EURASIP Journal on Bioinformatics and Systems Biology
a SpringerOpen Journal

RESEARCH Open Access

Approximate maximum likelihood estimation for stochastic chemical kinetics

Aleksandr Andreychenko, Linar Mikeev, David Spieler and Verena Wolf*

Abstract

Recent experimental imaging techniques are able to tag and count molecular populations in a living cell. From these data mathematical models are inferred and calibrated. If small populations are present, discrete-state stochastic models are widely-used to describe the discreteness and randomness of molecular interactions. Based on time-series data of the molecular populations, the corresponding stochastic reaction rate constants can be estimated. This procedure is computationally very challenging, since the underlying stochastic process has to be solved for different parameters in order to obtain optimal estimates. Here, we focus on the maximum likelihood method and estimate rate constants, initial populations and parameters representing measurement errors.

Introduction

During the last decade stochastic models of networks of chemical reactions have become very popular. The reason is that the assumption that chemical concentrations change deterministically and continuously in time is not always appropriate for cellular processes. In particular, if certain substances in the cell are present in small concentrations the resulting stochastic effects cannot be adequately described by deterministic models. In that case, discrete-state stochastic models are advantageous because they take into account the discrete random nature of chemical reactions. The theory of stochastic chemical kinetics provides a rigorously justified framework for the description of chemical reactions where the effects of molecular noise are taken into account [1]. It is based on discrete-state Markov processes that explicitly represent the reactions as state-transitions between population vectors. When the molecule numbers are large, the solution of the deterministic description of a reaction network and the mean of the corresponding stochastic model agree up to a small approximation error. If, however, species with small populations are involved, then only a stochastic description can provide probabilities of events of interest such as probabilities of switching between different expression states in gene regulatory networks or the distribution of gene expression products. Moreover, even the mean behavior of the stochastic model can largely deviate from the behavior of the deterministic model [2]. In such cases the parameters of the stochastic model rather then the parameters of the deterministic model have to be estimated [3-5].

Here, we consider noisy time series measurements of the system state as they are available from wet-lab experiments. Recent experimental imaging techniques such as high-resolution fluorescence microscopy can measure small molecule counts with measurement errors of less than one molecule [6]. We assume that the structure of the underlying reaction network is known but the stochastic reaction rate constants of the network are unknown parameters. Then we identify rate constants that maximize the likelihood of the time series data. Maximum likelihood estimators are the most popular estimators since they have desirable mathematical properties. Specifically, they become minimum variance unbiased estimators and are asymptotically normal as the sample size increases.

Our main contribution consists in devising an efficient algorithm for the numerical approximation of the likelihood and its derivatives w.r.t. the stochastic reaction rate constants. Furthermore, we show how similar techniques can be used to estimate the initial molecule numbers of a network as well as parameters related to the measurement error. We also present extensive experimental results that give insights about the identifiability of certain parameters. In particular, we consider a simple gene expression model and the identifiability of reaction rate constants w.r.t. varying observation interval lengths and

*Correspondence: wolf@cs.uni-saarland.de
Computer Science Department, Saarland University, 66123 Saarbrücken, Germany

varying numbers of time series. Moreover, for this system we investigate the identifiability of reaction rate constants if the state of the gene cannot be observed but only the number of mRNA molecules. For a more complex gene regulatory network, we present parameter estimation results where different combinations of proteins are observed. In this way we reason about the sensitivity of the estimation of certain parameters w.r.t. the protein types that are observed.

Previous parameter estimation techniques for stochastic models are based on Monte-Carlo sampling [3,5] because the discrete state space of the underlying model is typically infinite in several dimensions and a priori a reasonable truncation of the state space is not available. Other approaches are based on Bayesian inference which can be applied both to deterministic and stochastic models [7-9]. In particular, approximate Bayesian inference can serve as a way to distinguish among a set of competing models [10]. Moreover, in the context of Bayesian inference linear noise approximations have been used to overcome the problem of large discrete state spaces [11].

Our method is not based on sampling but directly calculates the likelihood using a dynamic truncation of the state space. More precisely, we first show that the computation of the likelihood is equivalent to the evaluation of a product of vectors and matrices. This product includes the transition probability matrix of the associated continuous-time Markov process, i.e., the solution of the Kolmogorov differential equations (KDEs), which can be seen as a matrix-version of the chemical master equation (CME). Solving the KDEs is infeasible because of the state space of the underlying Markov model is very large or even infinite. Therefore we propose an iterative approximation algorithm during which the state space is truncated in an on-the-fly fashion, that is, during a certain time interval we consider only those states that significantly contribute to the likelihood. This technique is based on ideas presented in [12], but here we additionally explain how the initial molecule numbers can be estimated and how an approximation of the standard deviation of the estimated parameters can be derived. Moreover, we provide more complex case studies and run extensive numerical experiments to assess the identifiability of certain parameters. In these experiments we assume that not all molecular populations can be observed and estimate parameters for different observation scenarios, i.e., we assume different numbers of observed cells and different observation interval lengths. We remark that this article is an extension of a previously published extended abstract [13].

The article is further organized as follows: After introducing the stochastic model in Section"Discrete-state stochastic model", we discuss the maximum likelihood method in Section "Parameter inference" and present our approximation method in Section "Numerical approximation algorithm". Finally, we report on experimental results for two reaction networks in Section "Numerical results".

Discrete-state stochastic model

According to Gillespie's theory of stochastic chemical kinetics, a well-stirred mixture of n molecular species in a volume with fixed size and fixed temperature can be represented as a continuous-time Markov chain $\{\mathbf{X}(t), t \geq 0\}$ [1]. The random vector $\mathbf{X}(t) = (X_1(t), \ldots, X_n(t))$ describes the chemical populations at time t, i.e., $X_i(t)$ is the number of molecules of type $i \in \{1, \ldots, n\}$ at time t. Thus, the state space of $\mathbf{X}$ is $\mathbb{Z}_+^n = \{0, 1, \ldots\}^n$. The state changes of $\mathbf{X}$ are triggered by the occurrences of chemical reactions, which are of m different types. For $j \in \{1, \ldots, m\}$ let $\mathbf{v}_j \in \mathbb{Z}^n$ be the nonzero *change vector* of the j-th reaction type. Thus, if $\mathbf{X}(t) = \mathbf{x}$ and the j-th reaction is possible in $\mathbf{x}$, then $\mathbf{X}(t + dt) = \mathbf{x} + \mathbf{v}_j$ is the state of the system after the occurrence of the j-th reaction within the infinitesimal time interval $[t, t + dt)$.

Each reaction type has an associated *propensity function*, denoted by $\alpha_1, \ldots, \alpha_m$, which is such that $\alpha_j(\mathbf{x}) \cdot dt$ is the probability that, given $\mathbf{X}(t) = \mathbf{x}$, one instance of the j-th reaction occurs within $[t, t + dt)$. The value $\alpha_j(\mathbf{x})$ is proportional to the number of distinct reactant combinations in state $\mathbf{x}$ and to the reaction rate constant c_j. The probability that a randomly selected pair of reactants collides and undergoes the j-th chemical reaction within $[t, t + dt)$ is then given by $c_j dt$. The value c_j depends on the volume and the temperature of the system as well as on the microphysical properties of the reactant species.

Example 1. *We consider the simple gene expression model described in* [4] *that involves three chemical species, namely* DNA_{ON}, DNA_{OFF}, *and mRNA, which are represented by the random variables* $X_1(t)$, $X_2(t)$, *and* $X_3(t)$, *respectively. The three possible reactions are* $DNA_{\mathrm{ON}} \rightarrow DNA_{OFF}$, $DNA_{OFF} \rightarrow DNA_{\mathrm{ON}}$, *and* $DNA_{\mathrm{ON}} \rightarrow DNA_{\mathrm{ON}} + mRNA$. *Thus,* $\mathbf{v}_1 = (-1, 1, 0)$, $\mathbf{v}_2 = (1, -1, 0)$, $\mathbf{v}_3 = (0, 0, 1)$. *For a state* $\mathbf{x} = (x_1, x_2, x_3)$, *the propensity functions are* $\alpha_1(\mathbf{x}) = c_1 \cdot x_1$, $\alpha_2(\mathbf{x}) = c_2 \cdot x_2$, *and* $\alpha_3(\mathbf{x}) = c_3 \cdot x_1$. *Note that given the initial state* $\mathbf{x} = (1, 0, 0)$, *at any time, either the DNA is active or not, i.e.* $x_1 = 0$ *and* $x_2 = 1$, *or* $x_1 = 1$ *and* $x_2 = 0$. *Moreover, the state space of the model is infinite in the third dimension. For a fixed time instant* $t > 0$, *no upper bound on the number of mRNA is known a priori. All states* $\mathbf{x}$ *with* $x_3 \in \mathbb{Z}_+$ *have positive probability if* $t > 0$ *but these probabilities will tend to zero as* $x_3 \rightarrow \infty$.

The CME

For a state $\mathbf{x} \in \mathbb{Z}_+^n$ and $t \geq 0$, let $p(\mathbf{x}, t)$ denote the probability $\Pr(\mathbf{X}(t) = \mathbf{x})$, i.e., the probability that the process is

in state $\mathbf{x}$ at time t. Furthermore, let $\mathbf{p}(t)$ be the row vector with entries $p(\mathbf{x}, t)$ where we assume a fixed enumeration of all possible states.

