Hindawi Publishing Corporation
EURASIP Journal on Advances in Signal Processing
Volume 2010, Article ID 204612, 9 pages
doi:10.1155/2010/204612

Research Article

Inferring Parameters of Gene Regulatory Networks via Particle Filtering

Xiaohu Shen and Haris Vikalo

Department of Electrical and Computer Engineering, University of Texas at Austin, TX 78712, USA

Correspondence should be addressed to Haris Vikalo, hvikalo@ece.utexas.edu

Received 6 April 2010; Revised 9 July 2010; Accepted 24 August 2010

Academic Editor: Rui Kuang

Gene regulatory networks are highly complex dynamical systems comprising biomolecular components which interact with each other and through those interactions determine gene expression levels, that is, determine the rate of gene transcription. In this paper, a particle filter with Markov Chain Monte Carlo move step is employed for the estimation of reaction rate constants in gene regulatory networks modeled by chemical Langevin equations. Simulation studies demonstrate that the proposed technique outperforms previously considered methods while being computationally more efficient. Dynamic behavior of gene regulatory networks averaged over a large number of cells can be modeled by ordinary differential equations. For this scenario, we compute an approximation to the Cramer-Rao lower bound on the mean-square error of estimating reaction rates and demonstrate that, when the number of unknown parameters is small, the proposed particle filter can be nearly optimal.

1. Introduction

Gene regulatory networks (GRN) are systems comprising biomolecular components (genes, mRNA, proteins) that interact with each other and through those interactions determine gene expression levels, that is, determine the rate of gene transcription to mRNA [1–3]. The signals in GRN are carried by molecules. For instance, proteins which enable initiation of the gene transcription to mRNA (so-called *transcription factors*) can be considered as input signals. They bind to the so-called promoter regions adjacent to the regulated gene and, in doing so, enable an RNA Polymerase to perform the transcription. On the other hand, proteins that are translated from the mRNA can be considered as output signals. Some of the created proteins may act as transcription factors themselves and upregulate or downregulate gene expressions, that is, activate or suppress the transcription process. This creates feedback loops in the network which allow direct or indirect self-regulation. An illustration of a possible segment of a regulatory pathway is shown in Figure 1.

Recent development of DNA and protein microarrays sparked a surge of interest in studying gene regulatory mechanisms. The excitement is due to the capability of the microarrays to conduct simultaneous tests of an entire genome of an organism. By testing a number of biological samples taken over a period of time, one can track the network dynamics. The experimental advances have been accompanied by the theoretical developments in modeling and computational studies of the networks. Combination of these research efforts provides critical information about the functionality of cells and organisms, reveals mechanisms of genetic diseases, enables optimization of diagnostic techniques and therapies, and provides aid in the process of drug discovery.

To enable the analysis of gene regulatory networks, we need accurate yet tractable models capturing their dynamical behavior. The molecular interactions in gene regulatory networks are inherently stochastic. For instance, the number of created proteins is a random variable due to thermal fluctuations in a cell which cause promotors to randomly switch between an active and a repressed state. The fluctuations in the number of proteins are enhanced by the protein degradation which is a stochastic process itself. This, along with several other sources of randomness, call for probabilistic modeling of gene regulatory networks. However, a very

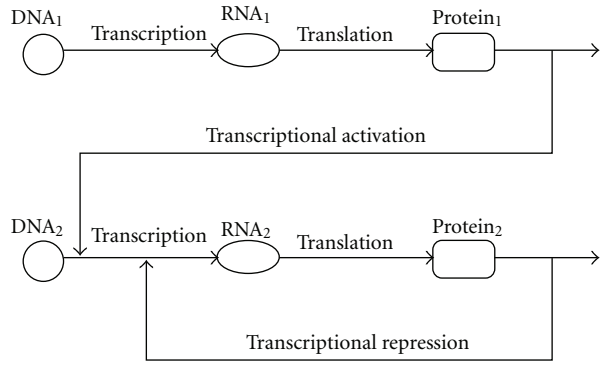

FIGURE 1: An illustration of a possible segment of a regulatory pathway.

detailed description of a network may be difficult to analyze and often requires considerable computational efforts. Hence, several models with varying degrees of accuracy and complexity have been proposed. These models rely on representations via chemical master and chemical Langevin equations [4–6], and ordinary differential equations [7, 8] as well as Bayesian [9, 10] and Boolean [3, 11] networks. Having selected one of the above models, we are interested in finding its structure and parameters that provide the best explanation of the experimental data. This requires further computational studies and opens up questions related to, for example, stability and control of the network. However, inference problems in gene regulatory networks are often challenging, and the difficulty of a problem increases with the complexity of the model and the size of the network.

In this paper, we consider models of GRN based on chemical master equations and study the problem of estimating stochastic rate constants therein. Such models provide the most precise description of the network processes; however, they are also computationally the most demanding. We limit our focus on small-sized networks with a known structure but unknown rate constants. We approximate a chemical master equation by a related chemical Langevin equation [12] and employ a particle filter with the Markov Chain Monte Carlo move step to solve the rate estimation problem. Simulation studies demonstrate that the proposed technique outperforms previously considered methods while being computationally more efficient. Dynamic behavior of gene regulatory networks averaged over a large number of cells can be modeled by ordinary differential equations. For this scenario, we compute an approximation to the Cramer-Rao lower bound on the mean-square error of estimating reaction rates and demonstrate that, when the number of unknown parameters is small, the proposed particle filter can be nearly optimal.

The paper is organized as follows. Section 2 describes the chemical master equation model of a gene regulatory network and its approximation by a chemical Langevin equation. Section 3 presents the particle filtering algorithm for the estimation of the stochastic rate constants and compares its performance with prior work. In Section 4, a deterministic model based on ordinary differential equations is described, and the Cramer-Rao lower bound on the performance of estimating rate constants is computed. Finally, we conclude the paper in Section 5.

2. Models Based on Chemical Master and Chemical Langevin Equations

Consider a GRN comprising N molecular components. The network variables are the numbers of the molecules of each of the N species; generally, we are interested in the temporal changes of these variables. Denote the number of molecules of the ith network component at time t by $x_i(t)$; for convenience, collect the $x_i(t)$ into a vector $X(t)$, that is, denote $X(t) = [x_1(t) \cdots x_N(t)]^T$. Molecular reactions in a GRN are subject to significant spontaneous fluctuations. Consequently, the numbers of the molecular species $x_i(t)$ are inherently stochastic processes. We can model $X(t)$ as a Markov process with discrete states, where the time evolution of the state probabilities $P(X,t)$ is given by the chemical master equation

$$\frac{\partial P(X,t)}{\partial t} = \sum_{m=1}^{M} [a_m(X - \mathcal{V}_m)P(X - \mathcal{V}_m, t) - a_m(X)P(X,t)]. \quad (1)$$

In (1), M denotes the total number of reactions that are possible within the network (i.e., the number of the so-called *reaction channels*), and $\mathcal{V}_m = [v_{m1} \quad v_{m2} \quad \cdots \quad v_{mN}]^T$ is the vector describing change in the number of molecules of each of the N species due to the reaction in the mth reaction channel (e.g., v_{mi} is the change, either positive or negative, in the number of molecules of the ith network component due to the reaction in the mth channel). Moreover, $a_m(\cdot)$ in (1) is the so-called propensity function, that is, $a_m(\cdot)dt$ is the probability that during time interval $(t, t + dt)$ there is a reaction in the mth channel. The propensity function can further be expressed as $a_m(X(t)) = c_m h_m(X(t))$, where $c_m dt$ is the probability that one reaction takes place in $(t, t + dt)$ and $h_m(X(t))$ denotes the number of possible simultaneous reactions. (The coefficients c_m are often referred to as the stochastic rate constants. The function $h_m(X(t))$ counts all possible combinations of individual molecules that may lead to a reaction in the mth channel.) The chemical master equation is often used to simulate the Markov process $X(t)$ and enable computational studies of GRN. To this end, one may employ various stochastic simulation algorithms, originally proposed by Gillespie [4].

