Xiaodong Cai and Xiaodong Wang

Stochastic Modeling and Simulation of Gene Networks

A review of the state-of-the-art research on stochastic simulations

© EYEWIRE

In conjunction with experimental investigation, appropriate computational tools can substantially help researchers to uncover the mechanism underlying gene regulation and understand gene functionality. Several computational approaches with different levels of modeling detail have been developed to investigate the dynamics of gene networks [1]–[3]. With limited knowledge of network structure and experimental data, a Boolean network provides a *coarse* model, capable of predicting certain dynamic behavior of a gene network [4], [5]. The approach based on deterministic differential equations (DDEs) provides a *fine* model [1], [3] but only reflects the deterministic dynamics of a gene network averaged over many cells. The *finest* stochastic approach [6], [7], based on stochastic kinetics [8]–[10], can capture the stochasticity inherent in gene expression in a *single* cell. The power of the stochastic model lies in its completeness and attention to detail [2].

Stochasticity in gene expression is mainly due to a series of events that involve a small number of molecules of DNA, RNA, and proteins. As each of these molecular events is subject to significant thermal fluctuations, the amount of mRNA and protein expressed from a gene is a stochastic process, which is called noise by biologists. Although gene expression noise was noticed more than more than four decades ago [11], only recently it received much attention, since recent advances in technology have provided an impetus for novel experimental investigations (see [12]–[15] and the references therein). Gene expression noise can explain many biological phenomena, such as phenotypic variations in cells or organisms with the same genes and in the same environment [14], however, many questions related to gene expression noise remain unanswered [12]. While biological investigations of expression noise of a single gene or in a simple gene network have revealed some of the mechanisms by which cells control and exploit noise, a computational approach to modeling and simulating relatively large gene networks will shed light on many unanswered questions.

As stochasticity in gene expression has been clearly observed in experiments, it is apparent that precise modeling and simulation of a gene network should take into account this stochasticity. Stochastic kinetics can describe the stochastic behavior of coupled reactions [9], [10], and was shown to have a rigorous physical base [8]. Therefore, stochastic kinetics can be employed to characterize and simulate the dynamics of chemical reactions in gene expression. Recently, several gene networks have been simulated [6], [7], [16], using Gillespie's exact stochastic simulation algorithm (SSA) [9], [10]. However, Gillespie's SSA requires large computational power and quickly becomes unmanageable when the reaction system becomes relatively large. Development of efficient stochastic simulation algorithms has

1053-5888/07/$25.00©2007IEEE

been an intensive research topic recently, while stochastic modeling of gene network from both biological and computational perspectives is still at its infancy.

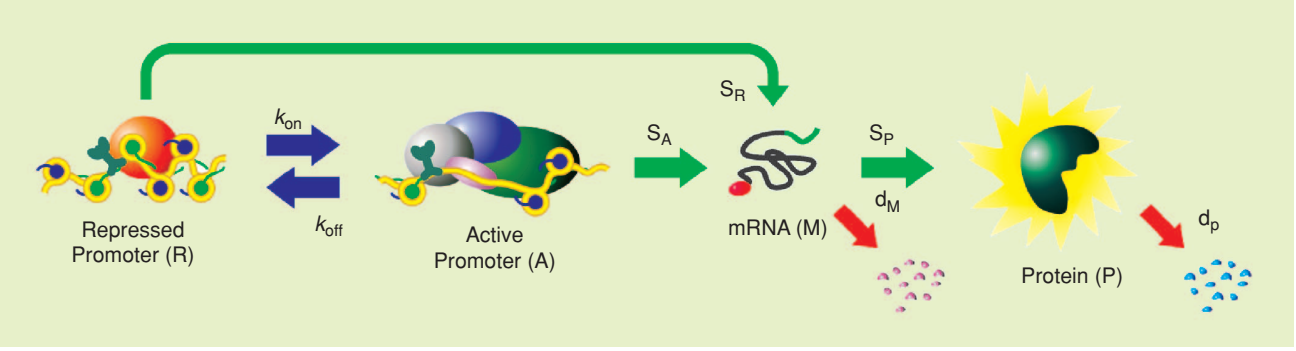
In this tutorial, we attempt to provide a comprehensive review of the state-of-the-art research on stochastic simulations. We also try to stimulate the interest of tackling the problem of stochastic simulation using statistical signal processing methods, as well as innovative thinking of stochastic modeling of gene networks from the viewpoint of signal processing.

STOCHASTIC MODELING OF GENE EXPRESSION

Figure 1 depicts a model of the expression of a single gene [13]. To initiate transcription, an RNA polymerase needs to bind to the promoter of a gene. In eucaryotes, an RNA polymerase requires a large set of proteins called transcription factors to position itself correctly at the promoter, and open the two strands of DNA [17]. As DNA in eucaryotes is packed into nucleosomes and higher order forms of chromatin structure, chromatin-modifying enzymes are also required to remodel chromatin so that an RNA polymerase can access the promoter. Consequently, the promoter is either in a repressed state in which an RNA polymerase cannot effectively bind to the promoter, or in an active state in which an RNA polymerase can bind to the promoter and efficiently initiate transcription. As activities of transcription factors and chromatin-modifying enzymes are subject to thermal fluctuations, the promoter randomly switches between these two states. A procaryotic gene, such as the lacZ gene in bacteria E. coli, can be controlled by an activator or/and a repressor [17]. As a result, a procaryotic gene can also randomly stay in either an inactive or active state. The parameters k_{on} and k_{off} in Figure 1 are deterministic rate constants used in conventional deterministic kinetics modeling the initiation of transcription. As we will discuss, the transition probability between two states is related to these deterministic rate constants. As shown in Figure 1, the gene is transcribed with a probability s_A per unit time, when the promoter is active, and with a much lower rate s_R per unit time, when the promoter is repressed. Due to the randomness present in the initiation of transcription and transcription process itself, the number of mRNA molecules transcribed from the gene is random.

In prokaryotes, ribosomes can bind to the mRNA as soon as it is accessible behind the transcribing RNA polymerase and start translation. On the other hand, in eukaryotes, mRNA molecules are transported from nucleus into cytoplasm and translated there. In the meantime, the mRNA can be bound and degraded by a multienzyme complex called degradosome. Therefore, an mRNA molecule is randomly translated to protein peptides by ribosomes, or, is degraded by degradosomes with certain probability that determines the rate constants d_M and s_P as shown in Figure 1. Finally, a functional protein can be targeted by a small polypeptide called ubiquitin, and be degraded by the proteasome. It is apparent that the amount of protein expressed from a gene is a random number, since the number of mRNA molecules is random as we discussed earlier, and the degradation and translation of the mRNA, as well as the degradation of protein itself, are random events. The model in Figure 1 is a simplified stochastic model for gene expression. More sophisticated models can be developed to characterize the gene expression in real cells, taking into account many additional factors, such as sequential assembly of the core transcription apparatus, pulsatile mRNA production due to reinitiation [18], and the scanning mechanism of ribosomes including leaky scanning and reinitiation in initiation of translation [19].

Genes and proteins are organized into extensive networks in e.g., signal pathways, that allow cells to respond and adapt to their environment. In a gene network, stochasticity in the expression of a particular gene can propagate to downstream genes, being amplified or attenuated. Particularly, negative feedback loops ubiquitous in biology can attenuate noise. Actually, some biologists found it useful to invoke analogies from signal processing when investigating gene expression noise [12], [20]. In terms of signal processing, a negative feedback loop in a biological pathway functions as a low-pass filter, and an integral feedback is similar to a band-pass filter [12], [20]. In our view, it is appropriate to model a gene network as a stochastic system, that involves many modules such as signal amplifiers, attenuators, integrators, negative and positive feedback loops, oscillators, and other possible components, similar to signal processing circuits. Since functional modules are a critical level of biological organization [21], such module-based modeling of gene networks not only is biologically meaningful, but also provides scalable models for large gene networks that can facilitate simulation and analysis.



[FIG1] A model of the expression of a single gene.

However, gene networks may have some fundamental differences with the stochastic systems we often deal with in signal processing and control systems. When we design a stochastic system, we often know the input and output relationship or the transfer function of the system. In the expression of a gene, the regulatory signal, such as transcription factor, can be regarded as the input signal, while the protein expressed from the gene is the output signal. However, it is not apparent how to characterize the input and output relationship for expression of a single gene, if such relationship exists. There are many questions that need to be answered, before we can model a gene network properly as a stochastic system that can be further analyzed.