Given $\mathbf{v}_1, \ldots, \mathbf{v}_m$, $\alpha_1, \ldots, \alpha_m$, and some initial populations $\mathbf{x}(0) = (x_1(0), \ldots, x_n(0))$ with $P(\mathbf{X}(0) = \mathbf{x}(0)) = 1$, the Markov chain $\mathbf{X}$ is uniquely specified and its evolution is given by the CME

$$\frac{d}{dt}\mathbf{p}(t) = \mathbf{p}(t)Q, \tag{1}$$

where Q is the infinitesimal generator matrix of $\mathbf{X}$ with $Q(\mathbf{x}, \mathbf{y}) = \alpha_j(\mathbf{x})$ if $\mathbf{y} = \mathbf{x} + \mathbf{v}_j$ and reaction type j is possible in state $\mathbf{x}$. Note that, in order to simplify our presentation, we assume here that all vectors $\mathbf{v}_j$ are distinct. All remaining entries of Q are zero except for the diagonal entries which are equal to the negative row sum. The ordinary first-order differential equation in (1) is a direct consequence of the Kolmogorov forward equation but standard numerical solution techniques for systems of first-order linear equations cannot be applied to solve (1) because the number of nonzero entries in Q typically exceeds the available memory capacity for systems of realistic size. If the expected populations of all species remain small (at most a few hundreds) then the CME can be efficiently approximated using projection methods [14-16] or fast uniformization methods [17,18]. The idea of these methods is to avoid an exhaustive state space exploration and, depending on a certain time interval, restrict the analysis of the system to a subset of states.

We are interested in the partial derivatives of $\mathbf{p}(t)$ w.r.t. a certain parameter λ such as reaction rate constants c_j, $j \in \{1, \ldots, m\}$ or initial populations $x_i(0)$, $i \in \{1, \ldots, n\}$. Later, they will be used to maximize the likelihood of observations and to find optimal parameters. In order to explicitly indicate the dependence of $\mathbf{p}(t)$ on λ we may write $\mathbf{p}_\lambda(t)$ instead of $\mathbf{p}(t)$ and $p_\lambda(\mathbf{x}, t)$ instead of $p(\mathbf{x}, t)$. We define the row vector $\mathbf{s}_\lambda(t)$ as the derivative of $\mathbf{p}_\lambda(t)$ w.r.t. λ, i.e.,

$$\mathbf{s}_\lambda(t) = \frac{\partial \mathbf{p}_\lambda(t)}{\partial \lambda} = \lim_{\Delta \to 0} \frac{\mathbf{p}_{\lambda+\Delta}(t) - \mathbf{p}_\lambda(t)}{\Delta}.$$

We denote the entry in $\mathbf{s}_\lambda(t)$ that corresponds to state $\mathbf{x}$ by $s_\lambda(\mathbf{x}, t)$. Note that we use bold face for vectors. By (1), we find that $\mathbf{s}_\lambda(t)$ is the solution of the system of ODEs

$$\frac{d}{dt}\mathbf{s}_\lambda(t) = \mathbf{s}_\lambda(t)Q + \mathbf{p}_\lambda(t)\frac{\partial}{\partial \lambda}Q, \tag{2}$$

when choosing $\lambda = c_j$ for $j \in \{1, \ldots, m\}$. In this case, the initial condition is $s_\lambda(\mathbf{x}, 0) = 0$ for all $\mathbf{x}$ since $p(\mathbf{x}, 0)$ is independent of c_j. If the unknown parameter is the i-th initial population, i.e., $\lambda = x_i(0)$, then we get

$$\frac{d}{dt}\mathbf{s}_\lambda(t) = \mathbf{s}_\lambda(t)Q, \tag{3}$$

with initial condition $\mathbf{s}_\lambda(0) = \frac{\partial}{\partial \lambda}\mathbf{p}_\lambda(0)$ since Q is independent of $x_i(0)$. Similar ODEs can be derived for higher order derivatives of the CME.

Parameter inference

Following the notation in [4], we assume that observations of the reaction network are made at time instances $t_1, \ldots, t_R \in \mathbb{R}_{\geq 0}$ where $t_1 < \cdots < t_R$. Since it is unrealistic to assume that all species can be observed, we assume w.l.o.g. that the species are ordered such that we have observations of $X_1, \ldots, X_d$ for some fixed d with $1 \leq d \leq n$, i.e. $O_i(t_\ell)$ is the observed number of species i at time t_ℓ for $i \in \{1, \ldots, d\}$ and $\ell \in \{1, \ldots, R\}$. Let $\mathbf{O}(t_\ell) = (O_1(t_\ell), \ldots, O_d(t_\ell))$ be the corresponding vector of observations. Since these observations are typically subject to measurement errors, we assume that $O_i(t_\ell) = X_i(t_\ell) + \epsilon_i(t_\ell)$ where the error terms $\epsilon_i(t_\ell)$ are independent and identically normally distributed with mean zero and standard deviation σ. Note that $X_i(t_\ell)$ is the true population of the i-th species at time t_ℓ. Clearly, this implies that, conditional on $X_i(t_\ell)$, the random variable $O_i(t_\ell)$ is independent of all other observations as well as independent of the history of $\mathbf{X}$ before time t_ℓ.

We assume further that we do not know the values of the rate constants $\mathbf{c} = (c_1, \ldots, c_m)$ and our aim is to estimate these constants. Similarly, the initial populations $\mathbf{x}(0)$ and the exact standard deviation σ of the error terms are unknown and must be estimated. We remark that it is straightforward to extend the estimation framework such that a covariance matrix for a multivariate normal distribution of the error terms is estimated. In this way, different measurement errors of the species can be taken into account as well as dependencies between error terms.

Let f denote the joint density of $\mathbf{O}(t_1), \ldots, \mathbf{O}(t_R)$ and, by convenient abuse of notation, for a vector $\mathbf{x}_\ell = (x_1, \ldots, x_d)$ let $\mathbf{X}(t_\ell) = \mathbf{x}_\ell$ represent the event that $X_i(t_\ell) = x_i$ for $1 \leq i \leq d$. In other words, $\mathbf{X}(t_\ell) = \mathbf{x}_\ell$ means that the populations of the observed species at time t_ℓ equal the populations of vector $\mathbf{x}_\ell$. Note that this event corresponds to a set of states of the Markov process since d may be smaller than n. More precisely, $\Pr(\mathbf{X}(t_\ell) = \mathbf{x}_\ell) = \sum_{\mathbf{y}: y_i = x_i, i \leq d} p(\mathbf{y}, t_\ell)$. Now the likelihood of the observation sequence $\mathbf{O}(t_1), \ldots, \mathbf{O}(t_R)$ is given by

$$\begin{aligned}\mathcal{L} &= f\left(\mathbf{O}(t_1), \ldots, \mathbf{O}(t_R)\right) \\ &= \sum_{\mathbf{x}_1} \cdots \sum_{\mathbf{x}_R} f\left(\mathbf{O}(t_1), \ldots, \mathbf{O}(t_R)\right) \mid \\ &\qquad\qquad \mathbf{X}(t_1) = \mathbf{x}_1, \ldots, \mathbf{X}(t_R) = \mathbf{x}_R) \\ &\quad \Pr(\mathbf{X}(t_1) = \mathbf{x}_1, \ldots, \mathbf{X}(t_R) = \mathbf{x}_R).\end{aligned} \tag{4}$$

Note that $\mathcal{L}$ depends on the chosen rate parameters $\mathbf{c}$ and the initial populations $\mathbf{x}(0)$ since the probability measure $\Pr(\cdot)$ does. Furthermore, $\mathcal{L}$ depends on σ since the density f does. When necessary, we will make this dependence explicit by writing $\mathcal{L}(\mathbf{x}(0), \mathbf{c}, \sigma)$ instead of $\mathcal{L}$. We now

seek constants $\mathbf{c}^*$, initial populations $\mathbf{x}(0)$ and a standard deviation σ^* such that

$$\mathcal{L}(\mathbf{x}(0)^*, \mathbf{c}^*, \sigma^*) = \max_{\mathbf{x}(0),\sigma,\mathbf{c}} \mathcal{L}(\mathbf{x}(0), \mathbf{c}, \sigma) \tag{5}$$

where the maximum is taken over all $\sigma > 0$ and vectors $\mathbf{x}(0)$, $\mathbf{c}$ with all components strictly positive. This optimization problem is known as the maximum likelihood problem [19]. Note that $\mathbf{x}(0)^*$, $\mathbf{c}^*$ and σ^* are random variables because they depend on the (random) observations $\mathbf{O}(t_1), \ldots, \mathbf{O}(t_R)$.

