A MARKOV MODEL BASED ANALYSIS OF STOCHASTIC BIOCHEMICAL SYSTEMS

Preetam Ghosh*, Samik Ghosh, Kalyan Basu and Sajal K Das
Biological Networks Research Group,
Dept. of Comp. Sc. & Engg., University of Texas at Arlington, TX-76010
Email: {ghosh, sghosh, basu, das}@cse.uta.edu

The molecular networks regulating basic physiological processes in a cell are generally converted into rate equations assuming the number of biochemical molecules as deterministic variables. At steady state these rate equations gives a set of differential equations that are solved using numerical methods. However, the stochastic cellular environment motivates us to propose a mathematical framework for analyzing such biochemical molecular networks. The stochastic simulators that solve a system of differential equations includes this stochasticity in the model, but suffer from simulation stiffness and require huge computational overheads. This paper describes a new markov chain based model to simulate such complex biological systems with reduced computation and memory overheads. The central idea is to transform the continuous domain chemical master equation (CME) based method into a discrete domain of molecular states with corresponding state transition probabilities and times. Our methodology allows the basic optimization schemes devised for the CME and can also be extended to reduce the computational and memory overheads appreciably at the cost of accuracy. The simulation results for the standard Enzyme-Kinetics and Transcriptional Regulatory systems show promising correspondence with the CME based methods and point to the efficacy of our scheme.

1. INTRODUCTION

The research challenge of today is to develop a comprehensive modeling framework integrating molecular, genetic and pathway data for a quantitative understanding of physiology and behavior of biological processes at multiple scales. The complexity of the biological process at molecular level is enormous due to the vast number of molecular state spaces possible in a cell and the large number of state transitions. Computational cell biology currently models the biological system as an aggregate functional state where the underlying molecular transitions are not captured. Hence, these models can only provide understanding for some specific problems at a functional level but not at the molecular dynamics level.

Spatio-temporal models capturing the temporal and spatial dynamics of biological processes [1, 2] at a molecular level can be classified as follows:

(1) mesoscale dynamics,
(2) cellular/organ-level stochastic simulation
(3) rule based model

Existing quantum mechanics and molecular dynamics based models are limited in scope, as they cannot handle the complexity of an entire cell or a complex pathway. The former captures the random environment of the cell at electron level and is very useful to understand the structure of the macromolecules but can only handle about 1000 atoms. The molecular mechanics model uses force field methods to understand the function of the macromolecules. This model is used to study the binding site configurations for protein-protein or protein-DNA interactions and protein folding. Currently it can handle about 1 million atoms and hence is not sufficient to model a cell or complex pathways.

The models for mesoscale dynamics, and cellular/organ-level stochastic simulation focus on a narrow range of biological components such as the wave model [1] for ventricular fibrillation in human heart, neural network signaling model [2] to control the onset of sleep in humans, and simulation frameworks like E-Cell [4] and Virtual Cell [3]. Mesoscale dynamics deal with rate equation based kinetic models and uses continuous time deterministic techniques. This model solves a complex set of differential equations corresponding to chemical reactions of the pathways. Since a biological system involves a large number of such equations, the model can only solve a system of at most 1000 reactions.

To address the observed stochasticity in a biological system [11, 12] Gillespie [5] extended the rate based model to a stochastic simulation framework. This led to a few other variations such as Kitano's Cell Designer [10], DARPA's BioSpice [9], Cell Illustrator [8] etc. The computational overhead of this simulation

[*] Corresponding author.

forced the use of approximation techniques to solve the rate equations by sacrificing accuracy e.g. the Tau Leap algorithm [6, 7]. Gillespie's technique considers the biochemical system as a discrete Markov process but suffers from the following limitation:

- It assumes that a biological system only consists of different biochemical reactions. Hence, each reaction event is abstracted by the experimentally determined rate constant and cannot incorporate the pertinent details of that biological event. For example, ideally a protein-ligand docking event should incorporate some details of the protein/ligand docking site location which is considered by our protein-ligand docking model presented in [24]. Because our models presented in [25, 23, 24] are parametric, we can easily estimate the kinetic parameters even in cases where such experimental data are not available.

Due to the large number of protein complexes in a cell, these stochastic simulation models lead to a combinatorial explosion in the number of reactions, thus making them unmanageable for complex metabolic and signaling pathway problems. The simulation model we propose here builds on the Gillespie technique and allows for many novel approximation techniques which cannot be implemented in the Gillespie simulation. Moreover, the flexibility of using different mathematical abstractions for different biological events make our technique more attractive than the π-calculus [31, 32] modeling technique.

Finally, the rule based simulation [13] models the multi cell interactions at a molecular level and addresses the more complex host-pathogen interactions. It ignores the stochastic nature of biological functions and considers a set of rules derived from pathways. In this paper, we convert the biological process into a stochastic network and solve it as a stochastic network analysis problem. Stochastic discrete event simulation is another way of addressing this problem as we have described in [22].

2. STOCHASTIC BIOCHEMICAL SYSTEM ANALYSIS

In a stochastic biochemical system, the state of the system at any time is defined by the number of molecules of each type. The transition from one state to another is derived from the probability of the reactions at the current state and the resulting next state is the new molecular state. As the molecular reactions in a biological process occur due to the random collision of the molecules, the state transition parameters are random and the state space is discrete. Let us assume in a stochastic biochemical system there are M elementary (monomolecular or bimolecular) irreversible reaction channels, which react at random times. A monomolecular reaction converts a reactant molecule into one or more product molecules. A bimolecular reaction converts two reactant molecules into one or more product molecules. We can decompose a reaction channel that involves more than two reactant molecules into a cascade of elementary reaction channels and model a reversible reaction channel by two irreversible reaction channels. The state of a stochastic biochemical system at time t is characterized by the M-dimensional random vector

$$Z(t) = [Z_1(t)Z_2(t)...Z_M(t)]^T$$

where $Z_m(t) = z$, if the m^{th} reaction has occurred z times during the time interval $[0, t)$ and T denotes vector or matrix transposition. The random variable $Z_m(t)$ is referred to as the degree of advancement (DA) of the m^{th} reaction [14]. Also $X_n(t)$ denotes the number of molecules of the n^{th} reactant or product species present in the system at time t. By assuming N distinct species, we have

$$X(t) = [X_1(t)X_2(t)...X_N(t)]^T$$

Given that the biochemical system is at state $X(t) = x$ at time t, let $q_m(x)$ be the number of all possible distinct combinations of the reactant molecules associated with the m^{th} reaction channel when the system is at state x. Note that