Model (1) provides a very accurate description of the network dynamics [4]. However, since it tracks individual discrete events, it is often cumbersome for practical purposes. For instance, relying on (1) to infer the parameters of the network (i.e., the stochastic rate constants c_m) may in principle be possible [13]; however, it is computationally rather intensive to do so. Therefore, simplified network models are desirable. Under certain assumptions

(e.g., large $x_i(t)$, small dt), we may approximate (1) by the chemical Langevin equation,

$$X(t+dt) - X(t) = \sum_{m=1}^{M} \left[\mathcal{V}_m a_m(X(t))dt + \mathcal{V}_m \sqrt{a_m(X(t))dt}\,\mathcal{N}_m(0,1) \right], \tag{2}$$

where $\mathcal{N}_m(0,1)$ denote zero-mean, unit-variance, independent, identically distributed (iid) Gaussian random variables. By collecting vectors $\mathcal{V}_m$ into a stoichiometry matrix $S = [\mathcal{V}_1 \quad \mathcal{V}_2 \quad \cdots \quad \mathcal{V}_M]$, we can write (2) as

$$X(t+dt) - X(t) = S\mathbf{a}(X(t))dt + \left(SA(X(t))S^T\right)^{1/2} dW, \tag{3}$$

where dW denotes an M-dimensional Wiener process; vector $\mathbf{a}(X(t))$ is defined as

$$\mathbf{a}(X(t)) = [a_1(X(t)) \quad a_2(X(t)) \quad \cdots \quad a_M(X(t))]^T, \tag{4}$$

where

$$A(X(t)) = \operatorname{diag}\{a_1(X(t)), a_2(X(t)), \ldots, a_M(X(t))\}. \tag{5}$$

We should point out that while the chemical Langevin equation (2) may be used as a network model for the purpose of parameter estimation, in general it is not sufficiently accurate to provide reliable simulations of the network dynamics. To conduct computational studies of a GRN, we still need to model them using stochastic simulation algorithms.

Let us write the chemical Langevin equation (3) using the notation typically encountered in the literature on stochastic differential equations as

$$X(t+dt) - X(t) = \mu(X(t), \boldsymbol{\theta})dt + \sigma(X(t), \boldsymbol{\theta})dW, \tag{6}$$

where $\mu(X(t), \boldsymbol{\theta}) = S\mathbf{a}(X(t))$ denotes the drift, $\sigma(X(t), \boldsymbol{\theta}) = (SA(X(t))S^T)^{1/2}$ is the diffusion, and $\boldsymbol{\theta}$ is the vector of (generally unknown) parameters (i.e., the elements of $\boldsymbol{\theta}$ are the stochastic rate constants c_i). Our goal is to infer $\boldsymbol{\theta}$ from $X(t)$ observed at discrete time instances $t_i = i\Delta, 1 \le i \le L$, where L denotes the total number of observations. Assuming zero-mean Gaussian measurement noise with covariance matrix Σ, the collected observations have normal distribution of the form

$$y_i = y(i\Delta) \sim \mathcal{N}(X(i\Delta), \Sigma). \tag{7}$$

In [14], the authors find the best linear-model fit to the data presumed to be generated by (6), and then infer parameters based on the derived linear model. In [15, 16], the use of statistical mechanics tools for the estimation of the parameters of a network modeled by (6) was considered. In [17, 18], a Markov Chain Monte Carlo (MCMC) algorithm was employed to infer the network parameters. This approach provides sound estimate of the parameters, but it requires a very high computational effort. As an alternative, we propose the use of a particle filter with an MCMC move step. This we describe in the next section.

3. Particle Filter with Markov Chain Monte Carlo Move Step

We consider Bayesian approaches to inferring the unknown parameters in $\boldsymbol{\theta}$, which is treated as a random vector with a prior $p(\boldsymbol{\theta})$. Specifically, we rely on particle filtering methods to infer the posterior distribution $p(\boldsymbol{\theta} \mid y_{1:N})$, and then find the estimate $\widehat{\boldsymbol{\theta}}$ as the conditional mean of $p(\boldsymbol{\theta} \mid y_{1:N})$. Here $y_{1:N} = \{y_1, y_2, \ldots, y_N\}$ denotes the set of observations collected in the interval $[\Delta, N\Delta]$, where Δ denotes the sampling period and N denotes the total number of observations (e.g., y_n is the noisy observation collected at time $n\Delta$). The desired posterior distribution can be expressed as

$$p(\boldsymbol{\theta} \mid y_{1:N}) = \int p(x_{1:N}, \boldsymbol{\theta} \mid y_{1:N}) dx_{1:N}, \tag{8}$$

where $x_{1:N} = \{x_1, x_2, \ldots, x_N\}$ denotes the set of points of the process $X(t)$ corresponding to the observations in $y_{1:N}$ (e.g., $x_n = X(n\Delta)$), and $p(x_{1:N}, \boldsymbol{\theta} | y_{1:N})$ is given by

$$p(x_{1:N}, \boldsymbol{\theta} \mid y_{1:N}) \propto p(y_{1:N} \mid x_{1:N}, \boldsymbol{\theta}) p(x_{1:N} \mid \boldsymbol{\theta}) p(\boldsymbol{\theta}). \tag{9}$$

