Emerging Research in the Analysis and Modeling of Gene Regulatory Networks

Ivan V. Ivanov
Texas A&M University, USA

Xiaoning Qian
Texas A&M University, USA

Ranadip Pal
Texas Tech University, USA

A volume in the Advances in Medical Technologies and Clinical Practice (AMTCP) Book Series

Medical Information Science
REFERENCE
An Imprint of IGI Global

Published in the United States of America by
Medical Information Science Reference (an imprint of IGI Global)
701 E. Chocolate Avenue
Hershey PA 17033
Tel: 717-533-8845
Fax: 717-533-8661
E-mail: cust@igi-global.com
Web site: http://www.igi-global.com

Library of Congress Cataloging-in-Publication Data

Names: Ivanov, Ivan V., 1962- | Qian, Xiaoning, 1975- | Pal, Ranadip, 1980-
Title: Emerging research in the analysis and modeling of gene regulatory networks / Ivan V. Ivanov, Xiaoning Qian, and Ranadip Pal, editors.
Description: Hershey, PA : Medical Information Science Reference, [2016] | Includes bibliographical references and index.
Identifiers: LCCN 2016005966| ISBN 9781522503538 (hardcover) | ISBN 9781522503545 (ebook)
Subjects: LCSH: Genetic regulation--Computer simulation. | Gene regulatory networks.
Classification: LCC QH450 .E44 2016 | DDC 572.8/65--dc23 LC record available at https://lccn.loc.gov/2016005966

This book is published in the IGI Global book series Advances in Medical Technologies and Clinical Practice (AMTCP) (ISSN: 2327-9354; eISSN: 2327-9370)

British Cataloguing in Publication Data
A Cataloguing in Publication record for this book is available from the British Library.

All work contributed to this book is new, previously-unpublished material. The views expressed in this book are those of the authors, but not necessarily of the publisher.

Chapter 3
Modeling Stochastic Gene Regulatory Networks Using Direct Solutions of Chemical Master Equation and Rare Event Sampling

Youfang Cao
Los Alamos National Laboratory, USA

Anna Terebus
University of Illinois – Chicago, USA

Jie Liang
University of Illinois – Chicago, USA

ABSTRACT

Stochasticity plays important roles in many biological networks. A fundamental framework for studying the full stochasticity is the Discrete Chemical Master Equation (dCME). Under this framework, the combination of copy numbers of molecular species defines the microstate of the molecular interactions in the network. The probability distribution over these microstates provide a full description of the properties of a stochastic molecular network. However, it is challenging to solve a dCME. In this chapter, we will first discuss how to derive approximation methods including Fokker-Planck equation and the chemical Langevin equation from the dCME. We

DOI: 10.4018/978-1-5225-0353-8.ch003

also discuss the widely used stochastic simulation method. After that, we focus on the direct solutions to the dCME. We first discuss the Finite State Projection (FSP) method, and then introduce the recently developed finite buffer method (fb-dCME) for directly solving both steady state and time-evolving probability landscape of dCME. We show the advantages of the fb-dCME method using two realistic gene regulatory networks.

1. INTRODUCTION

Molecular components in biological systems, such as proteins, DNA, RNAs, and substrates, are often interacting with each other in a complex reaction network to perform certain biological functions. These networks of interacting biomolecules are at the heart of the regulations of many critical cellular processes, from the regulation of gene expression (Arkin, Ross, & McAdams, 1998; Hasty, Pradines, Dolnik, & Collins, 2000; Levin, 2003; McAdams & Arkin, 1997; Ozbudak, Thattai, Kurtser, Grossman, & van Oudenaarden, 2002), signal transduction (Samoilov, Plyasunov, & Arkin, 2005), to the differentiation of stem cells (Ogawa, 1989). The biological networks are intrinsically stochastic due to thermal fluctuations (McCullagh et al., 2009). The intrinsic stochasticity in these cellular processes originates from reactions involving small copy numbers of molecules. It frequently occur in a cell when molecular concentrations are in the range of 0.1 μM to 1 nM (typically from about 10 to 100 copies in a cell) (Arkin et al., 1998). For example, the regulation of transcriptions depends on the binding of often a few proteins to a promoter site; the synthesis of protein peptides on ribosome involves a small copy number of molecules; and patterns of cell differentiation depend on initial small copy number events. In these biological processes, fluctuations due to the stochastic behavior intrinsic in small copy number events play important roles.

The importance of stochasticity in cellular functions has been well recognized (Mettetal, Muzzey, Pedraza, Ozbudak, & van Oudenaarden, 2006; Ozbudak et al., 2002; Paulsson & Ehrenberg, 2000; Volfson et al., 2006; Zhou, Chen, & Aihara, 2005). Studies of network models show that stochasticity is important for magnifying signal, sharpening discrimination, and inducing multistability (Paulsson & Ehrenberg, 2000). Understanding the stochastic nature and its consequences for cellular processes involving molecular species of small copy numbers in a network is an important problem. A fundamental framework for studying the full stochasticity is the discrete chemical master equation (dCME). Under this framework, the combination of copy numbers of molecular species defines the microscopic state of the molecular interactions in the network. By explicitly treating microscopic states of reactants, reactions can all be effectively modeled as transitions between microstates, with transition rates determined by the physical properties of the molecules and the cell environment. The reaction trajectories can be modeled in the framework of the dCME, which describes continuous time Markov chains (CTMC) equivalent to the Kolmogorov equation. The probability distribution or probability landscape over these microstates and its time-evolving behavior provide a full description of the properties of a stochastic molecular network.

However, it is challenging to solve a dCME that involves a nontrivial number of species. Analytical solutions of the chemical master equation exist only for very simple cases (Vellela & Qian, 2009). A widely used method is to carry out Monte Carlo simulations of the chemical master equation using the Stochastic Simulation Algorithm (SSA), also known as Gillespie's algorithm (Gillespie, 1977). Although the SSA approach has found wide applications, it is ineffective in simulating rare events, as most of computing time is spent on following high-probability paths (Daigle, Roh, Gillespie, & Petzold, 2011; Roh, Daigle, Gillespie, & Petzold, 2011; Roh, Gillespie, & Petzold, 2010). A recently developed importance path sampling technique, the Adaptively Biased Sequential Importance Sampling (ABSIS) has been shown to significantly improve rare event sampling efficiency (Cao & Liang, 2013). Alternatively, the chemical master equation can also be approximated using the Fokker-Planck equation (FPE) and the chemical Langevin equation (CLE) (Gillespie, 2000).

Direct solutions to the dCMEs can provide exact results, however complete identification and characterization of the state space is prerequisite, which can rapidly become intractable due to the explosion of the size of the discrete state space. Currently, the state space of a network cannot be fully characterized in general. The conventional hypercube approach predefines the maximum copy number of each reactant, and bound the state space by the product of the maximum numbers. However, the size of state spaces generated with the hypercube approach will quickly inflate to enormity, which makes it intrinsically inefficient as many unreachable states from an initial condition are included. An alternative approach is to generate states by carrying out stochastic simulations. One can simply follow explicitly simulated reaction events to whatever microstates the system reaches, such as the *N-reachability* approach used in Ref (Munsky & Khammash, 2006). However, this approach cannot guarantee that all reachable states are included, especially when the system contains rare transitions. Therefore, it cannot guarantee full characterization of rare events.

Munsky and Khammash developed the finite state projection (FSP) method to directly solve the time evolution of the dCME by using an absorbing state to truncate the state space generated by the *N-reachability* approach (Munsky & Khammash, 2006; Munsky & Khammash, 2007). However, the absorbing state in the FSP method makes it unable to solve the steady state probability landscape of dCME, which is of great importance in studying the behavior of reaction networks.

In this chapter, we first discuss the basic theoretical framework of the probability landscape of a stochastic network and the underlying discrete chemical master equation (dCME). We then discuss the connections between dCME and other modeling methods often used in studying stochastic biological networks. We further discuss the formulation of the continuous chemical master equation (cCME), which approximates the dCME, as well as its further simplifications in the form of Fokker-Planck and Langevin models. This is followed by a discussion of the approach of Monte Carlo simulations to study stochastic network, with the Gillespie's stochastic simulation algorithm discussed in some details. We also discuss a few recent developments on rare event probability estimation using biased path sampling. After discussing the advantage and challenges to directly solve the dCME, we describe a recently developed approach to significantly reduce the size of the state space, and therefore efficiently solve the steady state and time evolution probability landscapes of the dCME, namely the *finite buffer* method for direct solution of dCME (Cao & Liang, 2008; Cao, Lu, & Liang, 2010). We then describe two realistic gene regulatory networks to show the effectiveness of the fb-dCME method.

2. DISCRETE CHEMICAL MASTER EQUATION FRAMEWORK FOR MODELING STOCHASTIC NETWORKS

2.1 Discrete Chemical Master Equation (dCME)

Here we first describe the chemical master equation on a discrete state space. The dCME describes the gain and loss of probability associated with each microstate due to chemical reactions. The chemical reactions can be thought as jump processes that bring the system from one combination of copy number of molecular species (micro state) to a different combination of copy number of molecular species once a reaction occurs. The dCME describes the change of probability of different microstates connected by such jump processes due to reactions.

Molecular species and reactions: We assume a system with m molecular species $\{X_1, X_2, \ldots, X_m\}$, where X_i is the *i*-th molecular species. There are n chemical reactions $\boldsymbol{R} = (R_1, R_2, \ldots, R_n)$ that can happen in the system, and each has a rate constant $\boldsymbol{r} = (r_1, r_2, \cdots, r_n)$. We denote the copy number of the -th molecular species as . The combination of the copy numbers at time *t* is a vector of nonnegative integers and is denoted as $\boldsymbol{x}(t) = (x_1(t), x_2(t), \ldots, x_m(t)) \in \mathbb{Z}_0^+$. We call

$\boldsymbol{x}(t)$ the *microstate* of the system at time t. The probability for the system to be in state $\boldsymbol{x}(t)$ is $p(\boldsymbol{x},t)$. The collection of all possible microstates consists of the *state space* of the system: $\boldsymbol{S}=\{\boldsymbol{x}(t)|t\to\infty\}$. Its size is denoted as $|\boldsymbol{S}|$. The probability distribution over the state space $\boldsymbol{S}$ at time is the *probability landscape* $\boldsymbol{p}(\boldsymbol{S},t)$. The time-evolving and steady state probability landscape provides a full description of the properties of a stochastic molecular network (Ao, Kown, & Qian, 2007; Cao & Liang, 2008; Cao et al., 2010; Kim & Wang, 2007; Schultz, Onuchic, & Wolynes, 2007).

Stoichiometry: A chemical reaction R_k takes the general form:

$$c_{1,k}X_1+c_{2,k}X_2+\ldots+c_{m,k}X_m\xrightarrow{r_k}c_{1,k}^{'}X_1+c_{2,k}^{'}X_2+\ldots+c_{m,k}^{'}X_m, \tag{1}$$

where r_k is the rate constant associated with reaction R_k. The reaction brings the system from the microstate $\boldsymbol{x}_j$ to $\boldsymbol{x}_i$. The difference between $\boldsymbol{x}_j$ and $\boldsymbol{x}_i$ is the stoichiometry vector $\boldsymbol{s}_k$ of reaction R_k:

$$\boldsymbol{s}_k=\boldsymbol{x}_i-\boldsymbol{x}_j=\left(c_{1,k}^{'}-c_{1,k},c_{2,k}^{'}-c_{2,k},\ldots,c_{m,k}^{'}-c_{m,k}\right). \tag{2}$$

Here $\boldsymbol{s}_k$ can admit 0 entries if a molecular species does not participate in the reaction, so $\boldsymbol{s}_k$ has the same dimension as that of the microstate.

As an example, the reaction $A+2B\to C$ reduces the number of *A* and *B* by 1 and 2, respectively, and increase the number of *C* by 1. Its stoichiometry vector has $c_A=-1$, $c_B=-2, c_C=+1$, and $s=(-1,-2,+1)$. If there are other molecular species, their coefficients are all 0 for this reaction. By treating microscopic states of reactants explicitly, linear and nonlinear reactions, such as synthesis, degradation, dimeric binding, and multimerization, can all be modeled as transitions between microstates.

Reaction rate: The rate of the k-th reaction that transition the system from state $\boldsymbol{x}_j$ to state $\boldsymbol{x}_i$ is determined by the intrinsic reaction rate constant r_k, and the available copy numbers of each reactant in the current state $\boldsymbol{x}_j$:

$$A_k\left(\boldsymbol{x}_i,\boldsymbol{x}_j\right)=A_k\left(\boldsymbol{x}_j\right)=r_k\prod_{l=1}^{m}\binom{x_l}{c_{l,k}}, \tag{3}$$

assuming the convention $\begin{pmatrix} 0 \\ 0 \end{pmatrix} = 1$. If the k-th reaction can lead the system from state $\boldsymbol{x}_j$ to state $\boldsymbol{x}_i$, we have $A_k\left(\boldsymbol{x}_i, \boldsymbol{x}_j\right) > 0$, otherwise $A_k\left(\boldsymbol{x}_i, \boldsymbol{x}_j\right) = 0$. In most cases, only one reaction connects two microstates. However, since in principle more than one reaction may connect two states, we have the overall reaction rate that brings the system from $\boldsymbol{x}_j$ to $\boldsymbol{x}_i$ as:

$$A\left(\boldsymbol{x}_i, \boldsymbol{x}_j\right) = \sum_{R_k \in \boldsymbol{R}} A_k\left(\boldsymbol{x}_i, \boldsymbol{x}_j\right),$$

where $A\left(\boldsymbol{x}_i, \boldsymbol{x}_j\right)$ represents the transition probability function per unit time from $\boldsymbol{x}_j$ to $\boldsymbol{x}_i$. Overall, we have the transition rate matrix: $\boldsymbol{A} = \left\{A\left(\boldsymbol{x}_i, \boldsymbol{x}_j\right)\right\}$, where the diagonal elements are defined as: $A\left(\boldsymbol{x}_i, \boldsymbol{x}_i\right) = -\sum_{i \neq j} A\left(\boldsymbol{x}_i, \boldsymbol{x}_j\right)$.

Discrete Chemical Maser Equation. The discrete chemical master equation can then be written as:

$$\frac{dp\left(\boldsymbol{x}, t\right)}{dt} = \sum_{\boldsymbol{x}' \neq \boldsymbol{x}} \left[A\left(\boldsymbol{x}, \boldsymbol{x}'\right) p\left(\boldsymbol{x}', t\right) - A\left(\boldsymbol{x}', \boldsymbol{x}\right) p\left(\boldsymbol{x}, t\right)\right]. \tag{4}$$

Note here we regard the probability $p\left(\boldsymbol{x}, t\right)$ of a microstate $\boldsymbol{x}$ is continuous in time, while the states are all discrete. We call this the discrete chemical master equation (dCME). The dCME in this form fully account for the probabilities of jumps between states, regardless whether the copy number components of $\boldsymbol{x}$ and $\boldsymbol{x}'$ are small or large. It gives a full account for the stochasticity due to small copy number events.