$$q_m(x) = \left\{ \begin{array}{cc} x_i, & \text{for monomolecular reactions} \\ x_i(x_i-1)/2, & \text{for bimolecular reactions} \\ & \text{with identical reactants} \\ x_i x_j, & \text{for bimolecular reactions} \\ & \text{with different reactants} \end{array} \right\}$$

for some $1 \leq i, j \leq N, i \neq j$. Moreover, let $c_m > 0$ be the probability per unit time that a randomly chosen combination of reactant molecules will react through the m^{th} reaction channel. This probability is known as the specific probability rate constant of the m^{th} reaction. Then, the probability that one m^{th} reaction will occur during a time interval $[t, t+dt)$ will approximately be equal to $\pi_m(x)dt$, for a sufficiently small dt, where

$$\pi_m(x) = c_m q_m(x),\ m \in M = \{1, 2, ..., M\},$$

is known as the propensity function of the m^{th} reaction channel [15, 16]. Note that, given the state $z(t)$ of the biochemical system at time t, we can uniquely determine the state $x(t)$ of the system at time t as

$$X_n(t) = g_n(Z(t)) = x_{0,n} + \sum_{m \in M} s_{nm} Z_m(t), \quad t \geq 0, \tag{1}$$

where $x_{0,n}$ is the initial number of molecules of the n^{th} species present in the cell at time $t = 0$ and s_{nm} is the stoichiometric coefficient. This coefficient quantifies the change in the number of molecules of the n^{th} molecular species caused by one occurrence of the m^{th} reaction. The state $z(t)$ cannot be determined from $x(t)$ in general since there might be several states $z(t)$ that lead to the same state $x(t)$. To distinguish $Z(t)$ from $X(t)$, all existing works on stochastic simulation refer to $Z(t)$ as the hidden state and to $X(t)$ as the observable state and use a hidden markov model to analyze the system.

The discrete-valued random process

$$Z = \{Z(t), t \geq 0\}$$

characterizes the dynamic evolution of the hidden state of a biochemical system. This process is specified by the probability mass function (PMF)

$$P_z(z;t) = Pr[Z(t) = z | Z(0) = 0],$$

for every $t \geq 0$. Simple probabilistic arguments show that $P_z(z;t)$ satisfies the following first-order differential equation [17]:

$$\frac{\partial P_z(z;t)}{\partial t} = \sum_{m \in M} \alpha_m(z - e_m) P_z(z - e_m; t) - \alpha_m(z) P_z(z;t),$$

for $t > 0$, with initial condition $P_z(0;0) = 1$, where e_m is the m^{th} column of the $M \times M$ identity matrix and

$$\alpha_m(z) = \pi_m(g(z)) = c_m q_m(g(z)),$$
$$g(z) = [g_1(z) g_2(z) ... g_N(z)]^T$$

This is the well-known forward Kolmogorov differential equation [18–20] governing the stochastic evolution of a continuous-time Markov chain. In computational biochemistry, Eqn. 1 is referred to as the chemical master equation (CME) [14]. It turns out that Z is a multivariate birth process [18, 20] and X is a multivariate birth-death process.

3. OUR MARKOV CHAIN FORMULATION

We replace the hidden markov model by a Markov Chain based approach to model a composite biochemical system. Note that the system only represents biochemical reactions or protein-ligand docking events in the cell. Thus in the Markov Chain, each state transition occurs due to *one* reaction/docking event. If multiple reaction/docking events are possible, then the state transitions can occur due to any one of these events and hence we can have multiple transition paths to the next state. The *states* in the Markov Chain are defined as the number of molecules of the different components in the biological system, i.e., $X(t) = [X_1(t), X_2(t), ..., X_N(t)]$. For example, consider the following biochemical system:

$$R_1 : X_1 + X_2 \longrightarrow X_3; \qquad R_2 : X_2 + X_4 \longrightarrow X_5$$

where, X_1, X_2, X_4 are proteins and X_3, X_5 denote the docked complexes. Then each state in the Markov Chain will have 5 tuples corresponding to the number of molecules of these 5 components. The corresponding Markov Chain with the possible state transitions is shown in Fig 1. Note that each transition signifies either an R_1 or an R_2 type of event. Thus, the total number of edges coming out of each node is given by the possible number of reaction/docking events (and equivalently the number of differential equations) considered in the system.

3.1. The MFPT concept

Assuming first order kinetics, the probability that a particle has reached the final state at some time t is given by $P_f(t) = 1 - e^{-kt}$, where k is the rate, and $P_f(t)$ is the probability of reaching a final state by time t. By running many independent simulations shorter than $1/k$, we can estimate the cumulative distribution $P_f(t)$, and fit the value for the rate, k. The mean first passage time is the average time when a particle reaches the final state for the first time, given that it is in an initial state at $t = 0$,

$$MFPT = \int_{t=0}^{\infty} (\frac{d}{dt} P_f(t)) t \, dt = \int_{t=0}^{\infty} kte^{-kt} dt = \frac{1}{k}$$

3.2. Computing the state transition probabilities and times

Note that computing the MFPT requires an estimate of each state transition probability along with the time taken for the transition. Because, each

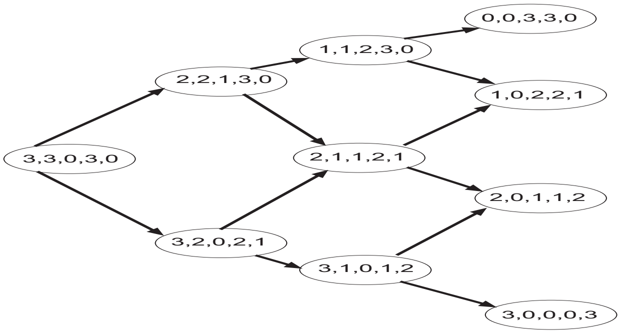

Fig. 1. Markov Chain formulation with 3 molecules each of X_1, X_2, X_4 and no X_3, X_5 molecules initially.

