Theoretical Computer Science 412 (2011) 2188–2206

Contents lists available at ScienceDirect

Theoretical Computer Science

journal homepage: www.elsevier.com/locate/tcs

Probabilistic approximations of ODEs based bio-pathway dynamics☆

Bing Liu [a,*], David Hsu [a,b], P.S. Thiagarajan [a,b]

[a] NUS Graduate School for Integrative Sciences and Engineering, National University of Singapore, Singapore
[b] Department of Computer Science, National University of Singapore, Singapore

ARTICLE INFO

Keywords:
Computational systems biology
Modeling
ODEs
Dynamic Bayesian networks

ABSTRACT

Bio-chemical networks are often modeled as systems of ordinary differential equations (ODEs). Such systems will not admit closed form solutions and hence numerical simulations will have to be used to perform analyses. However, the number of simulations required to carry out tasks such as parameter estimation can become very large. To get around this, we propose a discrete probabilistic approximation of the ODEs dynamics. We do so by discretizing the value and the time domain and assuming a distribution of initial states w.r.t. the discretization. Then we sample a representative set of initial states according to the assumed initial distribution and generate a corresponding set of trajectories through numerical simulations. Finally, using the structure of the signaling pathway we encode these trajectories compactly as a dynamic Bayesian network.

This approximation of the signaling pathway dynamics has several advantages. First, the discretized nature of the approximation helps to bridge the gap between the accuracy of the results obtained by ODE simulation and the limited precision of experimental data used for model construction and verification. Second and more importantly, many interesting pathway properties can be analyzed efficiently through standard Bayesian inference techniques instead of resorting to a large number of ODE simulations. We have tested our method on ODE models of the EGF-NGF signaling pathway [1] and the segmentation clock pathway [2]. The results are very promising in terms of accuracy and efficiency.

1. Introduction

Quantitative mathematical models are needed for understanding the dynamics of complex biological processes that govern various intra- and inter-cellular functions. Here we focus on signaling pathways which typically sense extra-cellular or internal signals and in response, activate a cascade of intra-cellular reactions. A multitude of signaling pathways govern and coordinate the behavior of cells and many disease processes arise from defects in signaling pathways. Hence it is important to develop quantitative dynamic models for this class of bio-pathways.

A signaling pathway can be viewed as a network of bio-chemical reactions and one often models this network as a system of Ordinary Differential Equations (ODEs) [3]. The equations describe specific bio-chemical reactions while the variables represent the concentration levels of molecular species (genes, RNAs, proteins).

However, signaling pathways usually involve a large number of bio-chemical reactions. Hence the corresponding systems of ODEs will not admit closed form solutions. Instead, one will have to numerically generate trajectories to study the dynamics. Further, the quantitative observations of the system will have very limited precision. Specifically, the initial

☆ This research has been partially supported by the Singapore Ministry of Education (MOE) grant T208A2104.
* Corresponding author. Tel.: +65 65167997.
E-mail addresses: lb@nus.edu.sg (B. Liu), dyhsu@comp.nus.edu.sg (D. Hsu), thiagu@comp.nus.edu.sg (P.S. Thiagarajan).

0304-3975/$ – see front matter © 2011 Elsevier B.V. All rights reserved.
doi:10.1016/j.tcs.2011.01.021

concentration levels of the various proteins and rate constants will often be available only as *intervals* of values. Further, experimental data in terms of the concentration levels of a few proteins at a small number of time points will also be available only in terms of intervals of values. In addition, the data will often be gathered using a population of cells. Consequently, when numerically simulating the ODE model, one must resort to Monte Carlo methods to ensure that sufficiently many values from the relevant intervals are being sampled. As a result, basic tasks such as model validation, parameter estimation and sensitivity analysis will require the generation of a large number of trajectories. This motivates our main goal: Approximating the dynamics of systems of ODEs modeling bio-chemical networks.

We start with such a system of ODEs and fix a suitable discretization of the value domains of the variables and rate constants into finite sets of intervals. We also fix a discretization of the time domain into a finite number of time points. We assume a prior distribution of the initial states. Usually, this prior will consist of a uniform distribution over some of the discretized intervals of values of the variables and the rate constants. Then by sampling the prior distribution of initial states, we numerically pre-compute and store a representative subset of trajectories induced by the ODEs dynamics. The key idea is to exploit the dependencies/independencies in the pathway structure to compactly encode these trajectories as a time-variant dynamic Bayesian network [4]. The resulting approximation is called the Bayesian Dynamics Model (BDM). We then view the BDM as an approximation of the ODEs dynamics and perform analysis tasks on this simpler model.

Since the trajectories are grouped together through the discretization, our method bridges the gap between the accuracy of the results obtained by ODE simulation and the limited precision of experimental data used for model construction and validation. In addition, the BDM represents the dependencies between the variables more explicitly in the graph structure of the underlying dynamic Bayesian network (DBN). More crucially, many interesting pathway properties can be analyzed efficiently through standard Bayesian inference techniques, instead of resorting to large scale numerical simulations.

Admittedly, there is a one-time computational cost incurred to construct the BDM. But this cost can be easily amortized by performing multiple analysis tasks using the BDM. We have tested our method on a model of the EGF-NGF signaling pathway [1] which determines the stimulation-dependent PC12 cell fate. We have also constructed a BDM model of the segmentation clock pathway [2] which governs the periodic formation of the vertebral precursors. The results we obtain are very promising in terms of both accuracy and efficiency.

1.1. Related work

A variety of qualitative and quantitative computational models have been proposed in the recent years to study bio-pathways [5–8]. The quantitative models can be broadly classified into two types. In one approach, the *number* of molecules of each kind is kept track of and stochastic simulations are used to advance the system state one reaction at a time. In the second approach, one tracks the *concentrations* of the molecular species of each kind and ODEs are used to construct the models. One then deploys deterministic numerical simulations to study the dynamics. Clearly, both approaches are needed to cover different contexts.

Discretization is a common approach which has been adapted in many modeling formalisms. For instance, discretized approximations of stochastic models are studied in [9–11]. In these works, the dynamics of a process-algebra-based description of a bio-pathway is given in terms of a Continuous Time Markov Chain (CTMC) which is then discretized using the notion of *levels* to ease analysis. Apart from the fact that our starting point is a system of ODEs, a crucial additional step that we take is to exploit the structure of the pathway to factor the dynamics into a dynamic Bayesian network. We then perform analysis tasks on this more compact representation. In a similar vein, our model is more compact than the graphical model of a network of non-homogenous Markov chains studied in [14].

For sure, our dynamic Bayesian network based model may be viewed as a factored Markov chain. In this sense, a crucial component of our construction mirrors the technique of factoring a Hidden Markov Model (HMM) as a dynamic Bayesian network by decomposing a system state into its constituent variables [15]. This connection leads us to believe that the techniques proposed in [16], as well as the verification techniques reported in [17,18] can be adapted to our setting.

Analyzing CTMC models such as PEPA requires stochastic simulations that are often computationally intensive [19]. We note however, in our setting, though BDM is a probabilistic graphical model, we do not have to resort to stochastic simulations. The inferencing algorithm we use (the Factored Frontier Algorithm [20]), in one sweep, gathers information about the statistical properties of the family of trajectories encoded by the BDM.

1.2. Organization and contents

The rest of the paper is organized as follows. After the preliminaries in the next section, we describe our method for constructing the BDM approximation in Section 3. In Section 4, we present techniques for performing tasks such as basic inferencing, parameter estimation and global sensitivity analysis using the BDM model. In the subsequent section we present two case studies. In the final section, we summarize the paper and discuss future work.

This paper is an improved and expanded version of the work presented in [21]. In addition, we present here a second case study consisting of the segmentation clock pathway model (Section 5.2). Furthermore, we discuss sampling methods (Section 3.3.1) and present a new sampling method called equation sampling. We later compare its performance (Section 5.1.5) with the sampling technique used in [21]. Finally, we have varied the number of intervals used to discretize the value spaces and compared the accuracies of the resulting approximations (Section 5.1.4).

Additional material containing figures and tables supporting the work presented here can be found at [22]. One can also obtain from here our open source tool for the BDM's construction and analysis.

2. Preliminaries

We first develop the notions leading to the observation that the flows (vector fields) that arise as the solution to our systems of ODEs will be measurable functions. This will secure the mathematical basis for our approximation. The relevant material can be found in [23–26].

Let $\mathbb{N}$ denote the set of non-negative integers. Assume that X and Y are metric spaces [27]. A function $f : X \to Y$ is said to be of class C^k, where $k \in \mathbb{N}$, if the derivatives $f', f'', \ldots, f^{(k)}$ exist and are continuous. Thus, the class C^0 consists of all continuous functions and the class C^1 consists of all continuously differentiable functions.

A **σ-algebra** over a set X is a nonempty collection of subsets of X that is closed under complementation and countable unions. The **Borel σ-algebra** on a topological space X, denoted as $\mathcal{B}_X$, is the minimal σ-algebra containing all the open sets of X.

A **probability space** is a triple $(\Omega, \mathcal{F}, \mathbf{P})$ consisting of a set Ω, a σ-algebra $\mathcal{F}$ over Ω, and a function $\mathbf{P} : \mathcal{F} \to [0, 1]$ such that: (i) $\mathbf{P}(\Omega) = 1$; (ii) if $\{A_w\}_{w \in W}$ is a countable family of pairwise disjoint sets in $\mathcal{F}$, then $\mathbf{P}(\cup_w A_w) = \sum_w \mathbf{P}(A_w)$.

Let X and Y be nonempty sets and $\mathcal{M}$ and $\mathcal{N}$ be σ-algebras of subsets of X and Y respectively. A function $f : X \to Y$ is said to be $(\mathcal{M}, \mathcal{N})$**-measurable** if

$$E \in \mathcal{N} \Rightarrow f^{-1}(E) \in \mathcal{M}.$$

The following fact is crucial for our purposes.