CHARACTERIZING THE STOCHASTIC DYNAMICS OF GENE NETWORKS

Suppose that gene expression and other activities in a gene network involve N molecular species $\{S_1, \ldots, S_N\}$ that chemically interact through M reaction channels $\{R_1, \ldots, R_M\}$. We specify the dynamic state of this chemical system by the state vector $\mathrm{X}(t) = [X_1(t), \ldots, X_N(t)]^T$, where $X_n(t)$, $n = 1, \ldots, N$, is the number of S_n molecules at time t, and $(\cdot)^T$ represents the transpose of the vector in parentheses. As in [9], [22], [23], the dynamics of reaction R_m are defined by a state-change vector $\boldsymbol{\nu}_m = [\nu_{1m}, \ldots, \nu_{Nm}]^T$, where ν_{nm} gives the changes in the S_n molecular population produced by one R_m reaction, and a propensity function a_m together with the fundamental premise of stochastic chemical kinetics

$$a_m(\mathrm{x})dt \triangleq \text{the probability, given } \mathrm{X}(t) = \mathrm{x}, \text{ that one reaction } R_m \text{ will occur in the next infinitesimal time interval } [t, t+dt). \quad (1)$$

It is instructive to consider the following simple example involving $N = 5$ molecular species and $M = 3$ reactions:

$$R_1 : S_1 + S_2 \to S_3 + S_4, \quad R_2 : 2S_3 \to S_5, \quad R_3 : S_4 \to S_2. \quad (2)$$

In this example, we have $\boldsymbol{\nu}_1 = [-1, -1, +1, +1, 0]^T$, $\boldsymbol{\nu}_2 = [0, 0, -2, 0, +1]^T$, and $\boldsymbol{\nu}_3 = [0, +1, 0, -1, 0]^T$.

Define the probability rate constant c_m as the probability that a randomly selected combination of R_m reactant molecules react in a unit time period [9]. Let $h_m(\mathrm{x})$ be the number of distinct combinations of R_m reactant molecules in the system at time t when $\mathrm{X}(t) = \mathrm{x}$, then the propensity function is given by $a_m(\mathrm{x}) = c_m h_m(\mathrm{x})$. In example (2), $a_1(\mathrm{x}) = c_1 x_1 x_2$, $a_2(\mathrm{x}) = c_2 x_3(x_3 - 1)/2$, and $a_3(\mathrm{x}) = c_3 x_4$. As argued in [24], we typically only need to consider elementary reactions including bimolecular and monomolecular reactions, such as those in (2), since trimolecular reactions in a fluid are usually the combined result of two bimolecular reactions and one monomolecular reaction. The probability rate constant c_m can be calculated from the conventional deterministic reaction rate k_m [10]. For monomolecular reactions, we have $c_m = k_m$, and for bimolecular reactions, we have $c_m = k_m/\Omega$, when two reactants are from different molecular species as in R_1 of example (2), and $c_m \approx k_m/(2\,\Omega)$, when two reactants are the same as in R_2 of example (2), where Ω is the volume of the system.

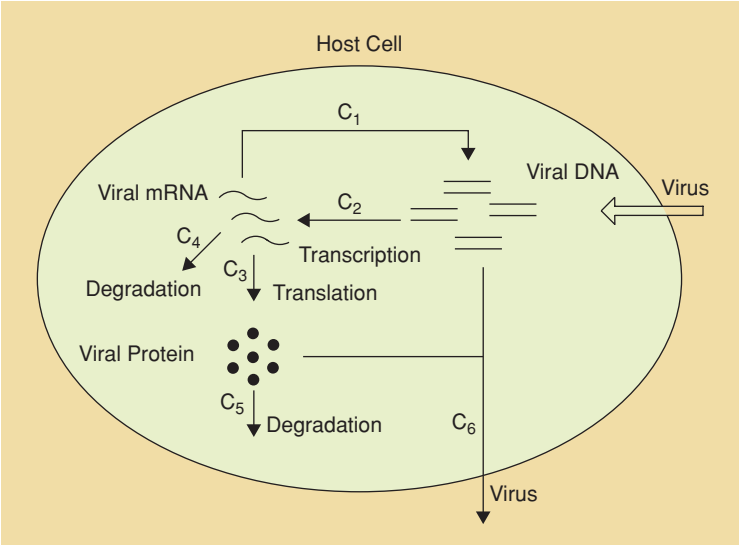
As an example of modeling gene expression, we now consider a simplified model of intracellular viral infection depicted in Figure 2 [25]–[27]. Based on several assumptions and simplifications, the viral infection process is modeled by the following six reactions.

$$\begin{aligned} &\mathrm{R}_1 : \mathrm{RNA} \xrightarrow{c_1} \mathrm{DNA}, && \mathrm{R}_2 : \mathrm{DNA} \xrightarrow{c_2} \mathrm{RNA} + \mathrm{DNA}, \\ &\mathrm{R}_3 : \mathrm{RNA} \xrightarrow{c_3} \mathrm{P} + \mathrm{RNA}, && \mathrm{R}_4 : \mathrm{RNA} \xrightarrow{c_4} \emptyset, \\ &\mathrm{R}_5 : \mathrm{P} \xrightarrow{c_5} \emptyset, && \mathrm{R}_6 : \mathrm{DNA} + \mathrm{P} \xrightarrow{c_6} \mathrm{V}, \end{aligned} \quad (3)$$

where we denote the viral DNA, mRNA, protein, and virus as DNA, RNA, P, and V, respectively. For the clarity of illustration, we assume that the genome of virus consists of double stranded DNA, although it can be other different nucleic acids. During initial infection of a cell, one viral DNA molecule is inserted into the cell. The DNA is transcribed to mRNA (R_2), while the mRNA molecule can be used as a template to replicate the viral DNA (R_1), and also translated to protein (R_3). In the meantime, the mRNA and protein molecules are degraded (R_4 and R_5). Some viral DNA molecules are packed into viral proteins to form viral structure and exit the host cell (R_6). The stochastic rate constants are given by [27]: $c_1 = 1\ \mathrm{day}^{-1}$, $c_2 = 0.025\ \mathrm{day}^{-1}$, $c_3 = 100\ \mathrm{day}^{-1}$, $c_4 = 1\ \mathrm{day}^{-1}$, $c_5 = 1.99\ \mathrm{day}^{-1}$, and $c_6 = 11.25 \times 10^{-6}\ \mathrm{day}^{-1}$. We will simulate this example using the exact SSA in a later section.

THE CHEMICAL MASTER EQUATION

As the probability of a reaction occurs in the infinitesimal time interval $[t, t+dt)$ is only dependent upon the state $\mathrm{X}(t)$ at



[FIG2] **A simplified model of intracellular viral infection.**

time t, it is clear that $\mathbf{X}(t)$ is a Markov process with discrete states, or a jump Markov process. The time evolution of the state probability $P(\mathbf{x}, t)$ of this Markov process is governed by the chemical master equation (CME) [9]:

$$\frac{\partial P(\mathbf{x}, t)}{\partial t} = \sum_{m=1}^{M} [a_m(\mathbf{x} - \boldsymbol{\nu}_m)P(\mathbf{x} - \boldsymbol{\nu}_m, t) - a_m(\mathbf{x})P(\mathbf{x}, t)]. \tag{4}$$

The CME essentially says that the rate of change in $P(\mathbf{x}, t)$ is equal to the probability of entering the state $\mathbf{x}$ minus the probability of leaving the state $\mathbf{x}$ in unit time. A rigorous derivation of the CME is given in [9], based on the fundamental premise (1). While the CME exactly describes the evolution of $P(\mathbf{x}, t)$ with time, unfortunately, we can solve the CME to obtain $P(\mathbf{x}, t)$ only in rare case. However, under certain conditions that we will discuss next, $\mathbf{X}(t)$ is approximately determined by a stochastic differential equation, the chemical Langevin equation (CLE), which is more tractable analytically or numerically.

THE CHEMICAL LANGEVIN EQUATION

Let $K_m(\mathbf{x}, \tau)$, for any $\tau > 0$, be the number of R_m reactions that occur in the time interval $[t, t + \tau]$. The state vector at time $t + \tau$ will be

$$\mathbf{X}(t + \tau) = \mathbf{x} + \sum_{m=1}^{M} \boldsymbol{\nu}_m K_m(\mathbf{x}, \tau). \tag{5}$$

The probability distribution function of $K_m(\mathbf{x}, \tau)$ is generally difficult to obtain. But we can have an excellent approximation to it, if the following condition is satisfied:

C1: The value of τ is small enough to ensure that there is no significant change in the propensity functions of all reactions, i.e., $a_m(\mathbf{X}(t')) \approx a_m(\mathbf{x})$, $\forall t' \in [t, t + \tau]$, $\forall m \in [1, M]$.