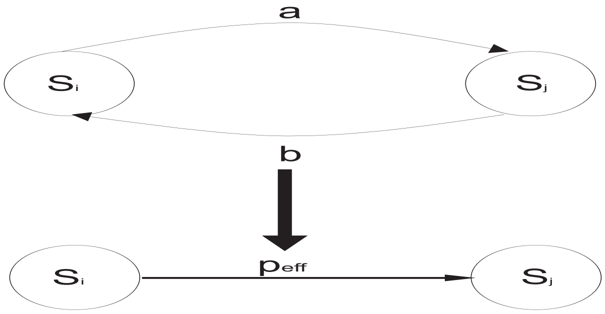

Fig. 2. A simple birth-death model for reversible reactions.

state transition signifies either a reaction or docking, we can find the state transition probabilities and times from the *batch models* of the reaction [23, 25] and docking [24] events using concepts from collision theory. The batch model incorporates the number of molecules of each reactant present before the start of the reaction/docking events. This makes each state transition depend upon the current state that the system is in. Note that the batch model estimates the time of reaction/docking as a random variable following a Gamma distribution when few reactant molecules are present in the system. However, as the number of reactant molecules increase, the mean-to-standard deviation ratio for time becomes close to 1 signifying an exponential distribution. Also, note that [24] reports that the docking time is primarily affected by the collision theory component. Hence the batch models of [23, 25] are also applicable to the docking events.

3.2.1. *Monomolecular reactions*

The time taken for monomolecular reactions can be simply computed from the experimentally determined reaction rate constant for the reaction. Denoting the reaction rate constant by k_{R_3}, the probability of reactions of type R_3 (denoted by P_{R_3}) is given by:

$$R_3: \; X_6 \rightarrow X_7 + X_8; \quad P_{R_3} = [X_6]k_{R_3}\tau$$

where $[X_6]$ denotes the concentration of X_6 type of molecules and τ denotes a infinitely small time step (generally in the order of $\sim 10^{-8}$ secs). Note that this definition of the monomolecular reaction probability is exactly the same as that used for solving the CME and can be defined as the probability of a reaction of type R_3 occurring in time τ.

Time taken for completing R_3 (denoted by T_{R_3}) can be estimated from the rate constant as follows:

$$T_{R_3} = \frac{1}{[X_6]k_{R_3}}$$

In [23, 25] we have shown that the reaction time is a *random variable* following an exponential distribution when there are sufficient number of molecules in the system. Hence, we assume that the monomolecular reaction completion time also follows an exponential distribution with mean T_{R_3}.

3.2.2. *Bimolecular reactions*

We use the batch model developed in [25] for computing the probability of reaction and first and second moments of the reaction completion times. Considering reaction R_1, the probability and time can be estimated as:

$$P_{R_1} = \frac{n_1 n_2 r_{12}^2 \tau}{V}\sqrt{\frac{8\pi k_B T}{m_{12}}}\, e^{\frac{-E_{A12}}{k_B T}}; \quad T_{R_1} = \frac{\tau}{p_{R_1}}$$

where, n_1, n_2 are the numbers of X_1 and X_2 type molecules present in the cell, r_{12} is the collision radius computed as the sum of the radii of X_1 and X_2 molecules (which are assumed to be spherical), m_{12} is the reduced mass computed as $m_{12} = \frac{m_1 m_2}{m_1 + m_2}$ (where m_1, m_2 are the masses in gm of X_1 and X_2 type molecules), V is the cell volume, T is the temperature (in Kelvin), k_B is the Boltzmann's constant $= 1.381 \times 10^{-23} kg\ m^2/s^2/K/molecule$ and E_{A12} is the activation energy required for reaction R_1. T_{R_1} denotes the *mean* of the reaction completion time which is assumed to follow an exponential distribution. Note that the Gillespie simulator also considers the reaction time to be a random variable following the exponential distribution.

In [23], we have shown that the mean of the reaction time (T_{R_1}) is actually equal to the time reported by the rate equation based model. Hence, denoting the rate of reaction R_1 by k_{R_1}, we have:

$$T_{R_1} = \frac{1}{n_1 n_2 k_{R_1}}$$

Hence the probability of reaction can also be computed if one does not know the activation energy for any specific reaction but the rate constant is known.

As before, reactions involving multiple copies of any molecule type can be represented by a cascade of elementary reactions of the above types.

3.2.3. *Reversible Reactions*

The Gillespie simulator considers reversible reactions as two separate reactions. This increases the complexity of the system as more number of reactions need to be handled. Also, in our Markov Chain based model, a reversible reaction will involve a double edge between any two nodes making the MFPT computations difficult. Hence we can approximately characterize reversible reactions using a simple birth-death model as shown in Fig 2.

Let us denote the forward and backward transition probabilities between any two states S_i and S_j by a and b respectively. We need to compute the *effective* probability that the reaction proceeds in the forward direction denoted by P_{eff} such that the double edge can be replaced by a single edge driving the reaction in the forward direction with probability P_{eff}. However, the time for the forward reaction still remains the *same* and can be computed as above. The computation of P_{eff} will be different for the monomolecular and bimolecular reaction scenarios. In general, P_{eff} can be expressed by:

$$P_{eff} = P(S_i) \times a - P(S_j) \times b$$

where, $P(S_i)$ and $P(S_j)$ are the probabilities of being in states S_i and S_j respectively. However, $P(S_i)$ and $P(S_j)$ does not simply depend on a and b, but also on the transition probabilities of edges into and out of nodes S_i and S_j making the P_{eff} estimation quite complicated. In the following, we show two approximate schemes of computing P_{eff} for monomolecular and bimolecular reactions.

Monomolecular reactions: Consider reversible reactions of type R_1, i.e., $X_1 + X_2 \leftrightarrow X_3$. In this case, the probabilities of forward and backward reactions (a and b) can be computed as discussed before. We approximate P_{eff} as $P_{eff} = a - b$ in such cases. Note that this approximation assumes that $P(S_i) \approx P(S_j)$ for all the reversible reactions in the system. While this indeed is a gross simplification of the reversible reaction kinetics, the results obtained show that it is not overly restrictive. Moreover, when $a \approx b$, we assume that the reversible reaction attains equilibrium and make node S_i a sink i.e., no further state transitions can originate from this node.

Bimolecular reactions: Consider reversible reactions of type R_4 as follows:

$$R_4: \quad X_9 + X_{10} \leftrightarrow X_{11} + X_{12}$$

Here also we can use the above approximation of $P(S_i) \approx P(S_j)$ and compute $P_{eff} = a - b$.