Proposition 1. [26] *If X and Y are metric spaces and $f : X \to Y$ is continuous, then f is $(\mathcal{B}_X, \mathcal{B}_Y)$-measurable.*

2.1. ODEs and flows

Through the rest of this section and the next one, we assume a set of ODEs $\dot{x}_i(t) = f_i(\mathbf{x}(t), \mathbf{p})$ involving the variables $\{x_1, x_2, \ldots, x_n\}$. Each variable $x_i(t)$ is a real-valued function of t with the domain of t being the set of reals. $\{p_1, p_2, \ldots, p_m\}$ is the set of real-valued parameters. We will require the ODEs to be *autonomous* in the sense t does not appear explicitly in any f_i. In our setting, we will often be interested in studying the dynamics for different combinations of values for the parameters. Hence it will be convenient to treat them also as variables. However they will be time-invariant; once their values are fixed at $t = 0$, these values will not change through the passage of time. Consequently, we will implicitly assume m additional differential equations of the form $\dot{p}_j(t) = 0$ with j ranging over $\{1, 2, \ldots, m\}$. We will often let $\mathbf{v}$ range over $\mathbb{R}^n_+$, the values space of the variables and $\mathbf{k}$ range over $\mathbb{R}^m_+$, the values space of the parameters and $\mathbf{z}$ range over $\mathbb{R}^{n+m}_+$, the combined values space. In vector form, our system of autonomous ODEs may be represented as $\mathbf{Z}' = F(\mathbf{Z})$. We shall assume that the ODEs will be modeling mass action or Michaelis-Menten kinetics [28]. However, our method will be applicable for more general types of reaction kinetics too.

Based on the preceding remarks, we can assume $f_i : \mathbb{R}^{n+m}_+ \to \mathbb{R}_+$ to be of the form:

$$\sum_{j=1}^{r_i} c_j n_{ij} g_j,$$

where r_i is the number of reactions associated with species x_i and $c_j = -1$ ($c_j = +1$) if x_i is a reactant(product) of the jth reaction. Further, the quantities $n_{ij} \in \mathbb{Z}$ denote the stoichiometric coefficients and g_j are rational functions of the form $g_j = p_\alpha x_a x_b$ (mass action) or $g_j = p_\alpha x_a x_b/(p_\beta + x_a)$ (Michaelis-Menten) with $a, b \in \{1, 2, \ldots, n\}$ and $\alpha, \beta \in \{1, 2, \ldots, m\}$, describing the kinetic rates of the corresponding reactions. This leads us to assume that $f_i \in C^1$ for each i and hence $F : \mathbb{R}^{n+m}_+ \to \mathbb{R}^{n+m}_+$ can also be assumed to be a C^1 function. Furthermore, the variables representing the concentration level of a species within a single cell as well as the parameters capturing the reaction rates will take values from a bounded interval. Hence the domain of F can be restricted to a bounded region $\mathcal{D}$ of $\mathbb{R}^{n+m}_+$.

Given $\mathbf{z}_0 = (\mathbf{v}_0, \mathbf{k})$ where $\mathbf{v}_0$ specifies the initial values of the variables and $\mathbf{k}$ specifies the parameters values, the system of ODEs will have a unique solution since $F \in C^1$ [23]. We shall denote this solution by $\mathbf{Z}(t)$ with $\mathbf{Z}(0) = \mathbf{z}_0$ and $\mathbf{Z}'(t) = F(\mathbf{Z}(t))$. We are guaranteed that $\mathbf{Z}(t)$ will be a C^0-function [23].

It will be convenient to define the flow $\Phi : \mathbb{R}_+ \times \mathcal{D} \to \mathcal{D}$ of $\mathbf{Z}' = F(\mathbf{Z})$ for arbitrary initial vectors $\mathbf{z}$. It will be a C^0-function given by: $\Phi(t, \mathbf{z}) = \mathbf{Z}(t)$ with $\Phi(0, \mathbf{z}) = \mathbf{Z}(0) = \mathbf{z}$ and $\partial(\Phi(t, \mathbf{z}))/\partial t = F(\Phi(t, \mathbf{z}))$ for all t [23]. Further, $\Phi(t, \cdot)$ will be bijective.

Since the flow Φ is C^0, i.e. continuous and $\mathcal{D} \subseteq \mathbb{R}^{n+m}$ is a metric space we are assured that $\Phi(t, \cdot)$ is $(\mathcal{B}_\mathcal{D}, \mathcal{B}_\mathcal{D})$-measurable by Proposition 1. In what follows, we use Φ_t to denote $\Phi(t, \cdot)$ and summarize the above observations via:

Proposition 2. *Suppose $\mathbf{Z}' = F(\mathbf{Z})$ is an autonomous system of ODEs with F in C^1 and with the domain of F being a bounded region $\mathcal{D}$ of $\mathbb{R}^{n+m}_+$. Then there exists a unique flow $\Phi : \mathbb{R}_+ \times \mathcal{D} \to \mathcal{D}$ for arbitrary initial vectors $\mathbf{z}$ satisfying: $\Phi(t, \mathbf{z}) = \mathbf{Z}(t)$*

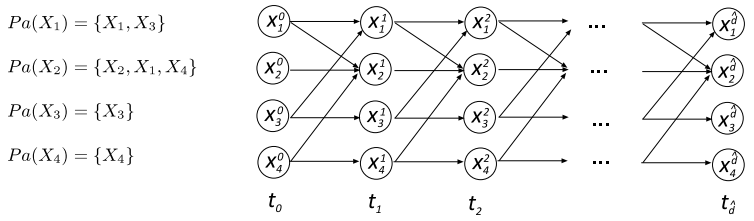

Fig. 1. A DBN example.

with $\Phi(0, \boldsymbol{z}) = \boldsymbol{Z}(0) = \boldsymbol{z}$ and $\partial(\Phi(t, \boldsymbol{z}))/\partial t = F(\Phi(t, \boldsymbol{z}))$ for all t. Further, $\Phi(t, \cdot)$ will be in C^0 and hence $\mathcal{B}_{\mathcal{D}}$-measurable. As a result, for all $t \in \mathbb{R}$:

$$B \in \mathcal{B}_{\mathcal{D}} \Rightarrow \Phi_t^{-1}(B) = \{\boldsymbol{z} \in \mathcal{D} \mid \Phi(t, \boldsymbol{z}) \in B\} \in \mathcal{B}_{\mathcal{D}}.$$

2.2. Markov Chains and dynamic Bayesian networks

A **Markov Chain** [29] is a pair $(S, \{p_{ij}\})$ where $S = \{s_1, s_2, \ldots, s_{\hat{n}}\}$ is set of states and $p_{ij} \in [0, 1]$ are the transition probabilities with $\Sigma_{j=1}^{\hat{n}} p_{ij} = 1$ for every i. Thus if the system is in state s_i at t then it will be in state s_j at $t+1$ with probability p_{ij}. Given an initial probability distribution Ψ^0 over S at $t = 0$, viewed as an $1 \times \hat{n}$-row vector, the probability distribution Ψ^k over S at $t = k$ will be given by $(\Psi^0)T^k$ where T is the $\hat{n} \times \hat{n}$ transition probability matrix with $T_{ij} = p_{ij}$.

A **Bayesian network** [15] is a finite acyclic directed graph $BN = (V, E)$ which has a finite-valued random variable X_v and a conditional probability table CPT_v associated with each node v. The entries in CPT_v will be of the form $Pr(X_v = x | X_{v_1} = x_1, X_{v_2} = x_2, \ldots, X_{v_j} = x_j)$ where $\{v_1, v_2, \ldots, v_j\}$ is the set of *parents* of v given by $Pa(v) = \{u \mid (u, v) \in E\}$. BN represents — often compactly — the joint probability distribution over the random variables $\{X_v\}_{v \in V}$ given by: $Pr(X_{v_1} = x_1, X_{v_2} = x_2, \ldots, X_{v_{\check{n}}} = x_{\check{n}}) = \prod_{i=1}^{\check{n}} Pr(X_{v_i} = x_i | X_{vi1} = x_{i1}, X_{vi2} = x_{i2}, \ldots, X_{vij} = x_{ij})$ with $Pa(v_i) = \{v_{i1}, v_{i2}, \ldots, v_{ij}\}$.

Dynamic Bayesian networks (DBNs) are Bayesian networks that model temporal evolution of systems whose (local states) are modeled as random variables [4]. There are many variants of dynamic Bayesian networks. We will be dealing with a restricted class of time-variant two-slice dynamic Bayesian networks. They will be of the form $(B_0, \{B_{\rightarrow}^d\}_{d=1}^{\hat{d}}, Pa)$, where B_0 defines the initial probability distributions $\{Pr(\mathbf{X}_i^0)\}$ of the random variables $\{X_i\}_{i=1}^{l}$. And $\{B_{\rightarrow}^d\}$ are two-slice temporal Bayesian networks for the time points $\{t^1, \ldots, t^{\hat{d}}\}$. The nodes of the Bayesian network $B_{\rightarrow}^d$ denoted V^d is given by $V^d = \{X_i^{d-1} \mid 1 \leq i \leq l\} \cup \{X_i^d \mid 1 \leq i \leq l\}$ (here we are identifying the nodes with the random variables associated with them). The edge relation E^d will be the subset of $\{X_i^{d-1} \mid 1 \leq i \leq l\} \times \{X_i^d \mid 1 \leq i \leq l\}$ satisfying $(X_j^{d-1}, X_i^d) \in E^d$ iff $X_j \in Pa(X_i)$. As might be expected, $Pa : \mathbf{X} \to 2^{\mathbf{X}}$ with $\mathbf{X} = \{X_i \mid 1 \leq i \leq l\}$. Each node X_i^d will also a conditional probability table CPT_i^d associated with it with entries of the form $Pr(X_i^d = x \mid X_{i1}^{d-1} = x_{i1}, \ldots X_{ij}^{d-1} = x_{ij})$, where $Pa(X_i) = \{X_{i1}, \ldots X_{ij}\}$.

Thus the way the nodes of the $(d+1)$th layer are connected to the nodes of the dth layer will remain invariant. However CPT_i^{d+l} will be, in general, different from CPT_i^d. An example of such a dynamic Bayesian network is shown in Fig. 1. To avoid clutter we have not shown the CPTs associated with each node. In our setting these dynamic Bayesian networks will represent an associated Markov chain in a factored form.

3. The Construction of the Bayesian dynamics model

Conceptually, our approximation technique consists of three major steps. First we discretize the value spaces of the variables and parameters into a finite sets of intervals. We also discretize the time domain of interest into a finite number of time points. Finally we assume a prior distribution of initial values over the intervals. As a result, the flow defined by our system of ODEs will induce a Markov chain $\mathcal{MC}_{ideal}$.