Condition C1 generally can be satisfied when x_n, $n = 1, \ldots, N$, are sufficiently large, since reactions occurred in $[t, t + \tau]$ will cause negligible changes in the state vector $\mathbf{X}(t')$, $\forall t' \in [t, t + \tau]$. As all the propensity functions essentially remain constant in $[t, t + \tau]$, all reactions occur independently. Then, it can be shown that each $K_m(\mathbf{x}, \tau)$ is an independent Poisson random variable with mean $a_m(\mathbf{x})\tau$ [22], [23]. It is well known that a Poisson random variable can be approximated by a Gaussian random variable with the same mean and variance, if its mean is much larger than one. To approximate $K_m(\mathbf{x})\tau$ by a Gaussian random variable, we impose another condition on τ:

C2: The value of τ is large enough to ensure that the expected number of occurrences of each reaction R_m in $[t, t + \tau]$ is much larger than 1, i.e., $a_m(\mathbf{x})\tau \gg 1$.

We can regard any time interval τ that satisfies both C1 and C2 as a macroscopic infinitesimal and denote it simply by dt, and recall that $\mathbf{x}$ is the value of $\mathbf{X}(t)$. Then, (5) can be approximated by the following stochastic differential equation [24]:

$$\mathbf{X}(t + dt) = \mathbf{X}(t) + \sum_{m=1}^{M} \boldsymbol{\nu}_m a_m(\mathbf{X}(t))dt + \sum_{m=1}^{M} \boldsymbol{\nu}_m \sqrt{a_m(\mathbf{X}(t))dt}\mathcal{N}_m(t), \tag{6}$$

where $\mathcal{N}_m(t)$, $m = 1, \ldots, M$, are independent standard Gaussian random variables with zero mean and unit variance. Notice that when we approximate a Poisson random variable with a Gaussian random variable, we in effect convert the discrete-state Markov process $\mathbf{X}(t)$ to a continuous-state Markov process. Equation (6) is the standard-form CLE for the continuous Markov process $\mathbf{X}(t)$.

While the CLE has been used to simulate gene expression [12], it is also pointed out in [28] that such a CLE approach is theoretically unsound. Now, it is clear that only when conditions C1 and C2 are both satisfied, the CLE approach can provide results with acceptable accuracy.

THE DETERMINISTIC REACTION RATE EQUATION

In large chemical reaction systems where both $X_n(t)$, $\forall n$ and the system volume Ω are large, the conventional deterministic reaction rate equation (RRE) is often used to describe the system dynamics. The deterministic RRE can be derived from the CLE, when we go to the thermodynamic limit, in which the number of molecules in the system and the system volume Ω both go to ∞ in such a way that the species concentrations remain constant. Since the RRE is more commonly written in terms of the species concentrations, let us denote $Y_n(t) = X_n(t)/\Omega$ as the concentration of S_n. Dividing both sides of (6) by Ω, we can show that the random term in (6) vanishes in the thermodynamic limit, and if we further let $dt \to 0$, (6) implies the following conventional deterministic RRE [24]:

$$\frac{d\mathbf{Y}(t)}{dt} = \sum_{m=1}^{M} \boldsymbol{\nu}_m \tilde{a}_m(\mathbf{Y}(t)), \tag{7}$$

where $\tilde{a}_m(\mathbf{Y}(t)) = a_m(\mathbf{Y}(t))/\Omega$. It is important to know under which conditions we can use the CLE or RRE, since a hybrid approach, involving CME, as well as CLE or RRE, is sometimes employed to simulate the dynamics of chemical reaction systems, as we will discuss later.

STOCHASTIC SIMULATION ALGORITHMS

EXACT SSA

Although it is difficult to solve the CME in general, Gillespie developed an SSA to simulate the Markov process $\mathbf{X}(t)$. Like the CME, Gillespie's SSA is based on the fundamental premise (1). Therefore, the realizations of $\mathbf{X}(t)$ generated from Gillespie's SSA adhere to a probability model identical to that obtained by the CME. For this reason, Gillespie's simulation algorithm is called the *exact* SSA. There are three different but statistically equivalent methods for exact SSA: Gillespie's direct method (DM) [10], Gillespie's first reaction method (FRM) [9], and the next reaction method (NRM) of Gibson and Bruck [29].

THE DIRECT METHOD

For a chemical system in a given state $\mathbf{X}(t) = \mathrm{x}$ at time t, Gillespie's DM SSA answers the following two questions: i) when will the next reaction occur? and ii) which reaction will occur? Specifically, the SSA simulates the occurrence of the following event:

$$\begin{aligned} E \ :& \text{ no reaction occurs in the time interval} \\ & [t, t+\tau], \text{ and a reaction } R_\mu \\ & \text{occurs in the infinitesimal time interval} \\ & (t+\tau, t+\tau+d\tau). \end{aligned} \tag{8}$$

Clearly, τ and μ are random variables; it is not difficult show that they are independent. The probability density functions (pdfs) of τ and μ are, respectively, given by [10]

$$p(\tau) = a_0(\mathrm{x})\exp(-a_0(\mathrm{x})\tau), \quad \tau > 0, \tag{9}$$

$$p(\mu) = a_\mu(\mathrm{x})/a_0(\mathrm{x}), \quad \mu = 1, \ldots, M, \tag{10}$$

where $a_0(\mathrm{x}) = \sum_{m=1}^{M} a_m(\mathrm{x})$. It is easy to generate τ and μ from two independent uniform random variables directly according to their pdfs. The SSA based on the DM can be summarized as follows:

Algorithm 1: Exact SSA—Direct Method [10]

1) Initialization (set the initial number of molecules, set $t \leftarrow 0$).
2) Calculate the propensity function, $a_m(\mathrm{x})$, $m = 1, \ldots, M$.
3) Generate τ and μ according to their pdfs in (9) and (10).
4) Set $t \leftarrow t+\tau$, and update the state vector $\mathbf{X}(t) \leftarrow \mathbf{X}(t) + \boldsymbol{\nu}_\mu$.
5) Go to step 2, or else stop.

THE FIRST REACTION METHOD

Let us consider M independent events:

$$\begin{aligned} E \ :& \text{ no reaction } R_m \text{ occurs in the time interval} \\ & [t, t+\tau_m], \text{ and an } R_m \\ & \text{occurs in the infinitesimal time interval} \\ & (t+\tau_m, t+\tau_m+d\tau_m), \quad m = 1, \cdots, M. \end{aligned} \tag{11}$$

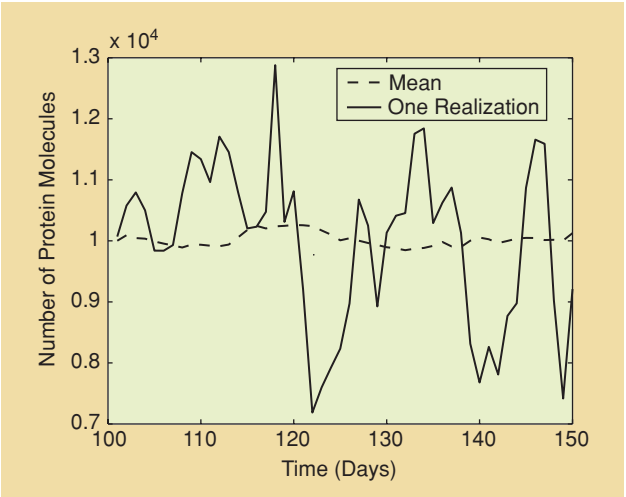
Notice that in the event E_m in (11), it is possible that a reaction other than R_m occurs in the time interval $[t, t+\tau_m]$, while in the event E in (8) no reaction occurs in $[t, t+\tau]$. The pdf of τ_m can be easily found to be an exponential distribution with parameter $a_m(\mathrm{x})$, i.e., $p(\tau_m) = a_m(\mathrm{x})\exp(-a_m(\mathrm{x})\tau_m)$, $\tau_m > 0$. If we independently generate τ_m, $m = 1, \ldots, M$, and take $\tau = \min\{\tau_1, \ldots, \tau_M\}$ and $\mu = \arg\min_m\{\tau_1, \ldots, \tau_M\}$, we essentially generate the event E in (8). Therefore, the FRM is equivalent to the DM. Compared with the DM, the FRM is not efficient, because it needs to generate more random variables. However, Gibson and Bruck transformed the FRM into an equivalent and more efficient NRM [29].