In matrix form, the dCME can be written as:

$$\frac{d\boldsymbol{p}\left(\boldsymbol{S}, t\right)}{dt} = \boldsymbol{A}\boldsymbol{p}\left(\boldsymbol{S}, t\right), \tag{5}$$

where $\boldsymbol{A} \in \mathbb{R}^{|\boldsymbol{S}| \times |\boldsymbol{S}|}$ is the rate matrix formed by the collection of all $A\left(\boldsymbol{x}_i, \boldsymbol{x}_j\right)$: $\boldsymbol{A} = \left\{A\left(\boldsymbol{x}_i, \boldsymbol{x}_j\right)\right\}, \boldsymbol{x}_i, \boldsymbol{x}_j \in \boldsymbol{S}$, and $\boldsymbol{p}\left(\boldsymbol{S}, t\right)$ is the probability distribution vector of the dCME over the state space $\boldsymbol{S}$ at time t. The dCME describes the gain and loss in probability associated with each microstate due to chemical reactions. These chemical reactions can be regarded as jump processes upon firings of reactions, which bring the system from one combination of copy number of molecular species to a different combination of copy number of molecular species. The dCME fully accounts for the stochastic jumps between states, regardless whether the copy numbers $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$ are small or large. The overall stochasticity due to small copy number events is therefore fully described. It provides a fundamental framework to study stochastic molecular networks (Gillespie, 1977; Van Kampen, 2007).

2.1.1 Continuous Chemical Master Equation

If we treat the state space as continuous, that is, we assume the amount of a molecular species x_i is measured by a real value (such as concentration) instead of a copy number, the micro-state $\boldsymbol{x}(t)$ becomes a real-valued vector $\boldsymbol{x}(t) \in \mathbb{R}^m$. We have the chemical master equation equivalent to Eqn. (4) on continuous state space as:

$$\frac{dp(\boldsymbol{x},t)}{dt} = \int_0^\infty \left[A(\boldsymbol{x},\boldsymbol{x}')\, p(\boldsymbol{x}',t) - A(\boldsymbol{x}',\boldsymbol{x})\, p(\boldsymbol{x},t) \right] d\boldsymbol{x}', \tag{6}$$

where the kernel $A(\boldsymbol{x},\boldsymbol{x}')$ represents the transition probability function per unit time from $\boldsymbol{x}'$ to $\boldsymbol{x}$. The CME in this form is equivalent to the Chapman-Kolmogorov equation frequently used to describe continuous Markov processes.

Remark. The continuous state space version of the CME requires a strong assumption. It is only appropriate if one can assume that the difference in the amount of molecules in neighboring states is infinitesimally small, which is valid only if the copy number of the molecular species in the system are much larger than 1, and larger than the changes in the numbers of molecules when a reaction occurs. The continuous CME therefore cannot be used when the total amount of molecules involved is very small, for example, in systems of single or a handful of particles. In these cases, the discrete CME should be used, as it does not contain any intrinsic singularity difficulties.

2.1.2 Relationship to Law of Mass Action Equations

Based on the above definition of biological networks, we can now derive the deterministic mass action equations from the discrete chemical master equation, which describe the time dynamics of the mean value of concentration of each molecular species. By multiplying the copy number of each molecular species x_i to both sides of the Eqn. (4), and taking summation over all elements from zero to infinity, we can obtain the average concentration x_i of species X_i:

$$\frac{d\langle x_i\rangle}{dt}=\frac{d\left(\sum_{\boldsymbol{x}\in\boldsymbol{S}} x_i p(\boldsymbol{x},t)\right)}{dt}=$$
$$\sum_{k=1}^{m}\left(\sum_{x_1=0}^{\infty}\sum_{x_2=0}^{\infty}\cdots\sum_{x_n=0}^{\infty} x_i A_k(\boldsymbol{x}-\boldsymbol{s}_k)\,p(\boldsymbol{x}-\boldsymbol{s}_k,t)\right)-\sum_{k=1}^{m}\left(\sum_{x_1=0}^{\infty}\sum_{x_2=0}^{\infty}\cdots\sum_{x_n=0}^{\infty} x_i A_k(\boldsymbol{x})\,p(\boldsymbol{x},t)\right) \tag{7}$$

After adjusting the indexes of the terms at the right hand side, we can obtain:

$$\frac{d\langle x_i\rangle}{dt}=\sum_{k=1}^{m}\left(\sum_{x_1=0}^{\infty}\sum_{x_2=0}^{\infty}\cdots\sum_{x_n=0}^{\infty}\boldsymbol{s}_{ik} A_k(\boldsymbol{x})\,p(\boldsymbol{x},t)\right). \tag{8}$$

As $\langle A_k(\boldsymbol{x})\rangle = A_k(\boldsymbol{x})\,p(\boldsymbol{x},t)$, we therefore obtain the deterministic equation of the mean concentration of molecular species X_i as:

$$\frac{d\langle x_i\rangle}{dt}=\sum_{k=1}^{m}\boldsymbol{s}_{ik}\langle A_k(\boldsymbol{x})\rangle. \tag{9}$$

The same equation can be derived from all molecular species $X_i, i=1,\cdots,n$. If we can assume the equality $\langle A_k(\boldsymbol{x})\rangle = A_k(\langle\boldsymbol{x}\rangle)$, then the above equation can be used to accurately describe the average behavior of each molecular species $X_i, i=1,\cdots,n$. However, equality $\langle A_k(\boldsymbol{x})\rangle = A_k(\langle\boldsymbol{x}\rangle)$ only holds when $A_k(\boldsymbol{x})$ is a linear function of $\boldsymbol{x}$. Whenever a reaction has two or more reactants, its reaction rate $A_k(\boldsymbol{x})$ becomes a nonlinear function of $\boldsymbol{x}$, therefore $\langle A_k(\boldsymbol{x})\rangle \neq A_k(\langle\boldsymbol{x}\rangle)$. According to van Kampen, a Taylor expansion of $A_k(\boldsymbol{x})$ can be used to approximate the $\langle A_k(\boldsymbol{x})\rangle$ (Van Kampen, 2007):

$$\langle A_k(\boldsymbol{x})\rangle = A_k(\langle\boldsymbol{x}\rangle)+\frac{1}{2}\left\langle(\boldsymbol{x}-\boldsymbol{x})^2\right\rangle A_k^{''}(\langle\boldsymbol{x}\rangle)+\cdots. \tag{10}$$

By truncating the second order and higher terms, we can approximate the mean reaction rates as:

$$\left\langle A_k\left(\boldsymbol{x}\right)\right\rangle \approx A_k\left(\left\langle \boldsymbol{x}\right\rangle\right). \tag{11}$$

Therefore, we obtain the deterministic mass action equation as:

$$\frac{dx_i}{dt} = \sum_{k=1}^{m} \boldsymbol{s}_{ik} A_k\left(\boldsymbol{x}\right). \tag{12}$$

We can rewrite the equation in matrix form as:

$$\boldsymbol{x}' = \boldsymbol{s}\boldsymbol{A}\left(\boldsymbol{x}\right), \tag{13}$$

where is the stoichiometry matrix of the reaction network, and $\boldsymbol{A}\left(\boldsymbol{x}\right) = \left(A_1\left(\boldsymbol{x}\right), \cdots, A_{\boldsymbol{m}}\left(\boldsymbol{x}\right)\right)^T$ is a vector of rate functions of all reactions in the network. The above mass action kinetic equations show the relationship between discrete chemical master equation and its corresponding deterministic equations. This is consistent with Kurtz and Keizer's theoretical studies about the relationship between the stochastic and deterministic models of chemical reactions (Keizer, 1977; Kurtz, 1972).

2.2 Approximations

2.2.1 Fokker-Planck Equation

Solving the CME requires knowledge of all details of the transition kernel $A\left(\boldsymbol{x}, \boldsymbol{x}'\right)$, including both the state space and the transition rates between them. Currently, there exist no general analytical solution to the CME, except for the simplest problems (Schultz et al., 2007; Vellela & Qian, 2007). Solving the CME numerically is also challenging. Nevertheless, for systems of small and moderate size, there exists algorithm that can enumerate optimally the state space, with $A\left(\boldsymbol{x}, \boldsymbol{x}'\right)$ fully characterized, and the discrete CME can be solved numerically (Cao & Liang, 2008).

One can approximate CME with the Fokker-Planck Equation (FPE). Similar to the continuous CME, the FPE also describe the evolution of probabilities of system states over time, but with the transition kernel in the CME replaced by a differential operator of second order. Unlike the discrete CME, the states of the system are continuous macroscopic states, and FPE does not describe the discrete jump processes (Van Kampen, 2007). We follow the disposition of van Kampen (Van Kampen,

2007) and briefly describe how FPE is related to CME, focusing on the additional assumptions and approximations involved beyond those made for continuous CME.

2.2.1.1 Assumptions of Fokker-Planck Equation

The first assumption is that the jumps between states must be small, namely, the "before" and the "after" states are in a close neighborhood: $\left|\boldsymbol{x}_j - \boldsymbol{x}_i\right| < \epsilon$, where $\epsilon \in \mathbb{R}$ is infinitesimal. Second, the transition probability varies slowly: $A(\boldsymbol{x},\boldsymbol{x}') \approx A(\boldsymbol{x}+\epsilon,\boldsymbol{x}'+\epsilon)$. Third, the probability $p(\boldsymbol{x},t)$ must also vary slowly: $p(\boldsymbol{x},t) \approx p(\boldsymbol{x}+\epsilon,t)$. A consequence of these assumptions is that the transition kernel $A(\boldsymbol{x},\boldsymbol{x}')$ is differentiable to a higher order.

With these assumptions, the first term in Eqn. (6) can be approximated, where the full detail of the transition kernel $A(\boldsymbol{x},\boldsymbol{x}')$ from $\boldsymbol{x}'$ to $\boldsymbol{x}$ is needed. The goal is to replace $A(\boldsymbol{x},\boldsymbol{x}')$ with its Taylor expansion centered around $\boldsymbol{x}$. For this, we first reparameterize $A(\boldsymbol{x},\boldsymbol{x}')$ as $A(\boldsymbol{x};\boldsymbol{s})$, where $\boldsymbol{s} = \boldsymbol{x} - \boldsymbol{x}'$. Eqn. (6) can be rewritten as:

$$\frac{dp(\boldsymbol{x},t)}{dt} = \int A(\boldsymbol{x}-\boldsymbol{s};\boldsymbol{s})\,p(\boldsymbol{x}-\boldsymbol{s},t)\,d\boldsymbol{s} - p(\boldsymbol{x},t)\int A(\boldsymbol{x};-\boldsymbol{s})\,d\boldsymbol{s}. \tag{14}$$

2.2.1.2 Kramers-Moyal Expansion

The first term of Eqn. (6) is then expanded around using Taylor expansion as:

$$\begin{aligned}\int A(\boldsymbol{x}-\boldsymbol{s};\boldsymbol{s})\,p(\boldsymbol{x}-\boldsymbol{s},t)\,d\boldsymbol{s} = \int A(\boldsymbol{x};\boldsymbol{s})\,p(\boldsymbol{x},t)\,d\boldsymbol{s} &- \sum_i \frac{\partial}{\partial x_i}\left[\int \boldsymbol{s}\cdot A(\boldsymbol{x};\boldsymbol{s})\,d\boldsymbol{s}\cdot p(\boldsymbol{x},t)\right] \\ &+ \frac{1}{2}\sum_{i,j}\frac{\partial^2}{\partial x_i \partial x_j}\left[\int \boldsymbol{s}^2\cdot A(\boldsymbol{x};\boldsymbol{s})\,d\boldsymbol{s}\cdot p(\boldsymbol{x},t)\right] \\ &- \frac{1}{3!}\sum_{i,j,k}\frac{\partial^3}{\partial x_i \partial x_j \partial x_k}\left[\int \boldsymbol{s}^3\cdot A(\boldsymbol{x},\boldsymbol{s})\,d\boldsymbol{s}\cdot p(\boldsymbol{x},t)\right] + \cdots\end{aligned} \tag{15}$$

Putting it back to Eqn. (6), and approximating by dropping higher than second order terms, we obtain the Fokker-Planck equation:

$$\frac{\partial p(\boldsymbol{x},t)}{\partial t} = -\sum_i \frac{\partial}{\partial x_i}\left[\int \boldsymbol{s}\cdot A(\boldsymbol{x};\boldsymbol{s})\,d\boldsymbol{s}\cdot p(\boldsymbol{x},t)\right] + \frac{1}{2}\sum_{i,j}\frac{\partial^2}{\partial x_i \partial x_j}\left[\int \boldsymbol{s}^2\cdot A(\boldsymbol{x};\boldsymbol{s})\,d\boldsymbol{s}\cdot p(\boldsymbol{x},t)\right] \tag{16}$$

Using a simpler notation, we have:

$$\frac{dp\left(\boldsymbol{x},t\right)}{dt}=-\sum_i\frac{\partial}{\partial x_i}\left[F\left(\boldsymbol{x}\right)p\left(\boldsymbol{x},t\right)\right]+\frac{1}{2}\sum_{i,j}\frac{\partial^2}{\partial x_i\partial x_j}\left[G\left(\boldsymbol{x}\right)p\left(\boldsymbol{x},t\right)\right], \tag{17}$$

Where $F\left(\boldsymbol{x}\right)=\int \boldsymbol{s}\cdot A\left(\boldsymbol{x};\boldsymbol{s}\right)d\boldsymbol{s}$ and $G\left(\boldsymbol{x}\right)=\int \boldsymbol{s}^2\cdot A\left(\boldsymbol{x};\boldsymbol{s}\right)d\boldsymbol{s}$.

2.2.2 Chemical Langevin Equation

When the macroscopic behavior of the system can be determined, a general approach to study stochastic dynamic network is to follow the Langevin equation by combining a deterministic term describing the macroscopic behavior of the network and a diffusion term describing the stochastic fluctuation.