If more than one sequence of observations is made, then the corresponding likelihood is the product of the likelihoods of all individual sequences. More precisely, if $\mathbf{O}^k(t_l)$ is the k-th observation that has been observed at time instant t_l where $k \in \{1, \ldots, K\}$, then we define $\mathcal{L}_k(\mathbf{x}(0), \mathbf{c}, \sigma)$ as the probability to observe $\mathbf{O}^k(t_1), \ldots, \mathbf{O}^k(t_R)$ and maximize

$$\prod_{k=1}^{K} \mathcal{L}_k(\mathbf{x}(0), \mathbf{c}, \sigma). \tag{6}$$

In what follows, we concentrate on expressions for $\mathcal{L}_k(\mathbf{x}(0), \mathbf{c}, \sigma)$ and $\frac{\partial}{\partial c_j}\mathcal{L}_k(\mathbf{x}(0), \mathbf{c}, \sigma)$. We first assume $K = 1$ and drop index k. We consider the case $K > 1$ later. In (4) we sum over all population vectors $\mathbf{x}_1, \ldots, \mathbf{x}_R$ of dimension d such that $\Pr(\mathbf{X}(t_\ell) = \mathbf{x}_\ell, 1 \le \ell \le R) > 0$. Since $\mathbf{X}$ has a large or even infinite state space, it is computationally infeasible to explore all possible sequences. In Section "Numerical approximation algorithm" we propose an algorithm to approximate the likelihoods and their derivatives by dynamically truncating the state space and using the fact that (4) can be written as a product of vectors and matrices. Let ϕ_σ be the density of the normal distribution with mean zero and standard deviation σ. Then

$$\begin{aligned} &f(\mathbf{O}(t_1), \ldots, \mathbf{O}(t_R) \mid \mathbf{X}(t_1) = \mathbf{x}_1, \ldots, \mathbf{X}(t_R) = \mathbf{x}_R) \\ &= \prod_{\ell=1}^{R} \prod_{i=1}^{d} f(O_i(t_\ell) \mid X_i(t_\ell) = x_{i\ell}) \\ &= \prod_{\ell=1}^{R} \prod_{i=1}^{d} \phi_\sigma(O_i(t_\ell) - x_{i\ell}), \end{aligned}$$

where $\mathbf{x}_\ell = (x_{1\ell}, \ldots, x_{d\ell})$. If we write $w(\mathbf{x}_\ell)$ for $\prod_{i=1}^{d} \phi_\sigma(O_i(t_\ell) - x_{i\ell})$, then the sequence $\mathbf{x}_1, \ldots, \mathbf{x}_R$ has "weight" $\prod_{\ell=1}^{R} w(\mathbf{x}_\ell)$ and, thus,

$$\mathcal{L} = \sum_{\mathbf{x}_1} \cdots \sum_{\mathbf{x}_R} \Pr(\mathbf{X}(t_1) = \mathbf{x}_1, \ldots, \mathbf{X}(t_R) = \mathbf{x}_R) \prod_{\ell=1}^{R} w(\mathbf{x}_\ell). \tag{7}$$

Moreover, for the probability of the sequence $\mathbf{x}_1, \ldots, \mathbf{x}_R$ we have

$$\Pr(\mathbf{X}(t_1) = \mathbf{x}_1, \ldots, \mathbf{X}(t_R) = \mathbf{x}_R) = p(\mathbf{x}_1, t_1) P_2(\mathbf{x}_1, \mathbf{x}_2) \ldots P_R(\mathbf{x}_{R-1}, \mathbf{x}_R)$$

where $P_\ell(\mathbf{x}, \mathbf{y}) = \Pr\big(\mathbf{X}(t_\ell) = \mathbf{y} \mid \mathbf{X}(t_{\ell-1}) = \mathbf{x}\big)$ for d-dimensional population vectors $\mathbf{x}$ and $\mathbf{y}$. Hence, (7) can be written as

$$\begin{aligned} \mathcal{L} = &\sum_{\mathbf{x}_1} p(\mathbf{x}_1, t_1) w(\mathbf{x}_1) \sum_{\mathbf{x}_2} P_2(\mathbf{x}_1, \mathbf{x}_2) w(\mathbf{x}_2) \ldots \\ &\sum_{\mathbf{x}_R} P_R(\mathbf{x}_{R-1}, \mathbf{x}_R) w(\mathbf{x}_R). \end{aligned} \tag{8}$$

Assume that $d = n$ and let P_ℓ be the matrix with entries $P_\ell(\mathbf{x}, \mathbf{y})$ for all possible states $\mathbf{x}, \mathbf{y}$. Note that P_ℓ is the transition probability matrix of $\mathbf{X}$ for time step $t_\ell - t_{\ell-1}$ and thus the general solution $e^{Q(t_\ell - t_{\ell-1})}$ of the Kolmogorov forward and backward differential equations

$$\frac{d}{dt}P_\ell = QP_\ell, \qquad \frac{d}{dt}P_\ell = P_\ell Q.$$

In this case, using $\mathbf{p}(t_1) = \mathbf{p}(t_0)P_1$ with $t_0 = 0$, we can write (8) in matrix-vector form as

$$\mathcal{L} = \mathbf{p}(t_0) P_1 W_1 P_2 W_2 \ldots P_R W_R \mathbf{e}. \tag{9}$$

Here, $\mathbf{e}$ is the vector with all entries equal to one and W_ℓ is a diagonal matrix whose diagonal entries are all equal to $w(\mathbf{x}_\ell)$ with $\ell \in \{1, \ldots, R\}$, where W_ℓ is of the same size as P_ℓ.

If $d < n$, then we still have the same matrix-vector product as in (9), but define the weight $w(\mathbf{x})$ of an n-dimensional population vector as

$$w(x_1, \ldots, x_n) = \prod_{i=1}^{d} \phi_\sigma(O_i(t_\ell) - x_i),$$

i.e. the populations of the unobserved species have no influence on the weight.

Since it is in general not possible to analytically obtain parameters that maximize $\mathcal{L}$, we use numerical optimization techniques to find $\mathbf{c}^*$, $\mathbf{x}(0)^*$ and σ^*. Typically, such techniques iterate over values of $\mathbf{c}$, $\mathbf{x}(0)$ and σ and increase the likelihood $\mathcal{L}(\mathbf{c}, \sigma)$ by following the gradient. Therefore, we need to calculate the derivatives $\frac{\partial}{\partial c_j}\mathcal{L}$, $\frac{\partial}{\partial x_i(0)}\mathcal{L}$ and $\frac{\partial}{\partial \sigma}\mathcal{L}$. For $\frac{\partial}{\partial c_j}\mathcal{L}$ we obtain

$$\begin{aligned} \frac{\partial}{\partial c_j}\mathcal{L} &= \frac{\partial}{\partial c_j}\left(\mathbf{p}(t_0) P_1 W_1 P_2 W_2 \ldots P_R W_R \mathbf{e}\right) \\ &= \mathbf{p}(t_0) \left(\sum_{\ell=1}^{R} \left(\frac{\partial}{\partial c_j} P_\ell \right) W_\ell \prod_{\ell' \neq \ell} P_{\ell'} W_{\ell'} \right) \mathbf{e}. \end{aligned} \tag{10}$$

The derivative of $\mathcal{L}$ w.r.t. $x_i(0)$ and σ is derived analogously. The only difference is that $\mathbf{p}(t_0)$ is dependent on $x_i(0)$ and $P_1, \ldots, P_R$ are independent of σ but $W_1, \ldots, W_R$

depend on σ. It is also important to note that expressions for partial derivatives of second order can be derived in a similar way. These derivatives can then be used for an efficient gradient-based local optimization.

For $K > 1$ observation sequences we can maximize the log-likelihood

$$\log \prod_{k=1}^{K} \mathcal{L}_k = \sum_{k=1}^{K} \log \mathcal{L}_k, \tag{11}$$

instead of the likelihood in (6). Note that the derivatives are then given by

$$\frac{\partial}{\partial \lambda} \sum_{k=1}^{K} \log \mathcal{L}_k = \sum_{k=1}^{K} \frac{\frac{\partial}{\partial \lambda} \mathcal{L}_k}{\mathcal{L}_k}, \tag{12}$$

where λ is c_j, $x_i(0)$ or σ. It is also important to note that only the weights $w(\mathbf{x}_\ell)$ depend on k, that is, on the observed sequence $\mathbf{O}^k(t_1), \ldots, \mathbf{O}^k(t_R)$. Thus, when we compute $\mathcal{L}_k$ based on (9) we use for all k the same transition matrices $P_1, \ldots, P_R$ and the same initial conditions $\mathbf{p}(t_0)$, but possibly different matrices $W_1, \ldots, W_R$.

Numerical approximation algorithm

In this section, we focus on the numerical approximation of the likelihood and the corresponding derivatives. Our algorithm calculates an approximation of the likelihood based on (9) by traversing the matrix-vector product from the left to the right. The main idea behind the algorithm is that instead of explicitly computing the matrices P_ℓ, we express the vector-matrix product $\mathbf{u}(t_{\ell-1})P_\ell$ as a system of ODEs similar to the CME (cf. Equation (1)). Note that even though P_ℓ is sparse the number of states may be very large or infinite, in which case we cannot compute P_ℓ explicitly. Let $\mathbf{u}(t_0), \ldots, \mathbf{u}(t_R)$ be row vectors that are obtained during the iteration over time points $t_0, \ldots, t_R$, that is, we define $\mathcal{L}$ recursively as $\mathcal{L} = \mathbf{u}(t_R)\mathbf{e}$ with $\mathbf{u}(t_0) = \mathbf{p}(t_0)$ and

$$\mathbf{u}(t_\ell) = \mathbf{u}(t_{\ell-1})P_\ell W_\ell \quad \text{for all } 1 \le \ell \le R,$$

where $t_0 = 0$. We solve R systems of ODEs

$$\frac{d}{dt}\tilde{\mathbf{u}}(t) = \tilde{\mathbf{u}}(t)Q \tag{13}$$

with initial condition $\tilde{\mathbf{u}}(t_{\ell-1}) = \mathbf{u}(t_{\ell-1})$ for the time interval $[\,t_{\ell-1}, t_\ell)$ where $\ell \in \{1, \ldots, R\}$. After solving the ℓ-th system of ODEs we set $\mathbf{u}(t_\ell) = \tilde{\mathbf{u}}(t_\ell)W_\ell$ and finally compute $\mathcal{L} = \mathbf{u}(t_R)\mathbf{e}$. We remark that this is the same as solving the CME for different initial conditions and due to the largeness problem of the state space we use the dynamic truncation of the state space that we proposed in previous work [17]. The idea is to consider only the most relevant equations of the system (13), i.e., the equations that correspond to those states $\mathbf{x}$ where the relative contribution $\tilde{u}(\mathbf{x}, t)/(\tilde{\mathbf{u}}(t_\ell)\mathbf{e})$ is greater than a threshold δ. Since during the integration the contribution of a state might increase or decrease we add/remove equations on-the-fly depending on the current contribution of the corresponding state. Note that the structure of the CME allows us to determine in a simple way which states will become relevant in the next integration step. For a small time step of length h we know that the probability being moved from state $\mathbf{x} - \mathbf{v}_j$ to $\mathbf{x}$ is approximately $\alpha_j(\mathbf{x} - \mathbf{v}_j)h$. Thus, we can simply check whether a state that receives a certain probability inflow receives more than the threshold. In this case we consider the corresponding equation in (13). Otherwise, if a state does not receive enough probability inflow, we do not consider it in (13). For more details on this technique we refer to [17].