THE NEXT REACTION METHOD

The NRM improves the efficiency of the FRM by exploiting the following two observations: i) each $a_m(\mathrm{x})$ is only affected by a few reactions and can be efficiently calculated in each step, and ii) τ_m, $m = 1, \ldots, M$, generated in a step can be reused in the next step. Towards this end, a data structure called *dependency graph* is defined to tell precisely which $a_m(\mathrm{x})$ should be updated after a reaction occurs. An *indexed priority queue* is also defined to properly reuse τ_m, $m = 1, \cdots, M$. The detailed description of the NRM can be found in [29]. After incorporating these two mechanisms into the NRM, it is argued in [29] that the FRM is more efficient than the DM, for loosely coupled chemical reaction systems where the firing of one reaction channel does not affect many other reactions.

However, a detailed analysis of CPU cost of both NRM and DM in [30] shows that maintaining and updating the data structure of the indexed priority queue in the NRM may require significantly large cost for some practical systems. An optimized direct method (ODM) is proposed in [30] to improve the efficiency of the DM. The ODM incorporates the dependency graph used in the NRM into the DM to reduce the cost of calculating the propensity functions, $a_m(\mathrm{x})$. It also properly reorders the index of reaction channels to reduce the cost of generating the reaction index μ. With these two optimization steps, the ODM is much more efficient than the original DM. It is argued in [30] that in practical systems that almost always have the multiscale nature, the ODM is preferable to the NRM.

We simulated the viral infection process depicted in Figure 2 using the exact SSA. Figure 3 depicts the number of molecules of viral protein from the 100th–150th day after infection. After initial infection, a cell may exhibit either a typical infection in which all species become populated or an aborted infection in which all species are eliminated from the cell [26]. We start to run simulation from the 100th day assuming that the cell is typically infected and using the following initial condition: the numbers of viral mRNA, DNA, and protein molecules are 20,



[FIG3] Number of protein molecules.

200, 10^4, respectively. The mean of these molecular numbers obtained from 10^4 simulation runs and the results of one simulation run are shown in Figure 3. It is seen that molecular numbers in one simulation run fluctuate around the mean numbers.

APPROXIMATE SSA

Although the exact SSA, that simulates every reaction event exactly, and one at time, is easy to implement, and produces realizations of $\mathbf{X}(t)$ with correct statistics, it is often too slow for simulating many practical systems. Several approximate methods have been developed to significantly speed up simulation by giving up some of the exactness of the SSA. The basic idea behind these approximate methods is that instead of simulating a single reaction per step, a number of reactions can occur in each simulation step. As one step leaps over many reactions, these approximate methods are known as leap methods including the τ-leap method [22], [23], the binomial τ-leap method [31], [32], and the K-leap method [33]. Since the exact SSA is based on the fundamental premise (1), one would expect that a leap method can provide an excellent approximation to the exact SSA, if the propensity functions $a_m(\mathbf{x})$ remain approximately constant in each leap.

THE τ-LEAP METHOD

In the τ-leap method, the step size of each leap, τ, is a deterministic number selected to satisfy condition C1 that is also referred to as leap condition. Once τ has been selected, the number of firings of a reaction channel R_m, $K_m(\mathbf{x}, \tau)$, is approximately a Poisson random variable with mean $a_m(\mathbf{x})\tau$ as shown earlier, and the state vector $\mathbf{X}(t)$ can be updated using (5). The question now is how to select the value of τ to satisfy the leap condition, given $\mathbf{X}(t) = \mathbf{x}$. Letting $\Delta a_m(\tau; \mathbf{x}) \triangleq a_m(\mathbf{X}(t+\tau)) - a_m(\mathbf{x})$, Gillespie imposed the following constraint to satisfy the leap condition C1 [22]:

$$|\Delta a_m(\tau; \mathbf{x})| \le \varepsilon a_0(\mathbf{x}), \quad \forall m = 1, \cdots, M, \tag{12}$$

where ε is a prespecified error control parameter satisfying $0 < \varepsilon \ll 1$. Since $\Delta a_m(\tau; \mathbf{x})$ is a random variable, it will be impossible to find a τ directly satisfying (12). Gillespie proposed to use a first-order Taylor expansion of $\Delta a_m(\tau; \mathbf{x})$ to approximate $\Delta a_m(\tau; \mathbf{x})$, and then bound the absolute mean and standard deviation of this approximate $\Delta a_m(\tau; \mathbf{x})$ by $\varepsilon a_0(\mathbf{x})$ [22], [23], which we will describe next in detail. Since we have $\Delta \mathbf{X}(\tau) \triangleq \mathbf{X}(t+\tau) - \mathbf{x} = \sum_{m=1}^{M} K_m(\tau; \mathbf{x})\boldsymbol{\nu}_m$, the first-order Taylor expansion of $\Delta a_m(\tau; \mathbf{x})$ can be found as $\Delta a_m(\tau; \mathbf{x}) \approx \sum_{m=1}^{M} f_{mm'}(\mathbf{x}) K_{m'}(\mathbf{x}, \tau)$, where

$$f_{mm'}(\mathbf{x}) \triangleq \left[\frac{\partial a_m(\mathbf{x})}{\partial \mathbf{x}}\right]^T \boldsymbol{\nu}_{m'}, \; m, m' = 1, \ldots, M. \tag{13}$$

Using this approximation, we can find the approximate mean and variance of $\Delta a_m(\tau; \mathbf{x})$ as

$$\begin{aligned} E[\Delta a_m(\tau; \mathbf{x})] &\approx \eta_m(\mathbf{x})\tau, \\ \mathrm{var}[\Delta a_m(\tau; \mathbf{x})] &\approx \sigma_m^2(\mathbf{x})\tau, \end{aligned} \tag{14}$$

where

$$\begin{aligned} \eta_m(\mathbf{x}) &\triangleq \sum_{m'=1}^{M} f_{mm'}(\mathbf{x}) a_{m'}(\mathbf{x}), \quad m = 1, \ldots, M, \\ \sigma_m^2(\mathbf{x}) &\triangleq \sum_{m'=1}^{M} f_{mm'}^2(\mathbf{x}) a_m(\mathbf{x}), \quad m = 1, \ldots, M. \end{aligned} \tag{15}$$

If we impose the following requirements: $|E[\Delta a_m(\tau; \mathbf{x})]| < \varepsilon a_0(\mathbf{x})$, $\sqrt{\mathrm{var}[\Delta a_m(\tau; \mathbf{x})]} < \varepsilon a_0(\mathbf{x})$, we obtain the value of τ that approximately satisfies the leap condition (12)

$$\tau = \min_{m \in [1,M]} \left\{ \frac{\varepsilon a_0(\mathbf{x})}{|\eta_m(\mathbf{x})|}, \frac{\varepsilon^2 a_0^2(\mathbf{x})}{\sigma_m^2(\mathbf{x})} \right\}. \tag{16}$$

The accuracy of τ-leaping will depend upon how well the leap condition is satisfied. If the reactant molecule populations are very large, it will take a very large number of reaction events to change the propensity functions significantly. In this case, we should be able to satisfy the leap condition with a choice for τ that allows for many reaction events to occur in $[t, t+\tau]$. On the other hand, if satisfying the leap condition turns out to require τ to be less than some small multiple (say 10) of $1/a_0(\mathbf{x})$ that is the expected step size in the exact SSA, only a very few reactions can be leaped over, and it would be faster to forego leaping and use the exact SSA. We summarize the τ-leap method as follows.

Algorithm 2: Approximate SSA: τ-Leap Method [22], [23]

1) Initialization (set the initial number of molecules, set $t \leftarrow 0$).
2) Calculate the propensity function, a_m, $m = 1, \ldots, M$.
3) Calculate τ from (13), (15), and (16).
4) If the τ value is less than some small multiple (say ten) of $1/a_0(\mathbf{x})$, then reject it and execute instead a moderate number (say 100) of successive exact SSA steps, and then go to step 2. Otherwise, accept τ and proceed to step 5.
5) For each $m = 1, \ldots, M$, independently generate K_m according to a Poisson distribution with mean $a_m(\mathbf{x})\tau$.
6) Set $t \leftarrow t + \tau$, and update the state vector vector $\mathbf{X}(t) \leftarrow \mathbf{X}(t) + \sum_{m=1}^{M} \boldsymbol{\nu}_m K_m$.
7) Go to step 2, or else stop.

