both a sum over all states and an integral over all nonnegative times. Iterating this recursion relation,

$$P(\cdot|k) \approx \mathcal{W}^k \circ P(\cdot|0).$$

which has the form of a Markov chain.

In these equations we have defined $P(a, t|b, k)$ to be the “just-reacted state probability”: the probability of being in state a at time t immediately after the k 'th reaction event, given that the state is b at time zero. These equations explicitly express the SSA algorithm as a discrete-time Markov chain representing a randomized algorithm. This expression is in accord with, for example, Theorem 10.1 of [38].

2.1 Simulation of SPG's with output parameters

Equation 5 receives a stronger interpretation when the result of a reaction event is not only a new time and a new population of objects, but also a new assignment of parameters to these objects. This occurs for \mathcal{W} operators that contain parameter-dependent rates, as in equation 1. A parameter-dependent rate function, $\rho([x_i], [y_j])$, can be factored into $k([x_i]) \times P([y_j] | [x_i])$, where P is a normalized conditional probability. The Markov chain \mathcal{W} can then be accurately sampled by (1) choosing the next reaction's identity and time, as in SSA, according to the input parameters rate functions, $k([x_i])$, and then (2) sampling the output parameters according to $P([y_j] | [x_i])$ given the input parameters of the chosen reaction.

3 Parameter Inference

Using SPGs and related stochastic processes for modeling applications requires an estimation of the model's parameters from the observations. Parameter inference may be achieved by evaluating the likelihood function

$P(\Theta|\Gamma)$ where Γ denotes the stochastic grammar and its parameters, and Θ denotes the observations or evidence. Prior information about the model's parameters may be incorporated, using Bayes' theorem, to establish the posterior distribution:

$$P(\Gamma|\Theta) = \frac{P(\Theta|\Gamma)P(\Gamma)}{P(\Theta)} = \alpha P(\Theta|\Gamma)P(\Gamma) \quad (6)$$

The marginal probability, $P(\Theta)$, is equivalent to a normalizing factor which will be discarded in our sampling algorithm. The evidence, Θ , may be in the form of full or partial observation of the underlying system state in certain time points. The likelihood function is decomposed according to the multi-reaction path and the observations probability (the error term):

$$P(\Theta|\Gamma) = \sum_r P(\Theta|a[r, \vec{\tau}])P(r|\Gamma) \quad (7)$$

where $\vec{\tau}$ denotes the observations time points, r denotes a multi-reaction path and $a[r, \vec{\tau}]$ denotes the system state at the observation time points as realized by the path r .

Usually, deriving an analytical solution for the likelihood function is not possible. We resort to a sampling algorithm that is based on the Markov Chain Monte Carlo (MCMC) principle.

3.1 MCMC algorithm

An MCMC algorithm [2] generates a sequence of samples, according to a Markov chain transition function, that converges to the desired (target) probability. The Markov chain transition function, $M(x|x')$, is designed so that its invariant probability is the target probability, $\pi(x)$:

$$\pi(x) = \int M(x|x')\pi(x')dx'$$

Furthermore, the Markov chain must be ergodic, *i.e.* it converges to the invariant probability regardless of the initial condition [20]. The chain satisfy the ergodic condition if it is both irreducible and aperiodic. A chain is irreducible when there is a non-zero probability of reaching every state from any other state. A reducible chain has more than one invariant probability. An aperiodic (with period of 1) chain is established when, for every state, there is a non-zero probability to stay.

A sufficient but not necessary condition [20; 2] for $\pi(x)$ to be the invariant probability of the chain is the following detailed balance:

$$\pi(x)M(x'|x) = \pi(x')M(x|x')$$

For our sampling task, we use the Metropolis-Hastings (MH) algorithm [15] which is an MCMC method. An MH step works as follows:

1. Generate a candidate sample x given the current sample, $x^* = x^{(i-1)}$, according to a proposal distribution $q(x|x^*)$.
2. (To maintain detailed balance) The candidate sample is accepted according to the probability:

$$\text{Min}\left[1, \frac{\pi(x)q(x^*|x)}{\pi(x^*)q(x|x^*)}\right]$$

If accepted $x^{(i)} = x$, otherwise $x^{(i)} = x^*$.