To evaluate (9), one needs to compute the joint density $p(x_{1:N} \mid \boldsymbol{\theta}) = p(x_N \mid x_{N-1}, \boldsymbol{\theta}) \cdots p(x_2 \mid x_1, \boldsymbol{\theta}) p(x_1 \mid \boldsymbol{\theta})$. In general, however, the transition densities

$$p(x_{n+1} \mid x_n, \boldsymbol{\theta}) = p(X((n+1)\Delta) \mid X(n\Delta), \boldsymbol{\theta}) \tag{10}$$

for the process (6) are not available in a closed form. The stochastic differential equation (6) can be discretized using the Euler-Maruyama scheme as

$$x_{n+1} = x_n + \mu(x_n, \boldsymbol{\theta})\Delta + \sigma(x_n, \boldsymbol{\theta})\delta W, \tag{11}$$

where δW denotes a zero-mean Gaussian distribution with covariance ΔI, and I denotes the identity matrix. Hence the transition density $p(x_n | x_{n-1}, \boldsymbol{\theta})$ can be approximated by a Gaussian distribution with mean $x_n + \mu(x_n, \boldsymbol{\theta})\Delta$ and covariance $\sigma(x_n, \boldsymbol{\theta})(\sigma(x_n, \boldsymbol{\theta}))^T\Delta$. However, the Euler-Maruyama approximation of the transition density is accurate only when Δ is small. If the sampling period is not sufficiently small, one can introduce the so-called missing values $z_{1:m} = \{z_1, z_2, \ldots, z_m\}$ which emulate the diffusion process between x_n and x_{n+1} (a distinct set of missing values is introduced for each n). The number of augmented missing values m is chosen such that the Euler-Maruyama approximation of the transition density between z_k and z_{k+1} is accurate, that is, m is chosen such that $p(z_{k+1} | z_k, \boldsymbol{\theta})$ can be closely approximated by a Gaussian distribution. It is straightforward to show that

$$\widetilde{\pi}\left(z_j \mid z_{j-1}, \boldsymbol{\theta}_{n-1}, y_n\right) = \mathcal{N}\left(z_{j-1} + \psi\frac{\Delta}{m}, \gamma\frac{\Delta}{m}\right), \tag{12}$$

where $\psi = \mu + \beta(\beta\Delta_j + \Sigma)^{-1}(y_n - [x_{n-1} + \mu\Delta_j])$, $\gamma = \beta - \beta(\beta\Delta_j + \Sigma)^{-1}\beta^T(\Delta/m)$, $\mu = S\mathbf{a}(x_{n-1})$, $\beta = SA(x_{n-1})S^T$, $\Delta_j = (m-j+1)(\Delta/m)$, and Σ denotes the covariance matrix of the measurement noise.

Introduction of the missing values enables propagating (9) by means of a particle filter, where the filter

relies on a Gaussian importance density (12). A simple sequential importance resampling (SIR) scheme provides asymptotically consistent estimates, that is, the approximation converges to the true value of the parameters as the number of particles grows. However, the SIR scheme often suffers from sample impoverishment and, therefore, has weak performance. To improve the sample diversity and the performance of the particle filter, we employ the importance sampling scheme with an MCMC move step. Specifically, we use the Metropolis-Hastings algorithm to decide whether a resampled particle will be accepted or not. For implementation details, we refer the reader to the formal algorithm given below.

Algorithm 1 (initialization). Set $n = 1$. Draw $\{\boldsymbol{\theta}_n^i, x_n^i\}_{i=1}^{N_s}$ from the prior density $\pi(\boldsymbol{\theta})\pi(x_n)$. Assign particle weights $\omega_n^i = \pi(y_n|x_n^i, \boldsymbol{\theta}_n^i)$, for $i = 1, 2, \ldots, N_s$, and normalize them.

Algorithm 2 (iterations). For $n \geq 2$.

(i) For $i = 1, \ldots, N_s$, draw missing data $\{z_k^i\}_{k=1}^m$ from an importance density

$$q\left(z_1, \ldots, z_m \mid \boldsymbol{\theta}_{n-1}^i, x_{n-1}^i, y_n\right) \tag{13}$$

obtained using the Euler approximation as

$$\pi\left(z_1 \mid x_{n-1}^i\right) \prod_{j=2}^{m} \widetilde{\pi}\left(z_j^i \mid z_{j-1}^i, \boldsymbol{\theta}_{n-1}^i, y_n\right), \tag{14}$$

where

$$\widetilde{\pi}\left(z_j^i \mid z_{j-1}^i, \boldsymbol{\theta}_{n-1}^i, y_n\right) = \mathcal{N}\left(z_{j-1}^i + \psi\frac{\Delta}{m}, \gamma\frac{\Delta}{m}\right), \tag{15}$$

$\psi = \mu + \beta(\beta\Delta_j + \Sigma)^{-1}(y_n - [x_{n-1}^i + \mu\Delta_j])$, $\gamma = \beta - \beta(\beta\Delta_j + \Sigma)^{-1}\beta^T(\Delta/m)$, $\mu = S\mathbf{a}(x_{n-1}^i)$, $\beta = SA(x_{n-1}^i)S^T$, $\Delta_j = (m - j + 1)(\Delta/m)$.

Set $x_n^i = z_m^i$ and update the particle weights as

$$\omega_n^i = \omega_{n-1}^i \pi\left(y_n \mid x_n^i, \boldsymbol{\theta}_{n-1}^i\right)\pi\left(z_1^i \mid x_{n-1}^i, \boldsymbol{\theta}_{n-1}^i\right) \times \frac{\prod_{j=2}^m \pi\left(z_j^i \mid z_{j-1}^i, \boldsymbol{\theta}_{n-1}^i\right)}{q\left(z_1, \ldots, z_m \mid \boldsymbol{\theta}_{n-1}^i, x_{n-1}^i, y_n\right)}. \tag{16}$$

(ii) (*Normalization*) Normalize the weights ω_n^i, and compute $N_{\text{eff}} = (1/\sum_{k=1}^{N_s}(\omega_n^k)^2)$.

(iii) (*Resampling*) If $N_{\text{eff}} < N_{\text{threshold}}$,

$$\left\{\boldsymbol{\theta}_n^{i*}, x_n^{i*}, \frac{1}{N_s}\right\}_{i*=1}^{N_s} = \text{Resample}\left(\left\{\boldsymbol{\theta}_n^i, x_n^i, \omega_n^i\right\}_{i=1}^{N_s}\right). \tag{17}$$

(iv) (*Resample move*) If resampling is performed in Algorithm 2(iii), then for $i = 1, \ldots, N_s$:

(a) Draw a candidate $\boldsymbol{\theta}_*$ from a kernel density $\mathcal{K}(\boldsymbol{\theta}) = \mathcal{N}(\boldsymbol{\theta}_n^i, h_{\text{opt}}S)$, where S is the empirical covariance of $\boldsymbol{\theta}$ in the previous step and h_{opt} is the smoothing parameter.

(b) Draw missing data $\{z_{k*}^i\}_{k*=1}^m$ from an importance density $q(z_{1*}, \ldots, z_{m*}|\boldsymbol{\theta}_*, x_{n-1}^i, y_n)$ and set $x_{n*}^i = z_{m*}^i$.

(c) Calculate the Metropolis-Hastings acceptance rate

$$\alpha = \frac{\pi(y_n \mid x_{n*}^i, \boldsymbol{\theta}_*)\pi\left(z_{1*} \mid x_{n-1}^i, \boldsymbol{\theta}_*\right)}{\pi\left(y_n \mid x_n^i, \boldsymbol{\theta}_n^i\right)\pi\left(z_1^i \mid x_{n-1}^i, \boldsymbol{\theta}_{n-1}^i\right)} \times \frac{\prod_{j=2}^m \pi\left(z_{j*}^i \mid z_{(j-1)*}^i, \boldsymbol{\theta}_*\right)}{\prod_{j=2}^m \pi\left(z_j^i \mid z_{j-1}^i, \boldsymbol{\theta}_{n-1}^i\right)} \frac{q\left(z_1, \ldots, z_m \mid \boldsymbol{\theta}_{n-1}^i, x_{n-1}^i, y_n\right)}{q\left(z_{1*}, \ldots, z_{m*} \mid \boldsymbol{\theta}_*, x_{n-1}^i, y_n\right)}. \tag{18}$$

(d) Set $(\boldsymbol{\theta}_n^i, x_n^i) = (\boldsymbol{\theta}_*, x_{n*}^i)$ with prob. $\min\{1, \alpha\}$.