An efficient method for selecting step size for the τ-leap method of [22] and [23] was recently proposed in [34]. Instead of using (12) to satisfy the leap condition C1, the authors of [34] propose to bound the *relative* change in all the propensity function by the same amount ε: $|\Delta a_m(\tau; \mathbf{x})| \le \varepsilon a_m(\mathbf{x}), \; \forall m = 1, \ldots, M$. They further show that these inequalities are approximately equivalent to the following set of inequalities: $|\Delta X_n(\tau)| \le \max\{\varepsilon_n x_n, 1\}$, $n = 1, \ldots, N$, where ε_n can be found from ε as discussed in [34]. Then, they choose the step size τ to satisfy the above inequalities appreciably. It is demonstrated in [34] that this step-size selection method is more efficient than that in the original τ-leap method.

THE BINOMIAL τ-LEAP METHOD

In the τ-leap method, the number of firings of each reaction channel $K_m(\mathbf{x}, \tau)$ in each leap is approximated by a Poisson random variable. As realizations of a Poisson random variable can be any nonnegative integer, we always run the risk that a reaction channel R_m fires so many times in one leap that more molecules of one of its reactants will be consumed than those are actually available. When this happens, the number of molecules of that reactant becomes negative, which is clearly undesirable. Tian and Burrage, and independently Chatterjee et al., proposed the binomial τ-leap method to cope with the problem of negative population [31], [32]. The binomial τ-leap method approximates the Poisson random variable in step 5 of Algorithm 2 by a binomial random variable, $\mathcal{B}(k_{m,\max}, p_m)$, with parameters $k_{m,\max}$ and $p_m = [a_m(\mathbf{x})\tau]/k_{m,\max}$. All other steps are the same. Several methods of choosing $k_{m,\max}$ were proposed to avoid the problem of negative number of molecules. Although the binomial τ-leap method improve simulation accuracy in some cases, the method in [32] cannot handle the case where more than two reaction channels share certain reactants, while the method in [31] may introduce bias.

In an alternative approach, Cao et al. modified the τ-leap method to avoid the problem of negative population [35]. Cao et al. classify the reaction channels into two categories: critical and noncritical reaction channels. If $k_{m,\max} = \min_{n\in[1,N],\, \nu_{nm}<0} \lfloor x_n/|\nu_{nm}| \rfloor$ is less than or equal to some critical value n_c, where $\lfloor x \rfloor$ denotes the greatest integer that is less than or equal to x, then R_m is critical; otherwise, it is noncritical. Typical value for n_c is between two and 20. A tentative step size τ' is calculated from (16), as in the τ-leap method; another tentative step size τ'' is generated from an exponential distribution with parameter $1/a_0^c(\mathbf{x})$, where $a_0^c(\mathbf{x})$ is the sum of the propensity functions of the critical reaction channels. Then the actual step size is chosen as $\tau = \min\{\tau', \tau''\}$. For all noncritical reaction channels $\{R_{m'}\}$, we generate $K_{m'}$ as a sample of Poisson random variable with mean $a_{m'}(\mathbf{x})\tau$. For critical reaction channels $\{R_{m''}\}$, if $\tau' < \tau''$, then $K_{m''} = 0, \forall m''$; if $\tau' \geq \tau''$, then we generate reaction index μ according to the probablility $p(\mu) = a_\mu(\mathbf{x})/a_0^c(\mathbf{x})$ and set $K_u = 1$ and other $K_{m''}$ to be zero. It is argued that the modified τ-leap method can well handle the problem of negative population, and is easier to implement than the binomial τ-leap method.

THE K-LEAP METHOD

In the τ-leap method, the number of firings of each reaction channel during a leap is unbounded, and thus, there is always a probability that the state vector $\mathbf{X}(t)$ undergoes a significantly large change during one leap, which will inevitably cause large changes in the propensity functions, thereby violating the leap condition. The dilemma of the τ-leap method is: how can a preselected step size τ, without knowing at least an upper bound on the number of reactions that will occur in the next leap, satisfy the leap condition well? We recently developed a K-leap method to avoid this dilemma by simulating the occurrence of $K \geq 1$ reactions during each leap [33]. Here K is a deterministic constant chosen to satisfy the leap condition, and after K is chosen, the time τ that is leaped over in a step is a random variable.

Denoting the number of firings of each reaction channel as K_m, $m = 1, \ldots, M$, we proved in [33] that τ is independent from $K_1, \ldots, K_M$ under the constraint $\sum_{m=1}^M K_m = K$, that is $p(K_1, \ldots, K_M, \tau | \sum_{m=1}^M K_m = K) = p(\tau | \sum_{m=1}^M K_m = K)$ $p(K_1, \ldots, K_M | \sum_{m=1}^M K_m = K)$. Moreover, we showed that $p(\tau | \sum_{m=1}^M K_m = K)$ is a Gamma pdf given by

$$p\left(\tau \mid \sum_{m=1}^{M} K_m = K\right) = \frac{a_0 \exp(-a_0\tau)(a_0\tau)^{K-1}}{(K-1)!}, \tau > 0, \quad (17)$$

while $p(K_1, \ldots, K_M | \sum_{m=1}^M K_m = K)$ is a multinomial pdf given by

$$p\left(K_1, \ldots, K_M \mid \sum_{m=1}^{M} K_m = K\right) = \frac{K!}{\prod_{m=1}^M K_m!} \prod_{m=1}^{M} \theta_m^{K_m}, \quad (18)$$

where $a_0 = \sum_{m=1}^M a_m$, and $\theta_m = a_m/a_0$, $m = 1, \ldots, M$.

Several methods of selecting K according to the leap condition have been proposed in [33]. Here, we give a K-selection method that is in spirit similar to τ-selection method in (16). Let us define $\boldsymbol{\theta} \triangleq [\boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_M]^T$, and $\mathbf{K} \triangleq [K_1, \ldots, K_M]^T$, then we have $E[\mathbf{K}] = K\boldsymbol{\theta}$. If we define a matrix $\mathbf{C}$ with $[\mathbf{C}]_{mm'} = -\theta_m\theta_{m'}$, for $m \neq m'$, and $[\mathbf{C}]_{mm} = \theta_m(1-\theta_m)$, where $[\mathbf{C}]_{mm'}$ denotes the entry on the mth row and the m'th column of $\mathbf{C}$, the covariance matrix of $\mathbf{K}$ is given by $\mathrm{cov}[\mathbf{K}] = K\mathbf{C}$. Letting $\mathbf{f}_m = [f_{m1}, \ldots, f_{mM}]^T$, $m = 1, \ldots, M$, where $f_{mm'}$ is given in (13), and

$$\begin{aligned} \eta_m(\mathbf{x}) &\triangleq \mathbf{f}_m^T\boldsymbol{\theta}, m = 1, \ldots, M, \\ \sigma_m^2(\mathbf{x}) &\triangleq \mathbf{f}_m^T\mathbf{C}\mathbf{f}_m, m = 1, \ldots, M. \end{aligned} \quad (19)$$

From the first-order Taylor expansion of $\Delta a_m(K, \mathbf{x})$, we obtain the following:

$$\begin{aligned} E[\Delta a_m(K; \mathbf{x})] &\approx \eta_m(\mathbf{x})K, \\ \mathrm{var}[\Delta a_m(K; \mathbf{x})] &\approx \sigma_m^2(\mathbf{x})K. \end{aligned} \quad (20)$$

Using the constraints $|E[\Delta a_m(K; \mathbf{x})]| < \varepsilon a_0(\mathbf{x})$ and $\sqrt{\mathrm{var}[\Delta a_m(K; \mathbf{x})]} < \varepsilon a_0(\mathbf{x})$, and (20), considering that the minimum value of K is 1, we obtain the value of K:

$$K = \max\left\{\min_{m\in[1,M]}\left\{\frac{\varepsilon a_0(\mathbf{x})}{|\eta_m(\mathbf{x})|}, \frac{\varepsilon^2 a_0^2(\mathbf{x})}{\sigma_m^2(\mathbf{x})}\right\}, 1\right\}. \quad (21)$$

When $K = 1$, our K-leap method becomes the exact SSA. Hence, our K-leap method can adaptively change from the exact SSA to an approximate leap method, whenever the leap condition allows to do so. We summarize the K-leap method in the following simulation algorithm.