The choice of the proposal distribution, q , is important for the convergence of the algorithm. On the one hand, a proposal distribution with small variance might constrain the MH method to a local region of the target distribution space. On the other hand, a proposal distribution with high variance will result in many rejections, which subsequently increase the correlation between samples. As a rule of thumb, the proposal distribution should be adjusted so that the rejection rate is approximately 0.5 [18; 20].

In most parameter inference applications, we are interested in the global maxima of the target distribution. For this task, a simulated annealing like strategy [8] can be integrated in the MH algorithm. The target distribution is modified to $\pi^{1/T_i}(x)$ where T_i denotes a decreasing cooling schedule $\lim_{i \rightarrow \infty} T_i = 0$. The algorithm is expected to initially explore a broad region of the state space and gradually confine the search to lower-energy areas.

3.2 Grammar sampling algorithm

We define the invariant (target) distribution of the MH method as the posterior distribution in Equation 6. A sample is an instantiation of the grammar parameters and a multi-reaction path. A possible choice for the proposal distribution can be obtained by omitting the observations probabilities from the target distribution:

$$q = P(r|\Gamma)P(\Gamma)$$

This proposal distribution may exhibit high variance since the candidate sample is independent of the current sample which may lead to high number of rejections. Since the grammar parameters are a fixed size set, a conditional probability on the current parameters values may be easily included in q :

$$q = P(r|\Gamma)f(\Gamma|\Gamma^*) \quad (8)$$

A Normal distribution which is centered around the current parameters values, Γ^* , is a natural choice for continuous variables.

Still, the multi-reaction path r is independent of the current path. We offer a modified proposal distribution for which each candidate path has the same set of reactions as the current path except for a randomly chosen time window (t', t'') . The reactions in the chosen time window are sampled according to the SPG probability conditioned on the state at time t' . In other words, the SSA algorithm is executed over the subinterval (t', t'') where the initial state is the state reached by the current path at time t' , $a(r^*, t')$. The probability of the candidate path in the subinterval window (denoted as $r'[t', t'']$) is:

$$P(r'[t', t''] \mid \text{initial time} = t', \text{end time} = t'', \text{initial state} = a(r^*, t'), \Gamma) \quad (9)$$

The candidate multi-reaction path over the whole time interval, r , is defined as follows (where $+$ denotes concatenation of multi-reaction paths):

$$r = r^*[0, t'] + r'[t', t''] + r^*[t'', t_{\text{end}}] \quad (10)$$

Note that although the reactions in the last subinterval (t'', t_{end}) are identical to the current path, r^* , the states may be different due to the changes in the intermediate subinterval. The deterministic scheme for setting the reactions in the last subinterval may lead to improbable paths. For example, the reactions in $r'[t', t'']$ could remove all water molecules by decomposition ($2\text{H}_2\text{O} \longrightarrow 2\text{H}_2 + \text{O}_2$) even though there is another decomposition reaction that occurs in $r^*[t'', t_{\text{end}}]$ before the production of any other water molecules. However, the target probability (π) of such an improbable path will be zero and it will be rejected.

Now, the modified proposal distribution is defined as:

$$q(r'; t', t'') = P(r'[t', t''] \mid t', t'', a(r^*, t'), \Gamma) P(t', t'') f(\Gamma|\Gamma^*) \quad (11)$$

Note that for a proper MH proposal distribution, the time window (t', t'') should be integrated out. However, the MCMC method would preserve

detailed balance even when picking a random time window (according to $P(t', t'')$) instead of integrating over them. This construction of an MCMC sampler is known as a mixture of transitions [20]. In general, MH with mixture of transitions can be applied in order to modify each variable, or block of variables, separately. In that case, a variable, or block of variables, is randomly selected (usually according to uniform distribution) followed by an MH transition over the chosen variable space. Such an algorithm is applicable when the proposal distribution can be decomposed according to individual transitions. The current proposal distribution (Equation 11) can be decomposed into a transition over a random window in the multi-reaction path, and a transition over the grammar parameters (or individual parameters):

$$\begin{aligned} q_1(r'; t', t'') &= P(r'[t', t''] | t', t'', a(r^*, t'), \Gamma^*) P(t', t'') \\ q_2(\Gamma | \Gamma^*) &= f(\Gamma | \Gamma^*) \end{aligned} \tag{12}$$