From the dCME, we can derive the average behavior of the network using the Mayo expansion as:

$$\frac{d\overline{\boldsymbol{x}}\left(t\right)}{dt}=\sum_x\frac{d\left[x\cdot p\left(\boldsymbol{x},t\right)\right]}{dt}=\frac{d\sum_x\left[x\cdot p\left(\boldsymbol{x},t\right)\right]}{dt}, \tag{18}$$

where $\overline{\boldsymbol{x}}\left(t\right)=\sum_x x\cdot p\left(\boldsymbol{x},t\right)$ is the vector of expected mean value of the amount of molecular species. Van Kampen and others have shown that the following formula could be derived from the dCME for the mean amount of molecular species (Gillespie, 2007; Melykuti, Burrage, & Zygalakis, 2010; Van Kampen, 2007):

$$\frac{d\overline{\boldsymbol{x}}\left(t\right)}{dt}=\sum_{k\in\boldsymbol{R}}s_k\overline{A}_k\left(\boldsymbol{x}\right), \tag{19}$$

where s_k and $A_k\left(\boldsymbol{x}\right)$ are defined in Eqn. (2) and Eqn. (3). This gives the same expression as that of corresponding ordinary differential equations (ODE) of mass action kinetics of the same chemical reactions. The detailed derivation is shown in following sections.

Random fluctuations in the copy numbers of molecules occur because of the random jumps due to the spontaneous firing of reactions. Such reactions will introduce changes to the copy number of molecular species, *e.g.*, by the amount of $\boldsymbol{s}_k$ of the k-th reaction. Assuming that the jump is small, namely,

$\boldsymbol{x}\left(t+\Delta t\right)=\boldsymbol{x}\left(t\right)+\boldsymbol{s}_k \approx \boldsymbol{x}\left(t\right)$ if reaction $\boldsymbol{s}_k$ occurs during an infinitesimally small time interval Δt. This assumption would result in unchanged reaction rate:

$$A_k\left(\boldsymbol{x}\left(t+\Delta t\right)\right)\approx A_k\left(\boldsymbol{x}\left(t\right)\right)=r_k\prod_{l=1}^{n}\binom{x_l}{c_{lk}}. \tag{20}$$

With this assumption, the vector of the amount of molecular species $\boldsymbol{x}\left(t+\Delta t\right)$ at time $t+\Delta t$ can be written as:

$$\boldsymbol{x}\left(t+\Delta t\right)=\boldsymbol{x}\left(t\right)+\sum_{R_k\in\boldsymbol{R}} n_k\left(\boldsymbol{x},\Delta t\right)\cdot\boldsymbol{s}_k, \tag{21}$$

where $n_k\left(\boldsymbol{x},\Delta t\right)$ is the number of reaction R_k occurs during the period Δt. Based on the assumption $\boldsymbol{x}\left(t+\Delta t\right)=\boldsymbol{x}\left(t\right)$, the copy numbers of molecular species in Eqn. (21) do not change during Δt, and the reaction rates also do not change during Δt. Therefore, all reactions occurring during Δt could be considered independent of each other.

This assumption is valid only if the copy numbers in $\boldsymbol{x}\left(t\right)$ are all large, so the stoichiometry coefficients c_i forming the jump vector $\boldsymbol{s}_k$ are all comparatively small. This assumption clearly breaks down when the copy number of molecular species is not significantly larger than the stoichiometry coefficients, and therefore cannot be used to describe systems of a handful of particles.

A reasonable model for the random variable $n_k\left(\boldsymbol{x},\Delta t\right)$ is that of the Poisson distribution: $n_k \sim P\left(\lambda_k\right)$ with $\lambda_k = A_k\left(\boldsymbol{x}\left(t\right)\right)\cdot\Delta t$. With the additional assumption that Δt is sufficiently long such that a large number (>>1) of reactions occur during Δt, the Poisson distribution for the number of spontaneous reactions can be approximated by a Gaussian random variable (Gillespie, 2000). The approximation will be very accurate when λ_k is sufficiently large, *e.g.*, $\lambda_k > 1000$. Therefore, we now have $n_k \sim N\left(\mu,\sigma^2\right)$, with $\mu=\sigma^2=\lambda_k$, or alternatively, $n_k \sim \lambda_k + \lambda_k^{\frac{1}{2}}N\left(0,1\right)=A_k\left(\boldsymbol{x}\left(t\right)\right)\cdot\Delta t+\left[A_k\left(\boldsymbol{x}\left(t\right)\right)\cdot\text{”}\,t\right]^{\frac{1}{2}}N\left(0,1\right)$. With these assumptions, the fluctuations of the amount of molecules follow *m* independent Gaussian processes, one for each reaction:

$$\boldsymbol{x}\left(t+\Delta t\right)=\boldsymbol{x}\left(t\right)+\sum_{R_k\in\boldsymbol{R}}A_k\left(\boldsymbol{x}\left(t\right)\right)\cdot\Delta t\cdot\boldsymbol{s}_k+\sum_{R_k\in\boldsymbol{R}}\left[A_k\left(\boldsymbol{x}\left(t\right)\right)\cdot\Delta t\right]^{\frac{1}{2}}\cdot\boldsymbol{s}_k\cdot N\left(0,1\right). \tag{22}$$

This leads to the following equation:

$$\frac{\partial\boldsymbol{x}(t)}{dt}=\sum_{R_k\in\boldsymbol{R}}A_k\left(\boldsymbol{x}(t)\right)\cdot\boldsymbol{s}_k+\sum_{R_k\in\boldsymbol{R}}\left[A_k\left(\boldsymbol{x}(t)\right)\right]^{\frac{1}{2}}\cdot\boldsymbol{s}_k\cdot\left[\frac{1}{\partial t^{\frac{1}{2}}}\cdot N\left(0,1\right)\right]. \tag{23}$$

We denote $G\left(t\right)=\frac{1}{\partial t^{\frac{1}{2}}}\cdot N\left(0,1\right)=N\left(0,\frac{1}{\partial t}\right)$, we have the Langevin equation:

$$\frac{\partial\boldsymbol{x}(t)}{\partial t}=\sum_{R_k\in\boldsymbol{R}}A_k\left(\boldsymbol{x}(t)\right)\cdot\boldsymbol{s}_k+\sum_{R_k\in\boldsymbol{R}}\left[A_k\left(\boldsymbol{x}(t)\right)\right]^{\frac{1}{2}}\cdot\boldsymbol{s}_k\cdot G\left(t\right). \tag{24}$$

This is same as the chemical Langevin equation described by Gillespie (Gillespie, 2000).

2.2.3 Accuracy of Chemical Fokker-Planck and Langevin Equation

As shown above, due to the second-order truncation in the Kramers-Moyal expansion during the derivation of the Fokker-Planck and Langevin equation from the chemical master equation, the accuracy and error of these equations need to be clarified. Using the system-size expansion, Grima *et al.* (Grima, Thomas, & Straube, 2011) have shown that the estimates for the mean and variance of concentrations using the chemical Fokker-Planck equation are accurate to order $\Omega^{-\frac{3}{2}}$ for reaction networks which do not obey detailed balance, and at least accurate to order Ω^{-2} for systems obeying detailed balance, where Ω is the characteristic size of the system.

2.3 Monte-Carlo Simulation of the dCME

A widely used method to study stochastic networks is to carry out Monte Carlo simulations. By following the trajectories of reactions, one can gather statistics of reaction events at different time to gain understanding of the network behavior (Gillespie, 1977). We discuss the underlying algorithm, called the *stochastic simulation algorithm* (SSA), which is also known as *Gillespie's* algorithm.

Reaction probability: We denote the probability that after the last reaction at *t*, the next reaction, which happens to be the k-th reaction, occurs during an infinitesimally small time interval dt after a time interval Δt of no reaction occurring to be:

$$p\left[\boldsymbol{x}\left(t+\Delta t+dt\right),\Delta t,R_k|\boldsymbol{x}\left(t\right)\right]dt. \tag{25}$$

If we divide the time interval Δt into *H* equal subintervals, and assume that the occurrence of reactions following a Poisson process, the probability that none of the *n* reactions have occurred during the time prior to the end of a small time interval $\epsilon = \Delta t / H$ is:

$$\prod_{k=1}^{n}\left[1-A_k\left(\boldsymbol{x}(t)\right)\epsilon\right]\approx\sum_{k=1}^{n}\left[1-A_k\left(\boldsymbol{x}(t)\right)\epsilon\right]. \tag{26}$$

As the probability of no reactions for each of the *H* intervals is the same, no reactions have occurred during $” t$ is:

$$\lim_{H\to\infty}\sum_{k=1}^{n}\left[1-A_k\left(\boldsymbol{x}(t)\right)\epsilon\right]^{H}=e^{-A(\boldsymbol{x}(t))” t}. \tag{27}$$

As no reaction has occurred during Δt, the microstate remains the same. Therefore, the instantaneous state transition probability for reaction R_k occurring during the infinitesimal interval $\left[t+\Delta t, t+\Delta t+dt\right]$ is $A_k\left[\boldsymbol{x}\left(t+\Delta t\right)\right]dt = A_k\left(\boldsymbol{x}\left(t\right)\right)dt.$ We have:

$$p\left[\boldsymbol{x}\left(t+\Delta t+dt\right),\Delta t,R_k|\boldsymbol{x}\left(t\right)\right]=A_k\left[\boldsymbol{x}\left(t\right)\right]e^{-A(\boldsymbol{x}(t))\Delta t}dt. \tag{28}$$

Reaction trajectory. Let the microstate of the system at time *t* to be $\boldsymbol{x}\left(t\right)$. After time interval Δt, reaction R_k occurs at an infinitesimally small time interval dt at $t+\Delta t$, and the system is brought to the state $\boldsymbol{x}\left(t+\Delta t+dt\right)$. We assume the reactions are occurring instantaneously, *i.e.* $dt \to 0$, so we use $\boldsymbol{x}\left(t+\Delta t\right)$ to denote the new microstate after the reaction occurred in dt. We can observe the trajectory of a sequence of such reactions. Starting from state $\boldsymbol{x}\left(0\right)$ at time $t_0 = 0$, after a series of time intervals $\left(\Delta t_0, \Delta t_1, \cdots, \Delta t_{T-1}\right)$, the system reaches the state $\boldsymbol{x}\left(t_T\right)$, after traversing the sequence of states $\left(\boldsymbol{x}\left(t_0\right),\boldsymbol{x}\left(t_1\right),\cdots,\boldsymbol{x}\left(t_{T-1}\right)\right)$, with a reaction sequence $\left(R_{k_0},R_{k_1},\cdots,R_{k_{T-1}}\right)$ occurring along the way. Let $\left(t_1,t_2,\cdots,t_T\right)$ be the sequence of time points when a reaction occurs. The trajectory of reactions can be denoted as:

$$\left(\boldsymbol{x}\left(t_0\right);\boldsymbol{x}\left(t_1\right),R_{k_0};\cdots;\boldsymbol{x}\left(t_{T-1}\right),R_{k_{T-2}};\boldsymbol{x}\left(t_T\right),R_{k_{T-1}}\right).$$

Alternatively, we can denote the time intervals by its increments $\left(\Delta t_0,\Delta t_1,\cdots,\Delta t_{T-1}\right)$.

Probability of a reaction trajectory. Assuming a Markovian process, namely, future reactions depends only on the current state but not on any past state, the probability associated with a time trajectory is:

$$\begin{aligned}&\pi\left[\boldsymbol{x}\left(t_0\right);\boldsymbol{x}\left(t_1\right),R_{k_0};\cdots;\boldsymbol{x}\left(t_{T-1}\right),R_{k_{T-2}};\boldsymbol{x}\left(t_T\right),R_{k_{T-1}}\right]\\&=\pi\left[\boldsymbol{x}\left(t_1\right),\Delta t_0,R_{k_0}|\boldsymbol{x}\left(t_0\right)\right]\cdot\pi\left[\boldsymbol{x}\left(t_2\right),\Delta t_1,R_{k_1}|\boldsymbol{x}\left(t_1\right)\right]\cdots\pi\left[\boldsymbol{x}\left(t_T\right),\Delta t_{T-1},R_{k_{T-1}}|\boldsymbol{x}\left(t_{T-1}\right)\right].\end{aligned}\tag{29}$$

In principle, the probability of starting from state $\boldsymbol{x}\left(t_0\right)$ and reaching state $\boldsymbol{x}\left(t_T\right)$ can then be obtained by integrating over all possible paths:

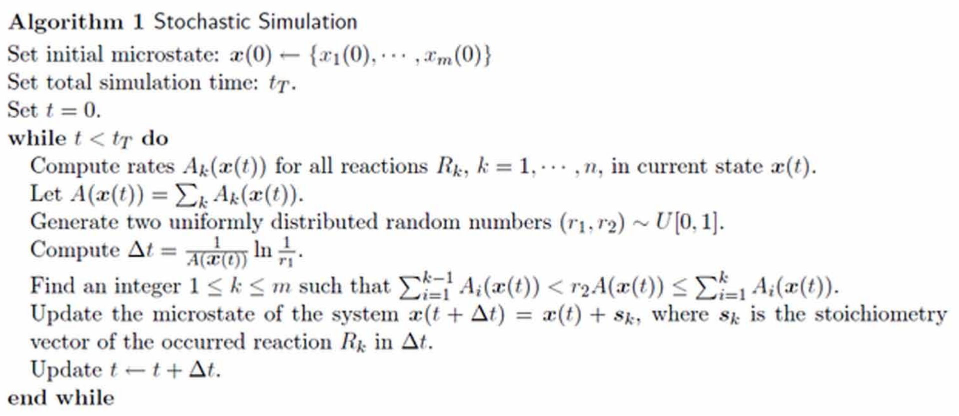
$$\pi\left[\boldsymbol{x}\left(t_T\right)|\boldsymbol{x}\left(x_0\right)\right]=\sum_{\left(\Delta t_0,\cdots,\Delta t_{T-1}\right);\left(R_{k_0},\cdots,R_{k_{T-1}}\right)}\pi\left[\boldsymbol{x}\left(t_0\right);\boldsymbol{x}\left(t_1\right),R_{k_0};\cdots;\boldsymbol{x}\left(t_{T-1}\right),R_{k_{T-2}};\boldsymbol{x}\left(t_T\right),R_{k_{T-1}}\right]\tag{30}$$

2.3.1 Stochastic Simulation Algorithm

If we can generate many independent samples of reaction trajectories that follow a proper probabilistic model starting from the same initial state, we can study the behavior of the stochastic network by analyzing these sample trajectories. The stochastic simulation algorithm (SSA) or Gillespie's algorithm was designed to perform such simulations (Gillespie, 1977). It is summarized in the Algorithm in Figure 1.