Algorithm 3: Approximate SSA: K-Leap Method [33]

1) Initialization (set the initial number of molecules, set $t \leftarrow 0$).

2) Calculate the propensity function, a_m, $m = 1, \cdots, M$.
3) Calculate K from (21).
4) If $K = 1$, execute an exact SSA step, and go to step 6.
5) If $K > 1$, generate τ according to the Gamma pdf (18), and generate K_m, $m = 1, \cdots, M$, according to the multinomial pdf (18).
6) Set $t \leftarrow t + \tau$, and update the state vector vector $\mathbf{X}(t) \leftarrow \mathbf{X}(t) + \sum_{m=1}^{M} \boldsymbol{\nu}_m K_m$.
7) Go to step 2, or else stop.

MULTISCALE STOCHASTIC SIMULATION

In gene networks and some other chemical reaction systems, certain reaction channels can fire much more frequently than others, i.e., some reaction channels are fast while the others are slow. For example, in the gene expression of the heat shock response of E. coli [36], [37], six out of 61 reaction channels are much faster than others. If the exact SSA is used to simulate such systems, the majority of the simulation time will be spent on fast reactions. However, it is often the case that the slow reactions have a greater impact on the behavior of the system. Simulating each of the fast reactions is often neither necessary nor useful, but only incurs a huge computational burden. Approximate leaping methods are also not efficient to simulate such multiscale systems, since the step size of each leap will most likely be very small, limited by fast reactions.

Multiscale stochastic simulation methods aim to efficiently simulate multiscale systems by legitimately skipping over the fast reactions and explicitly simulating only the slow reactions [25], [26], [38]–[40]. Let us first look at the quasi-steady-state approach of Rao and Arkin [40]. Rao and Arkin partition all molecular species into intermediate and primary species. If a species is highly reactive, it is an intermediate species, otherwise, it is a primary species. We also call intermediate species fast species and primary species slow species. Let the generic state vector at t be $\mathbf{x} = [\mathbf{x}_s^T, \mathbf{x}_f^T]^T$ where $\mathbf{x}_s$ and $\mathbf{x}_f$ are the state vectors of slow and fast species, respectively; correspondingly, the state change vector is partitioned as $\boldsymbol{\nu}_m = [(\boldsymbol{\nu}_m^s)^T, (\boldsymbol{\nu}_m^f)^T]^T$, where $\boldsymbol{\nu}_m^s$ and $\boldsymbol{\nu}_m^f$ are state change vectors associated with slow and fast species, respectively. We can write the probability $P(\mathbf{x}; t)$ as $P(\mathbf{x}; t) = P(\mathbf{x}_f \,|\, \mathbf{x}_s; t) P(\mathbf{x}_s; t)$. Using the chain rule of differentiation, the CME in (4) becomes

$$\begin{aligned} P(\mathbf{x}_s; t) \frac{dP(\mathbf{x}_f \,|\, \mathbf{x}_s; t)}{dt} &+ P(\mathbf{x}_f \,|\, \mathbf{x}_s; t) \frac{dP(\mathbf{x}_s; t)}{dt} \\ &= \sum_{m=1}^{M} \Bigg[a_m \left(\mathbf{x}_s - \boldsymbol{\nu}_m^s, \mathbf{x}_f - \boldsymbol{\nu}_m^f \right) \\ &\quad \times P \left(\mathbf{x}_f - \boldsymbol{\nu}_m^f \,|\, \mathbf{x}_s - \boldsymbol{\nu}_m^s; t \right) P \left(\mathbf{x}_s - \boldsymbol{\nu}_m^s; t \right) \\ &\quad - a_m(\mathbf{x}_s, \mathbf{x}_f) P(\mathbf{x}_f \,|\, \mathbf{x}_s; t) P(\mathbf{x}_s; t) \Bigg]. \end{aligned} \tag{22}$$

Summing both sides of (22) with respect to $\mathbf{x}_f$, and noticing that $\sum_{\mathbf{x}_f} dP(\mathbf{x}_f \,|\, \mathbf{x}_s; t)/dt = d\left[\sum_{\mathbf{x}_f} P(\mathbf{x}_f \,|\, \mathbf{x}_s; t)\right]/dt = 0$, we obtain the CME for the slow species:

$$\begin{aligned} \frac{dP(\mathbf{x}_s; t)}{dt} = \sum_{m=1}^{M} \Big[&\bar{a}_m \left(\mathbf{x}_s - \boldsymbol{\nu}_m^s \right) \\ &\times P \left(\mathbf{x}_s - \boldsymbol{\nu}_m^s; t \right) - \bar{a}_m(\mathbf{x}_s) P(\mathbf{x}_s; t) \Big], \end{aligned} \tag{23}$$

where

$$\bar{a}_m(\mathbf{x}_s) = \sum_{\mathbf{x}_f} a_m(\mathbf{x}_s, \mathbf{x}_f) P(\mathbf{x}_f \,|\, \mathbf{x}_s; t) \tag{24}$$

is the conditional mean of $a_m(\mathbf{x}_s, \mathbf{x}_f)$ with respect to the conditional probability $P(\mathbf{x}_f \,|\, \mathbf{x}_s; t)$.

Rao and Arkin made a quasi-steady-state assumption (QSSA) [40], which says that

$$\frac{dP(\mathbf{x}_f \,|\, \mathbf{x}_s; t)}{dt} \approx 0, \tag{25}$$

i.e., $P(\mathbf{x}_f \,|\, \mathbf{x}_s; t)$ is approximately independent of t, and can be written as $P(\mathbf{x}_f \,|\, \mathbf{x}_s)$. Under this QSSA, Rao and Arkin obtained an approximate CME for the slow species, which is the same as (23) except that the equality sign is replaced by an approximate equality sign, and

$$\bar{a}_m(\mathbf{x}_s) = \sum_{\mathbf{x}_f} a_m(\mathbf{x}_s, \mathbf{x}_f) P(\mathbf{x}_f \,|\, \mathbf{x}_s). \tag{26}$$

But here, we show that the CME (23) for the slow species can be exactly derived from the original CME. The CME (23) is in fact mathematically valid for any partition of the molecular species. However, without a proper assumption, such as the QSSA, the propensity function $\bar{a}_m(\mathbf{x}_s)$ in (24) is affected by the time-dependent pdf $P(\mathbf{x}_f \,|\, \mathbf{x}_s; t)$, which implies that we cannot simulate the time evolution of the slow species independently from the fast species, based on the CME (23). With the QSSA, we now can simulate the dynamics of the slow species independently from the fast species, if $\bar{a}_m(\mathbf{x}_s)$ in (26) can be calculated. Before we describe such a simulation method, let us take a look at another approach, proposed by Cao et al. in [38], to characterizing multiscale systems, because it provides more insights into multiscale systems.

Cao et al. first partition all reaction channels into two sets: reaction channels whose propensity functions are usually much larger than the propensity functions of all the other reaction channels are called fast, and all the other reaction channels are called slow. The molecular species are also correspondingly partitioned into two sets: those species whose population is affected by some fast reactions are called fast species, and any other species are called slow species. Denote the state vector $\mathbf{X}(t) = [\mathbf{X}_s(t)^T, \mathbf{X}_f(t)^T]^T$, where $\mathbf{X}_s(t)$ represents the slow species, and $\mathbf{X}_f(t)$ represents the fast species. Cao et al. defined a virtual fast process $\hat{\mathbf{X}}_f(t)$ which is basically the $\mathbf{X}_f(t)$ with all slow reaction channels turned off. They then made the following equilibrium assumption: i) the virtual system has a stochastic partial equilibrium, mathematically,

$\lim_{t\to\infty} P(\mathrm{x}_f, t) = P(\mathrm{x}_f, \infty)$, where $P(\mathrm{x}_f, \infty)$ is a well-behaved and time-independent pdf; and ii) compared with the occurrence time of the slow reactions, the transient time τ_{relax} (called the relaxation period) for the virtual reactions to reach equilibrium is negligible. Essentially, the equilibrium assumption of Cao et al. is equivalent to the QSSA of Rao and Arkin in the context under consideration. The equilibrium assumption is concerned with the reactions, while the QSSA is concerned with the state, but the partition of the reaction system into fast and slow reactions is equivalent to the partition of the system into fast and slow species. Therefore, Cao et al. and Rao and Akin basically characterize the multiscale systems equivalently, although Cao et al. specify the equilibrium assumption and the partition of the system more rigorously and precisely. Under the equilibrium assumption, Cao et al. show that the propensity function $a_m(\mathrm{x}_s)$ is independent of x_f:

$$\bar{a}_m(\mathbf{x}_s) = \sum_{\mathbf{x}_f} a_m(\mathbf{x}_s, \mathbf{x}_f) P(\mathbf{x}_f; \infty). \tag{27}$$

Since under the equilibrium assumption, the states of the slow species obey the CME in (22) with the propensity function given in (27), while the states of the fast species have a stationary distribution, we can simulate the states of the slow species using the exact SSA, while generating the states of the fast species from the stationary distribution. Define $\bar{a}_0(\mathrm{x}_s) = \sum_{m\in\mathcal{M}_s} \bar{a}_m(\mathrm{x}_s)$, where $\mathcal{M}_s$ denotes the set of indexes of the slow reaction channels. Then an exact SSA step needs to generate a time τ according to an exponential pdf with parameter $\bar{a}_0(\mathrm{x}_s)$, and generate a reaction index μ according to the probability $p(\mu) = \bar{a}_\mu(\mathrm{x}_s)/\bar{a}_0(\mathrm{x}_s)$, $\mu \in \mathcal{M}_s$. We summarize the multiscale simulation algorithm in the following algorithm.