The proposal distributions (q_1, q_2) should be designed according to the Markov chain ergodicity requirement. If q_2 is a Normal distribution which is centered in the current parameter value then the chain is aperiodic and irreducible (since each parameter value is reachable). The window proposal is aperiodic since there is a non-zero probability to stay in the same path, *i.e.* $q_1(r^*; t', t'') > 0$. If the set of possible windows, according to $P(t', t'')$, covers the whole time interval then the proposal distribution is irreducible (since each path is reachable from any other path).

3.3 Inference in SPGs with output parameters

An SPG rule may have output parameters, *i.e.* parameters of right hand side objects which are not given as input on the left hand side of the rule. Such rules are useful for modeling stochastic behavior of objects with spatial information, as in the example of Equation 2. SPGs that include output parameters may describe processes with an unbounded number of unique objects. The current inference method can be applied for such parameterized grammar models although this could be inefficient or even infeasible for grammar rules that denote processes over uncountable spaces (as in Equation 2). A new multi-reaction path can be accepted, by the MCMC method, only if all the required unique objects are available for the reactions in the last subinterval (see Equation 10). Therefore the probability of accepting a new multi-reaction path for a process of unbounded number of unique objects might be infinitesimal. A discretized representation of continuous

parameter space is a possible solution that sacrifices accuracy for feasibility and efficiency. Another promising direction is to include information about the output parameters in the proposal distribution, so that parametrically probable trajectory changes are usually proposed and high acceptance rate is maintained.

3.4 Computational complexity

The computational complexity of the MCMC procedure depends on the sampler’s convergence speed and the time and space complexities of each sampling step.

There are several results on the convergence time of MH samplers for simple problems. [16] shows that the convergence time of the MH algorithm for sampling multivariate Normal densities can be bounded by a polynomial in the number of free parameters. Therefore MH samplers have an advantage over importance and rejection sampling that suffer from the exponential curse of dimensionality [20; 2]. The method proposed here requires more complicated probability distributions that may even have variable structure. The set of free parameters should include the set of reactions’ times and types that constitute a multi-reaction path and is therefore unbounded with potentially unbounded cost (though bounded with high probability). However, for many stochastic processes in which the reactions’ times and types are strongly correlated, the convergence time may be bounded. We observe rough convergence for a multi-reaction path with 250 reactions (figure 1) in about 100 iterations (see figures 2 and 3).

A sampling step, in our MCMC algorithm, includes a simulation algorithm run over the random time window and calculation of the probability function over the complete multi-reaction path. We now discuss the computational bounds on selecting each random reaction event in the simulation algorithm, and on the total number of reaction events. These bounds govern the cost of the probability function’s calculation as well.

The simulation algorithm maintains a data structure of pending reactions and their rates. Each reaction step consists of a search for the new executed reaction, execution of a reaction, creation of new pending reactions, and the modification of affected pending reactions. Define $\hat{r}(t)$ as the set of pending reactions at time t and $M(t)$ as the set of unique objects that exist at time t . Note that $|\hat{r}(t)|$ is bounded by $\sum_k \binom{M(t)}{|\text{reactant}(k)|}$, where $\text{reactant}(k)$ is the set of LHS reactant elements (or input elements) of the k th grammar rule. The unique objects can be stored in an array in which each entry

holds a link to the set of relevant pending reactions (all the reactions for which the current object is the first reactant). In many stochastic processes (for example, various gene regulation network models), the total number of unique objects over the entire process, $M = \cup_t(M(t))$, is finite, relatively small and known in advance. Thus, the objects' array can include the entire set of unique objects and the space complexity is $O(\max_t |\hat{r}(t)| + |M|)$. In other cases, where $|M|$ is too large or unpredictable, an alternative for the array is a hash table of fixed size.

The search for a random reaction may take $O(|\hat{r}(t)|)$, but using a binary tree reduces the search time to $O(\log |\hat{r}(t)|)$. The tree leaves refer to pending reactions whereas the inner nodes store the total rate and number of reactions (leaves) in their subtree. New pending reactions are inserted to the subtree (right or left) that has fewer leaves. That way, the binary tree is balanced throughout the simulation with logarithmic height.