Generating random variables for selecting the time interval $\boldsymbol{\Delta t}$ *and reaction* $\boldsymbol{R_k}$. A key component in Gillespie's algorithm is to generate a pair of random variables $\left(\Delta t,k\right)$. The Δt will be the time interval until next reaction occurs, and the number *k* will specify the reaction R_k that will actually occur next. We have: $\pi\left(\Delta t,k\right)=\pi_1\left(\Delta t\right)\cdot\pi_2\left(R_k|\Delta t\right)$, where $\pi_1\left(\Delta t\right)$ is the probability that the next

Figure 1. Stochastic Simulation Algorithm



```
Algorithm 1 Stochastic Simulation
Set initial microstate: $\boldsymbol{x}(0) \leftarrow \{x_1(0), \cdots, x_m(0)\}$
Set total simulation time: $t_T$.
Set $t = 0$.
while $t < t_T$ do
    Compute rates $A_k(\boldsymbol{x}(t))$ for all reactions $R_k$, $k = 1, \cdots, n$, in current state $\boldsymbol{x}(t)$.
    Let $A(\boldsymbol{x}(t)) = \sum_k A_k(\boldsymbol{x}(t))$.
    Generate two uniformly distributed random numbers $(r_1, r_2) \sim U[0,1]$.
    Compute $\Delta t = \frac{1}{A(\boldsymbol{x}(t))} \ln \frac{1}{r_1}$.
    Find an integer $1 \le k \le m$ such that $\sum_{i=1}^{k-1} A_i(\boldsymbol{x}(t)) < r_2 A(\boldsymbol{x}(t)) \le \sum_{i=1}^{k} A_i(\boldsymbol{x}(t))$.
    Update the microstate of the system $\boldsymbol{x}(t + \Delta t) = \boldsymbol{x}(t) + \boldsymbol{s}_k$, where $\boldsymbol{s}_k$ is the stoichiometry
    vector of the occurred reaction $R_k$ in $\Delta t$.
    Update $t \leftarrow t + \Delta t$.
end while
```

reaction, regardless which specific ones, will occur in time interval $\left[t + \Delta t, t + \Delta t + dt\right]$, and $\pi_2(R_k \mid \Delta t)$ is the probability that the next reaction will be the k-th reaction. As $\pi_1\left(\Delta t\right) = \sum_i \pi\left(i, \Delta t\right)$, where $\pi\left(i, \Delta t\right)$ is the probability that reaction R_i occurs at time $t + \Delta t + dt$, we have $\pi_2\left(R_k | \Delta t\right) = \dfrac{\pi\left(k, \Delta t\right)}{\sum_i \pi\left(i, \Delta t\right)}$.

As we assume a model of Poisson process, we have:

$$\pi_1\left(\Delta t\right) = A\left(\boldsymbol{x}\left(t\right)\right) e^{-A(\boldsymbol{x}(t))\Delta t}, \text{and } \pi_2\left(R_k | \Delta t\right) = \frac{A_k\left(\boldsymbol{x}\left(t\right)\right)}{A\left(\boldsymbol{x}\left(t\right)\right)}. \tag{31}$$

That is, if we can generate a random variable Δt following $\pi_1\left(\Delta t\right)$, and another random integer k according to $\pi_2\left(R_k | \Delta t\right)$, the resulting pair $\left(\Delta t, k\right)$ will follow the desired distribution $\pi\left(\Delta t, k\right)$.

Assume we can generate a random number r following the uniform distribution $r \sim U\left[0,1\right]$. A general approach to obtain a random variable ρ that follows a distribution F is to calculate the transformation of r through the inverse function F^-: $\rho = F^-\left(r\right)$. Since $\pi_1\left(\Delta t\right) = e^{-A(\boldsymbol{x}(t))\Delta t}$, we can have:

$$\Delta t = \frac{1}{A\left(\boldsymbol{x}\left(t\right)\right)} \ln \frac{1}{r_1}, \text{where } r_1 \sim U\left[0,1\right]. \tag{32}$$

To sample the next reaction R_k, we can again first generate a uniformly distributed random number $r_2 \sim U\left[0,1\right]$. We can then take the k -th reaction such that:

$$\sum_{i=1}^{k-1} A_i\left(\boldsymbol{x}(t)\right) < r_2 A\left(\boldsymbol{x}(t)\right) \leq \sum_{i=1}^{k} A_i\left(\boldsymbol{x}(t)\right), \text{where } A\left(\boldsymbol{x}(t)\right) = \sum_{i=1}^{m} A_i\left(\boldsymbol{x}(t)\right). \tag{33}$$

Another approach to generate a pair of random number $\left(\Delta t, k\right)$ is to first calculate the probability at time $t + \Delta t$ for a reaction R_i to occur during an infinitesimal time interval $\left[t + \Delta t, t + \Delta t + dt\right]$, assuming that there were no changes between $\boldsymbol{x}\left(t\right)$ and $\boldsymbol{x}\left(t + \Delta t\right)$, namely, there is no reaction occurred. We can generate a tentative reaction time Δt_l for reaction R_l as:

$$\Delta t_l = \frac{1}{A_l\left(\boldsymbol{x}\left(t\right)\right)} \ln \frac{1}{r_l}, \text{where } r_l \sim U\left[0,1\right]. \tag{34}$$

From this set of random number pairs $\left(\Delta t_l, l\right)$ for all $l = 1, \cdots m$, we select the pair of random numbers with the shortest Δt, at which the next reaction R_k would occur:

$$\Delta t = \min\left(\Delta t\right), \text{and } k = \arg_l \min\left(\Delta t_l\right). \tag{35}$$

Remark. There are a number of issues in carrying out studies using stochastic simulation, as adequate sampling is challenging when the network becomes complex. There is no general guarantee that simulation can provide a full account of the network stochasticity, as it is difficult to determine whether simulations are extensive enough for accurate statistics. It is also difficult to determine whether adequate sampling has been achieved for individual trajectory. In addition, it is often difficult to characterize rare events that may be biologically important, as simulations follow high probability paths. Much recent work has been focused on improving SSA, for example, by introducing data structure so the generation of the two random variables of Δt and reaction *k* is more efficient. In addition, an approach to speed up SSA is to find the best time step Δt such that the copy numbers of the molecular species, hence the reaction rates, do not change much, so the simulation can leap forward with large time step (Cao, Gillespie, & Petzold, 2006).

2.3.2 Biased Sampling for Rare Events

Recent interests in introducing bias in selection of the next reaction, and in altering the reaction rate showed promise in improved sampling of rare events (Daigle et al., 2011; Kuwahara & Mura, 2008; Roh et al., 2011). The techniques of importance sampling and reweighting can improve sampling efficiency significantly. They have been widely used in equilibrium sampling where the condition of detailed balance holds (Torrie & Valleau, 1977). However, stochastic processes in reaction networks are generally not time reversible and the condition of detailed balance is not valid. Kuwahara and Mura developed the weighted SSA (wSSA) algorithm by applying the importance sampling technique to study stochastic reaction networks, in which each reaction rate is biased by a pre-determined constant, with the overall summation of reaction rates unchanged (Kuwahara & Mura, 2008). As the probability for reaction selection can be biased such that rare events are sampled more frequently while the time scale of the underlying reactions is maintained, significantly improved sampling efficiency for rare events was reported (Daigle et al., 2011; Kuwahara & Mura, 2008; Roh et al., 2011). However, the choice of bias constants strongly affects the effectiveness of wSSA. When there are many reactions and the network is complex, the heuristic approach of determining bias constants by examining the reactions does not work (Kuwahara & Mura, 2008). As there is no general guidance in how bias constants should be chosen, poor choices may lead to estimations that are less accurate than the original SSA (Daigle et al., 2011).

Daigle *et al.* developed the doubly-weighted SSA (dwSSA) algorithm, in which a multilevel cross-entropy (CE) method is used iteratively to provide estimates of bias constants (Daigle et al., 2011). This is achieved by running long trial simulations until a fraction of the sampled trajectories reaches the target states (Daigle et al., 2011). With this automated estimation, both reaction selection and the underlying time scale of reactions can be biased (Daigle et al., 2011).

A drawback of methods using constant biases such as wSSA and dwSSA is that the bias coefficients are global and state-independent, and are not influenced by the concentrations of molecules which evolve with time. As the apparent rate of a reaction can vary dramatically depending on the copy number of molecules, the degree of bias for a reaction therefore need to be adjusted according to the available copy numbers of reactants. With globally fixed bias constants, a network with reactions of a wide range of rates will have over- and under-biased reactions, depending on the states of the system. As a result, estimated properties of a network will have large variance, making these methods unsuitable for complex networks (Roh et al., 2010).

Roh *et al.* developed a state-dependent biasing wSSA method (swSSA) (Roh et al., 2010). By empirically classifying reactions into groups of favored, disfavored, and neutral reactions, biases in selection probability for reactions in the first two groups are calculated in a state-dependent fashion. The swSSA method can have better estimation accuracy and efficiency than the wSSA method (Roh et al., 2010), at the expense of about twice as many biasing parameters as that of the wSSA (Roh et al., 2010). Roh *et al.* further developed the state-dependent doubly weighted SSA method (sdwSSA), where reactions are further grouped into bins according to their selection probabilities, and are assigned different bias constants, which are automatically estimated using the cross-entropy method (Roh et al., 2011). However, the number of parameters to be estimated using sdwSSA is much larger than that of wSSA, dwSSA, and swSSA. For example, about 20 bias constants need to be estimated for a simple reversible isomerization system with only two reactions (Roh et al., 2011). Estimating a large number of bias constants needed for complex networks becomes difficult.

2.3.3 Adaptively Biased Sequential Importance Sampling for Rare Events (ABSIS)

To more efficiently sample rare events in biological networks, a novel biased sampling method named *adaptively biased sequential importance sampling* (ABSIS) has been developed by adaptively adjusted bias of reactions based on look-ahead strategy. The method has shown that barrier-crossing can be engineered for efficient and accurate sampling of rare events (Cao & Liang, 2013). Based on the principle of sequential importance sampling, the ABSIS approach adopts the look-ahead strategy, a technique well-established in polymer and protein studies (Liang, Zhang, & Chen, 2002; Lin, Chen, & Liu, 2009; Meirovitch, 1982, 1988), to gather future information for design of bias parameters to enable effective barrier crossings (Liang et al., 2002; Lin et al., 2009; Liu, Chen, & Logvinenko, 2001; Meirovitch, 1982, 1988). By enumerating short paths from the current state, bias coefficients are can be dynamically generated according to the need for barrier-crossing to effectively steer the sample paths towards the rare event destination. Unlike the dwSSA and sdwSSA methods, in which biases are fixed constants after parameter estimation, the biases in ABSIS for each reaction is dynamically determined based on exact calculation of the total probability of short κ-step forward- and backward-moving reaction paths, without the need of binning reaction rates. Reactions with higher probability of forward-moving are then encouraged, and reactions with higher probability of backward-moving are discouraged. Regardless of the number of reactions in the networks, the ABSIS only need to assign two bias parameters for the whole network: the degree to encourage forward-moving reactions and the degree to discourage backward-moving reactions, which both can be estimated through an efficient parameter estimation algorithm.

3. DIRECT SOLUTIONS TO THE dCME

The discrete chemical master equation provides a fundamental framework for studying the full stochasticity of biological networks (Gillespie, 1977). Under the definition of biological networks, the combination of copy numbers of molecular species defines the microstate of the molecular interactions in the network. By treating microscopic states of molecular species explicitly, linear and nonlinear reactions (such as synthesis, degradation, dimeric binding, and multimerization) can all be effectively modeled as transitions between microstates, with transition rates determined by the physical properties of the molecules and the cell environment. The probability distribution or potential landscape (Ao et al., 2007; Cao & Liang, 2008; Cao et al., 2010) over these microstates and its time-evolving and steady state behavior provide a full description of the properties of a stochastic molecular network.

3.1 Finite State Projection (FSP) Method for Direct Solution of Time Evolution

As biological networks often involve synthesis reactions, such as protein synthesis, the size of the state space can be potentially infinite. The finite state projection (FSP) method is based on a truncated projection of the state space and uses numerical techniques to compute direct solution to the dCME [29, 33]. Munsky and Khammash made two insightful observations. Denote two sets of indices of the microstates being chosen as J_1 and J_2, and assume $J_1 \subseteq J_2$. The truncated state spaces obtained by selecting states from J_1 and J_2 are $\boldsymbol{S}_{J_1}$ and $\boldsymbol{S}_{J_2}$, and the corresponding rate matrices are $\boldsymbol{A}_{J_1}$ and $\boldsymbol{A}_{J_2}$, respectively. The first observation is:

$$\left[e^{\boldsymbol{A}_{J2}}\right]_{J_1} \geq e^{\boldsymbol{A}_{J1}} \geq 0. \tag{36}$$

This assures that the time evolution probability distribution solved on the larger state space $\boldsymbol{S}_{J_2}$ is not smaller than that on $\boldsymbol{S}_{J_1}$:

$$\left[e^{\boldsymbol{A}_{J2}t}\right]_{J_1} \boldsymbol{p}\left(\boldsymbol{S}_{J_1},0\right) \geq e^{\boldsymbol{A}_{J1}t} \boldsymbol{p}\left(\boldsymbol{S}_{J_1},0\right). \tag{37}$$

This inequality implies that by increasing the size of the selected subset of states, the approximation improves monotonically. The second observation is, if one obtains a truncated state space by selecting states contained in the index set J, and if $1^T e^{t\boldsymbol{A}_J} p\left(\boldsymbol{S}_J,0\right) \geq 1-\epsilon$ for $\epsilon > 0$, then:

$$e^{tA_J}\boldsymbol{p}\left(\boldsymbol{S}_J,0\right)\leq \boldsymbol{p}\left(\boldsymbol{S}_J,t\right)\leq e^{tA_J}\boldsymbol{p}\left(\boldsymbol{S}_J,0\right)+\epsilon\mathbf{1}. \tag{38}$$

That is, starting with the initial probability vector $\boldsymbol{p}\left(\boldsymbol{S}_J,0\right)$ on the truncated state space, we can compute the probability vector in the truncated space as $e^{tA_J}\boldsymbol{p}\left(\boldsymbol{S}_J,0\right)$ at time t using the truncated rate matrix $\boldsymbol{A}_J$. If the inner-product of this vector with 1 is no less than $1-\epsilon$, the difference of this vector from the projected true vector $\boldsymbol{p}\left(\boldsymbol{S}_J,t\right)$ of the true probability $\boldsymbol{p}\left(\boldsymbol{S},t\right)$ is also no more than $\epsilon\mathbf{1}$. This inequality guarantees that the approximation obtained with truncated state space will never exceed the actual solution, and its error is bounded by ϵ (Munsky & Khammash, 2006).

These key observations led to the Finite State Projection algorithm, which iteratively adds more states to an initial truncated state space, until the approximation error is within a prescribed bound (Munsky & Khammash, 2006). The original Finite State Projection method was further extended (Munsky & Khammash, 2007), and it was recommended that the initial non-sparse probability vector $\boldsymbol{p}\left(\boldsymbol{S},0\right)$ should be determined by running a few steps of stochastic simulation discussed in a previous section.

Although the error due to state space truncation can be estimated (Munsky & Khammash, 2006), there are no systematic guidance and general strategy as to which states and how many of them should be incorporated to a finite projection to most effectively improve the approximation accuracy (Munsky & Khammash, 2006). Furthermore, the introduction of an absorption state where all truncated states are projected to will lead to accumulation of errors as time proceeds. As the absorbing state would eventually absorb all probability mass, the FSP method therefore is not appropriate for computing the steady state probabilistic landscape, as the approximation of the absorbing state will lead to errors that increase to 1 in the steady state.