Since the vectors $\tilde{\mathbf{u}}(t_\ell)$ do not sum up to one, we scale all entries by multiplication with $1/(\tilde{\mathbf{u}}(t_\ell)\mathbf{e})$. This simplifies the truncation of the state space using the significance threshold δ since after scaling it can be interpreted as a probability. In order to obtain the correct (unscaled) likelihood, we compute $\mathcal{L}$ as $\mathcal{L} = \prod_{\ell=1}^{R} \tilde{\mathbf{u}}(t_\ell)\mathbf{e}$. For our numerical implementation we used a threshold of $\delta = 10^{-15}$ and handle the derivatives of $\mathcal{L}$ in a similar way. To shorten our presentation, we only consider the derivative $\frac{\partial}{\partial c_j}\mathcal{L}$ in the sequel of the article. Iterative schemes for $\frac{\partial}{\partial \sigma}\mathcal{L}$ and $\frac{\partial}{\partial x_i(0)}\mathcal{L}$ are derived analogously. From (10) we obtain $\frac{\partial}{\partial c_j}\mathcal{L} = \mathbf{u}_j(t_R)\mathbf{e}$ with $\mathbf{u}_j(t_0) = \mathbf{0}$ and

$$\mathbf{u}_j(t_\ell) = (\mathbf{u}_j(t_{\ell-1})P_\ell + \mathbf{u}(t_{\ell-1})\frac{\partial}{\partial c_j}P_\ell)W_\ell \quad \text{for all } 1 \le \ell \le R,$$

where $\mathbf{0}$ is the vector with all entries zero. Thus, during the solution of the ℓ-th ODE in (13) we simultaneously solve

$$\frac{d}{dt}\tilde{\mathbf{u}}_j(t) = \tilde{\mathbf{u}}_j(t)Q + \tilde{\mathbf{u}}(t)\frac{\partial}{\partial c_j}Q \tag{14}$$

with initial condition $\tilde{\mathbf{u}}_j(t_{\ell-1}) = \mathbf{u}_j(t_{\ell-1})$ for the time interval $[\,t_{\ell-1}, t_\ell)$. As above, we set $\mathbf{u}_j(t_\ell) = \tilde{\mathbf{u}}_j(t_\ell)W_\ell$ and obtain $\frac{\partial}{\partial c_j}\mathcal{L}$ as $\mathbf{u}_j(t_R)\mathbf{e}$.

Solving (13) and (14) simultaneously is equivalent to the computation of the partial derivatives in (2) with different initial conditions. Numerical experiments show that the approximation errors of the likelihood and its derivatives are of the same order of magnitude as those of the transient probabilities and their derivatives. For instance, for a finite-state enzymatic reaction system that is small enough to be solved without truncation we found that the maximum absolute error in the approximations of the vectors $\mathbf{p}(t)$ and $\mathbf{s}_\lambda(t)$ is 10^{-8} if the truncation threshold is $\delta = 10^{-15}$ (details not shown).

In the case of K observation sequences we repeat the above algorithm in order to sequentially compute $\mathcal{L}_k$ for $k \in \{1, \ldots, K\}$. We exploit (11) and (12) to compute the total log-likelihood and its derivatives as a sum of individual terms. In a similar way, second derivatives can be

approximated. Obviously, it is possible to parallelize the algorithm by computing $\mathcal{L}_k$ in parallel for all k.

In order to find values for which the likelihood becomes maximal, global optimization techniques can be applied. Those techniques usually use a heuristic for different initial values of the parameters and then follow the gradient to find local optima of the likelihood. In this step the algorithm proposed above is used since it approximates the gradient of the likelihood. The approximated global optimum is then chosen as the minimum/maximum of the local optima, i.e, we determine those values of the parameters that give the largest likelihood. Clearly, this is an approximation and we cannot guarantee that the global optimum was found. Note that this would also be the case if we could compute the exact likelihood. If, however, a good heuristic for the starting points is chosen and the number of starting points is large, then it is likely that the approximation is accurate. Moreover, since we have approximated the second derivative of the log-likelihood, we can compute the entries of the Fisher information matrix and use this to approximate the standard deviation of the estimated parameters, i.e., we consider the square root of the diagonal entries of the inverse of a matrix H which is the Hessian matrix of the negative log-likelihood. Assuming that the second derivative of the log-likelihood is computed exactly, these entries asymptotically tend to the standard deviations of the estimated parameters.

We remark that the approximation proposed above becomes unfeasible if the reaction network contains species with high molecule numbers since in this case the number of states that have to be considered is very large. A numerical approximation of the likelihood is, as the solution of the CME, only possible if the expected populations of all species remain small (at most a few hundreds) and if the dimension of the process is not too large. Moreover, if many parameters have to be estimated, the search space of the optimization problem may become unfeasibly large. It is however straightforward to parallelize local optimizations starting from different initial point.

Numerical results

In this section we present numerical results of our parameter estimation algorithm applied to two models, the simple gene expression in Example 1 and a multi-attractor model. The corresponding SBML files are provided as Additional files 1 and 2. For both models, we generated time series data using Monte-Carlo simulation where we added white noise to represent measurement errors, i.e. we added random terms to the populations that follow a normal distribution with mean zero and a standard deviation of σ. Our algorithm for the approximation of the likelihood is implemented in C++ and linked to MATLAB's optimization toolbox [20] which we use to minimize the negative log-likelihood. The global optimization method (Matlab's GlobalSearch [21]) uses a scatter-search algorithm to generate a set of trial points (potential starting points) and heuristically decides when to perform a local optimization. We ran our experiments on an Intel Core i7 at 2.8 GHz with 8 GB main memory.

Simple gene expression

For our first model, the simple gene expression as introduced in Example 1, we chose the same parameters as Reinker et al.[4] multiplied by a factor of 10, i.e., $\mathbf{c} = (0.270, 1.667, 4.0)$ and as the initial condition we have ten mRNA molecules and the DNA is inactive. We generated K observation sequences of length $T = 100.0$ and observed all species at R equidistant observation time points. We added white noise with standard deviation $\sigma = 1.0$ to the observed mRNA molecule numbers at each observation time point. For the case $K = 5, R = 100$ we plot the generated observation sequences in Figure 1. We estimated the reaction rate constants, the initial molecule numbers, and the parameter σ of the measurement errors for the case $K = 5, R = 100$ where we chose the interval $[10^{-5}, 10^{3}]$ as a constraint for the rate constants, the interval $[0, 100]$ for the initial number of mRNA molecules and $[0, 5]$ for σ. Since we use a global optimization method, the running time of our method depends on the number of trial points generated by GlobalSearch. In Figure 2 we plot the trial points (red points) and local optimization runs (differently colored lines) for the case of 10 (a), 100 (b) and 1000 (c) trial points. The intersection of the dashed blue lines represents the location of the original parameters. In the case of ten trial points, the running time was about one minute and the local optimization was performed only once. In the case of 100 and 1000 trial points, the running times were about 22 min and 1.9 h, respectively and several local optimization runs converged in nearly the same point. However, we remark that in general the landscape of the target function might have multiple local minima and require more trial points resulting in longer running times.

We ran experiments for varying values of K and R ($K, R \in \{1, 2, 5, 10, 20, 50, 100\}$) to get insights whether for this network it is more advantageous to have many observation sequences with long observation intervals or few observation sequences with a short time between two successive observations. In addition, we ran the same experiments with the restriction that only the number of mRNA molecules was observable but not the state of the gene. In both cases we approximated the standard deviations of our estimators as a measure of quality by repeating our estimation procedure 100 times and by the Fisher information matrix as explained at the end of the previous section. We used 100 trial points for the global optimization procedure and chose tighter constraints than above

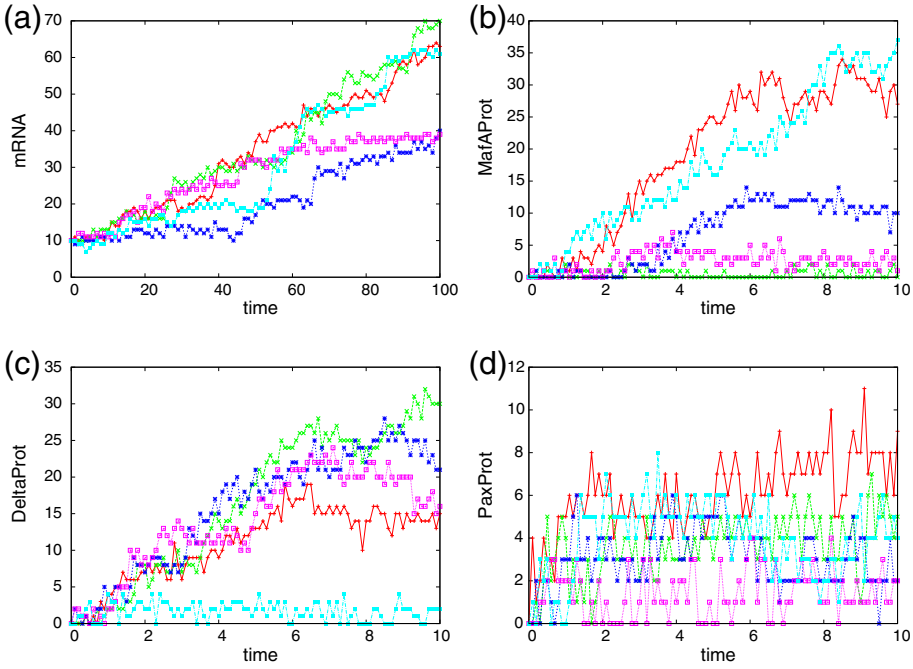

Figure 1 Time series data. Generated observation sequences for the gene expression **(a)** and multi-attractor **(b)**–**(d)** models. Each plot shows $K = 5$ sequences with $R = 100$ time points.

for the rate constants ($[0.01, 1]$ for c_1 and $[0.1, 10]$ for c_2, c_3) to have a convenient total running time.