Once a reaction, denoted as \hat{r}^* , is executed, the algorithm updates the affected pending reactions which are accessed directly by the array (or hash table) structure. The set of affected reactions is $\tilde{r} = \{r_i | a \in \hat{r}^* \wedge a \in r_i\}$, or in words, the set of reactions that share an object, a , with the executed reaction. Therefore this procedure takes $O(|\tilde{r}|)$ time.

Finally, the total number of reactions events depends on the total time T , the rates of each reaction $\rho(\hat{r}_i)$, and the number of possible reactants' groups per reaction $\prod_{j \in \hat{r}_i} N_j$ (where N_j is the number of identical objects that match the j th reactant). A bound on the number of reaction events is : $O(T * \max_i(\rho(\hat{r}_i) \prod_{i \in \hat{r}_j} N_j))$.

4 Illustrative Example

As an illustrative example of the inference algorithm, we use a simple model of chemical reactions for synthesis and decomposition $A + B \xrightleftharpoons[k_d]{k_s} C$, which can be expressed in SPG syntax:

```
grammar Chemical-Reaction {
  {A, B} → C with    $k_s$ 
  C → {A, B} with   $k_d$ 
}
```

After setting the rate variables, we generated multi-reaction paths using an SPG interpreter and simulator [39]. Figure 1 plots the time sequence of molecule states for a single stochastic multi-reaction path.

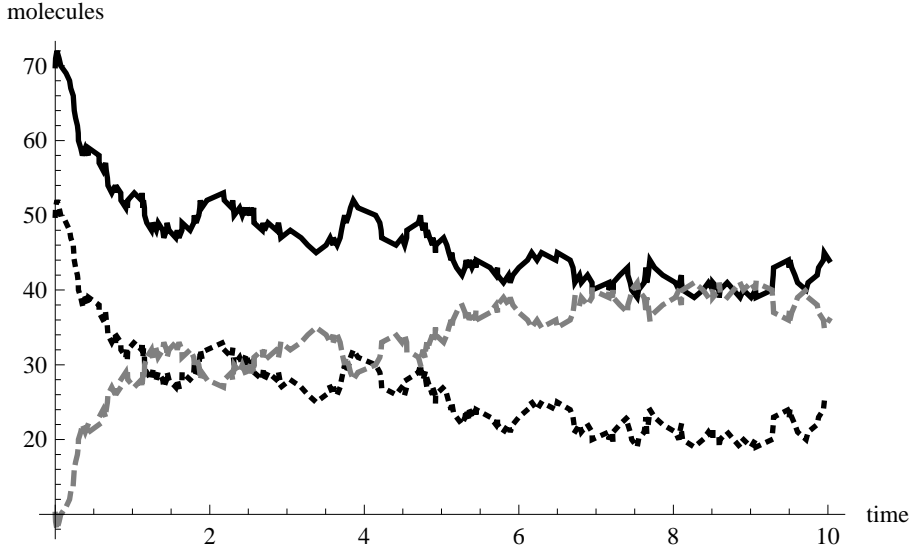


Figure 1: Plots of the time sequence of molecule states for a single stochastic multi-reaction path. Y axis - molecule quantity (A-solid black line,B-dotted black line, and C-dashed gray line). X axis - time. The synthesis and decomposition rates are $k_s = 0.01$, $k_d = 0.3$

The input for the inference algorithm is a sequence of molecule-number states in 20 evenly spaced time points. The observations conditional probability (data error term) has a Normal distribution shape with an input width (variance) parameter σ_i . Figure 2 and 3 present five trajectories of the MCMC inference algorithm for a single observations sequence. Figure 2 shows the convergence of the rate parameter where the true parameter value is 0.3. Figure 3 shows the convergence of the multi-reaction path in each MCMC iteration to the observed molecules states.

The inference algorithm uses different transitions for the rate variable and for the multi-reaction path, as defined in Equation 12. This has the benefit of potentially higher acceptance probability. The variance of the parameters proposal distribution and window size distribution were adjusted during the execution, in order to maintain a low rejection rate (below 0.5).