3.2 Challenges to Direct Solutions: Enormous State Space

A complete identification and characterization of the space of the microstates is prerequisite to directly solve the dCME and obtain the full probability landscapes of a network. However, it is challenging to enumerate an optimal state space of a network.

A straight-forward method is to enumerate all microstates in the whole hypercube space limited by the maximum copy numbers of reactants. The size of the state space enumerated using such a hyper-cube method is bounded by the product of the maximum copy numbers, which can be quickly inflated to enormity. For example, if there are 10 species, and each with a maximum of 30 molecules in the network, without taking into consideration of the details of the network, the hyper-cube approach will generate a state space with $30^{10} = 5.9 \times 10^{14}$ states, which would easily exceed the memory capacity of most computers. However, the state space can be significantly reduced if detailed reaction schemes are considered during state space enumeration. There are many states that the system can never reach from a certain initial state. There are many states that no reaction can happen, therefore not needed. Alternatively, one can follow explicitly simulated reaction events by carrying out stochastic simulations. However, this approach cannot guarantee that all reachable states will be explored, therefore cannot guarantee full characterization of rare events. Moreover, it can be highly inefficient in enumerating microstate, because most of computing time will be spent on revisiting those high-probability states.

To address this challenging issue, Cao and Liang have developed a novel method, namely the *finite buffer* method (Cao & Liang, 2008), to significantly reduce the size of the state space of arbitrary biological network. The state space enumeration algorithm has been shown to be optimal in terms of time and space complexity. The state space enumerated using the *finite buffer* method has been dramatically reduced comparing with the conventional hyper-cube approach (*e.g.* a reduction of factor 10^9 can be achieved in the MAPK network). Furthermore, the finite buffer method uses a reflective boundary, rather than the absorbing boundary state in the FSP method, therefore both time-evolving and steady state probability landscapes can be solved using the state space generated with the *finite buffer* method. Furthermore, the reflective boundary adopted in the finite buffer method allows the quantification for the truncation error in the steady state probability landscape (to be published).

3.3 State Space Enumeration and Transition Rate Matrix Construction

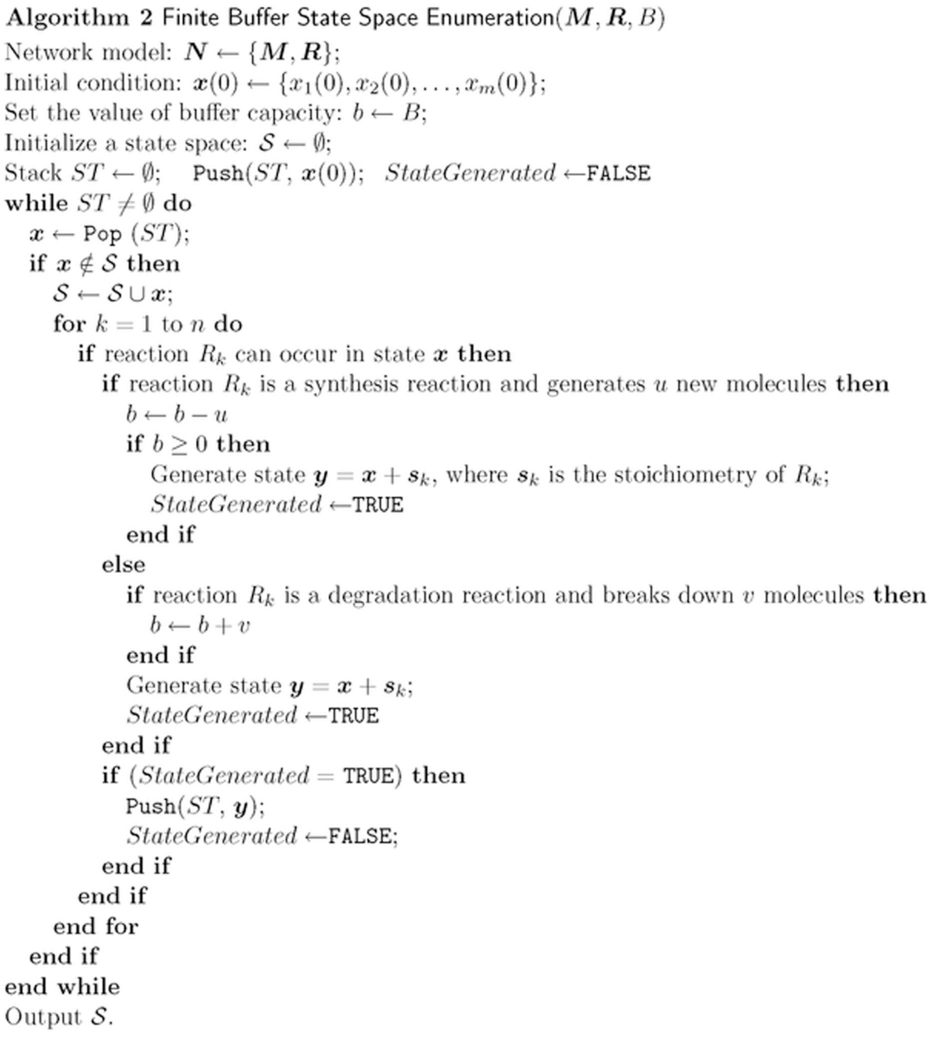
The technique of optimally enumerating microstates for a given initial condition now allows certain realistic systems to be studied using dCME, under the condition of finite buffer (Cao & Liang, 2008). Below we describe how microstates can be optimally enumerated. For a network with n molecular species and m reactions, we generate all microstates that the network can reach starting from a given initial condition, under the *finite buffer* constraint. We use a buffer queue of finite capacity to represent a reservoir of molecular tokens. Each time when a synthesis reaction occurs, a buffer token is spent to generate the new molecule, and each time when a degradation reaction happens, a buffer token is released and deposited back to the buffer queue. Synthesis reaction is allowed to occur only if the buffer queue is not exhausted. This is necessary due to the limitation of computing resources. As the microstate of a specific combination of copy numbers is $\boldsymbol{x} = (x_1, \cdots, x_m)$, we add an additional virtual buffer species B and its copy number b to denote the number of currently available tokens in the buffer queue, namely, the number of net new molecules that can still be synthesized at current microstate. Therefore, the microstate vector of the network can be augmented as $\boldsymbol{x} = (x_1, \cdots, x_m, b)$ by adding the copy number of buffer token B. A synthesis reaction can only occur when $b > 0$ in the *finite buffer* state enumeration algorithm.

Under these conditions, the set of all possible microstates that can be reached from an initial condition constitute the state space of the system. The set of allowed transitions is $\boldsymbol{T} = (T_{i,j})$, in which $t_{i,j}$ maps the microstate $\boldsymbol{x}_j$ to the microstate $\boldsymbol{x}_i$ by one reaction. The initial state of the reaction system is given as: $\boldsymbol{x}(0) = (x_1(0), x_2(0), \cdots, x_m(0), b(0))$, where $x_i(0)$ is the initial copy number of the i-th molecular species at time $t = 0$, and $b(0)$ is the predefined buffer capacity.

The *finite buffer* algorithm for enumerating the state space is summarized in the Algorithm in Figure 2. The state space and the reactions constitute a huge network with microstates as nodes and reactions as edges. The state space enumeration can then be projected to a network traversal problem. After initialization, the *finite buffer* algorithm starts with the given initial microstate $\boldsymbol{x}(0)$ to generate all possible microstates with no consideration of reaction rates. Each reaction is examined in turn to determine if this reaction can occur from the current microstate. If so, and if the buffer is not used up, the state that this reaction leads to is generated. If the newly generated state has never been visited before, we declare it as a new state and add it to our collection of states for the state space. We repeat this process for all new states, with the aid of a stack data structure (Cormen, Leiserson, & Rivest, 1990). This process terminates when all new states are exhausted (Cao & Liang, 2008). Note that a queue data structure can also be used in place of the stack. Both data structure generate the exact same state space, but with different sequences, as using the stack corresponds to a depth-first search (DFS) of the network, while the queue corresponds to a breadth-first search (BFS) (Cormen et al., 1990).

Under the finite buffer constraint, the time complexity of this algorithm is optimal. Since only unseen state will be pushed onto the stack, every state is pushed and popped at most once, and each state will be generated/visited at most twice before it is popped from the stack. As access to each state and to push/pop operations take $O(1)$ time, the total time required for the stack operations is $O(|\Omega|)$, where Ω is the state space. As the *finite buffer* algorithm examines all reactions for

Figure 2. The finite buffer state space enumeration algorithm



```
**Algorithm 2** Finite Buffer State Space Enumeration($\boldsymbol{M}, \boldsymbol{R}, B$)
Network model: $\boldsymbol{N} \leftarrow \{\boldsymbol{M}, \boldsymbol{R}\}$;
Initial condition: $\boldsymbol{x}(0) \leftarrow \{x_1(0), x_2(0), \ldots, x_m(0)\}$;
Set the value of buffer capacity: $b \leftarrow B$;
Initialize a state space: $\mathcal{S} \leftarrow \emptyset$;
Stack $ST \leftarrow \emptyset$;   Push($ST$, $\boldsymbol{x}(0)$);   $StateGenerated$ ←FALSE
while $ST \neq \emptyset$ do
  $\boldsymbol{x} \leftarrow$ Pop ($ST$);
  if $\boldsymbol{x} \notin \mathcal{S}$ then
    $\mathcal{S} \leftarrow \mathcal{S} \cup \boldsymbol{x}$;
    for $k = 1$ to $n$ do
      if reaction $R_k$ can occur in state $\boldsymbol{x}$ then
        if reaction $R_k$ is a synthesis reaction and generates $u$ new molecules then
          $b \leftarrow b - u$
          if $b \geq 0$ then
            Generate state $\boldsymbol{y} = \boldsymbol{x} + \boldsymbol{s}_k$, where $\boldsymbol{s}_k$ is the stoichiometry of $R_k$;
            $StateGenerated$ ←TRUE
          end if
        else
          if reaction $R_k$ is a degradation reaction and breaks down $v$ molecules then
            $b \leftarrow b + v$
          end if
          Generate state $\boldsymbol{y} = \boldsymbol{x} + \boldsymbol{s}_k$;
          $StateGenerated$ ←TRUE
        end if
        if ($StateGenerated$ = TRUE) then
          Push($ST$, $\boldsymbol{y}$);
          $StateGenerated$ ←FALSE;
        end if
      end if
    end for
  end if
end while
Output $\mathcal{S}$.
```

each reached state, the complexity of total time required is $O\left(m\left|\Omega\right|\right)$, where m is usually a modest constant (e.g. < 50). Based on the same argument, it is also easy to see that the algorithm is optimal in storage, as only valid states and valid transitions are recorded. Using this algorithm, all states reachable from an initial condition within the finite buffer constraint will be accounted for, and no irrelevant states will be included. Furthermore, all possible transitions will be recorded, and no infeasible transitions will be attempted (Cao & Liang, 2008).

In the *finite buffer* algorithm, the reaction rates $\{A_{i,j}\}$ between any two microstates $\boldsymbol{x}_j$ and $\boldsymbol{x}_i$ in the transition rate matrix $\boldsymbol{A}$ for the dCME are computed following the same approach outlined in references (Gillespie, 1977; Munsky & Khammash, 2006; Schultz et al., 2007). We give further details in later sections on how this is done for two realistic networks. The transition rate matrix $\boldsymbol{A}$ is highly sparse. We will also discuss numerical methods for finding the solutions to the dCME.

With this optimal method for dramatically reducing the state space and enumerating the microstates of a finite system, numerical methods for solving large linear systems can be applied to efficiently solve the dCME. Realistic biological systems can now be directly studied, such as the decision network of phage lambda (Cao & Liang, 2008; Cao et al., 2010).

3.3.1 Open and Close Network, Buffer Capacity

The finite buffer method can be applied to efficiently enumerate state spaces for both open and closed networks. In closed networks, buffer queues are not necessary for state space enumeration, only the initial state is required. In open networks, the capacity of the buffer queue introduced in the finite buffer method to limit the synthesis reactions actually put a restriction on the total mass of the network. That is, the state space enumerated using the finite buffer method consists of microstates that can be reached from the initial state and with total mass smaller than the limit of buffer capacity. Note that, by introducing a buffer queue to control the synthesis reaction, an open network is effectively transformed into a closed network, if considering the buffer queue as a virtual molecule.

3.3.2 State Space Enumeration as a Graph Traverse Problem

The state space and the transitions under a given initial condition can be considered as a directed graph $\boldsymbol{G} = (\boldsymbol{S}, \boldsymbol{T})$, in which microstates are vertices, and allowed transitions between any two states are edges, *i.e.*, reactions connecting two states. Two vertices $\boldsymbol{x}_j \in \boldsymbol{S}$ and $\boldsymbol{x}_i \in \boldsymbol{S}$ are connected by a directed edge $t_{i,j} \in \boldsymbol{T}$ if and only if $\boldsymbol{x}_j$ can be transformed to $\boldsymbol{x}_i$ through a reaction. As only states reachable from the specified initial state through one or more steps of reactions are concerned, the directed graph G is a connected graph.

It can be shown that the finite buffer algorithm must generate a finite set of states when starting from an initial state with a specified buffer capacity. Our algorithm implicitly generates the graph $\boldsymbol{G}$. Due to the finite set of reactions R in the network,

$\boldsymbol{G}$ will have a finite number of total child branches at any finite steps away from the initial state. Suppose the algorithm will not terminate in finite steps. Since each state is only visited no more than twice in the finite buffer algorithm, $\boldsymbol{G}$ must have an unlimited depth. That is, there must exist an infinite path with all different microstate in the graph $\boldsymbol{G}$ that starts from the initial state and extends to infinity. This is impossible for any initial condition, as each molecular species has a limited initial copy number, and the buffer capacity limits the number of new molecules that can be synthesized in open systems. Therefore, such infinite paths do not exist. The finite buffer algorithm therefore must terminate in finite steps to generate a finite state space.

3.4 Steady State Probability Landscape

The steady state probability landscape over the state space can be computed by directly solving the linear equation $\boldsymbol{A\pi} = 0$, where $\boldsymbol{A}$ is the transition rate matrix constructed using the finite buffer method, and $\boldsymbol{\pi}$ is the steady state probability distribution over the enumerated state space. The steady state probability distribution $\boldsymbol{\pi}$ is a high-dimensional vector, it can be further projected to a lower dimensional space of biological interest. A number of iterative numerical techniques can be applied to solve the equation, such as the Gauss-Seidel method (Golub & Van Loan, 1996) and the bi-conjugate gradient stabilized method (BiCGSTAB) (Saad, 2003).