Algorithm 4: Multiscale Simulation—Partial Equilibrium Method [38]

1) Preparation: Partition the system into fast and slow reactions and species. Find the stationary pdf $P(\mathrm{x}_f, \infty)$ of the virtual process $\hat{\mathrm{x}}_f(t)$.

2) Initialization (set the initial number of molecules, set $t \leftarrow 0$).

3) Calculate the propensity function, $\bar{a}_m(\mathrm{x}_s)$, for slow reaction channels from (27).

4) Generate τ according to the pdf $p(\tau) = \bar{a}_0(\mathrm{x}_s)\exp(-\bar{a}_0(\mathrm{x}_s)\tau)$, $\tau > 0$, and generate μ according to the probability $p(\mu) = \bar{a}_\mu(\mathrm{x}_s)/\bar{a}_0(\mathrm{x}_s)$, $\mu \in \mathcal{M}_s$.

5) Set $t \leftarrow t + \tau$, and update the state vector: $\mathrm{x}_s \leftarrow \mathrm{x}_s + \boldsymbol{\nu}_\mu$ $\mathrm{x}_f \leftarrow$ a realization generated from the stationary pdf $P(\mathrm{x}_f, \infty)$.

6) Record $\mathrm{X}(t) = [\mathrm{x}_s^T, \mathrm{x}_f^T]^T$ as desired. Go to step 3, or else stop.

The most difficult part of Algorithm 4 is to compute $P(\mathrm{x}_f, \infty)$ which is used in calculating $\bar{a}_m(\mathrm{x}_s)$ and generating realizations of fast state vector. However, it is shown in [38] that only the first two moments of $P(\mathrm{x}_f, \infty)$ are needed to calculate $\bar{a}_m(\mathrm{x}_s)$. Hence the slow process $\mathrm{X}_s(t)$ can be simulated with only the first two moments of $P(\mathrm{x}_f, \infty)$ by ignoring the update on x_f in step 5.

Under the QSSA, the multiscale simulation algorithm of Rao and Arkin is the same as Algorithm 4, except the pdf $P(\mathrm{x}_f, \infty)$ in steps 1 and 5 is replaced by the pdf $P(\mathrm{x}_f|\mathrm{x}_s)$, and $\bar{a}_m(\mathrm{x}_s)$ in step 3 is calculated from (26). However, it is also difficult to find $P(\mathrm{x}_f \,|\, \mathrm{x}_s)$. It is suggested in [40] that $P(\mathrm{x}_f \,|\, \mathrm{x}_s)$ can be approximated by a Gaussian distribution and $\bar{a}_m(\mathrm{x}_s)$ replaced by $a_m(E[\mathrm{x}_f \,|\, \mathrm{x}_s], \mathrm{x}_s)$.

Instead of using the state vector $\mathrm{X}(t)$, Haseltine and Rawlings [26], as well as Goutsias [25], characterize the dynamic state of the system by using an $M \times 1$ random vector $\mathbf{Z}(t) = [Z_1(t), \ldots, Z_M(t)]^T$, where $Z_m(t) = z_m \geq 0$, if the mth reaction channel fires z_m times during the time interval $[0, t)$. Based on $\mathbf{Z}(t)$, they developed multiscale simulation methods that are in spirit similar to those in [38], [40]. It is most common to characterize the state of a chemical reaction systems by the population process $\mathrm{X}(t)$, instead of $\mathbf{Z}(t)$. Due to the space limitation, we will not further discuss the methods in [25], [26].

Since chemical reactions in gene expression are almost always of mutiscale nature, it is important to develop efficient multiscale SSAs. Although several promising multiscale SSAs have been proposed [25], [26], [38], [40], as we discussed earlier, some important issues need to be addressed. For example, in the QSSA approach of [40] and the partial equilibrium approach of [25], [26], [38], how can one efficiently compute the properties of the steady process? How can we dynamically partition the system? Can we incorporate a leap method into a multiscale SSA? Success in developing multiscale SSAs will be critical to making stochastic simulation widely applicable to gene networks and other chemical reaction systems.

CONCLUDING REMARKS

The comprehensive catalog of many known genomes and emerging high-throughput technologies, such as microarray [41], optical well arrays [42], and time-laps microscopy [43], that can provide simultaneous measurement of expression of thousands of genes, provide immense quantity of data, which enables us to investigate the genome as a system [44]. Such system-level investigation can reveal the system structures including the network of gene interactions and biochemical pathways and, more importantly, the network dynamics and functions. Therefore, systems biology is clearly an emerging field of tremendous importance. We believe that stochastic modeling, simulation, and analysis will play a very important role in system biology, since they provide a powerful tool to investigate the stochastic dynamics of gene networks and other chemical pathways. A signal processing approach will yield fruitful results in stochastic modeling, simulation, and analysis of gene networks.

ACKNOWLEDGMENTS

We would like to thank Prof. Mads Kærn at the University of Ottawa for providing us with his artwork for Figure 1.

AUTHORS

Xiaodong Cai (x.cai@miami.edu) received the B.S. degree from Zhejiang University, China, the M.Eng. degree from the National University of Singapore, Singapore, and the Ph.D. degree from the New Jersey Institute of Technology, Newark, all in electrical engineering. He was a member of Technical Staff at Lucent Technologies, New Jersey, and a senior system engineer at Sony technology center, California, in 2001. He was a postdoctoral researcher at the University of Minnesota, Minneapolis,from 2001 to 2004. Since August 2004, he has been with the Department of Electrical and Computer Engineering, University of Miami, Coral Gables. His research interests lie in the areas of statistical signal processing, communications, bioinformatics and computational system biology.

Xiaodong Wang (wangx@ee.columbia.edu) received the B.S. degree from Shanghai Jiao Tong University, Shanghai, China, the M.S. degree from Purdue University, and the Ph.D. degree from Princeton University, all in electrical engineering. From July 1998 to December 2001, he was on the faculty of the Department of Electrical Engineering, Texas A&M University. Since January 2002, he has been with the Department of Electrical Engineering, Columbia University. His research interests fall in the general areas of computing, signal processing and communications, and has published extensively in these areas. Among his publications is *Wireless Communication Systems: Advanced Techniques for Signal Reception* (Prentice Hall, 2003). He received the 1999 NSF CAREER Award and the 2001 IEEE Communications Society and Information Theory Society Joint Paper Award. He is an associate editor for *IEEE Transactions on Communications*, *IEEE Transactions on Wireless Communications*, *IEEE Transactions on Signal Processing*, and *IEEE Transactions on Information Theory*.

REFERENCES

[1] P. Smolen, D.A. Baxter, and J.H. Byrne, "Modeling transcriptional control in gene networks—Methods, recent results, and future directions," *Bull. Math. Biol.*, vol. 62, no. 2, pp. 247–292, 2000.

[2] J. Hasty, D. McMillen, F. Isaacs, and J.J. Collins, "Computational studies of gene regulatory networks: In numero molecular biology," *Nat. Rev. Genet.*, vol. 2, no. 4, pp. 268–279, Apr. 2001.

[3] H. de Jong, "Modeling and simulation of genetic regulatory systems: A literature review," *J. Comput. Biol.*, vol. 9, no. 1, pp. 67–103, 2002.