The results are depicted in Figure 3 for the fully observable system and in Figure 4 for the restricted system, where the state of the gene was not visible. In these figures we present the estimations of the parameters c_1, c_2, c_3, σ, and an estimation of the initial condition, i.e. the number of mRNA molecules at time point $t = 0$. Moreover, we give the total running time of the procedure (Figures 3f and 4f). Our results are plotted as a gray landscape for all combinations of K and R. The estimates are bounded by a red grid enclosing an environment of one standard deviation around the respective average over all 100 estimates that we approximated. The real value of the parameter is indicated by a dotted blue rectangle.

At first, we remark that neither the quality of the estimation nor the running time of our algorithm is significantly dependent on whether we observe the state of the gene in addition to the mRNA level or not. Moreover, concerning the estimation of all of the parameters, one can witness that the estimates converge more quickly against the real values along the K axis than the R axis and also the standard deviations decrease faster. Consequently, at least for the gene expression model, it is more advantageous to increase the number of observation sequences, than the number of measurements per sequence. For example, $K = 100$ sequences with only one observation each already provide enough information to estimate c_1 up to a relative error of around 2.1%. Unfortunately, in this case the computation time is the highest since we have to compute K individual likelihoods (one for each observation sequence). Moreover, if R is small then the truncation of the state space is less efficient. The reason is that we have to integrate for a long time until we multiply with the weight matrix W_ℓ. After this multiplication we decide which states contribute significantly to the likelihood and which states are neglected. We can, however, trade off accuracy against running time by varying K.

For the measurement noise parameter σ we see that it is more advantageous to increase R. Even five observation sequences with a high number of observations per sequence ($R = 100$) suffice to estimate the noise up to a relative error of around 10.2%. For the estimation of the initial conditions, both K and R seem to play an equally important role.

The standard deviations of the estimators give information about the accuracy of the estimation. In order to approximate the standard deviation we used statistics over 100 repeated experiments. In a realistic setting one would rather use the Fisher information matrix to approximate the standard deviation of the estimators since it is in most cases difficult to observe $100 \cdot K$ observation sequences of a real system. Therefore we compare the results of one experiment with K observation sequences and standard deviations approximated using the Fisher information matrix to the case where the experiment is

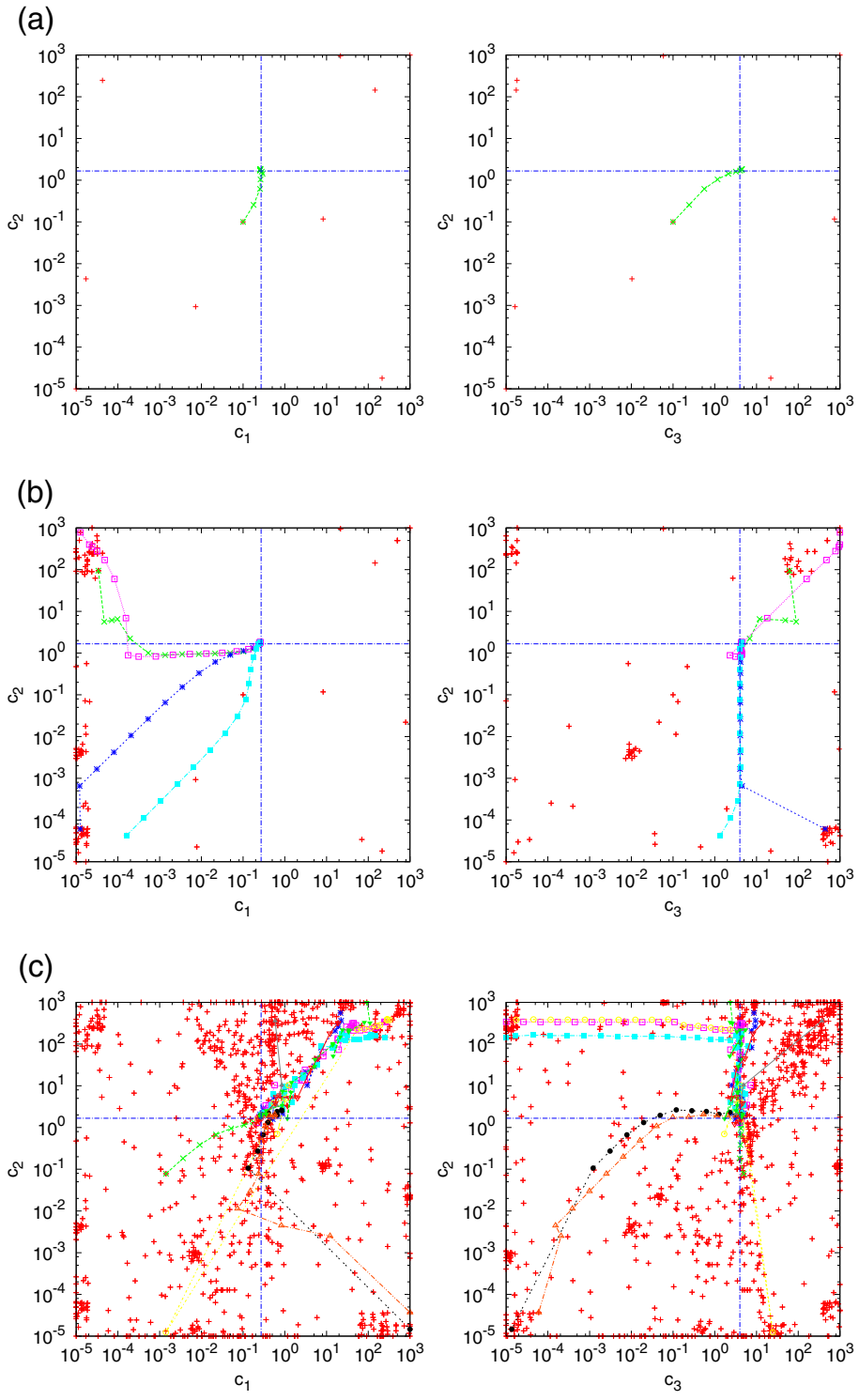

Figure 2 Start points and gradient convergence of the optimization procedure for the gene expression example: Red pluses show the potential start points. We use 10, 100, and 1000 start points in case **(a)**, **(b)**, and **(c)**, respectively. The markers that are connected by lines show the iterative steps of the gradient convergence while the dashed blue line shows the true values of the parameters. We chose $K = 5, R = 100$ and assume that the parameters are in the range $[10^{-5}, 10^{3}]$.

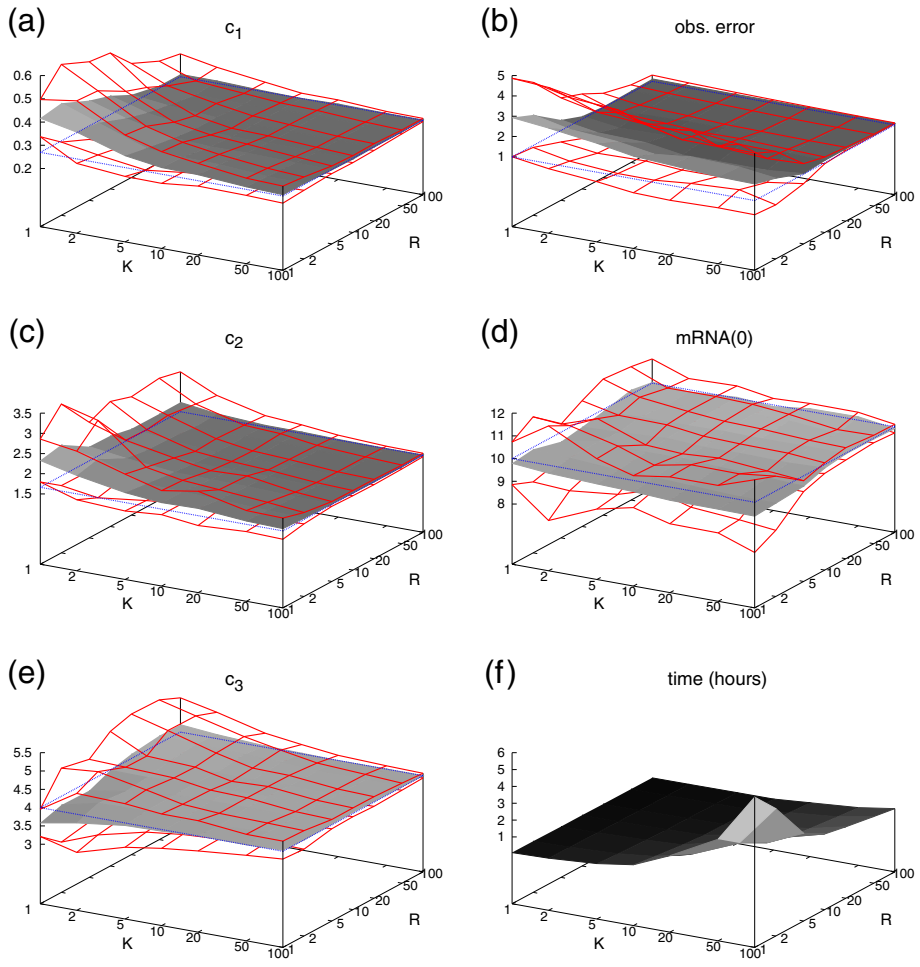

Figure 3 Results of the gene expression case study with observable gene state. The dotted blue rectangle gives the true value of c_1, c_2, c_3, σ (obs. error), and mRNA(0). The red grid corresponds to the approximated standard deviation of the estimators.