Alternatively, the transition rate matrix $\boldsymbol{A}$ can be transformed into a probability transition matrix $\boldsymbol{M}$ following a uniformization step: $\boldsymbol{M} = \boldsymbol{I} + \boldsymbol{A}\Delta t$, where $\boldsymbol{I}$ is the identify matrix, and $\Delta t \leq 1 / \max\left(-A_{ii}\right)$ is the discrete time unit (Kachalo, Lu, & Liang, 2006; Stewart, 1994). Therefore, the steady state probability distribution can be equivalently obtained by solving the equation $\boldsymbol{\pi} = \boldsymbol{M\pi}$, which corresponds to the eigenvector of $\boldsymbol{M}$ with respect to the eigenvalue . The Arnoldi method implemented in the software ARPACK can be used to solve the eigenvalue-eigenvector problem and compute the steady state distribution $\boldsymbol{\pi}$ (Lehoucq, Sorensen, & Yang, 1998).

3.5 Time Evolving Probability Landscape

The time evolving probability landscape derived from the dCME can be expressed in the form of a matrix exponential: $\boldsymbol{p}(t) = e^{\boldsymbol{A}t}\boldsymbol{p}(0)$, where $\boldsymbol{p}(0)$ is the initial probability landscape and $\boldsymbol{A}$ is the transition rate matrix over the enumerated state space. Once $\boldsymbol{p}(0)$ is specified, $\boldsymbol{p}(t)$ can be calculated using numerical methods such as the Krylov subspace projection method, *e.g.*, as implemented in the EXPOKIT package by Sidje (Sidje, 1998). Other state-of-the-art numerical techniques can also be applied (Kazeev, Khammash, Nip, & Schwab, 2014).

3.6 Software Availability and Online Simulation Tools

A C++ language implementation for the finite buffer method software package has been made publicly available. One can download the source codes and executable binary programs from the URL http://tanto.bioe.uic.edu/dcme/. The software package includes four separate programs to perform the functions of (1) finite buffer state space enumeration, (2) transition matrix construction, (3) steady state probability landscape solution using the Successive Over Relaxation method, and (4) time evolution probability landscape solution using the EXPOKIT method (Sidje, 1998), respectively. The website also includes simple examples to demonstrate the formats of input files.

For easier use of the tool, an online GUI simulation tool has been built recently for the finite buffer method at the nanoHUB computing grid. The online tool is also publicly available. One can open the URL https://nanohub.org/tools/fbsdcme in any Java supported browser, and click the "Lunch Tool" button to start the finite buffer software. To use the finite buffer software, one needs to input a SBML format reaction network file and an initial condition file. The format for the initial condition file has been demonstrated in the finite buffer method website (http://tanto.bioe.uic.edu/dcme/). In the SBML file of reaction network, the buffer species must be explicitly included in the species list, and also incorporated into the reactant list of all synthesis reactions and the product list of all degradation reactions.

4. EXAMPLES OF REALISTIC GENE REGULATORY NETWORKS

We now use two biological examples, namely the genetic toggle switch and the phage lambda lysogeny and lysis circuit, to show how stochastic gene regulatory networks can be modeled by directly solving the underlying discrete chemical master equation. We show how state spaces are enumerated and transition rate matrices are constructed using the finite buffer algorithm. We also show steady state and time evolving probability landscapes computed using the fb-dCME method, and we discuss interesting biological relevance. We show that the finite buffer dCME method can be used to answer important biological questions in stochastic gene regulatory networks.

4.1 Genetic Toggle Switch

Toggle switch is one of the smallest genetic networks that can present bistability (Blake, KAErn, Cantor, & Collins, 2003; Schultz et al., 2007). It is a small network consisting of two genes, say, A and B. Single copies of gene A and gene B in the chromosome each encoding a protein product. The protein product of each gene represses the other gene: When two protein monomers associate, they bind to the appropriate operator site and repress the transcription of the other gene.

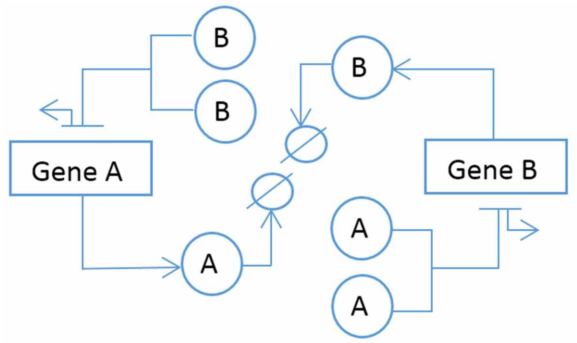
The molecular species and the network topology of a toggle switch model are shown in Figure 3. The reactions include: the synthesis and degradation of proteins A and B, with reaction constants denoted as s and d, respectively; the binding and unbinding of the operator site of one gene by the protein products of the other gene at rate b and u, respectively (Eqn. (39)). Here the reactions rates are chosen as: $d = 1s^{-1}$, $s = 100s^{-1}$, $u = 0.1s^{-1}$, $b = 0.00001s^{-1}$. The binding states of the two operator sites are "on-on/unbound-unbound", "on-off/unbound-bound", "off-on/bound-unbound", and "off-off/bound-bound". The synthesis rates of both proteins A and B depend on the binding state of the operator sites (Cao & Liang, 2008; Schultz et al., 2007).

$$GeneA \xrightarrow{s} GeneA + ProteinA,$$

$$GeneB \xrightarrow{s} GeneB + ProteinB,$$

$$ProteinA \xrightarrow{d} \varnothing, \; ProteinB \xrightarrow{d} \varnothing,$$

Figure 3. Genetic toggle switch



$$2ProteinA + GeneB \overset{b}{\rightarrow} BoundGeneB,$$

$$2ProteinB + GeneA \overset{b}{\rightarrow} BoundGeneA,$$

$$BoundGeneB \overset{u}{\rightarrow} 2ProteinA + GeneB,$$

$$BoundGeneA \overset{u}{\rightarrow} 2ProteinB + GeneA. \tag{39}$$

4.1.1 State Space Enumeration

We first calculate the state spaces under the initial condition of 1 copy of unbound gene A, 1 copy of unbound gene B, 0 copies of bound gene A and bound gene B, and 1 copies of their protein products. We set the buffer size to different copies of total protein A and protein B combined that can be synthesized. When the buffer capacity is 20, the size of the state space is 764. At buffer capacity of 200, 400, and 800 copies of proteins, the size increases to 79,604 states, 319,204 states, and 1,278,404 states, respectively. A comparison of state space size between the *finite buffer* method and the conventional hypercube method is shown in Table 1. The finite buffer method has shown dramatic reductions in state space sizes. Moreover, the factor of reduction is increasing with the buffer capacity. In the following computing results, we use the buffer capacity 300, and the finite buffer method generates

Table 1. State space sizes of toggle switch network using different methods

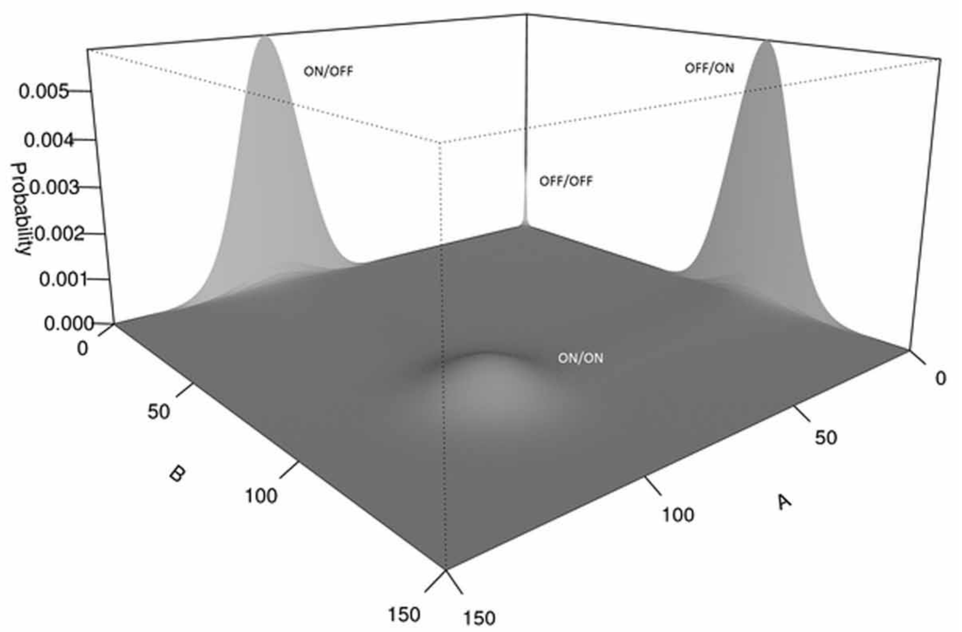
Buffer Capacity	Finite Buffer Method	Hypercube Method	Reduction Factor
50	4,904	1.5625×10^{10}	3.1862×10^{6}
100	19,804	1.0×10^{12}	5.0495×10^{7}
200	79,604	6.4×10^{13}	8.0398×10^{8}
300	179,404	7.29×10^{14}	4.0635×10^{9}
400	319,204	4.096×10^{15}	1.2832×10^{10}

a state space with 179,404 states. Therefore, the transition rate matrix $\boldsymbol{A}$ is a 179,404 dimensional sparse matrix.

4.1.2 Steady State Probability Landscape with Four Distinct Peaks

The exact probably landscape of the toggle switch model at steady state can be computed numerically. We choose the initial condition as: 1 copy of unbound gene A, 1 copy of unbound gene B, 0 copies of bound gene A and bound gene B, 0 copies of their protein products, and the buffer size for the total protein A and protein B combined to be $b = 300$, meaning the total copy number of proteins A and B combined in the system cannot exceed 300 at any time. We can increase or decrease the buffer capacity to include or exclude states. We then enumerate the state space of the toggle switch using the finite buffer algorithm. The steady state probability landscape of the network can then be computed (Figure 4). The Figure 4 shows the steady state probability landscape projected on the 2D space of protein A and B copy numbers. With these chosen parameters, the toggle switch exhibits a clear multi-stability in the steady state probability landscape. The toggle switch shows four distinct macroscopic states, corresponding to the “ON/ON”, “ON/OFF”, “OFF/

Figure 4. Steady state probability landscape of toggle switch

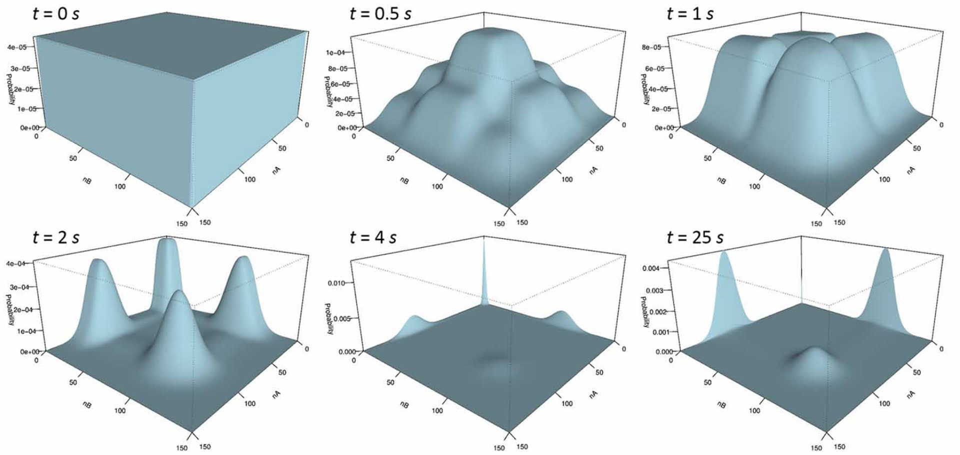


ON" and "OFF/OFF" configurations of the genetic circuits. Among them, the on/off and off/on states are more prominent, and the on/on state has a low probability, and the off/off state is severely unstable showing a very narrow and spiky peak (Cao & Liang, 2008). This result has important biological implications in cancer biology (Shiraishi, Matsuyama, & Kitano, 2010) and stem cell biology (Bu et al., 2013).

4.1.3 Time Evolving Probability Landscapes

Using the same state space and transition rate matrix, we can also directly compute the exact time-evolving probably landscape of the toggle switch at different time points. Starting from the uniform distribution, the time-evolving probability landscape is computed using the EXPOKIT package with 0.01 second time interval. Figure 5 shows the snapshots of the time-evolving probability landscape at 6 different time points: $t = 0$, 0.5, 1, 2, 4, and 25 seconds, respectively. The time-evolving probability landscapes have shown complex behavior of probability mass flows on the projected A-B 2D plane. At the very beginning, the probability of all states are equal ($t = 0$), and then a multi-layered peak arises in the center of the plane ($t = 0.5$), which is quickly separated into four thick peaks ($t = 1$). Gradually, the four thick peaks form the four different macroscopic states ($t = 2$, 4, and 25), and eventually converge to the steady state probability landscape ($t = 25$). Interestingly, the relative heights of the four peaks are changing during the time evolution. For example, four peaks have nearly same probabilities at about $t = 2$,

Figure 5. Time evolving probability landscapes of toggle switch starting from uniform distribution



but the probability of the off/off state becomes a few folds higher than the other three peaks at $t = 4s$, and eventually the probabilities of the on/off and off/on states take most of the probability mass. The dynamic evolution of the landscape over time may be important in explaining biological behaviors of genetic circuits.

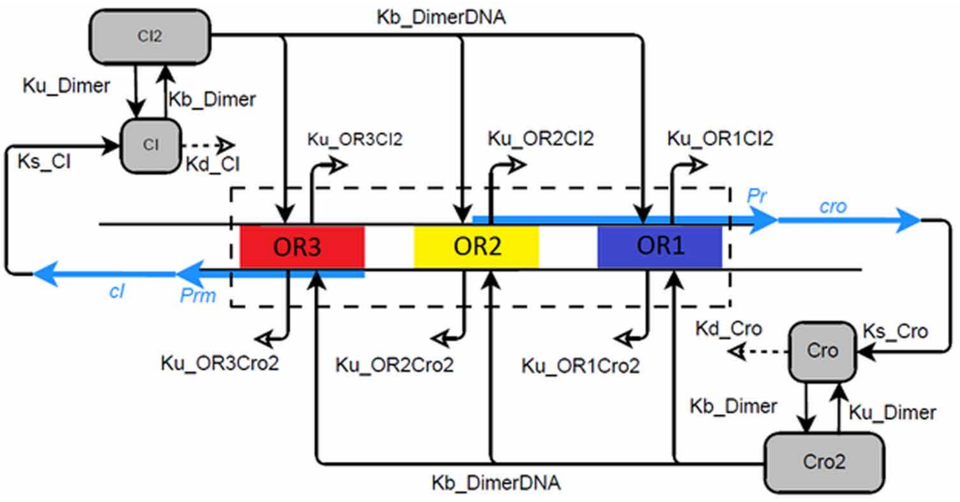
4.2 Phage Lambda Lysogeny-Lysis Epigenetic Switch

Bacteriophage lambda is a virus that infects *E. coli* cells. Of central importance is the molecular circuitry that controls phage lambda to choose between two productive modes of development, namely, the lysogenic phase and the lytic phase. In the lysogenic phase, phage lambda represses its developmental function, integrates its DNA into the chromosome of the host *E. coli* bacterium, and is replicated in cell cycles for potentially many generations. When threatening DNA damage occurs, for example, when UV irradiation increases, phage lambda switches from the epigenetic state of lysogeny to the lytic phase and undergoes massive replications in a single cell cycle, releases $50 \sim 100$ progeny phages upon lysis of the *E. coli* cell. This switching process is called prophage induction (Ptashne, 2004).