3.3. Pruning the Markov Chain

As mentioned before, we will estimate the time taken to reach *any* node in the markov chain by using the MFPT. Hence, we consider each node in the chain as a sink to compute its MFPT. Also, it has to be ensured that every node in the Markov Chain is able to reach the sink. Otherwise, since these nodes will have an infinite mean first passage time, calculations done on the Markov Chain will fail. We identify the nodes that can reach the sink by performing a depth first search from the sink over the incoming edges, and marking all nodes that are reachable. The nodes that were not marked can be simply deleted, thus ensuring that all nodes in the Markov Chain can reach a node in the final state. Next, we normalize the probabilities on all the edges so that on each node, the sum of the probabilities for all outgoing edges is one as follows:

$$P_{ij}^{new} = \frac{P_{ij}}{\sum_{edge_k} P_{ik}}$$

The probability on each edge equals the number of times that transition was made divided by the total number of transitions from that node.

3.4. Computing the total probability of reaching a final state

The Markov Chain consists of a set of nodes and a set of transitions or edges between these nodes. Each edge has a probability associated with it as well as the time taken to traverse this edge. We define the P_{sink} of a node as the probability that the system starting in the initial state would reach the sink state before reaching the initial state again. Following [21] we will use the Markov Chain to calculate the P_{sink} values. The P_{sink} can be defined conditionally based on the first transition made from the node as follows:

$$P_{sink}(node_i) = \sum_{transition(i,j)} P(transition(i,j)) \\ \times P_{sink}(node_i | transition(i,j))$$

where the sum is over all possible transitions (that are mutually exclusive) from $node_i$. The possible transitions from $node_i$ are simply all of the edges leading from $node_i$, and the probability of each of these transitions is the P_{ij} values defined previously. This satisfies the above condition. $P_{sink}(node_i | transition(i,j))$ is simply the P_{sink} of $node_j$ which results in the following equations:

$$P_{sink}(node_i) = \sum_{edge_{ij}} P_{ij} P_{sink}(node_j),$$
$$P_{sink}(node_i) = 1, \quad node_i \in sink,$$
$$P_{sink}(node_i) = 0, \quad node_i \in source$$

Thus the probability of reaching any node in the chain can be estimated by a simple recursive procedure that traverses the chain. Note that in the worst case, the chain becomes a tree, where each node can traverse to M *different* new nodes (M being the number of reactions considered). Hence the worst case time complexity of traversing the chain is $O(V+E) \approx O(E)$, where V, E are the number of vertices and edges of the chain. This is because the number of edges is generally greater than the number of vertices in the chain. In the worst case we might have a tree where $E = V - 1$. Also, as the probability has to be computed for each node in the chain, we have an overall complexity of $O(VE)$.

3.5. Computing the MFPT for reaching the final state

We define the mean first passage time (MFPT) of any node in the chain as the average time taken to reach that node (considered the sink) from the first node in the chain. The MFPT is defined conditionally based on the first transition made from any node:

$$MFPT(node_i) = \sum_{transition_{ij}} P(transition(i,j)) \\ \times MFPT(node_i | transition(i,j))$$

where the sum is over all possible transitions from $node_i$. The MFPT of $node_i$ given that a transition to $node_j$ was made, is the time taken to go from $node_i$ to $node_j$ added to the MFPT from $node_j$:

$$MFPT(node_i) = \sum_{edge_{ij}} P_{ij}(time_{ij} + MFPT(node_j)) \tag{2}$$

where the sum is over all edges leading from $node_i$. Also, we can define the initial conditions as follows:

$$MFPT(node_i) = \infty, \quad node_i \notin sink$$
$$MFPT(node_i) = 0, \quad node_i \in sink$$

Note that *time* is a random variable, and hence cannot be added as shown in the equations above. Hence we need to compute the convolution of exponential distributions that has to replace a simple addition of this random variable. Equivalently, it should be understood that *the MFPT is no longer fixed, but is also a random variable.*

We need general expressions for the following two types of convolutions of exponential distributions:

(1) General expression for $n+1$-fold convolution of exponential variables from an n-fold convolution for the $(time_{ij} + MFPT(node_i))$ component of Eqn 2:

$$f_n = a_1^n e^{-\frac{x}{T_1}} + a_2^n e^{-\frac{x}{T_2}} + ... + a_n^n e^{-\frac{x}{T_n}}$$
$$f_{n+1} = \frac{T_1}{T_1 - T_{n+1}} a_1^n e^{-\frac{x}{T_1}} + \frac{T_2}{T_2 - T_{n+1}} a_2^n e^{-\frac{x}{T_2}} \\ + ... + \frac{T_n}{T_n - T_{n+1}} a_n^n e^{-\frac{x}{T_n}} - [\frac{T_1}{T_1 - T_{n+1}} a_1^n \\ + \frac{T_2}{T_2 - T_{n+1}} a_2^n + ... + \frac{T_n}{T_n - T_{n+1}}] e^{-\frac{x}{T_{n+1}}}$$

$$\Rightarrow f_{n+1} = a_1^{n+1} e^{-\frac{x}{T_1}} + a_2^{n+1} e^{-\frac{x}{T_2}} + ... \\ + a_{n+1}^{n+1} e^{-\frac{x}{T_{n+1}}}$$

where, $T_1, T_2, ..., T_n$ denote the means of the reaction times of each edge of the n-fold convolution (convolution of the times for n edges gives an n-fold convolution), and $T_{n+1} = time_{ij}$ in the $(time_{ij} + MFPT(node_i))$ component of Eqn 2. While the above expression gives the general distribution for the $n+1$-fold convolution, the first and second moments can also be generically expressed as follows:

$$\text{First Moment} = F^{n+1} = a_1^{n+1}(T_1)^2 + a_2^{n+1}(T_2)^2 \\ + ... + a_{n+1}^{n+1}(T_{n+1})^2$$

$$\text{Second Moment} = S^{n+1} = a_1^{n+1}(T_1)^3 + a_2^{n+1}(T_2)^3 + ... + a_{n+1}^{n+1}(T_{n+1})^3$$

After a few manipulations it can be shown that the first and second moments of this general distribution reduces to:

$$F^{n+1} = T_1 + T_2 + ... + T_{n+1};$$
$$S^{n+1} = S^n + T_{n+1}(\sum_{i=1}^{n+1} T_i);$$
$$S^1 = (T_1)^2$$

(2) General expression for a convolution between an n-fold convolution (f_n) and an m-fold convolution (g_m) for the $(\sum_{edge_{ij}})$ component of Eqn 2:

$$f_n \otimes g_m = \sum_{j=1}^{m} \sum_{i=1}^{n} a_i^n a_j^m \left(\frac{e^{-\frac{x}{T_i^n}} - e^{-\frac{x}{T_j^m}}}{\frac{1}{T_j^m} - \frac{1}{T_i^n}} \right)$$

Note that the above expression contains $m + n$ terms in total and the first and second moments of this general distribution can also be computed in a similar manner as before.