This Markov chain cannot be computed explicitly. Hence as a second step, we compute an approximation of this Markov chain by sampling the initial states according to the prior sufficiently many times, and generating a trajectory for each of the sampled initial states. Then by a simple process of counting tied to the discretized value space and time domain, we obtain the Markov Chain denoted $\mathcal{MC}_{approx}$ as an approximation of $\mathcal{MC}_{ideal}$.

However, the number of states of $\mathcal{MC}_{approx}$ will be exponential in the number of variables. To get around this, in the third step, we exploit the pathway structure (i.e. the way the variables are coupled to each other in the system of ODEs) to represent $\mathcal{MC}_{approx}$ compactly as a dynamic Bayesian network. This representation of $\mathcal{MC}_{approx}$ is our final object and will be called the Bayesian Dynamics Model (BDM).

This three step procedure is however just a conceptual framework. We shall construct the BDM *directly* from the given system of ODEs instead of passing through a Markov chain. We now proceed with a more technical description the steps

involved in constructing the BDM model. In doing so, we shall assume that we are given the system of ODEs $\dot{x}_i(t) = f_i(\mathbf{x}(t), \mathbf{p})$ with n variables and m rate parameters specified in the previous section with the associated notations and assumptions.

3.1. The Markov chain $\mathcal{MC}_{ideal}$

Pathways models are usually validated by experimental data available only for a few time points with the concentrations measured at the final time point typically signifying the steady state value. Hence we assume the dynamics is of interest only for discrete time points and that too only up to a maximal time point. We denote these time points as $\{t_0, t_1, \ldots, t_{max}\}$. It is *not* necessary to uniformly discretize the time domain. However, to simplify the notations of the following sections, we fix a time step $\Delta t > 0$ and the time points of interest is assumed to be the set $\{d \cdot \Delta t\}$ with d ranging over $\{0, 1, \ldots, \hat{d}\}$. Thus $\hat{d} \cdot \Delta t$ is the maximal time point of interest.

Next we assume that the values of the variables can be observed with only finite precision and accordingly partition the range of each variable x_i into L^i intervals $[v_i^{min}, v_i^1), [v_i^1, v_i^2), \ldots, [v_i^{L_i-1}, v_i^{max}]$. We denote this set of intervals as $\mathcal{I}_i$. We also similarly discretize the range of each parameter p_j into a set of intervals denoted as $\mathcal{I}_{n+j}$. The set $\mathcal{I} = \{\mathcal{I}_i\}_{1\le i\le n} \cup \{\mathcal{I}_{n+j}\}_{1\le j\le m}$ is called the **discretization**. Again, we wish to emphasize that the value space can be discretized non-uniformly and our constructions will go through.

As pointed out earlier, the initial values as well as the rate constants (even when they are known) will be given not as point values but as distributions (usually uniform) over the intervals defined by the discretization. We correspondingly assume we are given a prior distribution in the form of a probability density function Υ^0 capturing the initial values.

For example, suppose we are given that the initial values are uniformly distributed within a hypercube $\hat{I}_1 \times \hat{I}_2 \times \cdots \times \hat{I}_{n+m}$, where $\hat{I}_i \in \mathcal{I}_i$ for each i. Let $\hat{I}_i = [l_i, u_i)$ and $\hat{w}_i = u_i - l_i$. Then the corresponding prior probability density function Υ^0 will be given by:

$$\Upsilon^0(\mathbf{z}) = \begin{cases} \frac{1}{\hat{w}_1 \cdot \hat{w}_2 \cdot \ldots \cdot \hat{w}_{n+m}} & \text{if } \mathbf{z} \in \hat{I}_1 \times \hat{I}_2 \times \cdots \times \hat{I}_{n+m}, \\ 0 & \text{otherwise.} \end{cases}$$

The associated probability space we have in mind is $(\mathcal{D}, \mathcal{B}_{\mathcal{D}}, \mathbf{P}^0)$ where $\mathcal{B}_{\mathcal{D}}$ is the Borel σ-algebra over $\mathcal{D}$; the minimal σ-algebra containing the open sets of $\mathcal{D}$ under the usual topology. $\mathbf{P}^0$ is the probability distribution induced by Υ^0 and is given by:

$$\mathbf{P}^0(B) = \int_B \Upsilon^0(\mathbf{z}) \mathrm{d}\mathbf{z}, \text{ for every } B \in \mathcal{B}_{\mathcal{D}}.$$

Further, $TRAJ_{ideal} = \{\Phi_t(\mathbf{z})\}_{t\ge 0}$ with $\mathbf{z}$ ranging over $\hat{I}_1 \times \hat{I}_2 \times \cdots \times \hat{I}_{n+m}$ is the family of trajectories starting from all the possible points in this hypercube. As before, Φ is the flow induced by the system ODEs.

Φ is measurable by Proposition 1. Hence we can define the probability distribution $\mathbf{P}^t$ over $\mathcal{B}_{\mathcal{D}}$ for every t as:

$$\mathbf{P}^t(B) = \mathbf{P}^0(\Phi_t^{-1}(B)), \quad \text{for every } B \in \mathcal{B}_{\mathcal{D}}.$$

Let v be a real number in the range of x_i. We define $[v]$ as the interval in which v falls. In other words, $[v] = I$ iff $v \in I$. Similarly, $[k] = J$ if $k \in J$ for a parameter value k of p_j with $J \in \mathcal{I}_{n+j}$.

Lifting this notation to the vector setting, if $\mathbf{z} = (v_1, v_2, \ldots, v_n, k_1, k_2, \ldots, k_m) \in \mathbb{R}_+^{n+m}$, we define $[\mathbf{z}] = ([v_1], [v_2], \ldots, [v_n], [k_1], \ldots, [k_m])$ and refer to it as a **discrete state**. An $\mathcal{MC}$**-state** is a pair $(\mathbf{s}, d)$, where $\mathbf{s}$ is a discrete state and $d \in \{0, 1, \ldots, \hat{d}\}$. We next define $Pr((\mathbf{s}, d)) = \mathbf{P}^{d\cdot\Delta t}(\{\mathbf{z} \mid \mathbf{z} \in I_1 \times I_2 \times \cdots \times I_{n+m}\})$, where $\mathbf{s} = (I_1, I_2, \ldots, I_{n+m})$. We term the $\mathcal{MC}$-state M to be *feasible* iff $Pr(M) > 0$.

The transition relation denoted as $\rightarrow$, between $\mathcal{MC}$-states is defined via: $M = (\mathbf{s}, d) \rightarrow M' = (\mathbf{s}', d')$ iff $d' = d + 1$ and both M and M' are feasible and there exist $\mathbf{z}_0, \mathbf{z}$, and $\mathbf{z}'$ such that $\Phi(d\cdot\Delta t,\ \mathbf{z}_0) = \mathbf{z}$ and $\Phi((d+1)\cdot\Delta t,\ \mathbf{z}_0) = \mathbf{z}'$. Furthermore, $[\mathbf{z}] = \mathbf{s}$ and $[\mathbf{z}'] = \mathbf{s}'$.

Let E, F denote, respectively, the event that the system is in the discrete state $\mathbf{s}$ at time $d \cdot \Delta t$ and in the discrete state $\mathbf{s}'$ at time $(d + 1) \cdot \Delta t$ for two feasible $\mathcal{MC}$-states $(\mathbf{s}, d \cdot \Delta t)$ and $(\mathbf{s}', (d + 1) \cdot \Delta t)$. Let $EF = E \cap F$ denote joint event $\{\mathbf{z}_0 \mid \Phi(d \cdot \Delta t, \mathbf{z}_0) \in \mathbf{s}, \Phi((d + 1) \cdot \Delta t, \mathbf{z}_0) \in \mathbf{s}'\}$. Consequently, we define the transition probability $Pr((\mathbf{s}, d) \rightarrow (\mathbf{s}', d')) = Pr(F|E) = Pr(EF)/Pr(E)$. Since $Pr(E) > 0$ this transition probability is well-defined.

Let $\mathcal{M} = \{M_1, M_2, \ldots, M_{\hat{n}}\}$ be the set of $\mathcal{M}$-states. We can now define the Markov chain $\mathcal{MC}_{ideal} = (\mathcal{M}, \{p_{ij}\})$ with transition probabilities $p_{ij} = Pr(M_i \rightarrow M_j)$ as above.

Example. A typical biochemical equation depicting an enzyme catalyzed reaction can be written as follows:

$$S + E \underset{k_2}{\overset{k_1}{\rightleftharpoons}} ES \xrightarrow{k_3} E + P. \tag{1}$$

As the basic component of signal transduction pathways [30], it accounts for one step in the transduction of a signaling cascade. In this reaction, the enzyme E binds reversibly to the substrate S, before converting it into the product P and

releasing it. The parameters k_1, k_2 and k_3 are the rate constants that govern the speed of these reactions. The corresponding ODE model will be:

$$\frac{dS}{dt} = -k_1 \cdot S \cdot E + k_2 \cdot ES$$
$$\frac{dE}{dt} = -k_1 \cdot S \cdot E + (k_2 + k_3) \cdot ES$$
$$\frac{dES}{dt} = k_1 \cdot S \cdot E - (k_2 + k_3) \cdot ES$$
$$\frac{dP}{dt} = k_3 \cdot ES.$$

Assuming that the range of each variable or parameter is: $S \in [0, 15]$, $E \in [0, 10]$, $ES \in [0, 10]$, $P \in [0, 15]$, $k_1 \in [0, 1]$, $k_2 \in [0, 1]$, $k_3 \in [0, 1]$ (for simplicity, we ignore all units in this example), we partition each range into 5 equal-sized intervals and form the discretization $\mathcal{I} = \{\mathcal{I}_S, \mathcal{I}_E, \mathcal{I}_{ES}, \mathcal{I}_P, \mathcal{I}_{k_1}, \mathcal{I}_{k_2}, \mathcal{I}_{k_3}\}$, where $\mathcal{I}_S = \mathcal{I}_P = \{[0, 3), [3, 6), [6, 9), [9, 12), [12, 15]\}$, $\mathcal{I}_E = \mathcal{I}_{ES} = \{[0, 2), [2, 4), [4, 6), [6, 8), [8, 10]\}$ and $\mathcal{I}_{k_1} = \mathcal{I}_{k_2} = \mathcal{I}_{k_3} = \{[0, 0.2), [0.2, 0.4), [0.4, 0.6), [0.6, 0.8), [0.8, 1]\}$. We fix the time step Δt to be 0.1 and fix the number of time points to be 100. We have adopted equal-sized intervals and fixed time steps only for convenience.