repeated 100 times. The results for varying values of K and R are given in Table 1 We observe that the approximation using the Fisher information matrix is in most cases close to the approximation based on 100 repetitions as long as K and R are not too small. This comes from the fact that the Fisher information matrix converges to the true standard deviation as the sample size increases.

Multi-attractor model

Our final example is a part of the multi-attractor model considered by Zhou et al. [22]. It consists of the three genes *MafA*, *Pax4*, and δ-gene, which interact with each other as illustrated in Figure 5. The corresponding proteins bind to specific promoter regions on the DNA and (de-)activate the genes. The reaction network has 2^3 different gene states, also called modes, since each gene can be on or off. It is infinite in three dimensions since for the proteins there is no fixed upper bound. The edges between the nodes in Figure 5 show whether the protein of a specific gene can bind to the promoter region of another gene. Moreover, edges with normal arrow heads correspond to binding without inhibition while the edges with line heads show inhibition.

We list all 24 reactions in Table 2 For simplicity we first assume that there is a common rate constant for all protein production reactions (p), for all protein degradations (d), binding (b), and unbinding (u) reactions. We further assume that initially all genes are active and no proteins are present. For the rate constants we chose $\mathbf{c} = (p, d, b, u) = (5.0, 0.1, 1.0, 1.0)$ and generated $K \in \{1, 5\}$ sample paths of length $T = 10.0$. We added normally distributed noise with zero mean and standard deviation $\sigma = 1.0$ to the protein levels at each of the $R = 100$ observation time points. Plots of the generated observation sequences are presented in Figure 1 b–d for the case $K = 5$. For the global optimization we used ten trial points. We chose the interval $[0.1, 10]$ as a constraint for

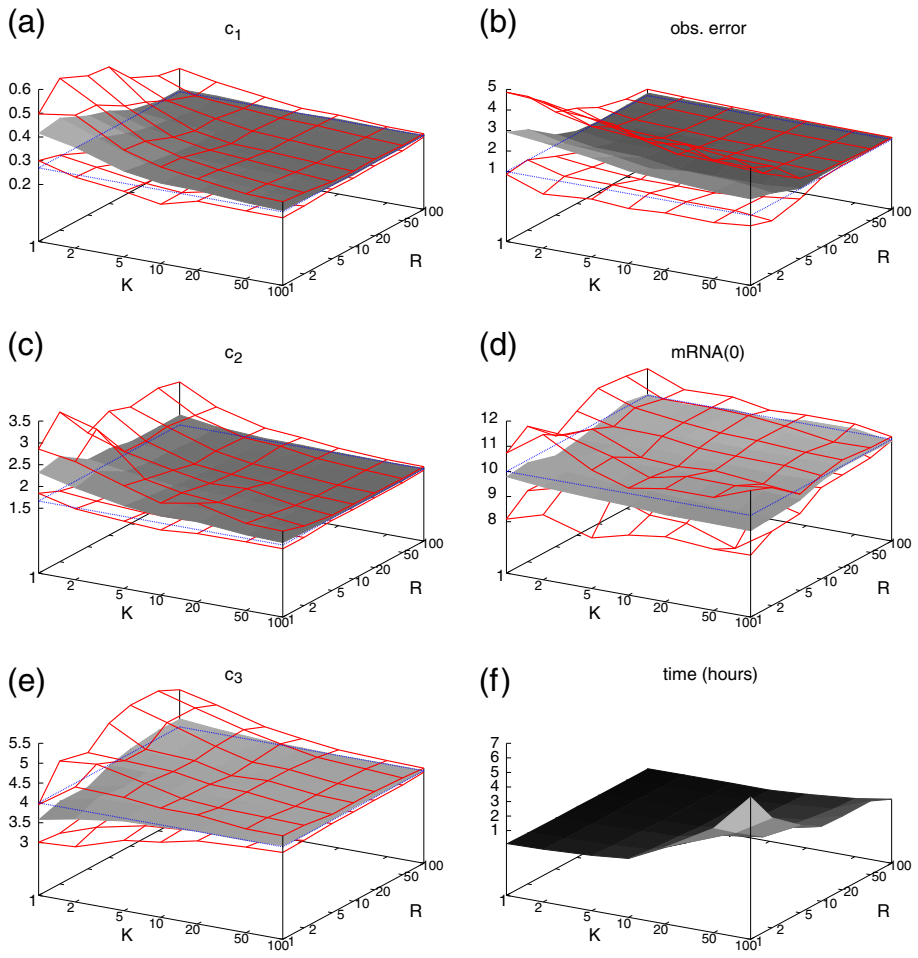

Figure 4 Results of the gene expression case study (as in Figure 3) but the state of the gene is not observed.

the rate constants p, b, u and the interval $[0.01, 1]$ for d. We estimated the parameters for all $2^3 - 1 = 7$ possibilities of observing or not observing the three protein numbers where at least one of them had to be observable. In addition we repeated the parameter estimation for the fully observable system where in addition to the three proteins also the state of the genes was observed. The results are depicted in Figure 6 where the x-axis of the plots refers to the observed proteins. For instance, the third entry on the x-axis of the plot in Figure 6 a shows the result of the estimation of parameter $c_1 = 5$ based on observation sequences where only the molecule numbers of the proteins MafAProt and DeltaProt were observed. For this case study, we used the Fisher information matrix

Table 1 Different approximations of the standard deviations of the estimators

Method	K	R	c_1	c_2	c_3	σ	mRNA(0)
Fisher inf. matrix	10	10	0.0545104	0.561963	0.935324	0.364339	0.639471
100 experiments			0.0358142	0.198700	0.262223	0.392884	0.490305
Fisher inf. matrix	20	20	0.0324508	0.299487	0.451476	0.174095	0.594820
100 experiments			0.0304157	0.167431	0.287471	0.134506	0.436059
Fisher inf. matrix	50	50	0.0139185	0.110709	0.152229	0.0440282	0.238033
100 experiments			0.0140331	0.078516	0.146232	0.0353837	0.183888
Fisher inf. matrix	100	100	0.00866066	0.0548249	0.0728129	0.0182564	0.208469
100 experiments			0.00691956	0.0430123	0.0641821	0.0217544	0.187968

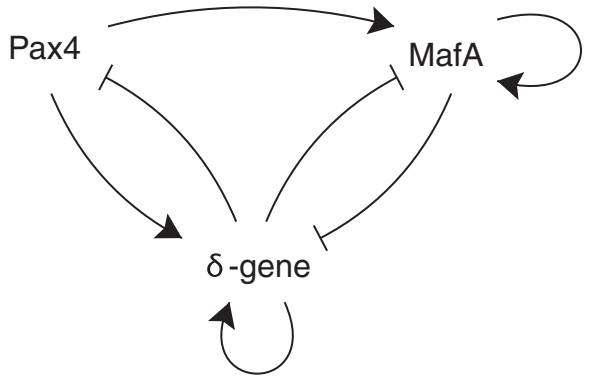

Figure 5 Illustration of the multi-attractor model.

to approximate the standard deviations of our estimators, plotted as bars in Figure 6 with the estimated parameter as midpoint. The fully observable case is labelled by "full".

We observe in Figure 6 that as expected the accuracy of the estimation and the running time of our algorithm is best when we have full observability of the system and gets worse with an increasing number of unobservable species. Still the estimation quality is very high when five observation sequences are provided for almost all combinations and parameters. When only one observation sequence is given ($K = 1$), the parameter estimation becomes unreliable and time consuming. This comes from the fact that the quality of the approximation highly depends on the generated observation sequence. It is possible to get much better and faster approximations with a single observation sequence. However, we did not optimize our results but generated one random observation sequence and ran our estimation procedure once based on this.

Recall that we chose common parameters p, d, b, u for production, degradation, and (un-)binding for all three protein species. Next we "decouple" the binding rates and estimate the binding rate of each protein independently. We illustrate our results in Figure 7. Again, in case of a single observation sequence ($K = 1$) the estimation is unreliable in most cases. If the true value of the parameter is unknown, then the high standard deviation shows that more information (more observation sequences) is necessary to estimate the parameter. In order to estimate the binding rate of PaxProt, we see that observing MafAProt yields the best result while for the binding rate of MafAProt observing PaxProt is best. Only for the binding rate of DeltaProt, the best results are obtained when the corresponding protein (DeltaProt) is observed. The running times of the estimation procedure are between 10 and 80 h, usually increase with K and depend on the observation sequences.