The molecular network that controls the choice between these two different physiological states has been studied extensively (Anderson & Yang, 2008; Arkin et al., 1998; Aurell, Brown, Johanson, & Sneppen, 2002; Jacob & Monod, 1961; Johnson et al., 1981; Little, Shepley, & Wert, 1999; Ptashne, 2004; Ptashne et al., 1976; Shea & Ackers, 1985). All of the major molecular components of the network have been identified, binding constants and reaction rates characterized, and there is a good experimental understanding of the general mechanism of the molecular switch (Ptashne, 2004). Theoretical studies have also contributed to the illumination of the central role of stochasticity (Arkin et al., 1998) and the stability of lysogen against spontaneous switching (Aurell et al., 2002; Zhu, Yin, Hood, & Ao, 2004).

To study how lysogeny is maintained and how it transitions to the lytic state, one can use a simplified stochastic model for the molecular regulatory network that controls the epigenetic switch in phage lambda (Figure 6) (Cao et al., 2010). Using a total of 54 biochemical reactions involving 13 molecular species, this model explicitly includes key components, essential reactions, and cooperativities of the phage lambda decision circuitry. The effects of UV irradiation can be modeled by increasing the CI degradation rates k_d due to the response of the SOS system. This epigenetic network model can reach around 1.7 million microstates. The steady state probability associated with each of these microstates can be computed from dCME after the microstates are enumerated using the finite buffer algorithm (Cao et al., 2010).

Figure 6. Phage lambda lysogeny-lysis genetic circuit

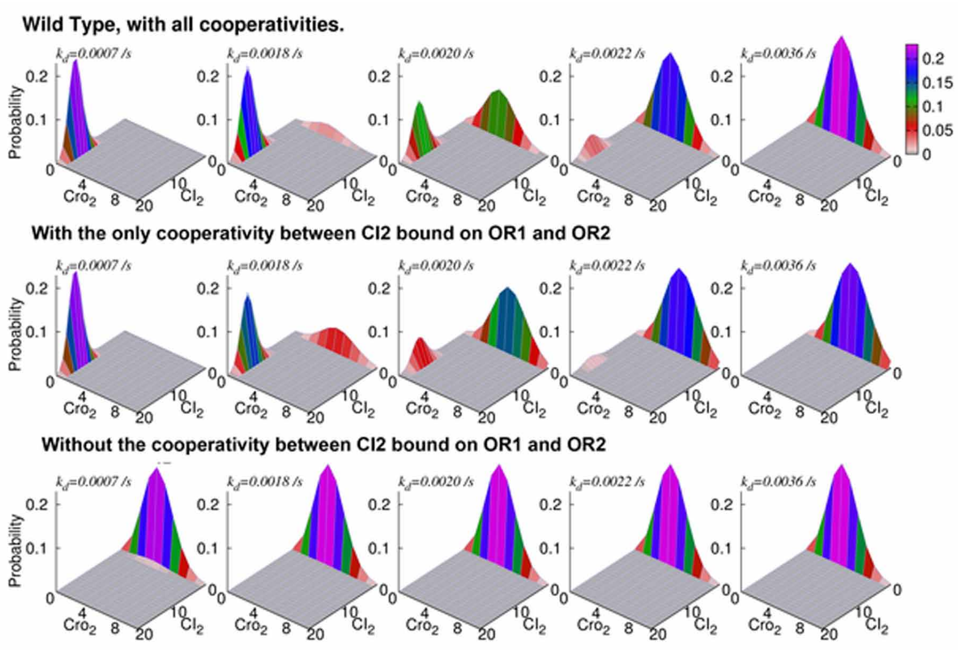


4.2.1 Steady State Probability Landscapes Can Help to Reveal Control Mechanisms

Figure 7 (row 1) shows the probability landscape of the phage lambda at five different UV irradiation conditions, each modeled with a different CI degradation rate k_d. Although there are 13 molecular species, the landscapes can be projected to the 2-dimensional subspace and record the total copy numbers of CI2 dimer and Cro2 dimer molecules. With a high copy number of CI2 repressor, the lysogenic phase of the phage lambda is maintained, whereas a high copy number of Cro2 protein signifies the lytic phase (Johnson et al., 1981). A clear picture of the landscape in lysogeny, at the start of transition, during mid-transition, at the end of transition, and in lysis can be seen (Figure 7).

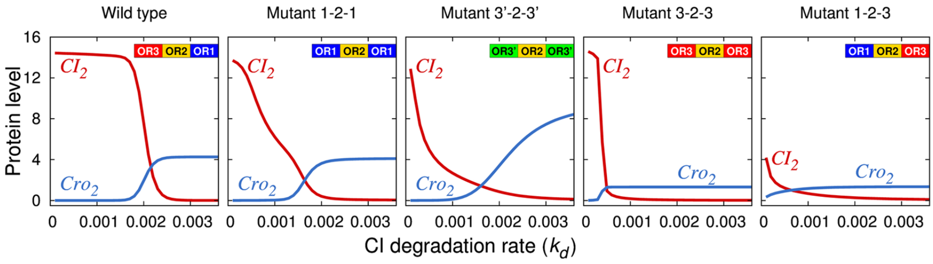
The stochastic network models can also be used to aid in understanding of the mechanism of how the decision network works. It is well known that cooperativity among proteins play important roles. After removing all cooperativities between neighboring proteins in the model, phage lambda cannot enter lysogeny regardless the dosage of the UV irradiation. However, when the cooperativity ΔG_{12} between two CI dimers when binding to operator sites OR1 and OR2 are restored, the lysogeny is largely restored (Figure 7, row 2). In contrast, if all other cooperativities are restored except ΔG_{12}, phage lambda still lacks the ability to enter the lysogeny phase (Figure 7, row 3). These calculations suggest that the cooperativity ΔG_{12} plays key roles in maintaining the stability of the circuit.

Figure 7. The probability landscape of the epigenetic circuits of lysogeny maintenance in phage lambda



An important property of biological stochastic network is its robustness against changes in the molecular components of the epigenetic network. Experimental studies showed that when the ordering of operator sites are changed, mutants of phage lambda have markedly different tolerance to UV irradiation. Calculations from solving the dCME model showed that the wild-type lysogeny has a high threshold towards lysis, and is overall insensitive to small fluctuation of UV dosage, if it is below certain threshold (Figure 8a). That is, the switching network of phage lambda is very stable and is strongly buffered with a high threshold against fluctuations in CI degradation rate due to environmental changes in UV irradiation. This high threshold against environmental fluctuations is important for the self-perpetuating nature of the epigenetic state of *E. coli* cells, allowing lysogeny to be passed on to its offspring. Once the degradation rate of CI reaches a threshold, phage lambda switches very efficiently to the lytic phase, and this efficiency is not built at the expense of stability against random fluctuation. Wild type phage lambda therefore can integrate signaling in the form of different CI degradation rates and can distinguish a true signal above the high threshold from random noise fluctuating below this threshold.

Figure 8. Instability, shallow threshold, and switching inefficiency of the network against fluctuation in UV irradiation in mutant phage lambda



In contrast, all mutant variants exhibit the behavior of a hair trigger, and require much less UV irradiation for the onset of lysis induction (Figure 8b-e). In addition, they are "leaky", and respond in a graded fashion towards increase UV irradiation, instead of the well-behaved threshold behavior observed in wild type phage lambda. In the case of mutant 1-2-3, the mutant phage lambda cannot enter lysogenic state. These results are in full agreement with experimental findings (Cao et al., 2010; Little et al., 1999).

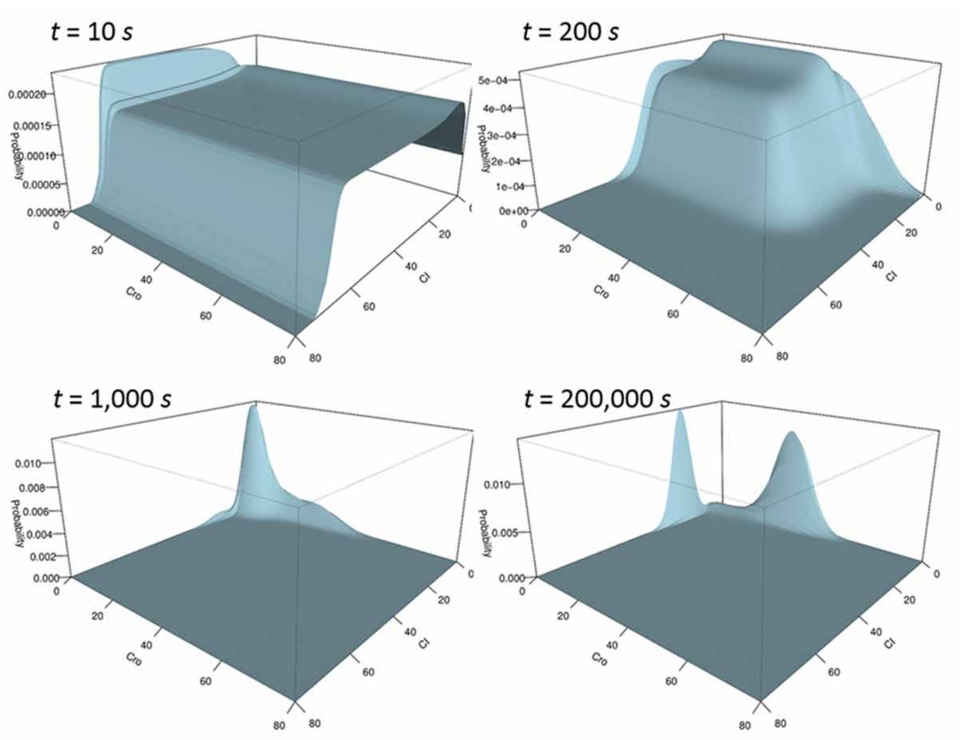
4.2.2 Time Evolving Probability Landscapes

The time-evolving probability landscape is computed using the EXPOKIT package starting from the uniform distribution. Figure 9 shows the snapshots of the time-evolving probability landscape at 4 different time points at: $t = 10$, 200, 1000, and $200,000$ second, respectively. The CI degradation rate is fixed as $k_d = 0.0025 s^{-1}$. The time-evolving probability landscapes have shown the behavior of probability mass flows on the projected CI-Cro plane. The probability mass first flows towards the state CI=0, Cro=0, and forms a peak there at about $t = 1000s$. However, two stable peaks are eventually formed separately, and the landscape converges to the steady state probability landscape ($t > 200,000s$).

5. DISCUSSIONS

In this chapter, we have discussed the significance of the discrete chemical master equation (dCME) as a theoretical framework for modeling nonlinear, biochemical reaction networks (Gillespie, 1977). This formulation provides a foundation to study stochastic phenomena in biological networks. Its role is analogous to that of the Schrödinger equation in quantum mechanics (Beard & Qian, 2008). Develop-

Figure 9. Time evolving probability landscapes of phage lambda genetic circuit at different time points



ing computational solutions to the dCME has important implications, just as the development of computational techniques for solving the Schrödinger equation for systems with many atoms is (Kohanoff, 2006; Kohn, Sham, & others, 1965). By computing the time-evolving probability landscape of cellular stochastic networks, we may gain understanding of the possible mechanisms of cellular states, as well as the inheritable phenotypes with a distributive epigenetic code, in which the network architecture and its landscape dictate the physiological meta-states of the cell under different conditions (Ptashne, 2007; Zhu et al., 2004).

Solving a given dCME, however, is a computationally challenging task at the present time. We have outlined several key difficulties, as well as some of the progresses that have been made so far. The finite buffer algorithm allows direct numerical solution to the dCME for both steady state and time-evolving probability landscapes. It can be applied to study stochasticity of systems with a handful of particles, as well as larger networks arising from very realistic biological problem, such as that of the lysogeny-lysis control circuit of the phage lambda (Cao et al., 2010). As an exact method, the finite buffer method can also be used to study model systems of finite size to gain insight into stochastic behavior of networks. The stochastic simulation algorithm offers the approach of studying the stochastic network through simulations (Gillespie, 1977). The formulation of stochastic differential equation such as the Langevin equation allows exploration of more complex stochastic systems, at the expense of less rigorous assumptions and perhaps more errors.

An important task is to integrate different stochastic methods for efficient computational solution of complex stochastic networks at large scale, with controlled accuracy. For example, one may apply the finite buffer algorithm to solve dCME directly for certain critical parts of the network, where rare events need to be assessed very accurately. Other parts of the network where general stochastic behavior needs to be determined can be studied using the Langevin equation. In addition, one may also wish to apply the stochastic simulation algorithm to certain parts of the network to probe their behavior. Furthermore, one may wish to apply ordinary differential equation (ODE) models to study parts of the system where copy numbers of molecules are large and there are little stochastic effects.

A great challenge is to develop a general strategy so the best methods can be applied to specific parts of the network and the results integrated to provide an overall picture of the stochastic dynamic behavior of the network. It would be desirable that the resulting errors due to approximations of varying degree are bounded within a tolerance level, while maintaining necessary computing speed and resource requirement. A further challenge is to develop such hybrid methods to compute the overall spatio-temporal stochastic dynamic properties of different cellular compartments, multi-cellular tissue, with consideration of different spatial distribution or gradient of molecules such as oxygen, nutrient, morphogens, and other signaling factors, all with stochasticity appropriately considered.