Suppose we are given a prior distribution that the initial values $\mathbf{z}_0 = (S, E, ES, P, k_1, k_2, k_3)$ are uniformly distributed within a hypercube $\mathbf{C} = [12, 15] \times [8, 10] \times [0, 2) \times [0, 3) \times [0.2, 0.4) \times [0.4, 0.6) \times [0.2, 0.4)$. We then have the prior probability density function Υ^0 given by:

$$\Upsilon^0(\mathbf{z}) = \begin{cases} \frac{1}{(15-12)(10-8)(2-0)(3-0)(0.4-0.2)(0.6-0.2)(0.4-0.2)} = \frac{125}{36} & \text{if } \mathbf{z} \in \mathbf{C}, \\ 0 & \text{otherwise.} \end{cases}$$

Thus, $M_0 = (\mathbf{s}_0 = ([12, 15], [8, 10], [0, 2), [0, 3), [0.2, 0.4), [0.4, 0.6), [0.2, 0.4)), 0)$ will be the initial $\mathcal{MC}$-state of the induced Markov chain $\mathcal{MC}_{ideal}$. Clearly $Pr(M_0) = 1$ since $Pr(M_0) = \mathbf{P}^0(\{\mathbf{z} \mid \mathbf{z} \in \mathbf{C}\}) = \int_{\{\mathbf{z} \mid \mathbf{z} \in \mathbf{C}\}} \Upsilon^0(\mathbf{z}) d\mathbf{z} = 1$.

3.2. The Markov Chain $\mathcal{MC}_{approx}$

The ODEs system will typically not admit a closed form solution. Hence Φ cannot be derived explicitly and as a consequence, $\mathcal{MC}_{ideal}$ can not be explicitly computed either. We can however compute an approximation of $\mathcal{MC}_{ideal}$ as follows.

We sample $\mathbf{z}$ (the initial state) a sufficiently large number of times, say N, according to the prior distribution $\mathbf{P}^0$ (we say more about N below). For each sampled initial $\mathbf{z}$, we determine through numerical integration the $\mathcal{M}$-states $[\Phi(d \cdot \Delta t, \mathbf{z})]$, with d ranging over $\{0, 1, \ldots, \hat{d}\}$. We also determine the transitions along this trajectory. Then through a simple counting process involving these N trajectories (as illustrated below), we compute a Markov chain that we refer to $\mathcal{MC}_{approx}$.

Example (*continued*)**.** Continuing with the enzyme-kinetic system example presented in previous sub-section, we randomly pick 1000 points from the hypercube $\mathbf{C}$ as initial states. Through numerical simulation, we generate 1000 trajectories starting from those points. Then $Pr((\mathbf{s}, d))$ is the fraction of the 1000 trajectories that are in the discrete state $\mathbf{s}$ at time point $d \cdot \Delta t$. $Pr((\mathbf{s}, d) \to (\mathbf{s}', d+1))$ is the fraction of the trajectories that are in $\mathbf{s}$ at time point $d \cdot \Delta t$ which land in $\mathbf{s}'$ at time point $(d+1) \cdot \Delta t$. For instance, suppose 132 trajectories are in the discrete state $\mathbf{s}_2$ at time point 0.1, and 12 of these 132 trajectories evolve to land in the discrete state $\mathbf{s}_3$ at time point 0.2. Then we have $Pr((\mathbf{s}_2, 1)) = 132/1000 = 0.132$ and $Pr((\mathbf{s}_2, 1) \to (\mathbf{s}_3, 2)) = 12/132 = 0.091$.

However, the number of states of this Markov chain will be exponential in n. As a result, for many signaling pathways, $\mathcal{MC}_{approx}$ will be simply too large. Hence we shall construct a time-variant two-slice DBN called the BDM to compactly represent $\mathcal{MC}_{approx}$. We shall however compute the BDM directly from the N sampled trajectories.

3.3. The BDM representation

The key observation is that the structure of the system of ODEs can be exploited to factorize $\mathcal{MC}_{approx}$ into a time-variant 2-slice DBN. This DBN will have $(n + m) \times (\hat{d} + 1)$ nodes. The node v will have associated with it a random variable X_i^d. This random variable will take as values the intervals in $\mathcal{I}_i$; the intervals into which the value space of x_i (in case $i \leq n$) or the parameter p_{i-n} (in case $i > n$) has been discretized. The superscript d will stand for the fact the probability distribution associated with X_i^d describes the probability of the value of the variable x_i (or the parameter p_{i-n}) falling into various intervals in $\mathcal{I}_i$ at time $d \cdot \Delta t$. In what follows, for convenience, we will use the same name to denote a node and the random variable associated with it. From the context it should be clear which role is intended. We now proceed with the construction of the DBN $(B_0, \{B_{\to}^d\}_{d=1}^{\hat{d}}, Pa)$.

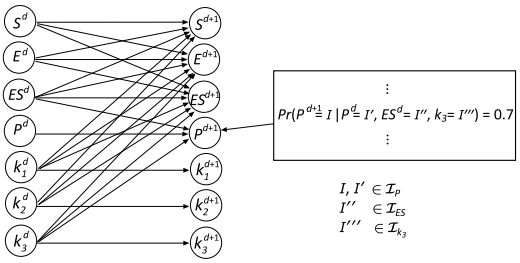

Fig. 2. A slice of the BDM of the enzyme-kinetic system.

We assume that the prior distribution of initial values of the variables and parameters are independent of each other. This is often a reasonable assumption. Even when the assumption is violated it is certainly reasonable to assume that marginal prior probabilities of each variable and parameter can be computed and thus $B^0 = \{Pr(X_i^0)\}_{i=1}^{n+m}$ can be computed. Next, the parent relation Pa is defined as follows. In doing so, it will be convenient to identify the variable x_i with X_i and the parameter p_j with X_{n+j}.

Suppose $z, z' \in \{x_1, x_2, \ldots, x_n, p_1, p_2, \ldots, p_m\}$. Then $z' \in Pa(z)$ iff $z' = z$ or z is a variable and z' appears in the right-hand side of the equation for $\mathrm{d}z/\mathrm{d}t$ in the system of ODEs.

Thus the structure of the ODEs and more precisely, the structure of the bio-chemical network induces the underlying graph of the DBN.

V^d, the set of nodes of the Bayesian network $B^d_{\rightarrow} = (V^d, E^d)$ will be: $V^d = \{X_i^{d-1} \mid 1 \le i \le n+m\} \cup \{X_i^d \mid 1 \le i \le n+m\}$. The edge relation E^d is defined in the obvious way now using the function Pa. To spell it out, it will be the subset of $\{X_i^{d-1} \mid 1 \le i \le n+m\} \times \{X_i^d \mid 1 \le i \le n+m\}$ satisfying $(X_j^{d-1}, X_i^d) \in E^d$ iff $X_j \in Pa(X_i)$.

Finally, suppose $Pa(x_i) = \{z_1, \ldots, z_l\}$. Then the conditional probability table (CPT) associated with the node X_i^d will have entries of the form $Pr(X_i^d = I \mid z_1^{d-1} = I^1, \ldots, z_l^{d-1} = I^l) = h$ with I ranging over $\mathcal{I}_i$ and I^j ranging over $\mathcal{I}_j$ for $1 \le j \le l$ and h ranging over $[0, 1]$. This entry captures probability of the value of the variable x_i (assuming $i \le n$) falling in the intervals I at time $d \cdot \Delta t$ given that at time $(d-1) \cdot \Delta t$, the value of the variable (parameter) z_j was in the interval I^j for $1 \le j \le l$. It is in this sense the dynamics defined by $\mathcal{MC}_{ideal}$ is captured in a factored form by the DBN.

Example (*continued*)**.** Fig. 2 shows two adjacent slices in the BDM of the enzyme-kinetic system. The structure of this BDM is derived from the ODEs presented in Section 3.1. For instance, the parent nodes of P^{d+1} are P^d, ES^d and k_3^d since P^d, P^{d+1} refer to the same variable P while ES, k_3 appear in the expression for $\mathrm{d}P/\mathrm{d}t$. As mentioned earlier, the parameters are assumed to retain their values during a run and hence we denote k_i^d as simply k_i and there will be no CPTs associated with these nodes. On the other hand, the CPT associated with the node P^{d+1} will have entries of the form $Pr(P^{d+1} = I \mid P^d = I', ES^d = I'', k_3 = I''') = h$, where $I, I' \in \mathcal{I}_P$, $I'' \in \mathcal{I}_{ES}$, $I''' \in \mathcal{I}_{k_3}$ and $h \in [0, 1]$. As illustrated in this example, the connectivity between the nodes in successive slices will remain invariant. However, due to the fact that the CPTs associated with the nodes capture the transition probabilities at different time points, they will be time variant.

$\mathcal{MC}_{approx}$ will have, in the worst case, $O((\hat{d}+1)K^n)$ states and $O(\hat{d}K^{2n})$ transitions, where K is the maximum of $|\mathcal{I}_i|$ with $1 \le i \le n+m$. In contrast, the number of nodes in the BDM representation is $O(\hat{d}(n+m))$ and the conditional probability table associated each node will have at most $O(K^{R+1})$ entries, where R is the maximal number of parents a node can have. Usually, the reactants in pathway models will be sparsely coupled to each other and hence R will be much smaller than n. For instance, in the first case study to be presented, $n = 32$ and $R = 5$.

To fill up the entries of the CPTs associated with the nodes we randomly choose N combinations of initial values for the variables and the parameters from their prior distribution as before. Since we assume that the initial values are independent of each other, the values of a variable/parameter can be sampled according to its marginal prior distribution. For instance, in our running example, we can randomly choose a value from $[12, 15]$ for S, a value from $[8, 10]$ for E, a value from $[0, 2)$ for ES, a value from $[0, 3)$ for ES, a value from $[0, 1]$ for k_1, a value from $[0, 1]$ for k_2, a value from $[0, 1]$ for k_3, and then form a vector of initial values.