The algorithm implements an exponentially decreasing cooling schedule for the rate parameter transition:

$$T_i = p T_{i-1} \quad \text{where } 0 < p < 1$$

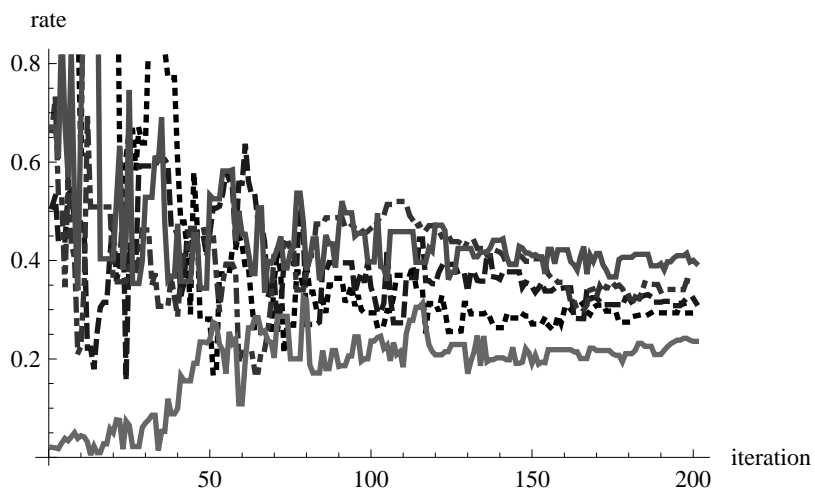


Figure 2: Trajectories of the MCMC inference algorithm: Y axis - inferred decomposition rate k_d . X Axis - iterations counter. The inferred decomposition rate converges toward the target rate value, $k_d = 0.3$

Initial attempts without a cooling schedule have resulted in rapid convergence to low-probability samples.

We have performed inference for different values of the decomposition rate k_d . The results are shown in Figure 4. For each rate value, five random multi-reaction paths were generated. We performed 25 inference runs where each is given one of the generated multi-reaction paths and a random initial rate value.

The variance of the observations conditional probability is important for the accuracy of the inference method. A search algorithm is more susceptible to get trapped in local minimas when the variance is too low. High variance may lead to the opposite effect since the penalty for improbable samples is insufficient.

Figure 5 displays the increase in standard deviation of inferred rates as we increase the variance of the observations' probability. The variance could be inferred as an additional parameter, but, in our experiments, it is held fixed at a given value.

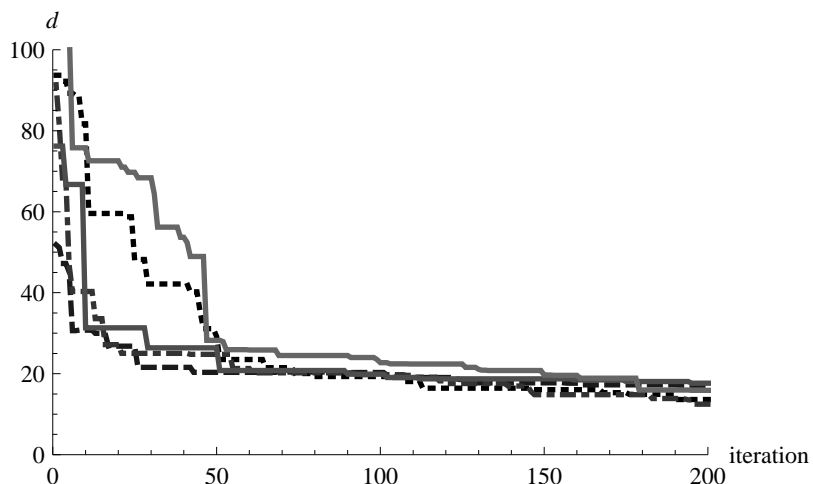


Figure 3: Trajectories of the MCMC inference algorithm: Y axis - Euclidean distance between the current sample’s molecule state and the input. X Axis - iterations counter.

5 Discussion

The method presented here does not require the construction of specialized jump transitions, which was required in [14], [5], [10], [1] and [33]. The modeling assumptions are all integrated in the declared grammar. The methods described in [5] and [10] require exact molecule counts over discrete time points, whereas in our method, the data is assumed to be noisy and the multi-reaction paths are weighted accordingly. Since the penalty for matching the data is less strict (and tunable), this method can be more flexible in traversing the multi-reaction path space.