Moreover, because of the simplified expression for the first moment of the MFPT, we can use the same expression as in Eqn 2 if one is only interested in the *mean* value of the MFPT itself. In the next section we report the results based on this mean value of the MFPT distribution. However, it is also possible to compute the exact MFPT distribution of each node in the chain.

It should be noted that the above expressions for the general distribution of the MFPT and corresponding first and second moments were derived assuming $T_i \neq T_j$, for all i, j. This will be true for most cases as it is quite unlikely that the mean of the reaction times are equal (because the mean also depends on the concentration of the reactant molecules and most states in the chain will have different concentrations of the particular reactants of the specific reaction). However, in certain cases, the mean reaction times might be equal and we need to add a small δ to make them different such that the above reactions remain valid. Consider a 2-fold convolution of exponentially distributed random variables with means T_1 and T_2. If $T_1 = T_2$, the general distribution takes the form $\frac{xe^{-\frac{x}{T_1}}}{T_1}$, and when $T_1 \neq T_2$, it is of the form $\frac{(e^{-\frac{x}{T_1}} - e^{-\frac{x}{T_2}})}{T_1 - T_2}$. However, with $\delta = T_1 - T_2$, we can show that

$$\lim_{\delta \to 0} \frac{(e^{-\frac{x}{T_1}} - e^{-\frac{x}{T_2}})}{T_1 - T_2} = \frac{xe^{-\frac{x}{T_1}}}{T_1}$$

Hence, smaller the value of δ, the more precise are the results obtained.

3.6. Approximating the Markov Chain: Reducing complexity at the cost of accuracy

In most cases, it is not possible to derive an analytical solution of the CME. The following approximation techniques have been proposed to reduce the complexity of the CME:

(1) Langevin approximation (LA) [16]: A useful approximation to the CME is obtained by assuming that there exists a time step dt such that the following two conditions are satisfied:

- Changes in the hidden system states that occur during time interval $[t, t + dt)$ do not appreciably affect the propensity functions.
- The expected number of occurrences of each reaction in a time interval $[t, t + dt)$ is much larger than one.

It can be shown that, under both conditions, the dynamic evolution of the hidden state process is governed by a simpler system of stochastic differential equations that can be solved by the Monte Carlo estimates.

(2) Linear Noise approximation (LNA) [26, 27]: Unfortunately, the LA method does not allow us to obtain an expression for the joint probability density function (PDF) of the hidden states. However, by using additional approximations, the hidden states can be characterized by a multivariate Gaussian PDF that can be solved numerically (e.g., by the standard Euler method) and is faster than the Monte Carlo method. However, both the LA and LNA methods require both conditions (shown above) to be satisfied simultaneously which is not possible in most biological systems.

(3) Poisson approximation (PA) [28]: A better approximation of the HMM is obtained by employing a time step dt satisfying the first condition, but may not necessarily satisfy the second one. Since reactions that occur during the time interval $[kdt, (k + 1)dt)$ will not appreciably change

the values of the propensity functions, these reactions will occur independently of each other. Moreover, the number of occurrences of the m^{th} reaction during $[kdt, (k+1)dt)$ is assumed to be a Poisson random variable.

(4) Mean-Field approximation (MFA) [29]: The PA method does not allow us to derive an expression for the joint PMF of the hidden states. However, it is possible to approximately characterize the hidden states by a PMF by the dynamic evolution of the normal Gibbs distribution. This method is superior to the LNA method for three main reasons:

- It is based on the more accurate Poisson approximation,
- its approximation accuracy does not depend on the cellular volume, and
- it does not require linearization of the underlying propensity functions.

(5) Stochastic quasi-equilibrium approximation (SQEA) [30]: Most often, reactions occur on vastly different time scales e.g., the transcription and translation reactions are typically slow reactions, whereas dimerization is a fast reaction. This means that transcription and translation may occur infrequently, whereas, dimerization may occur numerous times within successive occurrences of slow reactions.

In such cases, the Gillespie algorithm spends most of the time simulating fast reaction events. It may, however, be less important to know the activity of fast reactions in detail since the system's dynamic evolution may be mostly determined by the activity of the slow reactions. Hence, it is possible to approximate the CME by one that involves only slow reactions.

In our Markov model formulation, we do not have any hidden states as the chain can be appropriately characterized by the number of different molecule types present in the system (denoting the states of the chain), and each state transition is characterized by the corresponding reaction/docking events. Hence, most of the above techniques are not directly applicable to this formulation. However, we can employ the SQEA approach to substantially simplify the markov chain (with lesser number of states) making the MFPT computations faster. In this case, the states of the markov chain will have the same tuples as before, however the *state transitions will only be governed by the slow reactions.* During each state transition, the new state in the chain is computed depending on this slow reaction and also computing how many fast reactions can occur in that time and appropriately updating the molecule counts of the reactants in the fast reactions.

In fact this technique has a direct analogy to Gillespie's tau-leap algorithm, wherein, we can specify a certain time step Δt, and compute how many reactions (both fast and slow) occur within that period. Thus we can compute the next state and the markov chain will become a 1-dimensional chain thereby greatly reducing the complexity. Also the memory requirements for storing the Markov chain can be completely removed as the MFPT can be computed online as the chain progresses in time.

4. RESULTS AND ANALYSIS

4.1. Enzyme-Kinetics system

Figs 3-5 show the molecular distributions of the product (P) molecules with time for different number of enzyme (E) and substrate (S) molecules. Note that it is possible to report the exact molecular distributions of any molecule type in the system using our approach. The time axis reports the mean value of the MFPT (which is also a random variable as discussed earlier). Fig 8 compares the dependency of mean number of P type molecules on time with that reported from an exact simulation of the CME (obtained from Monte Carlo simulation of the differential equations in the system). Our results compare very well with the exact simulation for low number of molecules in the system. With large number of enzyme molecules present, the reactions occur very fast and the markov model formulation being driven in discrete time produces less accurate results. Nevertheless, it is computationally very fast and allows the study of more complicated systems (with large number of reactions and molecular types involved).