After picking N sample initial value vectors, we perform numerical integration to generate N trajectories and discretize those trajectories by the predefined intervals and compute the conditional probabilities for each node by simple counting. For example, suppose 132 trajectories hit $(P^0 = [0, 3), ES^0 = [0, 2), k_3 = [0.2, 0.4))$ at time 0 and 12 of them in turn hit $(P^1 = [3, 6))$ at time 0.1, then $Pr(P^1 = [3, 6) \mid P^0 = [0, 3), ES^0 = [0, 2), k_3 = [0.2, 0.4)) = 12/132 = 0.091$.

It is not difficult to show that $\mathcal{MC}_{approx}$ can be recovered from this DBN [31]. However one would rarely need the kind of global information provided by $\mathcal{MC}_{approx}$ to analyze the pathway dynamics. Hence one can almost always avoid having to explicitly construct this large entity.

3.3.1. Sampling methods

Since we have n variables and m parameters, the N sampled initial values vectors should be picked from a $(n+m)$-dimensional space. According to the prior distribution, the value of each variable or known parameter often lies in one

interval. For instance, the value of species E in our running example lies in [8, 10]. Thus, for existing models or for those for which the parameter estimation has already been carried out, we can randomly pick a value for each variable/parameter according to their marginal prior and then form a vector of initial values. We term this type of sampling as *direct* sampling.

However, the value of an unknown parameter will range over all its intervals. For instance, in our running example, the value of each unknown parameter should be sampled from 5 intervals. Thus if we have u unknown parameters whose value spaces have been discretized to K intervals each, one will require a sample size of $J \cdot K^u$ to ensure J samples for each possible combinations of interval values of the unknown parameters. We term this as *global* sampling. Thus the number of samples this method would require will be exponential in the number of unknown parameters. This will often be an unacceptably large number.

At the other end of the spectrum, we can pick J samples for each possible interval of variables or parameters. The idea is to randomly choose a point within the target interval I of, say, the variable x_i and then arbitrarily extend this point value to an $(n+m)$-dimensional vector over the allowed intervals of the other variables and parameters. We term this as *local* sampling and it will require a sample size of $(n+m) \cdot J \cdot K$ with coverage of J per interval. Hence local sampling will require a much smaller sample size. However, it cannot guarantee adequate coverage for *combinations* of interval values of the parameters.

To bring this out through an artificial but simple case, suppose x takes values in the intervals I_1 and I_2 while y takes values in the intervals I_3 and I_4. To get a local coverage of 100 samples per interval, we may pick 100 points from I_1 and extend it by combining it with a random value for y (which will fall in I_3 or I_4). Let S_1 be the set of such samples. Similarly let S_2 be the set of 100 samples obtained by picking 100 points from I_2 and extending each of them randomly to a value for y. Finally, let S_3 (S_4) be the 100 samples obtained by picking 100 values from I_3 (I_4) and extending each of them randomly to a value for x. In this way, with 400 samples we will be able to guarantee a minimum of 100 hits for each of the intervals $\{I_1, I_2, I_3, I_4\}$. However suppose the y-values of all the samples in S_1 (S_2) fall in S_3 (S_4) and the x-values of all the samples in $S_3(S_4)$ fall in $S_1(S_2)$. Then none of the 400 samples will fall in $I_1 \times I_4$ ($I_2 \times I_3$) and hence we will get 0-coverage for the *combination* of this pair of intervals!

To ensure that we are exploring the ODEs dynamics adequately, we need to ensure that all the possible combinations of interval values of unknown parameters governing any single equation are being sampled an adequate number of times. Otherwise, the probabilistic inference we need to to perform on the BDM model during parameter estimation and sensitivity analysis will have poor quality.

Hence to get good coverage in the presence of many unknown parameters, one will have to resort to more sophisticated methods. Here we propose a method called *equation* sampling by which numerical simulations can be carried out in the presence of unknown parameters while ensuring that the local dynamics defined by the individual equations are being explored adequately. To bring out the main idea, suppose the equation for the variable x_i involves the unknown parameters k_1 and k_2 and the values of k_1 (k_2) have been divided into three intervals I_1 (I_1'), I_2 (I_2') and I_3 (I_3'). Then for a specific combination of intervals, say I_2 and I_3' we pick 100 samples such that the k_1 value lies in I_2 and the k_2 value lies in I_3' for each of the samples. In this way we can pick 900 samples which ensure that there are at least 100 samples for each combination of interval values for k_1 and k_2. In general, we will be able provide a coverage of J samples for each possible combination of interval values of the unknown parameters in the equation for each variable with the help of $n \cdot J \cdot K^R$ samples, where R is the maximal number of unknown parameters appearing in *an equation*. Since the positive terms (negative terms) in the differential equation of a species describe the rates of reactions that producing (consuming) this species, equation sampling will provide of a coverage of all possible local conditions that determines the dynamics of a single species. Thus, with this type of sampling, the quality of model analysis tasks can be ensured with an acceptable sample size.

3.3.2. Optimizations

Various optimizations can be developed to reduce the practical complexity of the BDM construction. Specifically, the sampling process followed by the generation of a trajectory can be easily parallelized and executed on a computing cluster. In addition, the CPTs can be stored using a sparse representation. Yet another optimization is to split up a "fat" node with a large number of parents into nodes with smaller fan-in degrees and thus reduce R. As shown in Fig. 3, the reduction can be based on the form of the differential equation associated with the variable. Given that $\mathrm{d}ES/\mathrm{d}t = k_1 \cdot S \cdot E - (k_2 + k_3) \cdot ES$, we introduce two internal nodes X and Y, where X corresponds to the positive term of $\mathrm{d}ES/\mathrm{d}t$, namely, $k_1 \cdot S \cdot E$ and Y corresponds to the negative term $(k_2 + k_3) \cdot ES$. As a result, R can be reduced from 6 to 3. We note however, at present we consider this optimization only to reduce the sizes of the CPTs and not to reduce the number of samples when using the equation sampling method. More details can be found in the Supplementary Material [22].

For sure, the construction of the BDM will involve a significant computational effort but it is a one time cost and significant optimizations can be deployed. Moreover, once the BDM has been constructed, many of the analysis tasks can be performed very efficiently and the one time cost of constructing the BDM can be easily amortized. The experimental results presented in Section 5 will support this claim.

3.4. Error analysis

Since N is finite, there will be an error between the transition probabilities (also the $\mathcal{MC}$-state probabilities) computed using $\mathcal{MC}_{approx}$ and the ones defined by $\mathcal{MC}_{ideal}$. By the central limit theorem [25], this error can be probabilistically bounded. Let $\mathcal{MC}_{ideal} = (\mathcal{M}, \{p_{ij}\})$ and $\mathcal{MC}_{approx} = (\mathcal{M}, \{\hat{p}_{ij}\})$, we have

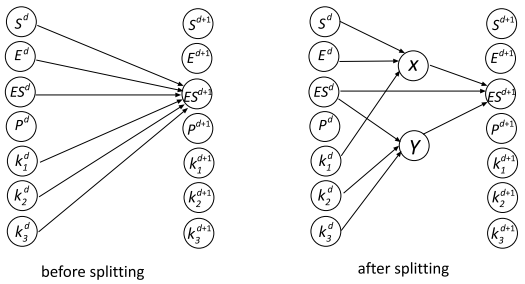

Fig. 3. Node splitting.

Proposition 3. *Suppose the number of samples is N. Then ϵ will be the error with probability c between $\hat{p}_{ij}$ in $\mathcal{MC}_{approx}$ and the corresponding p_{ij} in $\mathcal{MC}_{ideal}$ where*

$$\epsilon = \phi^{-1}\left(\frac{c+1}{2}\right)\sqrt{\frac{p_{ij}(1-p_{ij})}{N}}$$

with $\phi(x) = \frac{1}{\sqrt{2\pi}}\int_{-x}^{x} \mathrm{e}^{-y^2/2}\mathrm{d}y$.

Proof. Let X be a random variable such that $X = 1$ ($X = 0$ resp.) denote the event that a sample trajectory passes (not passes resp.) a discrete state $\mathbf{s}$ at time $d \cdot \Delta t$. Hence X will have a *Bernoulli distribution* with parameter p_{ij} with $\mu = p_{ij}$ and $\sigma^2 = p_{ij}(1-p_{ij})$. If $X_1, X_2, \ldots, X_N$ are the N measurements, by *Central Limit Theorem*, we have:

$$P\left\{-\epsilon \le \frac{\sum_{i=1}^{N} X_i}{N} - \mu \le \epsilon\right\} \approx 2\phi\left(\epsilon\frac{\sqrt{N}}{\sigma}\right) - 1$$

where ϵ is the error and $\phi(x) = \frac{1}{\sqrt{2\pi}}\int_{-x}^{x} \mathrm{e}^{-y^2/2}\mathrm{d}y$. Thus,

$$\epsilon = \phi^{-1}\left(\frac{c+1}{2}\right)\sqrt{\frac{p_{ij}(1-p_{ij})}{N}} \text{ with probability } c. \quad \square$$

Similarly, we have:

Proposition 4. *For each $M = (\mathbf{s}, d) \in \mathcal{M}$, its probability $\hat{Pr}(M)$ computed from $\mathcal{MC}_{approx}$ differs from the one $Pr(M)$ defined in $\mathcal{MC}_{ideal}$ by an error less than or equal to ϵ with probability c where:*

$$\epsilon = \phi^{-1}\left(\frac{c+1}{2}\right)\sqrt{\frac{Pr(M)(1-Pr(M))}{N}} \textit{ with probability } c,$$

with $\phi(x) = \frac{1}{\sqrt{2\pi}}\int_{-x}^{x} \mathrm{e}^{-y^2/2}\mathrm{d}y$.

Therefore, given an error bound ϵ and a confidence level c, we can compute N, the number of samples required to get an error less than or equal to ϵ with likelihood c. For instance, let $\epsilon = 0.01$ and $c = 0.95$. To estimate $Pr((\mathbf{s}_2, 1) \to (\mathbf{s}_3, 2))$, we need $\lceil(\sqrt{0.091(1-0.091)} \cdot \phi^{-1}((0.95+1/2)/0.01)^2\rceil = 3178$ samples. Further, this error will tend to 0 with probability 1 as N tends to ∞. There will be an additional error induced by the pth-order numerical integration method we use to compute the N trajectories. This error is $O(\Delta t^p)$ and will tend to 0 as Δt tends to 0 or p tends to ∞.