3.1. Computational Study of a Viral Infection Network. We demonstrate the performance of the proposed algorithm on a viral infection network previously studied in [19, 20]. The network comprises 6 reaction channels,

R_1: DNA + P $\rightarrow$ V,
R_2: DNA $\rightarrow$ RNA + DNA,
R_3: RNA $\rightarrow$ $\varnothing$,
R_4: RNA $\rightarrow$ DNA + RNA,
R_5: RNA $\rightarrow$ P + RNA,
R_6: P $\rightarrow$ $\varnothing$,

where P denotes viral protein molecules and V denotes synthesized viral cells. Reaction R_1 is the processes of producing viral cells from the viral DNA and protein. Reactions R_2 and R_5 are the transcription and translation process of the viral genes, respectively. Reaction R_4 models replication of a viral RNA template into a viral DNA.

For the purpose of parameter estimation, we assume that the above network evolves according to (3). However, the network is simulated via the Gillespie's algorithm, with the rate constants set to $[c_1\ \ c_2\ \ c_3\ \ c_4\ \ c_5\ \ c_6]^T = [11.25 \times 10^{-3}\ \ 0.25\ \ 0.5\ \ 1\ \ 2\ \ 1]^T$. We refer to the proposed particle filtering algorithm with MCMC move step as Algorithm 1, and employ it to estimate parameters in this network. The performance of Algorithm 1 is compared to the MCMC method proposed in [18], denoted for convenience as Algorithm 2. Algorithm 1 is performed with $N_s = 2 \times 10^4$, $m = 15$, and the resampling threshold $N_{\text{threshold}} = N_s/2$. Both algorithms use $N = 40$ noisy observations of the network states and employ the same initial sample distribution. The log values of the parameters $\log(\boldsymbol{\theta}_i)$ are initialized from the uniform distribution $\mathcal{U}(-4, 2)$, and the noise variance is assumed to be known. (Note that even though c_1 does not belong to the initialization range, the proposed technique accurately infers its value.) Figure 2 compares the mean square-error of estimating the parameters of the

viral infection network using Algorithms 1 and 2, obtained by performing 150 simulation runs. Clearly, the proposed Algorithm 1 outperforms Algorithm 2, while being roughly 5 times faster—the average running time of Algorithm 1 is 1030 seconds, while the average running time of Algorithm 2 is 5500 seconds (simulations in Matlab).

3.2. Computational Study of Prokaryotic Regulation. In this subsection, we illustrate the performance of the proposed algorithm when employed for estimating reaction rates in a network with 12 parameters. In particular, we consider estimation of the reaction rates in a GRN model of prokaryotic auto regulation. The system is characterized by the following 12 reactions [18]:

$$
\begin{aligned}
R_1&: P_1 + \mathrm{DNA}_1 \rightarrow P \cdot \mathrm{DNA}_1,\\
R_2&: P \cdot \mathrm{DNA}_1 \rightarrow P_1 + \mathrm{DNA}_1,\\
R_3&: P_1 + \mathrm{DNA}_2 \rightarrow P \cdot \mathrm{DNA}_2,\\
R_4&: P \cdot \mathrm{DNA}_2 \rightarrow P_1 + \mathrm{DNA}_2,\\
R_5&: \mathrm{DNA}_1 \rightarrow \mathrm{DNA}_1 + \mathrm{RNA}_1,\\
R_6&: \mathrm{RNA}_1 \rightarrow \mathrm{RNA}_1 + P_1,\\
R_7&: \mathrm{DNA}_2 \rightarrow \mathrm{DNA}_2 + \mathrm{RNA}_2,\\
R_8&: \mathrm{RNA}_2 \rightarrow \mathrm{RNA}_2 + P_2,\\
R_9&: \mathrm{RNA}_1 \rightarrow \varnothing,\\
R_{10}&: \mathrm{RNA}_2 \rightarrow \varnothing,\\
R_{11}&: P_1 \rightarrow \varnothing,\\
R_{12}&: P_2 \rightarrow \varnothing.
\end{aligned}
$$

Reactions $R_1 \sim R_4$ represent the reversible processes of repressor protein P_1 binding to DNA_1 and DNA_2. Reactions $R_5 \sim R_8$ are the transcription and translation processes of genes DNA_1 and DNA_2. Reactions $R_9 \sim R_{12}$ represent the degradation process of proteins and mRNAs in the system. The state vector X collects the numbers of components DNA_1, DNA_2, RNA_1, RNA_2, P_1, and P_2, and hence it is a 6-dimensional state vector.

Similar to the study of the viral infection network in the previous subsection, to infer the reaction rates, we assume that the above network evolves according to (3). However, the network is simulated via Gillespie's algorithm. In particular, we generate $N = 30$ noisy observations y_n, $1 \leq n \leq 30$, where the measurement noise is Gaussian with $\sigma^2 = 1$ (i.e., the noise variance matrix is $\Sigma = I$). The particle filter (Algorithm 1) is performed with $N_s = 2 \times 10^5$, $m = 20$, and the resampling threshold $N_{\text{threshold}} = N_s/5$. The log values of the parameters $\log(\boldsymbol{\theta}_i)$ are initialized from the uniform distribution $\mathcal{U}(-5, 1)$.

The reaction rates are inferred as the mean values of the distributions estimated by the particle filter. True values of the parameters and their estimates are shown in Table 1. When Algorithm 1 is performed with $m = 20$ and 2×10^5 MCMC iterations with a 3×10^4 burn-in period, the runtime of Algorithms 1 and 2 is comparable but the former is significantly more precise than the latter. In order to achieve similar performance, Algorithm 2 requires significantly higher complexity (10^6 MCMC iterations with a 3×10^4 burn-in period).

Table 1: True and estimated parameters for the two algorithms. Algorithm 2(i) employs 2×10^5 MCMC iterations, and Algorithm 2(ii) employs 10^6 iterations.