Figs 6-7 plots the probability distributions of the product molecules. The different bars at each possible molecular count value of the P type molecules correspond to the probability of reaching different states (from the initial state) in the Markov model having that number of P type molecules (and different molecular count values for the other entities in the system). It is again possible to compute the

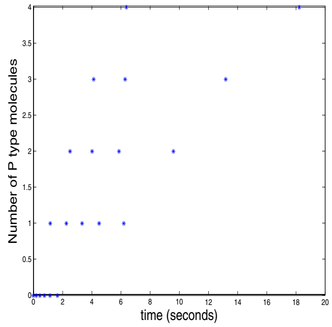

Fig. 3. Molecular distribution of P type molecules, with E=10, S=5.

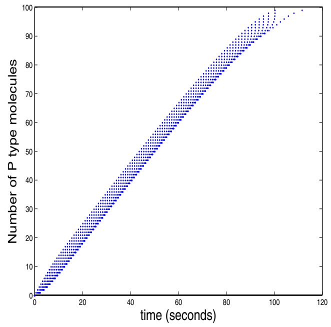

Fig. 4. Molecular distribution of P type molecules, with E=10, S=100.

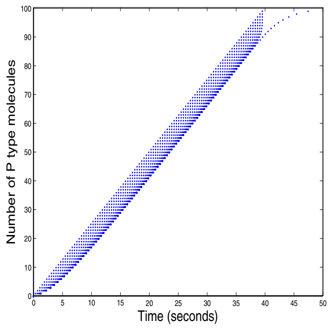

Fig. 5. Molecular distribution of P type molecules, with E=1000, S=100.

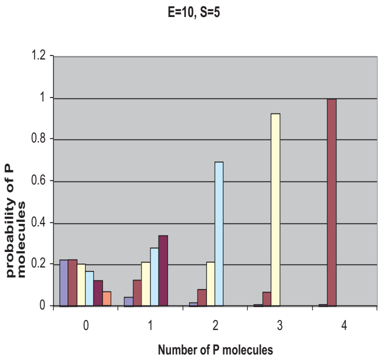

Fig. 6. Probability distribution of P type molecules, with E=10, S=5.

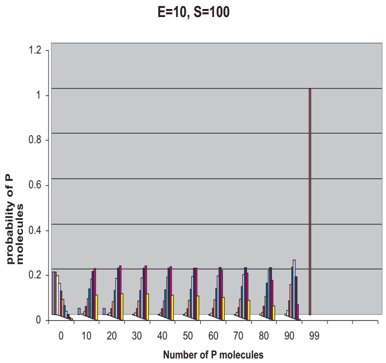

Fig. 7. Probability distribution of P type molecules, with E=10, S=100.

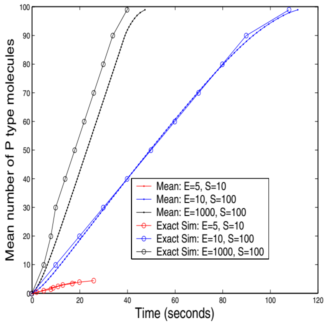

Fig. 8. Mean number of P type molecules, Our model Vs Exact Simulation.

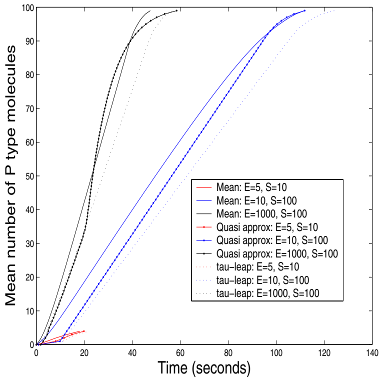

Fig. 9. Effects of SQEA and Tau-leaping approximations.

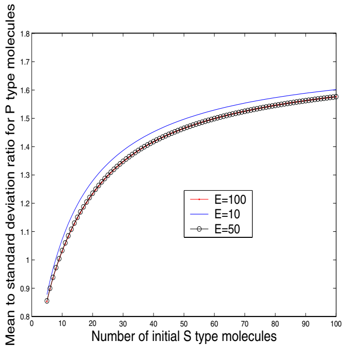

Fig. 10. Mean to standard deviation ratios of molecular distribution of P type molecules with constant number of enzyme molecules.

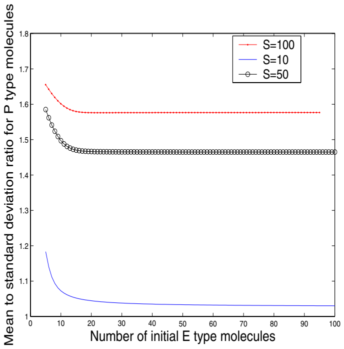

Fig. 11. Mean to standard deviation ratios of molecular distribution of P type molecules with constant number of substrate molecules.

complete distribution (not just the first and second moments) of all the different molecule types in the system with our formulation.

Fig 9 shows the effects of the SQEA (denoted by “quasi approx”) and tau-leap approximations to our markov model. The reversible reactions are considered fast reactions in our analysis. As expected, the SQEA approach provides a very accurate approximation of the mean number of product molecules whereas the tau-leap variation (with $\Delta t = 10^{-3}$ secs) provides the fastest (and most memory efficient) solution at the cost of accuracy.

Figs 10-11 plot the mean to standard deviation ratio of the molecular distribution of the product

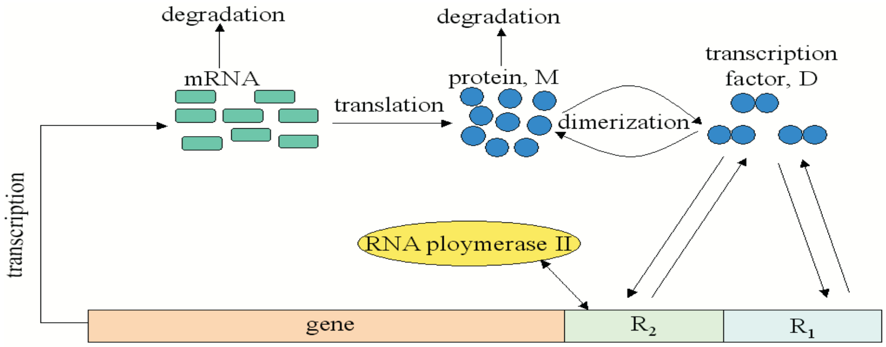

Fig. 12. A simple transcriptional regulatory system.