This error analysis is preliminary and more needs to be done as we remark in the concluding section.

4. Analysis methods

We now present some of the analysis techniques we have developed for the BDM representation. These techniques are founded on a basic Bayesian inference method realized via the *FF* (Factored Frontier) algorithm [20]. Specifically we develop parameter estimation and sensitivity analysis methods for the BDM model. Our goal here is not to develop new algorithms to solve these problems. Rather, we wish to demonstrate how standard techniques for tackling these problems can be adapted to the BDM framework in a straightforward manner.

4.1. Probabilistic inference

As pointed out earlier, although the dynamics defined by the ODEs is deterministic, to answer a basic query such as "*what will be the concentration of the protein x_i at time t?*" one will have to numerically generate a representative sample of trajectories and compute the average of the values for x_i at t yielded by the individual trajectories.

Using our BDM approximation, we can answer such a basic query and other more sophisticated queries by Bayesian inference. Specifically, given a Bayesian network, some observed evidence and some knowledge about the distribution of values of a set of variables, Bayesian inference aims to compute posterior distribution for a set of query variables. In our setting, the observed evidence will consist of known initial conditions and parameters as well as experimental data. Query variables will typically be selected random variables in the BDM. We adopt an approximate inference algorithm for DBNs known as the Factored Frontier (FF) algorithm [20].

The FF algorithm approximates, at each time point, joint distributions as products of marginal distributions. For example: $Pr(x_1^d, x_2^d, x_3^d) \approx \prod_{i=1}^{3} Pr(x_i^d)$. Hence the posterior distribution will be computed according to:

$$Pr(x_i^d|D) = \sum_I \left(Pr(x_i^d|Pa(x_i^d) = I) \prod_{u \in Pa(x_i^d)} Pr(u|D) \right). \tag{2}$$

Here $Pr(u|D)$ are the marginal distributions over the parents, D is the evidence, and $Pa(x_i)$ denotes the parents of x_i. The implementation of FF is straightforward. By storing $Pr(x_i^d|Pa(x_i^d))$ in the conditional probability tables and propagating $Pr(u|D)$ to the next time point, we can use (2) to compute $Pr(x_i^d|D)$. The time complexity of this algorithm is $O(\hat{d}(n+m)K^{R+1})$, where as before, K is the maximal number of intervals associated with a variable or rate constant's value domain and R is the maximal number of parents a node can have.

Using this algorithm, and with some additional computations, many queries can be answered. For instance, given the initial conditions, a single run of the FF algorithm will infer the marginal distributions of each variable at every time point. These probability distributions can then be used to validate the model by comparing them with experimental data. Flow cytometry data may provide direct information about the probability distributions of species concentration in a cell population. For such data, we may discretize it into distributions over intervals and supply it to the FF algorithm. On the other hand, western blot data, which is more common, will provide the averages of species concentration in a cell population. Suppose we have the data for x_i at time $d \cdot \Delta t$, denoted as $D_{x_i}^{d \cdot \Delta t}$. Note that the marginal distribution of x_i^d inferred by FF algorithm is over discrete values $\mathcal{I}_i$. To compute the real-valued "mean" of x_i^d that can be compared with $D_{x_i}^{d \cdot \Delta t}$, we identify each interval $I = [l, u)$ in $\mathcal{I}$ with its mid-point $(l + u)/2$. Then the expected value $E(x_i^d)$ can be computed and compared with $D_{x_i}^{d \cdot \Delta t}$.

4.2. Parameter estimation

Lack of knowledge about the parameters and hence the need to perform parameter estimation using limited data is a major bottleneck to pathway modeling. Current approaches to parameter estimation formulate it as a non-linear optimization problem [32]. A typical procedure will involve searching in a high dimensional solution space, in which each point represents a vector of parameter values. Whether a point is good or not is measured by an objective function, which will capture the difference between experimental data and prediction generated by simulations using the corresponding parameters.

For a large pathway model, one often needs to evaluate a very large number of solution points involving a numerical integration for each evaluation. This makes the process computationally intensive. The BDM representation allows us search for good parameter values in a hierarchical manner. Due to the discretized nature of the BDM, the solution space is transformed into a rectilinear grid tessellated by hyperrectangles that we call *blocks*. An important observation is that kinetic parameters are often robust [33]. In other words, the points around a good solution in the search space will also have relatively small objective values. Thus, instead of searching point by point in the solution space, we can first search for a few promising blocks and then take a closer look within these small blocks. Guided by this intuition, the general scheme of our "grid search" algorithm consists of two phases: (1) identify good blocks, (2) do a local search within candidate blocks. We note that phase (2) is necessary only when we aim to estimate parameters with finer granularity than the granularity of the BDM's discretization. Otherwise, one can skip phase (2) and return a probabilistic estimate (typically a Maximal Likelihood Estimate) of a combination of intervals of parameter values. For executing phase (1), we can apply any standard search algorithms over the discretized search space. As this space is much smaller than the original one, a simple direct search algorithm such as Hooke & Jeeves's search [34] can be adopted and the overall search process will only require a small number of evaluations of the objective function.

A block dictates a combination of intervals of parameter values. In order to evaluate the goodness of a block, we execute FF algorithm once by supplying the chosen parameter values — in terms of intervals — as evidence. Then the objective value can be computed by comparing the expected value of marginal distributions with the experimental data as described in

the previous subsection. Hence the parameter estimation process will only require a small number of executions of the FF algorithm and the overall running time will be significantly shortened.

Note that during BDM construction, if we do not have any knowledge about the prior distributions of unknown parameters, we can assume they are uniformly distributed within their ranges. Then after filling up all the entries of the CPTs of the BDM, the FF algorithm will be able to evaluate the goodness of any block in the discretized search space.

Example (*continued*)**.** Assume that only k_1 and k_2 are unknown parameters and that $k_3 \in [0.2, 0.4)$. Assume further that we have experimental data for S and P at time points $\{1, 2, 5, 10\}$. We then construct a BDM according to a prior distribution that the initial values $\mathbf{z}_0 = (S, E, ES, P, k_1, k_2, k_3)$ are uniformly distributed within the hypercube $[12, 15] \times [8, 10] \times [0, 2) \times [0, 3) \times [0, 1] \times [0, 1] \times [0.2, 0.4)$. Since $|\mathcal{I}_{k_1}| = |\mathcal{I}_{k_2}| = 5$, the solution space $[0, 1] \times [0, 1]$ is discretized into 25 blocks. In phase (1), we try to search for good blocks among the 25 blocks. Suppose we conduct a Hooke & Jeeves's search and evaluate blocks one by one. For instance, to evaluate block $([0.4, 0.6), [0.2, 0.4))$, we set $Pr(k_1 = [0.4, 0.6) = 1$ and $Pr(k_2 = [0.2, 0, 4) = 1$ (the distributions of S^0, E^0, ES^0, P^0, and k_3 are the same as the prior distribution for constructing the BDM) and execute FF algorithm. If the inferred distribution of S^{10} is $\{Pr(S^{10} = [12, 15]) = 0.6, Pr(S^{10} = [9, 12)) = 0.4\}$, we have $E(S^{10}) = (15 - 12)/2 \cdot 0.6 + (12 - 9)/2 \cdot 0.4 = 12.3$. Then we can compute the weighted square root error between $E(S^{10})$ and the available data. After searching suppose we find that $([0.2, 0.4), [0.4, 0.6))$ is the best block (i.e. it has the minimal objective value), we then can either execute phase (2) by searching within the solution space $[0.2, 0.4) \times [0.4, 0.6)$ or just return $\{k_1 \sim U(0.2, 0.4), k_2 \sim U(0.4, 0.6)\}$ as a probabilistic estimate (U stands for uniform distribution).

4.3. Global sensitivity analysis

Sensitivity analysis has been used to identify the critical parameters in signal transduction [35]. To overcome the limitations of traditional local sensitivity analysis methods, global methods have been proposed recently such as multi-parametric sensitivity analysis (MPSA) [36]. The MPSA procedure consists of: (1) draw samples from parameter space and for each combination of parameters, compute the weighted sum of squared error between experimental data and predictions generated by selected parameters; (2) classify the sampled parameter sets into two classes (good and bad) using a threshold error value; (3) plot the cumulative frequency of the parameter values associated with the two classes; (4) evaluate the sensitivities as the Kolmogorov–Smirnov statistic [37] of cumulative frequency curves.

Signaling pathway models often contain a large number of parameters. Hence it is necessary to sample a representative set from all possible combinations of parameter values. To improve this process, [38] adopts Latin Hypercube Sampling (LHS) since it requires fewer samples while guaranteeing that individual parameter ranges are evenly covered. Briefly, the range of each parameter is divided into K equal-sized intervals. Then for each parameter, one randomly samples K values, one from each interval of the parameter. Then to generate combinations of parameter values which are samples for MPSA, values are chosen in a random order from the K values for each parameter. This method helps to computationally manage the large number of parameters being varied simultaneously, while ensuring maximal sampling through each parameter dimension [39]. In our BDM setting, MPSA can be performed in a similar manner using LHS since the parameter space is discretized into blocks. In addition, the number of samples used to reach convergence is reduced since we can quickly evaluate the goodness of the whole block using the FF algorithm instead of having to draw samples from a block.

5. Two case studies

We have implemented the scheme for constructing the BDM in Java. To validate our techniques, we carried out two case studies. The first one involves a signaling network built by Brown et al [1], which aims to study the influence of the nerve growth factor (NGF) and the mitogenic epidermal growth factor (EGF) in rat pheochromocytoma (PC12) cells. The second case study deals with a signaling network with complicated dynamics studied by Goldbeter et al. [2] to investigate a remarkable example of biological rhythms, namely, the segmentation clock.

5.1. Case study 1: the EGF-NGF signaling pathway

PC12 cells are a valuable model system in neuroscience. They proliferate in response to EGF stimulation but differentiate into sympathetic neurons in response to NGF. This interesting phenomenon has been intensively studied [40]. It has been reported that the signal's specificity is correlated with different Erk dynamics. Specifically, a transient activation of Erk1/2 has been associated with cell proliferation, while a sustained activity has been linked to differentiation. How EGF and NGF affect the dynamics of active Erk through a network of intermediate signaling proteins is shown schematically in Fig. 4.