		Algorithm 1	Algorithm 2(i)	Algorithm 2(ii)
c_1	0.08	0.0707	0.0443	0.0869
c_2	0.82	0.8219	0.6726	0.7134
c_3	0.09	0.0597	0.1121	0.0650
c_4	0.9	0.5625	1.3913	0.5943
c_5	0.25	0.3283	0.1826	0.2862
c_6	0.1	0.1195	0.5800	0.0469
c_7	0.35	0.2875	0.9009	0.2561
c_8	0.3	0.4167	0.8943	0.3577
c_9	0.1	0.1197	0.1573	0.0985
c_{10}	0.1	0.1432	0.5097	0.2943
c_{11}	0.12	0.1178	1.2766	0.0984
c_{12}	0.1	0.1384	0.1669	0.1232
time(s)		5.9×10^4	5.3×10^4	2.5×10^5

4. A Deterministic Model of Gene Regulatory Networks

In reaction systems, where both the number of molecules and the system volume are large, due to averaging, the system dynamics can be described by a deterministic model. The same applies to modeling the dynamic behavior of a gene regulatory network averaged over a large number of cells. A deterministic model based on ordinary differential equations (ODE) is of the form [12, 21]

$$\frac{d\mathbf{x}(t)}{dt} = S\mathbf{a}(\mathbf{x}(t), \boldsymbol{\theta}), \tag{19}$$

where $\mathbf{x}(t)$ comprises real-valued and deterministic variables. On the other hand, the observation process is assumed to be corrupted by a Gaussian noise and hence the measurements are given by

$$y(t) = \mathbf{x}(t) + \mathbf{v}(t). \tag{20}$$

Typically, observations are collected at discrete time instances $t_i = i\Delta, 1 \leq i \leq L$, where L denotes the total number of observations. Therefore,

$$y(i\Delta) = \mathbf{x}(i\Delta) + \mathbf{v}(i\Delta), \quad 1 \leq i \leq L, \tag{21}$$

where $E\{\mathbf{v}(i\Delta)v(j\Delta)^T\} = \sigma_{\mathbf{v}}^2 I_N \delta_{ij}$.

To facilitate a simple estimation procedure, (19) can be discretized as

$$\mathbf{x}((i+1)\Delta) - \mathbf{x}(i\Delta) = \Delta \cdot S\mathbf{a}(\mathbf{x}(i\Delta), \boldsymbol{\theta}), \quad i = 1, \ldots, L-1. \tag{22}$$

It was pointed out in [22] that, under appropriate conditions, discretization induces smaller error than the measurement noise. In general, we assume that $\Delta \ll \sigma^2$. To estimate the unknown parameters $\boldsymbol{\theta}$ in (22), we employ the particle filtering with MCMC step (i.e., Algorithm 1 in Section 3). Since the state transitions in model (22) are deterministic

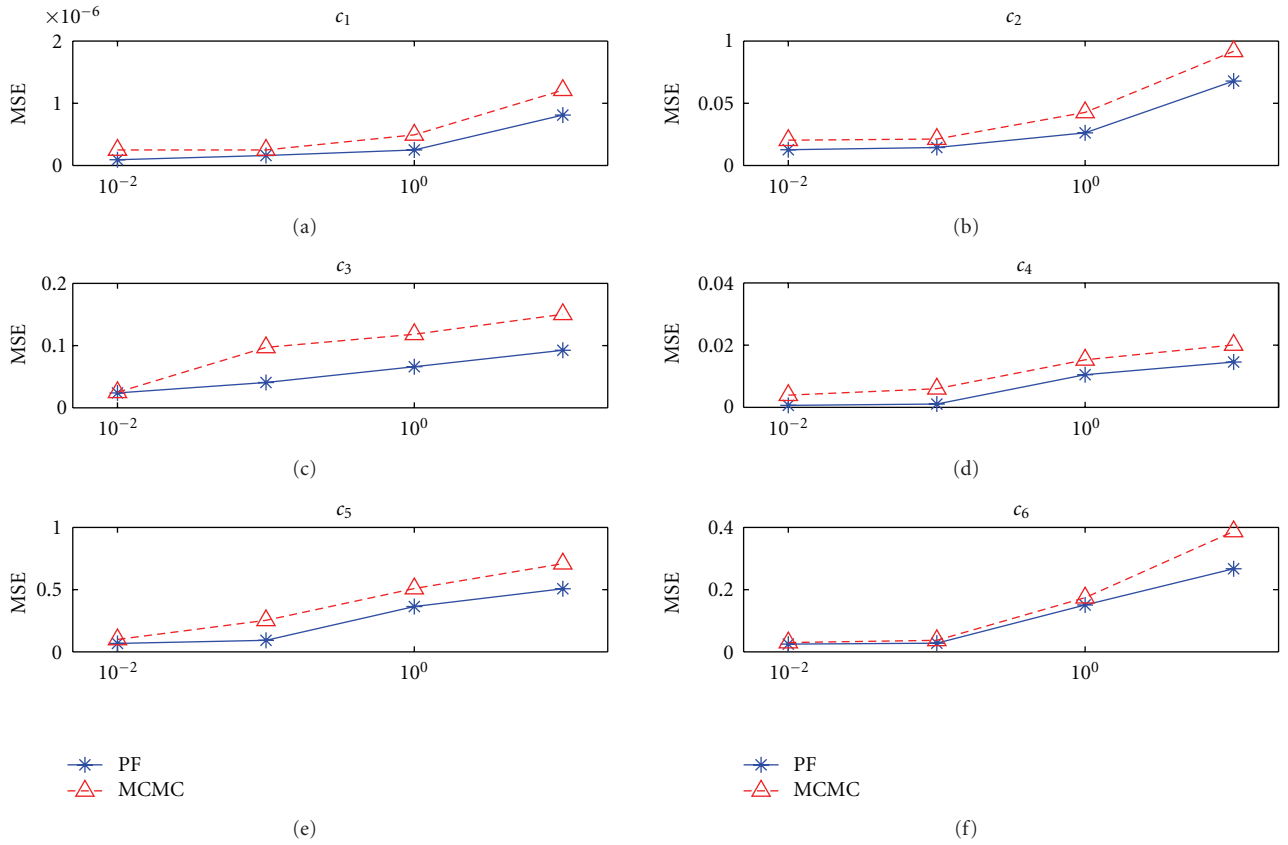

FIGURE 2: The mean-square-error performance comparison between Algorithm 1 (particle filter) and Algorithm 2 (MCMC) as a function of the variance of the observation noise σ^2 ($N_s = 2 \times 10^4$, $m = 15$, $N = 40$, $\Sigma = \sigma^2 I$).

(and not random, as in (3)), some of the steps of Algorithm 2 are simplified. In particular, steps (i) and (iv.b) of Algorithm 2 can be simplified in the following way: for each particle, instead of drawing a series of missing data $\{z_k^i\}_{k=1}^m$ from an importance distribution, we deterministically generate them from the previous state x_{n-1}^i as

$$\begin{aligned} z_1^i &= x_{n-1}^i + \frac{\Delta}{m} \cdot S\mathbf{a}\left(\mathbf{x}_{n-1}^i, \boldsymbol{\theta}_{n-1}^i\right), \\ z_K^i &= z_{k-1}^i + \frac{\Delta}{m} \cdot S\mathbf{a}\left(\mathbf{z}_{k-1}^i, \boldsymbol{\theta}_{n-1}^i\right), \\ &\qquad k = 2, \ldots, m, \end{aligned} \tag{23}$$

and $x_n^i = z_m^i$. The weights updating equation becomes

$$\omega_n^i = \omega_{n-1}^i \pi\left(y_n \mid x_n^i, \boldsymbol{\theta}_{n-1}^i\right). \tag{24}$$

Moreover, in Algorithm 2 (iv.c), the Metropolis-Hastings acceptance rate is simplified to

$$\alpha = \frac{\pi\left(y_n \mid x_{n*}^i, \boldsymbol{\theta}_*\right)}{\pi\left(y_n \mid x_n^i, \boldsymbol{\theta}_n^i\right)}. \tag{25}$$

Other steps of Algorithm 1 remain unchanged.