Dynamical grammars (DGs) are defined as an extension of Stochastic Parameterized Grammars (SPGs) that include continuous-time rules. Such rules contain continuous-time dynamics (in the form of ordinary or partial differential equations). DGs provide a multiscale modeling framework in which a system can be comprised of continuous and discrete elements. A DG simulation algorithm, which is an extension of the SPG simulation algorithm, can be derived directly from the Time-Ordered Product Expansion (TOPE). Therefore, the presented inference algorithm can be modified in order to infer parameters in DG models.

Other formulations related to SPG’s and DG’s include L-Systems [29],

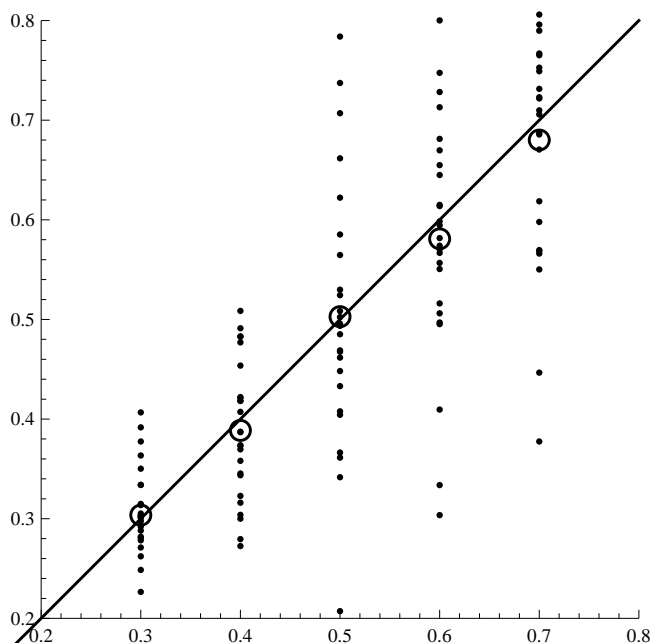


Figure 4: The distribution of inferred rate values for each target value, $k_d = \{0.3, 0.4, 0.5, 0.6, 0.7\}$: Each column of dots represents the 25 inference outcomes per target value (x position). The average inferred rate for each value is denoted by the circles.

Stochastic Pi-calculus (SPC) [28] and variants of Petri-nets (PNs) such as stochastic-colored PNs [9]. Both SPC and PNs were introduced for the study of concurrent computation and then applied to biology. Our inference scheme can be adopted to any of these formalisms. The main requirement of the scheme, which is to establish the probability of a multi-reaction path over some time window, can be derived according to the implicit or explicit probabilistic semantics of any of the above formalisms.

Acknowledgments Useful discussions with Yuanfeng Wang, Todd Johnson and David Orendorff about algorithms, and with Kimberly Gokoffski, Arthur Lander and Anne Calof about biological applications, are gratefully acknowledged. The work was supported in part by the National Science Foundation’s Frontiers in Biological Research (FIBR) program, award number EF-0330786, and the National Institutes of Health (NIH) awards numbers P50-GM76516 and P20-GM66051.

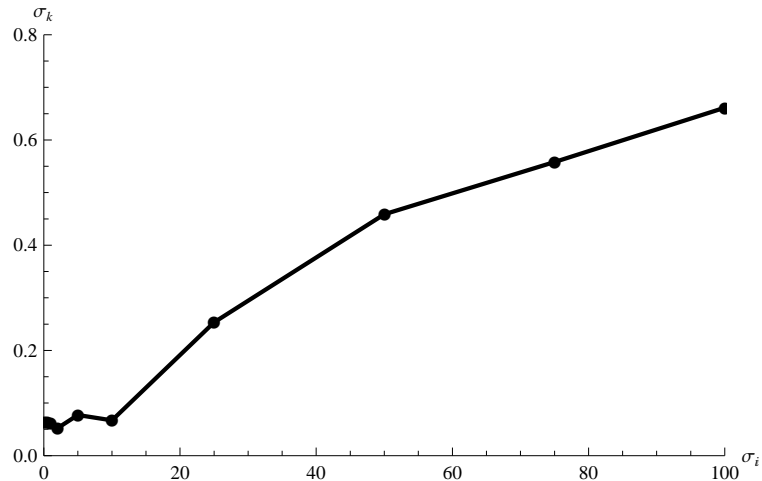


Figure 5: Standard deviation of inferred rates, σ_k , as a function of the observations' probability variance, σ_i . The synthesis and decomposition rates are $k_s = 0.01$, $k_d = 0.3$