REFERENCES

Anderson, L. M., & Yang, H. (2008). DNA looping can enhance lysogenic CI transcription in phage lambda. *Proceedings of the National Academy of Sciences of the United States of America*, *105*(15), 5827–5832. doi:10.1073/pnas.0705570105 PMID:18391225

Ao, P., Kown, C., & Qian, H. (2007). On the existence of potential landscape in the evolution of complex systems. *Complexity*, *12*(4), 19–27. doi:10.1002/cplx.20171

Arkin, A., Ross, J., & McAdams, H. H. (1998). Stochastic kinetic analysis of developmental pathway bifurcation in phage lambda-infected Escherichia coli cells. *Genetics*, *149*(4), 1633–1648. PMID:9691025

Aurell, E., Brown, S., Johanson, J., & Sneppen, K. (2002). Stability puzzles in phage lambda. *Physical Review E: Statistical, Nonlinear, and Soft Matter Physics*, *65*(5), 051914. doi:10.1103/PhysRevE.65.051914 PMID:12059600

Beard, D. A., & Qian, H. (2008). *Chemical biophysics: quantitative analysis of cellular systems*. Cambridge University Press. doi:10.1017/CBO9780511803345

Blake, W. J., KÆrn, M., Cantor, C. R., & Collins, J. J. (2003). Noise in eukaryotic gene expression. *Nature*, *422*(6932), 633–637. doi:10.1038/nature01546 PMID:12687005

Bu, P., Chen, K. Y., Chen, J. H., Wang, L., Walters, J., Shin, Y. J., & Shen, X. et al. (2013). A microRNA miR-34a-regulated bimodal switch targets Notch in colon cancer stem cells. *Cell Stem Cell*, *12*(5), 602–615. doi:10.1016/j.stem.2013.03.002 PMID:23642368

Cao, Y., Gillespie, D. T., & Petzold, L. R. (2006). Efficient step size selection for the tau-leaping simulation method. *The Journal of Chemical Physics*, *124*(4), 044109. doi:10.1063/1.2159468 PMID:16460151

Cao, Y., & Liang, J. (2008). Optimal enumeration of state space of finitely buffered stochastic molecular networks and exact computation of steady state landscape probability. *BMC Systems Biology*, *2*(1), 30. doi:10.1186/1752-0509-2-30 PMID:18373871

Cao, Y., & Liang, J. (2013). Adaptively biased sequential importance sampling for rare events in reaction networks with comparison to exact solutions from finite buffer dCME method. *The Journal of Chemical Physics*, *139*(2), 025101. doi:10.1063/1.4811286 PMID:23862966

Cao, Y., Lu, H. M., & Liang, J. (2010). Probability landscape of heritable and robust epigenetic state of lysogeny in phage lambda. *Proceedings of the National Academy of Sciences of the United States of America*, *107*(43), 18445–18450. doi:10.1073/pnas.1001455107 PMID:20937911

Cormen, T. H., Leiserson, C. E., & Rivest, R. L. (1990). *Introduction to algorithms*. Cambridge, MA: The MIT Press.

Daigle, B. J. Jr, Roh, M. K., Gillespie, D. T., & Petzold, L. R. (2011). Automated estimation of rare event probabilities in biochemical systems. *The Journal of Chemical Physics*, *134*(4), 044110. doi:10.1063/1.3522769 PMID:21280690

Gillespie, D. T. (1977). Exact stochastic simulation of coupled chemical reactions. *Journal of Physical Chemistry*, *81*(25), 2340–2361. doi:10.1021/j100540a008

Gillespie, D. T. (2000). The chemical Langevin equation. *The Journal of Chemical Physics*, *113*(1), 297–306. doi:10.1063/1.481811

Gillespie, D. T. (2007). Stochastic simulation of chemical kinetics. *Annual Review of Physical Chemistry*, *58*(1), 35–55. doi:10.1146/annurev.physchem.58.032806.104637 PMID:17037977

Golub, G. H., & Van Loan, C. F. (1996). *Matrix computations* (3rd ed.). Baltimore, MD: Johns Hopkins University Press.

Grima, R., Thomas, P., & Straube, A. V. (2011). How accurate are the nonlinear chemical Fokker-Planck and chemical Langevin equations? *The Journal of Chemical Physics*, *135*(8), 084103. doi:10.1063/1.3625958 PMID:21895155

Hasty, J., Pradines, J., Dolnik, M., & Collins, J. J. (2000). Noise-based switches and amplifiers for gene expression. *Proceedings of the National Academy of Sciences of the United States of America*, *97*(5), 2075–2080. doi:10.1073/pnas.040411297 PMID:10681449

Jacob, F., & Monod, J. (1961). Genetic regulatory mechanisms in the synthesis of proteins+. *Journal of Molecular Biology*, *3*(3), 318–356. doi:10.1016/S0022-2836(61)80072-7 PMID:13718526

Johnson, A. D., Poteete, A. R., Lauer, G., Sauer, R. T., Ackers, G. K., & Ptashne, M. (1981). lambda Repressor and cro--components of an efficient molecular switch. *Nature*, *294*(5838), 217–223. doi:10.1038/294217a0 PMID:6457992

Kachalo, S., Lu, H., & Liang, J. (2006). Protein folding dynamics via quantification of kinematic energy landscape. *Physical Review Letters*, *96*(5), 058106. doi:10.1103/PhysRevLett.96.058106 PMID:16487000

Kazeev, V., Khammash, M., Nip, M., & Schwab, C. (2014). Direct solution of the Chemical Master Equation using quantized tensor trains. *PLoS Computational Biology*, *10*(3), e1003359. doi:10.1371/journal.pcbi.1003359 PMID:24626049

Keizer, J. (1977). On the macroscopic equivalence of descriptions of fluctuations for chemical reactions. *Journal of Mathematical Physics*, *18*(7), 1316–1321. doi:10.1063/1.523422

Kim, K. Y., & Wang, J. (2007). Potential energy landscape and robustness of a gene regulatory network: Toggle switch. *PLoS Computational Biology*, *3*(3), e60. doi:10.1371/journal.pcbi.0030060 PMID:17397255

Kohanoff, J. (2006). *Electronic structure calculations for solids and molecules: theory and computational methods*. Cambridge Univ Pr. doi:10.1017/CBO9780511755613

Kohn, W., & Sham, L. et al. (1965). Self-consistent equations including exchange and correlation effects. *Physical Review*, *140*(4A), A1133–A1138. doi:10.1103/PhysRev.140.A1133

Kurtz, T. G. (1972). The relationship between stochastic and deterministic models for chemical reactions. *The Journal of Chemical Physics*, *57*(7), 2976–2978. doi:10.1063/1.1678692

Kuwahara, H., & Mura, I. (2008). An efficient and exact stochastic simulation method to analyze rare events in biochemical systems. *The Journal of Chemical Physics*, *129*(16), 165101. doi:10.1063/1.2987701 PMID:19045316

Lehoucq, R., Sorensen, D., & Yang, C. (1998). *Arpack users' guide: Solution of large scale eigenvalue problems with implicitly restarted Arnoldi methods*. Philadelphia: SIAM. doi:10.1137/1.9780898719628

Levin, M. D. (2003). Noise in gene expression as the source of non-genetic individuality in the chemotactic response of Escherichia coli. *FEBS Letters*, *550*(1-3), 135–138. doi:10.1016/S0014-5793(03)00857-3 PMID:12935899

Liang, J., Zhang, J., & Chen, R. (2002). Statistical geometry of packing defects of lattice chain polymer from enumeration and sequential Monte Carlo method. *The Journal of Chemical Physics*, *117*(7), 3511–3521. doi:10.1063/1.1493772

Lin, M., Chen, R., & Liu, J. S. (2009). *Lookahead Strategies for Sequential Monte Carlo*. Retrieved from Little, J. W., Shepley, D. P., & Wert, D. W. (1999). Robustness of a gene regulatory circuit. *The EMBO Journal*, *18*(15), 4299–4307.

Liu, J. S., Chen, R., & Logvinenko, T. (2001). A theoretical framework for sequential importance sampling and resampling. *Sequential Monte Carlo Methods in Practice*, 225-246.

McAdams, H. H., & Arkin, A. (1997). Stochastic mechanisms in gene expression. *Proceedings of the National Academy of Sciences of the United States of America*, *94*(3), 814–819. doi:10.1073/pnas.94.3.814 PMID:9023339

McCullagh, E., Farlow, J., Fuller, C., Girard, J., Lipinski-Kruszka, J., Lu, D., & El-Samad, H. et al. (2009). Not all quiet on the noise front. *Nature Chemical Biology*, *5*(10), 699–704. doi:10.1038/nchembio.222 PMID:19763097

Meirovitch, H. (1982). A new method of simulation of real chains: Scanning future steps. *Journal of Physics. A, Mathematical and General*, 15.

Meirovitch, H. (1988). Statistical properties of the scanning simulation method for polymer chains. *J. Chemical Phisics, 89*.

Melykuti, B., Burrage, K., & Zygalakis, K. C. (2010). Fast stochastic simulation of biochemical reaction systems by alternative formulations of the chemical Langevin equation. *The Journal of Chemical Physics*, *132*(16), 164109. doi:10.1063/1.3380661 PMID:20441260

Mettetal, J. T., Muzzey, D., Pedraza, J. M., Ozbudak, E. M., & van Oudenaarden, A. (2006). Predicting stochastic gene expression dynamics in single cells. *Proceedings of the National Academy of Sciences of the United States of America*, *103*(19), 7304–7309. doi:10.1073/pnas.0509874103 PMID:16648266

Munsky, B., & Khammash, M. (2006). The finite state projection algorithm for the solution of the chemical master equation. *The Journal of Chemical Physics*, *124*(4), 044104. doi:10.1063/1.2145882 PMID:16460146

Munsky, B., & Khammash, M. (2007). A multiple time interval finite state projection algorithm for the solution to the chemical master equation. *Journal of Computational Physics*, *226*(1), 818–835. doi:10.1016/j.jcp.2007.05.016

Ogawa, M. (1989). Hemopoietic stem cells: Stochastic differentiation and humoral control of proliferation. *Environmental Health Perspectives*, *80*, 199–207. doi:10.1289/ehp.8980199 PMID:2647480

Ozbudak, E. M., Thattai, M., Kurtser, I., Grossman, A. D., & van Oudenaarden, A. (2002). Regulation of noise in the expression of a single gene. *Nature Genetics*, *31*(1), 69–73. doi:10.1038/ng869 PMID:11967532

Paulsson, J., & Ehrenberg, M. (2000). Random signal fluctuations can reduce random fluctuations in regulated components of chemical regulatory networks. *Physical Review Letters*, *84*(23), 5447–5450. doi:10.1103/PhysRevLett.84.5447 PMID:10990965

Ptashne, M. (2004). A Genetic Switch: Phage Lambda Revisited (3rd ed.). Cold Spring Harbor Laboratory Press.

Ptashne, M. (2007). On the use of the word ‘epigenetic’. *Current Biology*, *17*(7), R233–R236. doi:10.1016/j.cub.2007.02.030 PMID:17407749

Ptashne, M., Backman, K., Humayun, M., Jeffrey, A., Maurer, R., Meyer, B., & Sauer, R. (1976). Autoregulation and function of a repressor in bacteriophage lambda. *Science*, *194*(4261), 156–161. doi:10.1126/science.959843 PMID:959843

Roh, M. K., Daigle, B. J. Jr, Gillespie, D. T., & Petzold, L. R. (2011). State-dependent doubly weighted stochastic simulation algorithm for automatic characterization of stochastic biochemical rare events. *The Journal of Chemical Physics*, *135*(23), 234108. doi:10.1063/1.3668100 PMID:22191865

Roh, M. K., Gillespie, D. T., & Petzold, L. R. (2010). State-dependent biasing method for importance sampling in the weighted stochastic simulation algorithm. *The Journal of Chemical Physics*, *133*(17), 174106. doi:10.1063/1.3493460 PMID:21054005

Saad, Y. (2003). *Iterative methods for sparse linear systems*. Siam. doi:10.1137/1.9780898718003

Samoilov, M., Plyasunov, S., & Arkin, A. (2005). Stochastic amplification and signaling in enzymatic futile cycles through noise-induced bistability with oscillations. *Proceedings of the National Academy of Sciences of the United States of America*, *102*(7), 2310–2315. doi:10.1073/pnas.0406841102 PMID:15701703

Schultz, D., Onuchic, J. N., & Wolynes, P. G. (2007). Understanding stochastic simulations of the smallest genetic networks. *The Journal of Chemical Physics*, *126*(24), 245102. doi:10.1063/1.2741544 PMID:17614590

Shea, M. A., & Ackers, G. K. (1985). The OR control system of bacteriophage lambda. A physical-chemical model for gene regulation. *Journal of Molecular Biology*, *181*(2), 211–230. doi:10.1016/0022-2836(85)90086-5 PMID:3157005

Shiraishi, T., Matsuyama, S., & Kitano, H. (2010). Large-scale analysis of network bistability for human cancers. *PLoS Computational Biology*, *6*(7), e1000851. doi:10.1371/journal.pcbi.1000851 PMID:20628618

Sidje, R. B. (1998). Expokit: A software package for computing matrix exponentials. *ACM Transactions on Mathematical Software*, *24*(1), 130–156. doi:10.1145/285861.285868

Stewart, W. J. (1994). *Introduction to the numerical solution of Markov chains* (Vol. 41). Princeton University Press Princeton.

Torrie, G. M., & Valleau, J. P. (1977). Nonphysical sampling distributions in Monte Carlo free-energy estimation: Umbrella sampling. *Journal of Computational Physics*, *23*(2), 187–199. doi:10.1016/0021-9991(77)90121-8

Van Kampen, N. G. (2007). Stochastic processes in physics and chemistry (3rd ed.). Elsevier Science and Technology.

Vellela, M., & Qian, H. (2007). A Quasistationary Analysis of a Stochastic Chemical Reaction: Keizer's Paradox. *Bulletin of Mathematical Biology*, *69*(5), 1727–1746. doi:10.1007/s11538-006-9188-3 PMID:17318672

Vellela, M., & Qian, H. (2009). Stochastic dynamics and non-equilibrium thermodynamics of a bistable chemical system: The Schlogl model revisited. *Journal of the Royal Society, Interface*, *6*(39), 925–940. doi:10.1098/rsif.2008.0476 PMID:19095615

Volfson, D., Marciniak, J., Blake, W. J., Ostroff, N., Tsimring, L. S., & Hasty, J. (2006). Origins of extrinsic variability in eukaryotic gene expression. *Nature*, *439*(7078), 861–864. doi:10.1038/nature04281 PMID:16372021

Zhou, T., Chen, L., & Aihara, K. (2005). Molecular communication through stochastic synchronization induced by extracellular fluctuations. *Physical Review Letters*, *95*(17), 178103. doi:10.1103/PhysRevLett.95.178103 PMID:16383875

Zhu, X. M., Yin, L., Hood, L., & Ao, P. (2004). Robustness, stability and efficiency of phage lambda genetic switch: Dynamical structure analysis. *Journal of Bioinformatics and Computational Biology*, *2*(4), 785–817. doi:10.1142/S0219720004000946 PMID:15617166