4.1. Cramer-Rao Lower Bound on the Mean-Square Error of Estimating Reaction Rates. Mean-square error of any estimation procedure can be bounded below by the Cramer-Rao lower bound (CRLB) [23]. In this section, we compute the CRLB on the estimation of reaction rates in the network described by (19) and (21). Collect the observations $y(i\Delta)$, $i = 1, 2, \ldots, L$, into a vector

$$\mathbf{y} = \begin{bmatrix} y(\Delta)^T & y(2\Delta)^T & \cdots & y(L\Delta)^T \end{bmatrix}^T. \tag{26}$$

The Cramer-Rao lower bound on the minimum mean-square error of estimating a parameter $\boldsymbol{\theta}_i$, given $\mathbf{y}$, is computed as

$$E\left(\hat{\boldsymbol{\theta}}_i - \boldsymbol{\theta}_i\right)^2 \geq \left[F^{-1}\right]_{ii}, \tag{27}$$

where the Fisher information matrix F is given by the negative of the expected value of the Hessian matrix of $\log p_{\mathbf{y}|\boldsymbol{\theta}}(\mathbf{y})$

$$F_{ij} = -E_{\mathbf{y}} \frac{\partial^2}{\partial\theta_i \partial\theta_j} \log p_{\mathbf{y}|\boldsymbol{\theta}}(\mathbf{y}). \tag{28}$$

From (21), it follows that $\mathbf{y}$ is a Gaussian vector with mean $\overline{y}(i\Delta) = \mathbf{x}(i\Delta)$ and covariance $R = \sigma_{\mathbf{v}}^2 I_{NL}$. Thus we have

$$p_{\mathbf{y}|\boldsymbol{\theta}}(\mathbf{y}) = \frac{1}{\sqrt{(2\pi)^{NL}|R|}} \exp\left[-\frac{1}{2}(\mathbf{y} - \overline{\mathbf{y}})^T R^{-1} (\mathbf{y} - \overline{\mathbf{y}})\right]. \tag{29}$$

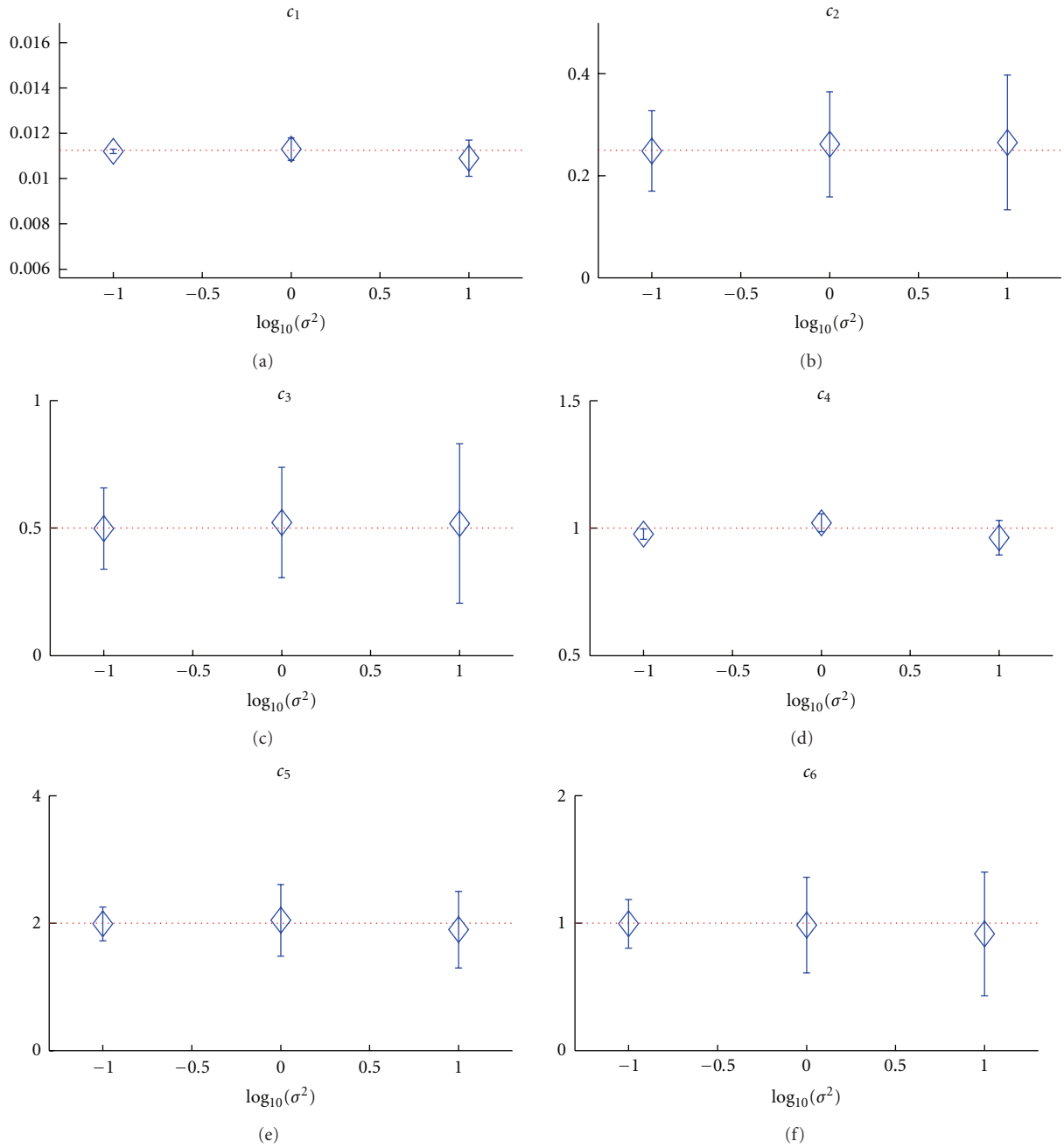

FIGURE 3: The mean and standard error of the particle filter estimator for the inference of reaction rates in a viral infection network, shown as a function of the variance of the observation noise (the number of particles used is $N_s = 10^4$; performance is averaged over 150 simulation runs).