References

- [1] C. Andrieu, J. Fernando G. de Freitas, and A. Doucet. Reversible jump MCMC simulated annealing for neural networks. pages 11–18. 2000. 3, 16
- [2] C. Andrieu, N. de Freitas, A. Doucet, and M. I. Jordan. An introduction to MCMC for machine learning. *Machine Learning*, 2003. 8, 9, 12
- [3] C. Archambeau, M. Opper, and J. Shawe-Taylor. NIPS 2006 workshop on dynamical systems, stochastic processes and Bayesian inference, 2006. <http://www.cs.ucl.ac.uk/staff/C.Archambeau/dsb.htm>. Held in Whistler, BC, Canada. 3
- [4] A. Bhan and E. Mjolsness. Static and dynamic models of biological networks. *Complexity*, 11:11–18, 2006. 2
- [5] R. J. Boys, D. J. Wilkinson, and T. B. L. Kirkwood. Bayesian inference for a discretely observed stochastic kinetic model. *Statistics and Computing*, 18(2):125–135, 2008. 3, 16

- [6] K. Chu, Y. Deng, and J. Reinitz. Parallel simulated annealing by mixing of states. *Journal of Computational Physics*, 148:646–662, 1999. 2
- [7] G. Eyink, J. Restrepo, and F. Alexander. A mean field approximation in data assimilation for nonlinear dynamics. *Physica*, D:347–368, 2004. 3
- [8] S. Geman and D. Geman. Stochastic relaxation, Gibbs distributions, and Bayesian restoration of images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 6(6):721–741, 1984. 9
- [9] H. Genrich. Predicate/transition nets. *Advances in Petri nets*, 1:208–247, 1986. 17
- [10] G. J. Gibson and E. Renshaw. Estimating parameters in stochastic compartmental models using Markov chain methods. *Math Med Biol*, 15:19–40, 1998. 3, 16
- [11] D. T. Gillespie. Approximate accelerated stochastic simulation of chemically reacting systems. *J. Phys.Chem.*, 115:1716–1733, 2001. 3
- [12] Daniel T. Gillespie. Exact stochastic simulation of coupled chemical reactions. *Journal of Physical Chemistry*, 81:2340–2361, 1977. 6
- [13] Andrew Golightly and Darren J. Wilkinson. Bayesian inference for stochastic kinetic models using a diffusion approximation. *Biometrics*, 61(3):781–788, 2005. 2
- [14] Peter J. Green. Reversible jump Markov chain Monte Carlo computation and Bayesian model determination. *Biometrika*, 82:711–32, 1995. 3, 16
- [15] W. K. Hastings. Monte Carlo sampling methods using Markov chains and their applications. *Biometrika*, 57:97–109, 1970. 3, 9
- [16] R. Kannan and Guangxing L. Sampling according to the multivariate normal density. *Foundations of Computer Science, 1996. Proceedings., 37th Annual Symposium on*, pages 204–212, 1996. 12
- [17] K.N. Kozlov and A.M. Samsonov. New migration scheme for parallel differential evolution. *5th International Conference on the Bioinformatics of Genome Regulation and Function (BGRS-2006)*, 2:141–144, 2006. 2