This model includes a common pathway to Erk through Ras shared by both the EGFR and NGFR, and also two important side branches through PI3K and C3G. This introduces multiple feedback loops leading to sophisticated dynamics. The ODE model of this pathway is available in the BioModels database [41]. It consists of 32 differential equations and 48 associated rate parameters (estimated from multiple sets of experimental data).

To construct the BDM, we first derived its graph from its ODEs (see Supplementary Table 1). We then discretized the ranges of each variable and parameter into 5 equal-size intervals and fixed the time step Δt to be 1 min. These choices were made mainly in order to proceed with the BDM construction smoothly but without trivializing the effort. Further, the

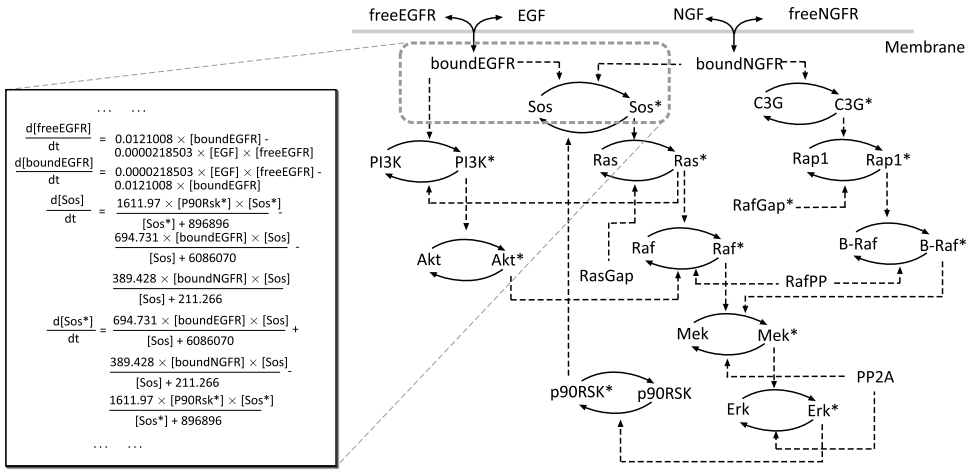

Fig. 4. EGF-NGF pathway [1].

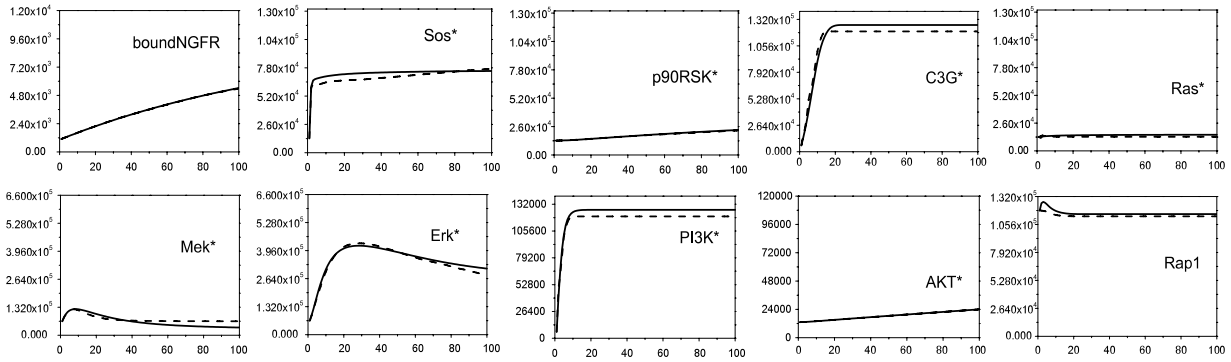

Fig. 5. Simulation results of EGF-NGF signaling pathway. Solid lines represent nominal profiles and dash lines represent BDM simulation profiles.

experimental data (western blot) is such that 5 uniform intervals seemed a reasonable choice. However our construction can be easily extended to non-uniform values intervals and time points. To fill up the conditional probability tables associated with the nodes, 3×10^6 trajectories were generated up to 100 mins by sampling initial states and parameters from the prior which are assumed to be uniform distributions over certain intervals (Supplementary Tables 2 and 3). Since we planned to study the effectiveness of our BDM based parameter estimation method (Section 5.1.2), we singled out 20 of the 48 parameters to be unknown (Supplementary Table 3) . When generating the 3×10^6 initial states, the sampled initial values of these parameters were chosen from their full range of possible values and not biased towards any specific intervals.

These samples were generated using the direct sampling method. We recall that in this method, the initial values of those trajectories are according to prior distribution (except for the parameters designated to be "unknown" as described above). Specifically, we randomly pick a value for each variable/parameter according to their marginal prior and then form a vector of initial values. The computational workload was distributed on 10 Opteron 2.2 GHz processors in a cluster. It took around 4 h to construct the BDM. All the subsequent experiments reported below were done using an Intel Xeon 2.8 GHz processor.

5.1.1. Probabilistic inference

To test the quality of our approximation, we implemented Monte Carlo integration for the ODE model to get good estimates by sampling and averaging. Specifically, we numerically generated a number of random trajectories — according to the prior — using ODEs and computed the average values of the variables at the chosen time points. Our experiments show that the average values converge when the number of random trajectories generated is roughly 10^4. The averaged trajectories projected to individual protein concentration time series values are termed to be the nominal simulation profiles. Using the implemented FF algorithm the mean of each variable over time was computed. In doing so, for the 20 parameters which were assumed to be unknown during the BDM construction process, their values were presented as specific intervals (derived from the original ODE model) in the form of evidence.

The time profiles resulting from the execution of the FF algorithm are termed to be the BDM-simulation profiles. As summarized in Fig. 5, our BDM-simulation profiles fit the nominal simulation profiles quite well for most of the cases.

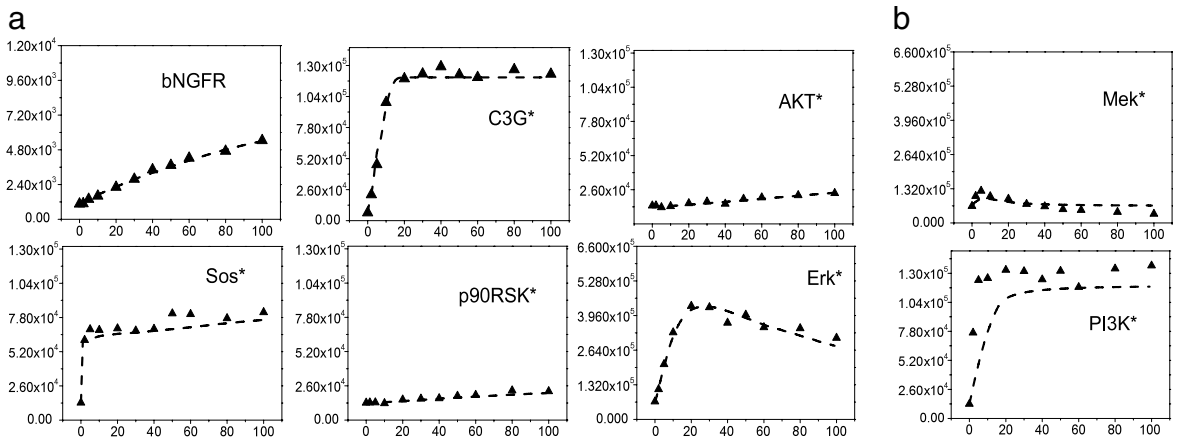

Fig. 6. Parameter estimation results. (a) BDM-simulation profiles vs. training data. (b) BDM-simulation profiles vs. test data.

In terms of running time, a single execution of FF inference required 0.08 s while generating a stable nominal profile requires 105.4 s. Thus, the total computation time will be sharply reduced for our approach when many such "queries" need to be answered.

5.1.2. Parameter estimation

In order to test the performance of the BDM-based parameter estimation method, we synthesized experimental time series data for 9 (out of 32) proteins {bounded EGFR, bounded NGFR, active Sos, active C3G, active Akt, active p90RSK, active Erk, active Mek, active PI3K}, measured at the time points {2, 5, 10, 20, 30, 40, 50, 60, 80, 100} (min). This data was synthesized using prior knowledge about initial conditions and parameters (see Supplementary Tables 2 and 3). To mimic western blot data which is cell population based, we first averaged 10^4 random trajectories generated by sampling initial states and rate constants, and then added observation noise with variance 5% to the simulated values. With the assumed measurement precision, these values were discretized into 5 intervals, which represent the concentration levels in western blot data. We reserved the data of 7 proteins for training the parameters and reserved the rest data for testing the quality of the estimated parameter values.

With 20 of the 48 parameters having been designated during the BDM construction as being unknown, the Hooke & Jeeves algorithm was implemented to search in the discretized parameter space. The estimated parameter values in terms of maximal likelihoods of certain combination of interval values (of the 20 unknown parameters) can be found in Supplementary Table 4. As shown in Fig. 6, the BDM-simulation profiles generated using the estimated parameters matches the training data as shown and also has good agreement with the test data.

We compared the efficiency and quality of our results with the following ODE based optimization algorithms: Levenberg–Marquardt (LM) [42], Genetic Algorithm (GA) [43], Stochastic Ranking Evolutionary Strategy (SRES) [44], and Particle Swarm Optimization (PSO) [45]. These optimization algorithms were executed using the COPASI [46] tool. We scored the resulting parameters obtained from all the algorithms using the weighted sum-of-squares *difference* between the experimental data and the corresponding simulation profiles (i.e. low scores correspond to low errors). The results are summarized in Fig. 7, which suggests that our method achieves a good balance between accuracy and performance. We also note that the cost of constructing the BDM representation gets rapidly amortized.

5.1.3. Global sensitivity analysis

We modified and implemented the MPSA method for the BDM setting. Using the same experimental data set introduced in previous subsection, the global sensitivities (K-S statistics) of the rate constants were computed. The results are shown in Fig. 8. The cumulative frequency distributions for the acceptable and unacceptable cases can be found in [22]. Specifically, the reactions involved in the phosphorylation of Erk (k_{23}), Mek (k_{17}), Akt (k_{34}) and p90RSK (k_{28}) have the highest sensitivities, indicating that these reactions affect the system behavior most directly. These results are consistent with previous findings [40,47].