Fig. 13. Terminology for the Transcriptional Regulatory System.

M	Protein (monomer)
D	Transcription factor (dimer)
RNA	mRNA
DNA	DNA template free of dimers
$DNA.D$	DNA template bound at R_1
$DNA.2D$	DNA template bound at R_1 and R_2

Fig. 14. Reactions Associated with the Transcriptional Regulatory System.

	Reaction	Rate Constant
1	$RNA \rightarrow RNA + M$	$0.043s^{-1}$
2	$M \rightarrow \emptyset$	$0.0007s^{-1}$
3	$DNA.D \rightarrow RNA + DNA.D$	$0.0715s^{-1}$
4	$RNA \rightarrow \emptyset$	$0.0039s^{-1}$
5	$DNA + D \rightarrow DNA.D$	$0.02s^{-1}$
6	$DNA.D \rightarrow DNA + D$	$0.4791s^{-1}$
7	$DNA.D + D \rightarrow DNA.2D$	$0.002s^{-1}$
8	$DNA.2D \rightarrow DNA.D + D$	$0.8765 \times 10^{-11}s^{-1}$
9	$M + M \rightarrow D$	$0.083s^{-1}$
10	$D \rightarrow M + M$	$0.5s^{-1}$

molecules with varying number of *substrate* and *enzyme* molecules respectively. With less number of substrates, the stochastic resonance is quite high in the system (as the ratio is less than 1). With higher number of substrates, the ratio saturates at 1.5 implying lesser stochasticity in the system. Also, the stochasticity is not very much dependent on the number of enzyme molecules in the system as depicted in Fig 11. Thus from these plots we can infer that the stochastic resonance in the molecular distribution of the product molecules is primarily governed by the number of substrate molecules in the system.

4.2. Transcriptional Regulatory System

We next show the results for a simple transcriptional regulatory system as shown in Fig 12. Protein M, synthesized by transcription of a gene, dimerizes to the transcription factor D, which may bind to the gene's regulatory region at two binding sites, R_1 and R_2. The promoter coincides with R_2. Binding of D at R_1 activates transcription of M. However, binding of D at R_2 excludes the RNA polymerase from binding at the gene's promoter and in this case transcription is repressed. Fig 13 presents the terminology used for the different components of this example system, whereas Fig 14 shows the list of reactions involved along with their respective rate constants [29].

In this section, we present the results for the well known Enzyme Kinetics system governed by the following three elementary reactions:

$$E.S \rightarrow P + E, \quad E + S \leftrightarrow E.S$$

The rate constant for the reversible reaction pair is set at $1s^{-1}$ and that for the first reaction is $0.1s^{-1}$.

In this system as well, we find very good agreement between the exact simulation results with that from our model. Thus, for reaction-pairs $\{5, 6\}$ $\{7, 8\}$ and $\{9, 10\}$ we choose the forward reactions as 6, 7 and 10 respectively and drive the Markov Chain formulation accordingly. The accuracy of our system suffers from this approximation (hence the difference from the exact simulation results).

It should be noted that these results were generated for a low number of the different molecule types in the system. As the number of molecules increase, the MFPT based results are further off from the ex-

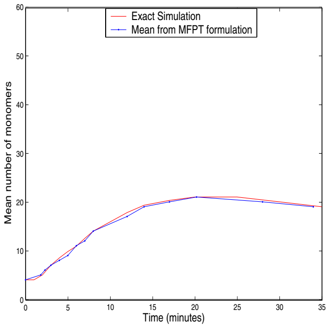

Fig. 15. Mean number of monomers: Exact Simulation Vs Our Model.

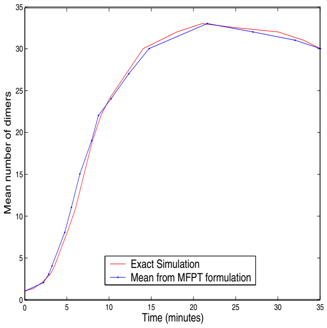

Fig. 16. Mean number of dimers: Exact Simulation Vs Our Model.

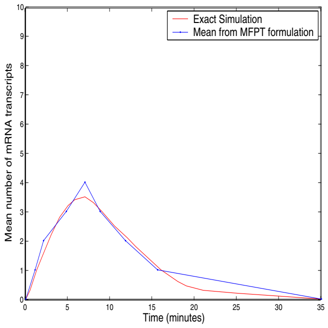

Fig. 17. Mean number of mRNA transcripts: Exact Simulation Vs Our Model.

act simulation results because of the approximations. Thus, our model allows for a computationally efficient implementation of a complex biochemical system simulation which can give accurate results when the number of molecules of the components in the system are small.

5. DISCUSSION

Here we make some comments regarding both the differential equation based and our discrete random process based approach for biological system modeling. The former approach is usually used to model the variations of the concentrations of biomolecules, where the latter models the variations of the number of biomolecules. As for any research problem for which there are a variety of feasible solutions, each of these approaches has its own pros and cons. For example, when the number of biomolecules is extremely large, it may not even be practical to use our discrete random process-based model because of the following reasons:

(1) the number of possible candidate states of a molecular entity, $X(t) \in \{0, 1, ...,$the maximum number of molecules$\}$, can be too huge to handle
(2) if a discretization strategy is used, accuracy of the model could be compromised.

No matter which model is used, some of the parameters (e.g., kinetic parameters for the differential equation based models) need to be estimated. The parametric models we have introduced for biochemical reactions and docking can estimate these parameters theoretically and can be used once we have sufficient fidelity in these models. However, the Markov model based approach presented in this paper will work for both cases i.e., by estimating the kinetic parameters through controlled experiments or by using the parametric models.

6. CONCLUSION AND FUTURE DIRECTIONS

We have introduced a Markov Chain based analysis technique as an alternative for complex biological process modeling. The main idea of this modeling is to transform the biological processes from a continuous deterministic process to a discrete random process. Because of its simplicity in comparison to solving numerically a large number of differential equations, our framework reduces the computational overhead and increases scalability considerably. We are currently working on a complex pathway model with many molecular types and with large number of molecules of each type to estimate the computational complexity. The main benefit of this analysis is to analyze the stochasticity of many reactions occurring together. Current experimental methods are not able to capture this measurement at a molecular level without special set-up.