[4] I. Shmulevich, E.R. Dougherty, and W. Zhang, "From Boolean to probabilistic Boolean networks as models of genetic regulatory networks," *Proc. IEEE*, vol. 90, no. 11, pp. 1778–1792, Nov. 2002.

[5] S.A. Kauffman, "Metabolic stability and epigenesis in randomly constructed genetics nets," *J. Theor. Biol.*, vol. 22, no. 3, pp. 437–467, 1969.

[6] A. Arkin, J. Ross, and H.H. McAdams, "Stochastic kinetic analysis of developmental pathway bifurcation in phage lambda-infected Escherichia coli cells," *Genetics*, vol. 149, pp. 1633–1648, 1998.

[7] H.H. McAdams and A. Arkin, "Stochastic mechanisms in gene expression," *Proc. Natl. Acad. Sci.*, vol. 94, pp. 814–819, 1997.

[8] D.T. Gillespie, "A rigorous derivation of the chemical master equation," *Physica A*, vol. 188, pp. 402–425, 1992.

[9] D.T. Gillespie, "Exact stochastic simulation of coupled chemical reaction," *J. Phys. Chem.*, vol. 81, no. 25, pp. 2340–2361, 1977.

[10] D.T. Gillespie, "A general method for numerically simulating the stochastic time evolution of coupled chemical reactions," *J. Comput. Phys.*, vol. 22, pp. 403–434, 1976.

[11] A. Novick and M. Weiner, "Enzyme induction as an all-or-none phenomenon," in *Proc. Nat. Academy Science*, vol. 43, 1957, pp. 553–566.

[12] C.V. Rao, D.M. Wolf, and A.P. Arkin, "Control, exploitation and tolerance of intracellular noise," *Nature*, vol. 420, pp. 231–237, Nov. 2002.

[13] M. Kærn, T.C. Elston, W.J. Blake, and J.J. Collins, "Stochasticity in gene expression: From theories to phenotypes," *Nature Rev.: Genetics*, vol. 6, pp. 451–464, June 2005.

[14] J.M. Raser and E.K. O'Shea, "Noise in gene expression: Origins, consequences, and control," *Science*, vol. 309, pp. 2010–2013, Sept. 2005.

[15] L. Cai, N. Friedman, and X.S. Xie, "Stochastic protein expression in individual cells at the single molecule level," *Nature*, vol. 440, pp. 358–362, Mar. 2006.

[16] A.M. Kierzek, "STOCKS: STOChastic Kinetic Simulations of biochemical systems with Gillespie algorithm," *Bioinformatics*, vol. 18, no. 3, pp. 470–481, 2002.

[17] B. Alberts, A. Johnson, J. Lewis, M. Raff, K. Roberts, and P. Walter, *Molecular Biology of the Cell*, 4th ed. New York: Garland, 2002.

[18] W.J. Blake, M. Kærn, C.R. Cantor, and J.J. Collins, "Noise in eukaryotic gene expression," *Nature*, vol. 422, pp. 633–637, Apr. 2003.

[19] M. Kozak, "Pushing the limits of the scanning mechanism for initiation of translation," *Gene*, vol. 299, no. 1-2, pp. 1–34, 2002.

[20] M. Samoilov, A. Arkin, and J. Ross, "Signal processing by simple chemical systems," *J. Phys. Chem. A*, vol. 106, no. 43, pp. 10205–10221.

[21] L.H. Hartwell, J.J. Hopfield, S. Leibler, and A.W. Murray, "From molecular to modular cell biology," *Nature*, vol. 402, pp. c47–c52, Dec. 2, 1999.

[22] D.T. Gillespie, "Approximate accelerated stochastic simulation of chemically reacting systems," *J. Chem. Phys.*, vol. 115, no. 4, pp. 1716–1733, 2001.

[23] D.T. Gillespie and L.R. Petzold, "Improved leap-size selection for accelerated stochastic simulation," *J. Chem. Phys.*, vol. 119, no. 6, pp. 8229–8234, 2003.

[24] D.T. Gillespie, "The chemical Langevin equation," *J. Chem. Phys.*, vol. 113, no. 1, pp. 297–306, July 2000.

[25] J. Goutsias, "Quasiequlibrium approximation of fast reaction kinetics in stochastic biochemical systems," *J. Chem. Phys.*, vol. 122, art. no. 184102, 2005.

[26] E.L. Haseltine and J.B. Rawlings, "Approximate simulation of coupled fast and slow reactions for stochastic chemical kinetics," *J. Chem. Phys.*, vol. 117, no. 15, pp. 6959–6969, Oct. 2002.

[27] R. Srivastava, L. You, J. Summers, and J. Yin, "Stochastic vs. deterministic modeling of intracellular viral kinetics," *J. Theor. Biol.*, vol. 218, no. 3, pp. 309–321, 2002.

[28] H.H. McAdams and A. Arkin, "It's a noisy business! Genetic regulation at the nanomolar scale," *Trends Genetics*, vol. 15, no. 2, pp. 65–69, Feb. 1999.

[29] M.A. Gibson and J. Bruck, "Exact stochastic simulation of chemical systems with many species and many channels," *J. Phys. Chem. A*, vol. 105, no. 9, pp. 1876–1889, 2000.

[30] Y. Cao, H. Li, and L.R. Petzold, "Efficient formulation of the stochastic simulation algorithm for chemically reacting systems," *J. Chem. Phys.*, vol. 121, no. 9, pp. 4059–4067, 2004.

[31] A. Chatterjee, D.G. Vlachos, and M.A. Katsoulakis, "Binomial distribution based τ-leap accelerated stochastic simulation," *J. Chem. Phys.*, vol. 122, art. no. 024112, 2005.

[32] T. Tian and K. Burrage, "Binomial leap methods for simulating stochastic chemical kinetics," *J. Chem. Phys.*, vol. 121, no. 1, pp. 10,356–10,364, 2004.

[33] X. Cai and Z. Xu, "K-Leap methods for accelerating stochastic simulation of chemically reacting systems," submitted for publication.

[34] Y. Cao, D.T. Gillespie, and L.R. Petzold, "Efficient stepsize selection for the tau-leap simulation method," *J. Chem. Phys.*, vol. 124, no. 4, art. no. 044109, 2006.

[35] Y. Cao, D.T. Gillespie, and L.R. Petzold, "Avoid negative population in explicit poisson tau-leaping," *J. Chem. Phys.*, vol. 123, art. no. 54104, 2005.

[36] Y. Cao, D. Gillespie, and L. Petzold, "Multiscale stochastic simulation algorithm with stochastic partial equilibrium assumption for chemically reacting systems," *J. Comput. Phys.*, vol. 206, no. 2, pp. 395–411, 2005.

[37] H. Kurata, H. El-Samad, T.-M. Yi, M. Khammash, and J. Doyle, "Feedback regulation of the heat shock response in E. coli," in *Proc. 40th IEEE Conf. Decision, Control*, Orlando, FL, Dec. 2001, pp. 837–842.

[38] Y. Cao, D.T. Gillespie, and L.R. Petzold, "The slow-scale stochastic simulation algorithm," *J. Chem. Phys.*, vol. 122, art. no. 14116, 2005.

[39] J. Puchalk and A.M. Kierzek, "Bridging the gap between stochastic and deterministic regimes in the kinetic simulations of the biochemical reaction networks," *Biophys. J.*, vol. 86, pp. 1357–1372, Mar. 2004.

[40] C.V. Rao and A.P. Arkin, "Stochastic chemical kinetics and the quasi-steady-state assumption: application to the Gillespie algorithm," *J. Chem. Phys.*, vol. 18, no. 11, pp. 4999–5010, 2003.

[41] G.A. Churchill, "Fundamentals of experimenal design for cDNA microarrays," *Nat. Genetics*, vol. 32, pp. 490–495, Dec. 2002.

[42] Y. Kuang, I. Biran, and D.R. Walt, "Simultaneously monitoring gene expression kinetics and genetic noise in single cells by optical well arrays," *Anal. Chem.*, vol. 76, no. 21, pp. 6282–6286, 2004.

[43] N. Rosenfeld, J.W. Young, U. Alon, P.S. Swain, and M.B. Elowitz, "Gene regulation at the single-cell level," *Science*, vol. 307, pp. 1962–1965, Mar. 2005.

[44] H.V. Westerhoff and B.O. Palsson, "The evolution of molecular biology into systems biology," *Nature Biotechnology*, vol. 22, no. 10, pp. 1249–1252, Oct. 2004. SP