- [18] J. Lam and J.-M. Delosme. An efficient simulated annealing schedule: Derivation. Technical Report 8816, Yale Electrical Engineering Department, 1988. 9
- [19] N. D. Lawrence and M. Rattray. Parameter estimation in systems biology, March 2007. Workshop, summarized in <http://www.cs.manchester.ac.uk/ai/pesb07/>. Hosted by the School of Computer Science, University of Manchester. 3
- [20] D. J. C. MacKay. *Information Theory, Inference and Learning Algorithms*. Cambridge University Press, Cambridge, U.K., 2003. 8, 9, 11, 12
- [21] G. Marnellos. *Gene Network Models Applied to Questions in Development and Evolution*. PhD thesis, Yale University, New Haven, CT, USA, 1997. 2
- [22] D. C. Mattis and M. L. Glasser. The uses of quantum field theory in diffusion-limited reactions. *Reviews of Modern Physics*, 70:979–1001, 1998. 5
- [23] P. Mendes and D. B. Kell. Non-linear optimization of biochemical pathways: applications to metabolic engineering and parameter estimation. *Bioinformatics*, 14:869–883, 1998. 2
- [24] E. Mjolsness, C. D. Garrett, J. Reinitz, and D. H. Sharp. Modeling the connection between development and evolution: Preliminary report. *Evolution and Biocomputation: Computational Models of Evolution*, 899:103–122, 1995. 2
- [25] E. Mjolsness, D. H. Sharp, and J. Reinitz. A connectionist model of development. *J. Theoret. Biol.*, 152(4):429–453, 1991. 2
- [26] E. Mjolsness and G. Yosiphon. Stochastic process semantics for dynamical grammars. *Annals of Mathematics and Artificial Intelligence*, 47(3–4), 2006. 4
- [27] Y. F. Nanfack, J. A. Kaandorp, and J. Blom. Efficient parameter estimation for spatio-temporal models of pattern formation: Case study of *Drosophila melanogaster*. *Bioinformatics*, 23:3356–3363, September 2007. 2

- [28] A. Phillips and L. Cardelli. A correct abstract machine for the stochastic pi-calculus. *In Proceedings of Concurrent Models in Molecular Biology (Bioconcur'04)*, 2004. 17
- [29] P. Prusinkiewicz and A. Lindenmeyer. *The Algorithmic Beauty of Plants*. 1990. 3, 16
- [30] N. Przulj, D. G. Corneil, and I. Jurisica. Modeling interactome: Scale-free or geometric? *Bioinformatics*, 20:3508–3515, 2004. 2
- [31] R. Raffard, K. Amonlirdviman, J.D. Axelrod, and C. Tomlin. An adjoint-based parameter identification algorithm applied to planar cell polarity signaling. *Automatic Control, IEEE Transactions on*, 53:109–121, Jan 2008. 2
- [32] J. Reinitz, E. Mjolsness, and D. H. Sharp. Model for cooperative control of positional information in *Drosophila* by *bcd* and maternal *hb*. *Journal of Experimental Zoology*, 271:47–56, 1995. 2
- [33] S. Reinker, R.M. Altman, and J. Timmer. Parameter estimation in stochastic biochemical reactions. *Systems Biology, IEE Proceedings*, 153:168–178, 2006. 3, 16
- [34] H. Risken. *The Fokker-Planck Equation*. Springer, 1984. 6
- [35] J. Schlecht, K. Barnard, E. Spriggs, and B. Pryor. Inferring grammar-based structure models from 3d microscopy data. *Computer Vision and Pattern Recognition. CVPR '07. IEEE Conference on*, 1:1–8, June 2007. 3
- [36] N. G. van Kampen. *Stochastic Processes in Physics and Chemistry*. North-Holland, 1981. 5
- [37] A. V. Werhli and Dirk Husmeier. Reconstructing gene regulatory networks with Bayesian networks by combining expression data with multiple sources of prior knowledge. *Statistical Applications in Genetics and Molecular Biology*, 6(1):1716–1733, 2007. 2
- [38] Darren J. Wilkinson. *Stochastic Modelling for Systems Biology*. Chapman & Hall/CRC Press, Boca Raton, Florida, 2006. 7
- [39] G. Yosiphon and E. Mjolsness. Plenum - a dynamical grammar interpreter/simulator, 2008. Software package written for Mathematica computer algebra system. Available at

<http://computableplant.ics.uci.edu/>. Last accessed June 19, 2008.
13

- [40] L. Zhang. *Dynamic Biological Signal Pathway Modeling and Parameter Estimation through Optimization*. PhD thesis, University of California, Irvine, 2008. 2