In Table 3 we list the results of estimating the production rate 5.0 in the multi-attractor model where we chose $R = 100$. More precisely, we estimated the production rate of each protein independently when the other two proteins were observed. Since the population of the PaxProt is significantly smaller than the populations of the other two proteins, its production rate is more difficult to estimate. The production rate of MafAProt is accurately estimated even if only a single observation sequence is considered. For estimating the production rate of DeltaProt, $K = 5$ observation sequences are necessary to get an accurate result.

Finally, we remark that for the multi-attractor model it seems difficult to predict whether for a given parameter the observation of a certain set of proteins yields a good accuracy or not. It can, however, be hypothesized that, if we want to accurately estimate the rate constant of a certain chemical reaction, then we should observe as many of the involved species as possible. Moreover, it is reasonable that constants of reactions that occur less often are more difficult to estimate (such as the production of PaxProt). In such a case more observation sequences are necessary to provide reliable information about the speed of the reaction.

Table 2 Chemical reactions of the multi-attractor model

PaxDna	$\xrightarrow{p}$	PaxDna + PaxProt
PaxProt	$\xrightarrow{d}$	∅
PaxDna + DeltaProt	$\xrightarrow{b}$	PaxDnaDeltaProt
PaxDnaDeltaProt	$\xrightarrow{u}$	PaxDna + DeltaProt
MafADna	$\xrightarrow{p}$	MafADna + MafAProt
MafAProt	$\xrightarrow{d}$	∅
MafADna + PaxProt	$\xrightarrow{b}$	MafADnaPaxProt
MafADnaPaxProt	$\xrightarrow{u}$	MafADna + PaxProt
MafADnaPaxProt	$\xrightarrow{p}$	MafADnaPaxProt + MafAProt
MafADna + MafAProt	$\xrightarrow{b}$	MafADnaMafAProt
MafADnaMafAProt	$\xrightarrow{u}$	MafADna + MafAProt
MafADnaMafAProt	$\xrightarrow{p}$	MafADnaMafAProt + MafAProt
MafADna + DeltaProt	$\xrightarrow{b}$	MafADnaDeltaProt
MafADnaDeltaProt	$\xrightarrow{u}$	MafADna + DeltaProt
DeltaDna	$\xrightarrow{p}$	DeltaDna + DeltaProt
DeltaProt	$\xrightarrow{d}$	∅
DeltaDna + PaxProt	$\xrightarrow{b}$	DeltaDnaPaxProt
DeltaDnaPaxProt	$\xrightarrow{u}$	DeltaDna + PaxProt
DeltaDnaPaxProt	$\xrightarrow{p}$	DeltaDnaPaxProt + DeltaProt
DeltaDna + MafAProt	$\xrightarrow{b}$	DeltaDnaMafAProt
DeltaDnaMafAProt	$\xrightarrow{u}$	DeltaDna + MafAProt
DeltaDna + DeltaProt	$\xrightarrow{b}$	DeltaDnaDeltaProt
DeltaDnaDeltaProt	$\xrightarrow{u}$	DeltaDna + DeltaProt
DeltaDnaDeltaProt	$\xrightarrow{p}$	DeltaDnaDeltaProt + DeltaProt

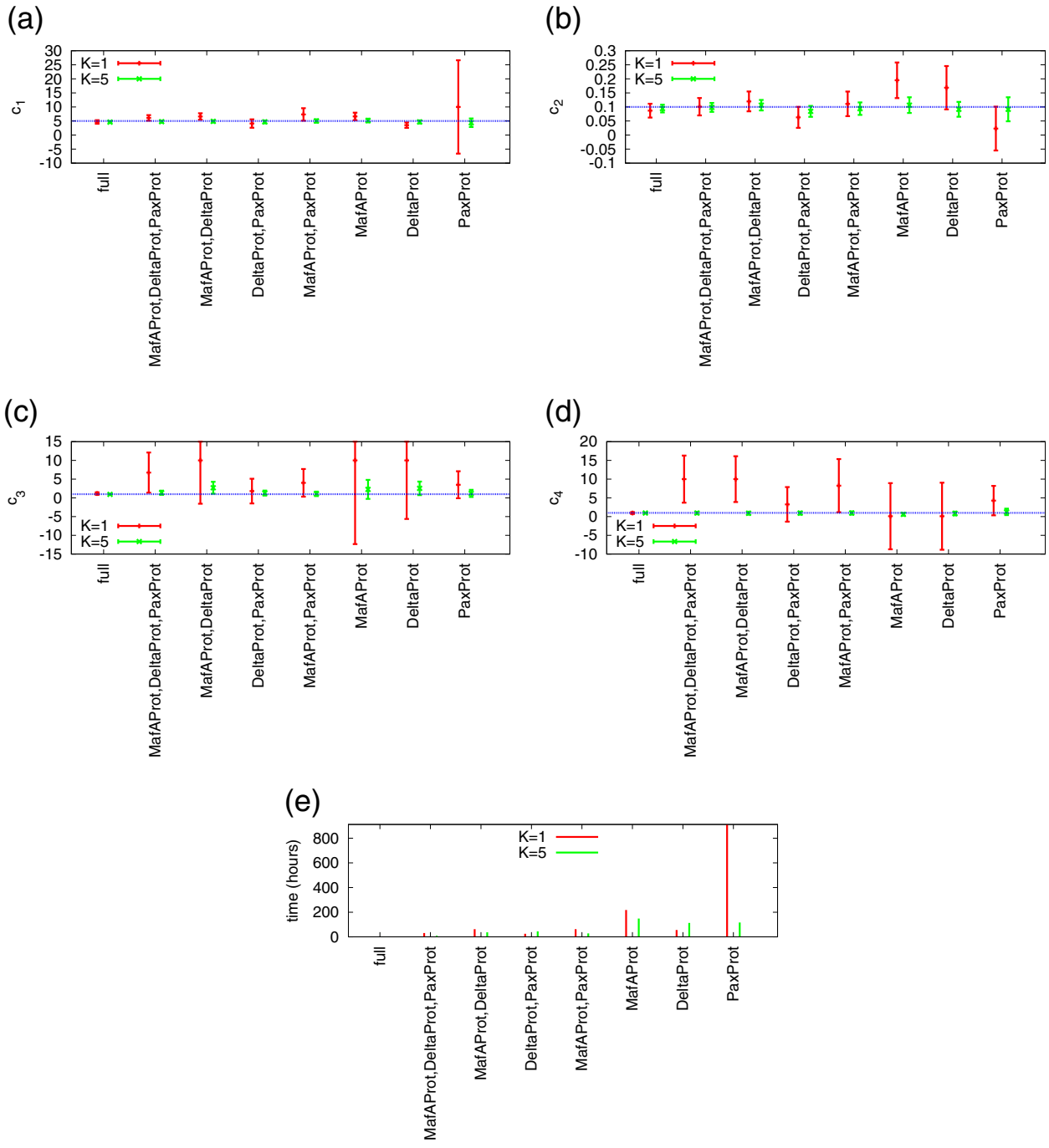

Figure 6 Parameter estimation results for the multi-attractor model. The *x*-axis shows the species that were observed during the estimation procedure. The dotted blue line corresponds to the true value of c_1, c_2, c_3, and c_4, respectively. The error bars in **(a)**–**(d)** show the mean (plus/minus the standard deviation) of the estimators. In **(e)** we plot the running time of the estimation procedure.

Conclusion

Parameter inference for stochastic models of cellular processes demands huge computational resources. We proposed an efficient numerical method to approximate maximum likelihood estimators for a given set of observations. We consider the case where the observations are subject to measurement errors and where only the molecule numbers of some of the chemical species are observed at certain points in time. In our experiments we show that if the observations provide sufficient information then parameters can be accurately identified. If only little information is available then the approximations of