Following similar derivations in [24], we obtain that

$$F_{ij} = \left(\frac{\partial \overline{\mathbf{y}}}{\partial \theta_i}\right)^T R^{-1} \left(\frac{\partial \overline{\mathbf{y}}}{\partial \theta_j}\right) + \frac{1}{2} \operatorname{tr}\left\{R^{-1}\frac{\partial R}{\partial \theta_i} R^{-1}\frac{\partial R}{\partial \theta_j}\right\}. \quad (30)$$

Since R is known, $\partial R/\partial \theta_i = 0$, and thus

$$F_{ij} = \left(\frac{\partial \overline{\mathbf{y}}}{\partial \theta_i}\right)^T R^{-1} \left(\frac{\partial \overline{\mathbf{y}}}{\partial \theta_j}\right). \quad (31)$$

Therefore, only $\partial \overline{\mathbf{y}}/\partial \theta_i$ is needed to evaluate F_{ij}. From (20), it follows that $\overline{y}(t) = x(t)$. Moreover,since $a_m(\mathbf{x}(t)) = \theta_m h_m(\mathbf{x}(t))$, we can write

$$\mathbf{a}(\overline{y}(t)) = \operatorname{diag}\{\boldsymbol{\theta}\} H(\overline{y}(t)), \quad (32)$$

where $H(\overline{y}(t)) = [h_1(\overline{y}(t)) \quad h_2(\overline{y}(t)) \quad \cdots \quad h_M(\overline{y}(t))]^T$. Taking derivatives of both side of (22), we obtain

$$\frac{\partial \overline{y}((i+1)\Delta)}{\partial \theta_m} = \frac{\partial \overline{y}(i\Delta)}{\partial \theta_m} + \Delta \cdot S \cdot E_i H(\overline{y}(i\Delta)) + \Delta \cdot S \operatorname{diag}\{\theta\} \frac{\partial H(\overline{y}(i\Delta))}{\partial \theta_m}, \quad (33)$$

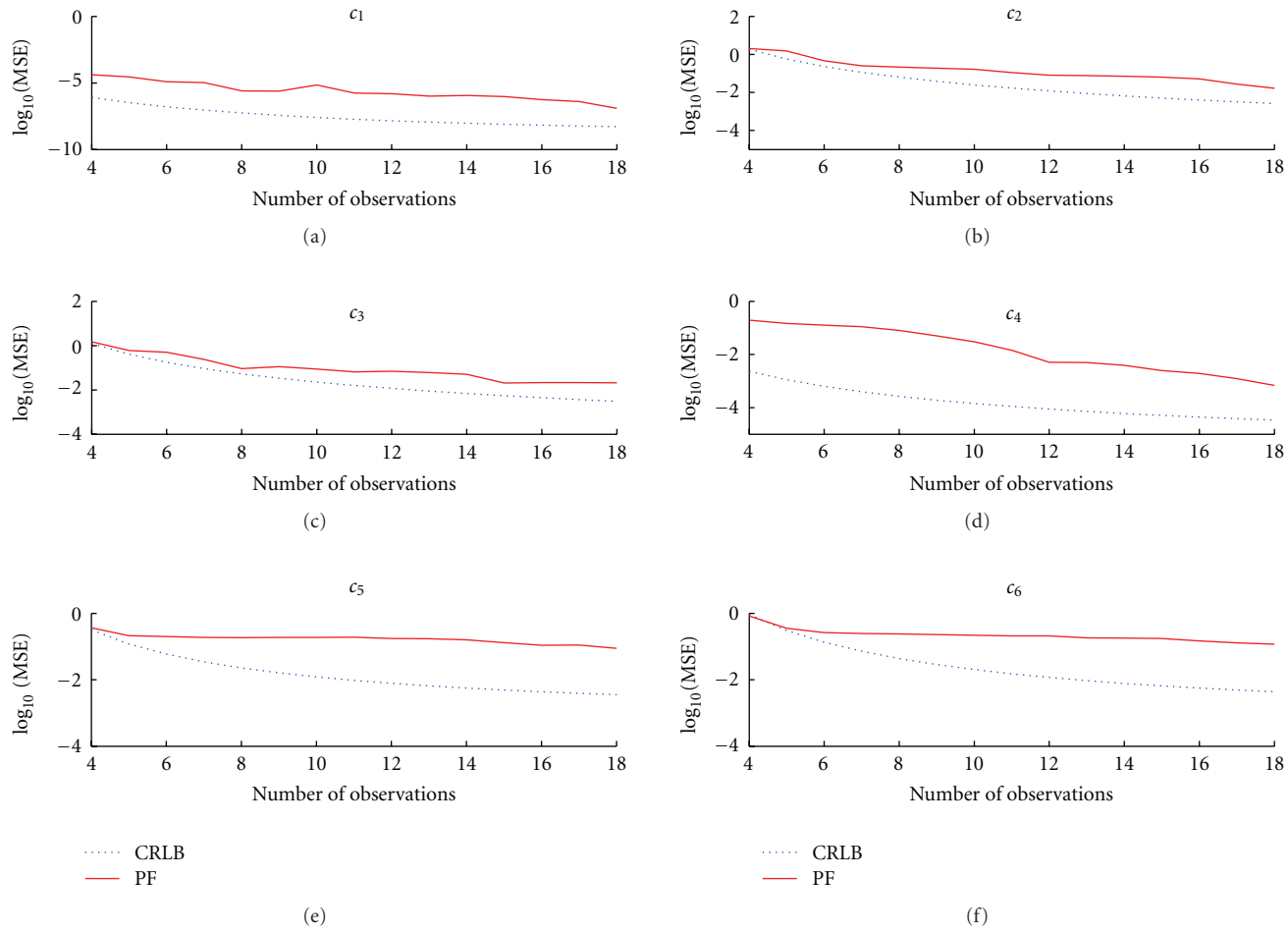

FIGURE 4: The CRLB and the average mean-square error of the particle filtering algorithm (the number of particles $N_s = 10^4$, noise covariance matrix $\Sigma = I$).

where E_i denotes the $M \times M$ matrix with all zero entries except the (i, i) entry which is equal to 1.

Notice that $\partial h_i(\overline{y}(i\Delta))/\partial\theta_m$ are functions of $\overline{y}(i\Delta)$ and $\partial h_i(\overline{y}(i\Delta))/\partial\theta_m$; therefore, we can recursively calculate $\partial\overline{y}((i+1)\Delta)/\partial\theta_m$ from $\partial\overline{y}(i\Delta)/\partial\theta_m$ and $\overline{y}(i\Delta)$. The value of $\overline{y}(t)$s can be obtained by numerically solving (19) (e.g., using *Mathematica*). This enables computation of $\partial\overline{\mathbf{y}}/\partial\theta_m$ and, therefore, the desired CRLB. (Note that the CRLB computed in this section assumes the discretized model (22); as $\Delta \to 0$, it approaches the true bound on estimating $\boldsymbol{\theta}$ in (19)).