The challenge in the model proposed here is the optimization of memory and computational speed of DFS and MFPT algorithms. Note that each node in the Markov Chain has an out-degree of M, where M is the number of reactions/docking considered in the system. The storage of an arbitrary graph with a large number of nodes and out-degree will have memory problems. It is also important to find appropriate simplifications and data structures to speed up the process. Can the chain be converted into a tree structure by eliminating (adding) pseudo nodes (edges) ? This will allow us to traverse the chain (during DFS or MFPT computations) in $O(log_M V)$ time. We have already stated that the tau-leap approximation on the chain reduces it to a 1-dimensional chain and the MFPT computations can be performed online. Also, can the tree structure be converted into a trie wherein the chain is compressed optimally thereby reducing the memory overheads ?

The complete cell model by this analysis may not be feasible due to the large number of molecules in the cell, but we expect that many complex biological systems can be modeled by this technique.

References

1. Making Sense of Complexity *Summary of the Workshop on Dynamical Modeling of Complex Biomedical Systems*, (2002).
2. Endy, D., and Brent, R. Modeling cellular behavior. *Nature.*, vol. 409, Jan 2001.
3. Loew, L. The Virtual Cell Project. *'In Silico' Simulation of Biological Processes (Novartis Foundation Symposium No. 247)*, Wiley, 207-221, 2002.
4. Tomita, M. et.al. The E-CELL Project: Towards Integrative Simulation of Cellular Processes. *New Generation Computing.*, (2000) 18(1): 1-12.
5. Gillespie, D. Exact stochastic simulation of coupled chemical reactions. *Journal of Physical Chemistry*, 81: 2340-2361.
6. Gillespie, D. Approximate accelerated stochastic simulation of chemically reacting systems. *Journal of Chemical Physics.*, 115(4): 1716-1733.
7. Rathinam, M., Petzold, L., Gillespie, D. Stiffness in Stochastic Chemically Reacting Systems: The Implicit Tau-Leaping Method. *Journal of Chemical Physics.*, 119 (24), 12784-12794, 2003.
8. Cell Illustrator, *www.fqspl.com.pl/life_science/cellillustrator/ci.htm*
9. BioSpice: open-source biology, *http://biospice.lbl.gov/home.html*
10. CellDesigner: A modeling tool of biochemical networks, *http://celldesigner.org/*
11. MacAdams, H., and Arkin. A. It is a noisy business! Genetic regulation at the nanomolar scale. *Trends in Genetics*, vol 15, pp 65-69, 1999.
12. Hasty, J., and Collins, J. Translating the Noise. *Nature, Genet.*, 2002, 31, 13-14.
13. Meier-Schellersheim, M., and Mack, G. SIMMUNE, a tool for simulating and analyzing immune system behavior. *CoRR cs.MA/9903017: (1999).*
14. vanKampen, N. Stochastic Processes in Physics and Chemistry. *Amsterdam: Elsevier*, 1992.
15. Gillespie, D. A Rigorous Derivation of the Chemical Master Equation. *Physica A*, vol. 188, pp. 404-425, 1992.
16. Gillespie, D. The Chemical Langevin Equation. *J. Chemical Physics*, vol. 113, no. 1, pp. 297-306, 2000.
17. Haseltine, E., and Rawlings, J. Approximate Simulation of Coupled Fast and Slow Reactions for Stochastic Chemical Kinetics. *J. Chemical Physics*, 117:15, pp. 6959-6969, 2002.
18. Karlin, S., and Taylor, H. A First Course in Stochastic Processes. *second ed. San Diego, Calif.: Academic Press*, 1975.
19. Karlin, S., and Taylor, H. A Second Course in Stochastic Processes. *San Diego, Calif.: Academic Press*, 1981.
20. Papoulis, A., and Pillai, S. Probability, Random Variables and Stochastic Processes. *fourth ed. New York: McGraw-Hill*, 2002.
21. Singhal, N. et al. Error analysis and efficient sampling in Markovian state models for molecular dynamics. *Jour. Of Chem. Physics.*, 2005.
22. Ghosh, S., Ghosh, P., Basu, K., Das, S., and Daefler, S. iSimBioSys: A Discrete Event Simulation Platform for 'in silico' Study of Biological Systems *Proc. of 39th IEEE Annual Simulation Symposium*, 2006, USA.
23. Ghosh, P., Ghosh, S., Basu, K., Das, S., and Daefler, S. An Analytical Model to Estimate the time taken for Cytoplasmic Reactions for Stochastic Simulation of Complex Biological Systems. *Proc. of the 2nd IEEE Granular Computing Conference*, 2006, USA.
24. Ghosh, P., Ghosh, S., Basu, K., Das, S., and Daefler, S. A stochastic model to estimate the time taken for Protein-Ligand Docking. *2006 IEEE Symposium on Computational Intelligence in Bioinformatics and Computational Biology (CIBCB)*, Sep. 2006, Canada.
25. Ghosh, P., Ghosh, S., Basu, K., Das, S., and Daefler, S. Stochastic Modeling of Cytoplasmic Reactions in Complex Biological Systems. *6th IEE International Conference on Computational Science and its Applications (ICCSA)*, May 8-11, 2006, Glasgow, UK.
26. Rao, C., Wolf, D., and Arkin, A. Control, Exploitation and Tolerance of Intracellular Noise. *Nature*, 420, pp. 231-237, 2002.
27. Raser, J., and O'Shea, E. Control of Stochasticity in Eukaryotic Gene Expression. *Science*, 304, pp. 1811-1814, 2004.
28. Cao, Y., Gillespie, D., and Petzold, L. Avoiding Negative Populations in Explicit Poisson Tau-Leaping. *J. Chemical Physics*, vol. 123, 054104, 2005.
29. Goutsias, J. A Hidden Markov Model for Transcriptional Regulation in Single Cells. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 3(1), 2006.
30. Goutsias, J. Quasiequilibrium approximation of fast reaction kinetics in stochastic biochemical systems. *J. Chemical Physics*, vol. 122, 184102, 2005.
31. Regev, A., Silverman, W., and Shapiro, E. Representation and simulation of biochemical processes using the π-calculus process algebra. *Proc. of the Pacific Symposium of Biocomputing (PSB 2001)*, 6: 459-470.
32. Regev, A., Silverman, W., and Shapiro, E. Representing biomolecular processes with computer process algebra: π-calculus programs of signal transduction pathways. *Proc. of the Pacific Symposium of Biocomputing 2000*, World Scientific Press, Singapore.