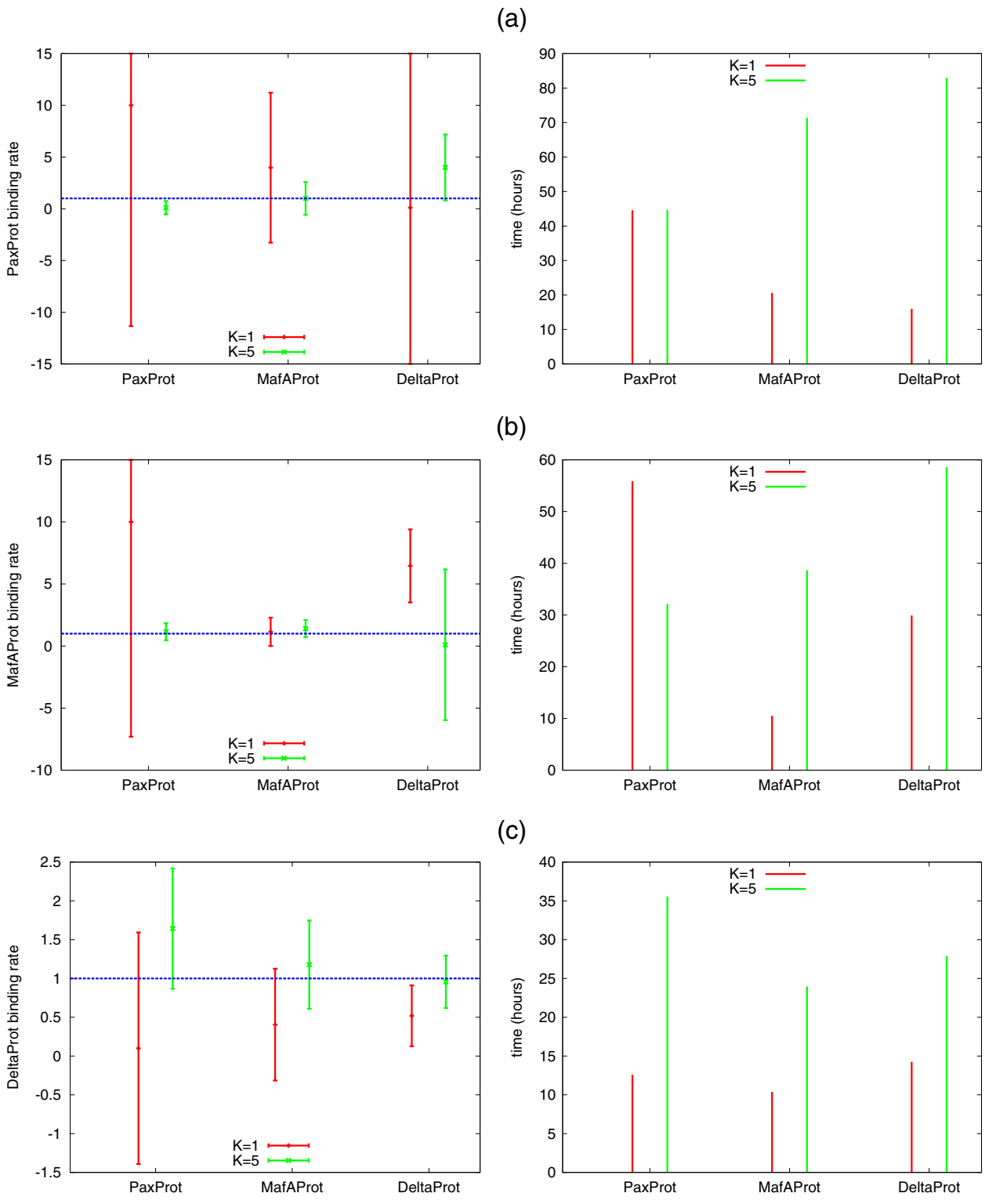

Figure 7 Results of the multi-attractor (as in Figure 6), but we estimate the binding rate of each protein independently.

the standard deviations of the estimators indicate whether more observations are necessary to accurately calibrate certain parameters.

As future work we plan a comparison of our technique to parameter estimation based on Bayesian inference. In addition, we will examine whether a combination of

Table 3 Production rate estimation in the multi-attractor model

Protein	*K*	Estimated rate constant	Standard deviation	Time (hours)	Observed proteins
PaxProt	1	10.0	13.6159	7.45	MafAProt, DeltaProt
	5	0.5693	2.1842	6.34	
MafAProt	1	4.9998	4.9884	11.62	PaxProt, DeltaProt
	5	5.4853	2.3873	13.86	
DeltaProt	1	2.5453	1.8075	4.35	PaxProt, MafAProt
	5	5.3646	1.4682	12.39	

methods based on prior knowledge and the maximum likelihood method is useful. Future plans further include parameter estimation methods for systems where some chemical species have small molecule numbers while others are high rendering a purely discrete representation infeasible. In such cases, hybrid models are advantageous where large populations are represented by continuous deterministic variables while small populations are still described by discrete random variables [23].

Additional files

Additional file 1: SBML file of the gene expression example.

- File name: genexpression.xml
- File format: SBML (see http://www.sbml.org/sbml/level2/version4)
- File extension: xml

Additional file 2: SBML file of the multiattractor model.

- File name: multiattractor.xml
- File format: SBML (see http://www.sbml.org/sbml/level2/version4)
- File extension: xml

Competing interests
The authors declare that they have no competing interests.

Acknowledgements
This research was been partially funded by the German Research Council (DFG) as part of the Cluster of Excellence on Multimodal Computing and Interaction at Saarland University and the Transregional Collaborative Research Center "Automatic Verification and Analysis of Complex Systems" (SFB/TR 14 AVACS).

Received: 8 January 2012 Accepted: 7 July 2012
Published: 18 July 2012

References

1. DT Gillespie, Exact stochastic simulation of coupled chemical reactions, J. Phys. Chem. **81**(25), 2340–2361 (1977)
2. A Loinger, A Lipshtat, NQ Balaban, O Biham, Stochastic simulations of genetic switch systems, Phys. Rev. E. **75**, 021904 (2007)
3. T Tian, S Xu, J Gao, K Burrage, Simulated maximum likelihood method for estimating kinetic rates in gene expression, Bioinformatics. **23**, 84–91 (2007)
4. S Reinker, R Altman, J Timmer, Parameter estimation in stochastic biochemical reactions, IEEE Proc. Syst. Biol. **153**, 168–178 (2006)
5. B Uz, E Arslan, I Laurenzi, Maximum likelihood estimation of the kinetics of receptor-mediated adhesion, J. Theor. Biol. **262**(3), 478–487 (2010)
6. I Golding, J Paulsson, S Zawilski, E Cox, Real-time kinetics of gene activity in individual bacteria, Cell. **123**(6), 1025–1036 (2005)
7. R Boys, D Wilkinson, T Kirkwood, Bayesian inference for a discretely observed stochastic kinetic model, Stat. Comput. **18**, 125–135 (2008)
8. JJ Higgins, Bayesian inference and the optimality of maximum likelihood estimation, Int. Stat. Rev. **45**, 9–11 (1977)
9. CS Gillespie, A Golightly, Bayesian inference for generalized stochastic population growth models with application to aphids, J. R. Stat. Soc. Ser. C. **59**(2), 341–357 (2010)
10. T Toni, D Welch, N Strelkowa, A Ipsen, M Stumpf, Approximate Bayesian computation scheme for parameter inference and model selection in dynamical systems, J. R. Soc. Interface. **6**(31), 187–202 (2009)
11. M Komorowski, B Finkenstädt, C Harper, D Rand, Bayesian inference of biochemical kinetic parameters using the linear noise approximation, J. R. Stat. Soc. Ser. **C 10**(343) (2009)
12. A Andreychenko, L Mikeev, D Spieler, V Wolf, in *Computer Aided Verification - 23rd International Conference, CAV 2011, Snowbird, UT , USA , July 14-20, 2011. Proceedings, Volume 6806 of Lecture Notes in Computer Science*. Parameter Identification for Markov Models of Biochemical Reactions, (Springer, Heidelberg, 2011), pp. 83–98
13. A Andreychenko, L Mikeev, D Spieler, V Wolf, in *Computational Systems Biology - 8th International Workshop, WCSB 2011, Zürich, Switzerland, June 6-8, 2011. Proceedings*. Approximate maximum likelihood estimation for stochastic chemical kinetics ((Tampere International Center for Signal Processing. TICSP series # 57, Tampere, Finland, 2011)
14. TA Henzinger, M Mateescu, V Wolf, in *Computer Aided Verification, 21st International Conference, CAV 2009, Grenoble, France, June 26 - July 2, 2009. Proceedings, Volume 5643 of Lecture Notes in Computer Science*. Sliding Window Abstraction for Infinite Markov Chains, (Springer, Heidelberg, 2009), pp. 337–352
15. B Munsky, M Khammash, The finite state projection algorithm for the solution of the chemical master equation, J. Chem. Phys. **124**, 044144 (2006)
16. K Burrage, M Hegland, F Macnamara, B Sidje, in *Proceedings of the Markov 150th Anniversary Conference*. A Krylov-based finite state projection algorithm for solving the chemical master equation arising in the discrete modelling of biological systems, (Boson Books, Bitingduck Press, Altadena, CA, USA, 2006), pp. 21–38
17. M Mateescu, V Wolf, F Didier, T Henzinger, Fast adaptive uniformisation of the chemical master equation, IET Syst. Biol. **4**(6), 441–452 (2010)
18. R Sidje, K Burrage, S MacNamara, Inexact uniformization method for computing transient distributions of Markov chains, SIAM J. Sci. Comput. **29**(6), 2562–2580 (2007)
19. L Ljung, *System Identification: Theory for the, User*, 2nd edn, (Prentice Hall, PTR, New Jersey, USA, 1998)
20. Global Optimization Toolbox: User's Guide (r2011b). Mathworks 2011. [www.mathworks.com/help/pdf_doc/gads/gads_tb.pdf]
21. Z Ugray, L Lasdon, JC Plummer, F Glover, J Kelly, R Marti, Scatter search and local NLP solvers: a multistart framework for global optimization, INFORMS J. Comput. **19**(3), 328–340 (2007)
22. JX Zhou, L Brusch, S Huang, Predicting pancreas cell fate decisions and reprogramming with a hierarchical multi-attractor model, PLoS ONE. **6**(3), e14752 (2011)
23. TA Henzinger, L Mikeev, M Mateescu, V Wolf, in *Computational Methods in Systems Biology, 8th International Conference, CMSB 2010, Trento, Italy, September 29 - October 1, 2010. Proceedings*. Hybrid numerical solution of the chemical master equation, (ACM, New York, USA, 2010), pp. 55–65

doi:10.1186/1687-4153-2012-9
Cite this article as: Andreychenko *et al.*: **Approximate maximum likelihood estimation for stochastic chemical kinetics.** *EURASIP Journal on Bioinformatics and Systems Biology* 2012 **2012**:9.