4.2. Computational Study of a Viral Infection Network. We illustrate the performance of the particle filter and compare it with the computed CRLB for the case of the viral infection network studied in Section 3.1. We assume that the network evolves according to the ODE model described in this section. The rate constants associated with reactions are, as before, $[c_1\ c_2\ c_3\ c_4\ c_5\ c_6]^T = [11.25 \times 10^{-3}\ 0.25\ 0.5\ 1\ 2\ 1]^T$. We apply the modified version of Algorithm 1 described in this section to estimate the rate constants and evaluate the corresponding CRLB. Note that, in this example, the stoichiometry matrix is given by

$$S = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & -1 & 0 & 0 & 0 \\ -1 & 0 & 0 & 1 & 0 & 0 \\ -1 & 0 & 0 & 0 & 1 & -1 \end{bmatrix}, \tag{34}$$

and hence

$$\frac{\partial H(\overline{y}(i\Delta))}{\partial \mathbf{c}_m} = \begin{bmatrix} \dfrac{\partial \overline{y}_3(i\Delta)}{\partial \mathbf{c}_m}\overline{y}_4(i\Delta) + \overline{y}_3(i\Delta)\dfrac{\partial \overline{y}_4(i\Delta)}{\partial \mathbf{c}_m} \\ \dfrac{\partial \overline{y}_3(i\Delta)}{\partial \mathbf{c}_m} \\ \dfrac{\partial \overline{y}_2(i\Delta)}{\partial \mathbf{c}_m} \\ \dfrac{\partial \overline{y}_2(i\Delta)}{\partial \mathbf{c}_m} \\ \dfrac{\partial \overline{y}_2(i\Delta)}{\partial \mathbf{c}_m} \\ \dfrac{\partial \overline{y}_4(i\Delta)}{\partial \mathbf{c}_m} \end{bmatrix}. \tag{35}$$

Figure 3 shows the mean and standard error of inferring the reaction rates using the proposed estimator, shown

as a function of the variance of the observation noise (discretization time: $\Delta = 0.1$, the number of particles: $N_s = 10000$, the noise variance: $\sigma^2 = 1$). Several of the parameters are estimated very accurately (e.g., c_1, c_4), while others have relatively large mean-square-error (e.g., c_2, c_6). Figure 4 compares the estimation mean-square error with the corresponding CRLB, plotted as a function of the number of measurements N used for the estimation. As indicated in Figure 4, the estimator performs close to the CRLB for several of the parameters (e.g., c_2, c_3), while for other parameters there is room for improvement.

5. Conclusions

In this paper, we studied the problem of estimating reaction rates in a gene regulatory network modeled by a chemical Langevin equation, that is, a high-dimensional stochastic differential equation. We proposed a solution which employs a particle filtering algorithm with Markov Chain Monte Carlo move step. Extensive simulation studies demonstrated that the proposed technique requires less computational complexity to achieve performance comparable to previously proposed methods. Moreover, we considered the deterministic description of the average network dynamics based on an ordinary differential equation model. For this scenario, we computed an approximate Cramer-Rao lower bound on the mean-square error of the estimation and demonstrated that, for some of the parameters, the proposed particle filter can be nearly optimal. The computed CRLB is indicative of the number of data points (i.e., the number of experiments) required to achieve a desired accuracy of inferring reaction rates. Further studies are needed to enable near-CRLB performance in the scenario of estimating a large number of unknown parameters.

Acknowledgment

This work was supported in part by the National Science Foundation under Grant no. CCF-0845730.

References

[1] W. F. Loomis and P. W. Sternberg, "Genetic networks," *Science*, vol. 269, no. 5224, p. 649, 1995.

[2] D. Thieffry, "From global expression data to gene networks," *BioEssays*, vol. 21, no. 11, pp. 895–899, 1999.

[3] R. Albert, "Boolean Modeling of Genetic Regulatory Networks," in *Complex Networks*, Springer, New York, NY, USA, 2004.

[4] D. T. Gillespie, "Exact stochastic simulation of coupled chemical reactions," *Journal of Physical Chemistry*, vol. 81, no. 25, pp. 2340–2361, 1977.

[5] D. T. Gillespie, "A rigorous derivation of the chemical master equation," *Physica A*, vol. 188, no. 1–3, pp. 404–425, 1992.

[6] H. H. McAdams and A. Arkin, "Stochastic mechanisms in gene expression," *Proceedings of the National Academy of Sciences of the United States of America*, vol. 94, no. 3, pp. 814–819, 1997.

[7] T. Chen, H. L. He, and G. M. Church, "Modeling gene expression with differential equations," in *Proceedings of Pacific Symposium on Biocomputing*, pp. 29–40, 1999.

[8] F. Grognard, H. de Jong, and J.-L. Gouze, *Piecewise-Linear Models of Genetic Regulatory Networks: Theory and Examples*, Lecture Notes in Control and Information Sciences (LNCIS), Springer, New York, NY, USA, 2007.

[9] N. Friedman, M. Linial, I. Nachman, and D. Pe'er, "Using Bayesian networks to analyze expression data," *Journal of Computational Biology*, vol. 7, no. 3-4, pp. 601–620, 2000.

[10] D. Heckerman, "A tutorial on learning with Bayesian networks," in *Learning in Graphical Models*, Kluwer Academic Publishers, Dordrecht, The Netherlands, 1998.

[11] S. A. Kauffman, "Metabolic stability and epigenesis in randomly constructed genetic nets," *Journal of Theoretical Biology*, vol. 22, no. 3, pp. 437–467, 1969.

[12] X. Cai and X. Wang, "Stochastic modeling and simulation of gene networks," *IEEE Signal Processing Magazine*, vol. 24, no. 1, pp. 27–36, 2007.

[13] R. J. Boys, D. J. Wilkinson, and T. B. L. Kirkwood, "Bayesian inference for a discretely observed stochastic kinetic model," *Statistics and Computing*, vol. 18, no. 2, pp. 125–135, 2008.

[14] K.-C. Chen, T.-Y. Wang, H.-H. Tseng, C.-Y. F. Huang, and C.-Y. Kao, "A stochastic differential equation model for quantifying transcriptional regulatory network in Saccharomyces cerevisiae," *Bioinformatics*, vol. 21, no. 12, pp. 2883–2890, 2005.

[15] J. Berg, "Dynamics of gene expression and the regulatory inference problem," *Europhysics Letters*, vol. 82, no. 2, Article ID 28010, 2008.

[16] A. Benecke, "Gene regulatory network inference using out of equilibrium statistical mechanics," *HFSP Journal*, vol. 2, no. 4, pp. 183–188, 2008.

[17] A. Golightly and D. J. Wilkinson, "Bayesian sequential inference for stochastic kinetic biochemical network models," *Journal of Computational Biology*, vol. 13, no. 3, pp. 838–851, 2006.

[18] A. Golightly, *Bayesian inference for nonlinear multivariate diffusion processes*, Ph.D. thesis, Newcastle University, 2006.

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

[20] J. Goutsias, "Quasiequilibrium approximation of fast reaction kinetics in stochastic biochemical systems," *Journal of Chemical Physics*, vol. 122, no. 18, Article ID 184102, 15 pages, 2005.

[21] D. T. Gillespie, "Chemical Langevin equation," *Journal of Chemical Physics*, vol. 113, no. 1, pp. 297–306, 2000.

[22] Z. Li, M. R. Osborne, and T. Prvan, "Parameter estimation of ordinary differential equations," *IMA Journal of Numerical Analysis*, vol. 25, no. 2, pp. 264–285, 2005.

[23] H. Cramer, *Mathematical Models of Statistics*, Princeton University Press, Princeton, NJ, USA, 1946.

[24] H. Vikalo, B. Hassibi, and A. Hassibi, "Limits of performance of quantitative polymerase chain reaction systems," *IEEE Transactions on Information Theory*, vol. 56, no. 2, pp. 688–695, 2010.