The MPSA method adopts Monte Carlo strategy for the ODE model. We recorded the running time of the algorithm until the K-S values converged. The total running time of the ODEs based MPSA method was about 22 h, while the MPSA method based on the BDM required only 34 min. Thus the cost of constructing the BDM can be easily recovered when one performs parameter estimation followed by sensitivity analysis.

5.1.4. Different discretizations

To evaluate the effects of different discretizations, we constructed BDMs for the EGF-NGF pathway by fixing K intervals for each variable, with K ranging over {3, 4, 5, 6, 7, 8}. We then computed the mean of each variable over time using the FF

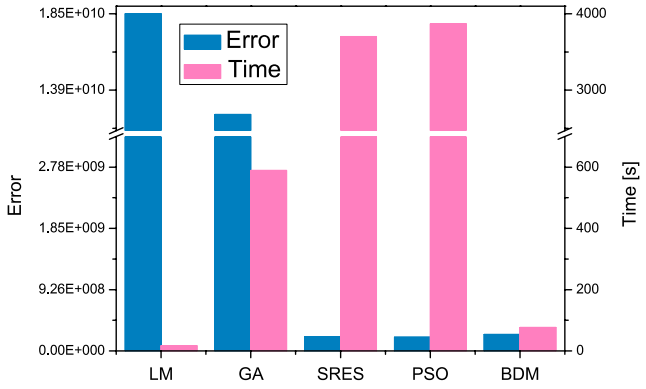

Fig. 7. Performance comparison of our parameter estimation method (BDM) and 4 other methods.

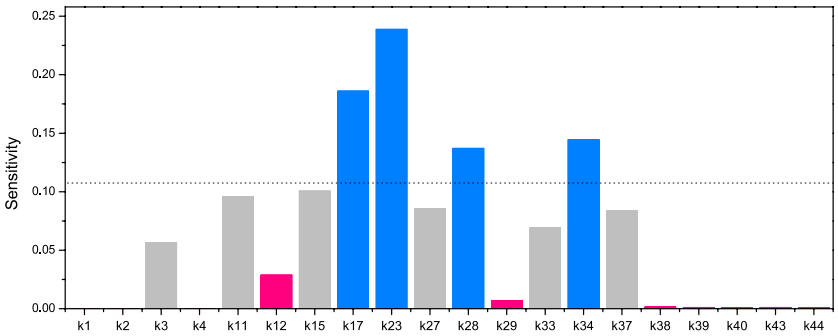

Fig. 8. Parameter sensitivities.

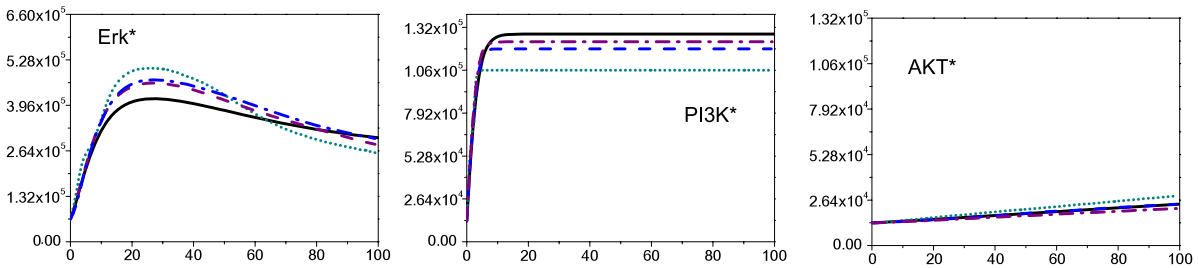

Fig. 9. The effects of different discretizations. Solid black lines represent nominal profiles, dash-dotted purple lines present BDM profiles with $K = 8$, dashed blue lines present BDM profiles with $K = 5$, dotted cyan lines present BDM profiles with $K = 3$. (b) Accuracy and efficiency comparison of different discretizations. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

algorithm for each BDM. The resulting profiles were compared with the nominal profiles. The comparison results are shown in Fig. 9. As might be expected, as K increases, the quality of our approximations will improve. However, since the time and space complexity of BDM based analysis depends on K, there is a tradeoff between efficiency and accuracy. To help decide on a good value of K, we scored the discretizations with different Ks using the weight sum-of-square difference between nominal profiles and BDM profiles, and measured the running time of a single FF inference. The results are summarized in Fig. 10 showing that discretizations with 5 or 6 intervals might be good choices, at least in the present context.

5.1.5. Equation sampling

We also implemented the equation sampling method described in Section 3 that provides a coverage of J samples for each possible combination of interval values of the unknown parameters in each equation. Using this method, we generated 495,000 trajectories to get a coverage of 1000 per combination. Fig. 11 shows the comparison of time profiles generated using the two sampling methods. The two set of profiles are nearly indistinguishable, suggesting that the equation sampling

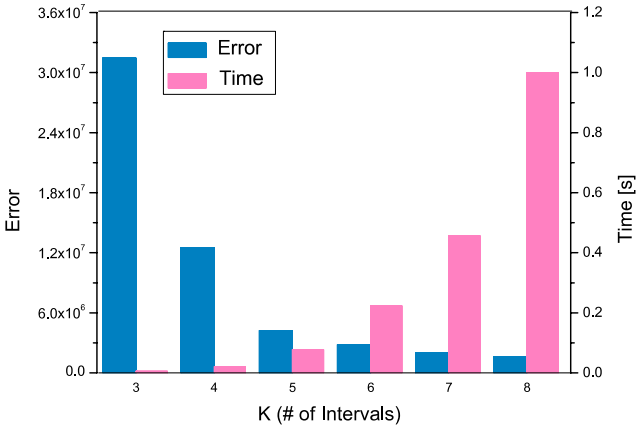

Fig. 10. Accuracy and efficiency comparison of different discretizations.

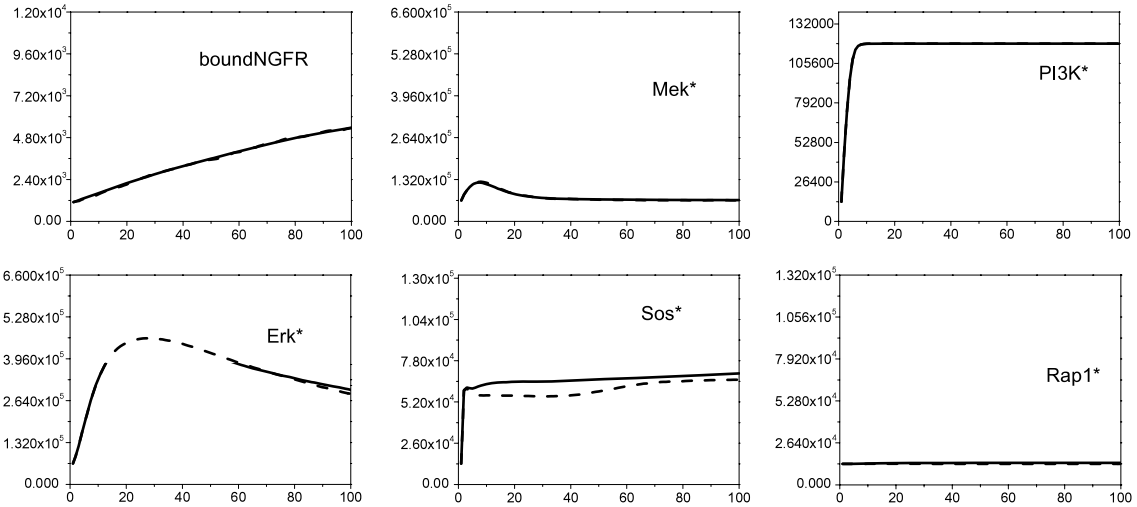

Fig. 11. The comparison of two sampling methods. Solid lines represent direct sampling with 3 millions samples and dash lines present J-coverage sampling with $J = 1000$.

can efficiently reduce the number of samples required. This also motivated us to conduct our next case study using this method.

5.2. Case study 2: segmentation clock network

In the developing vertebrate embryos, the segmental pattern of the spine is established when the somites are rhythmically produced. The periodic formation process of somites is governed by an oscillator called the segmentation clock, which drives the oscillatory expression of a large network of signaling genes [48]. The underlying signaling network proposed by Goldbeter et al. is shown in Fig. 12. It couples three oscillating pathways consisting of the FGF, Wnt and Notch signaling pathways, whose periodic behaviors are produced by negative feedback loops. The corresponding ODE model can be accessed in the BioModels database [41]. It includes 22 differential equations and 75 associated rate parameters. Again, anticipating our goal of evaluating the BDM based parameter estimation method, 40 of the 75 parameter values were singled out to be unknown. The rest of the experiments were conducted as described in our first case study.

We first constructed a BDM for the segmentation clock model. The graph of BDM is shown in Supplementary Table 5. Similar to the previous case study, we discretized the ranges of each variable and parameter into 5 equal-size intervals and fixed the time step Δt to be 5 min. To provide an equation of coverage of 1000 per combination, 2585,000 trajectories were generated up to 500 mins by sampling the prior (Supplementary Tables 6 and 7).

It is worth noting that even though equation sampling was used, a much larger sample size (in relation to the first case study in the context of equation sampling) is required due to the larger number of parameters and the "fatness" (i.e. the number of parameters appearing on the right-hand side) of some of the equations. The construction process consumed around 3.1 h on a cluster consisted of 10 processors.

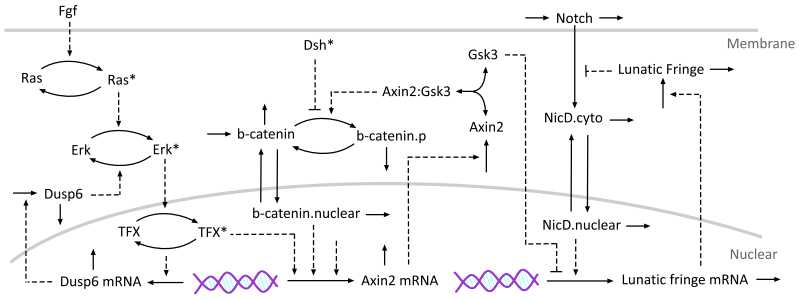

Fig. 12. Segmentation clock pathway [2].

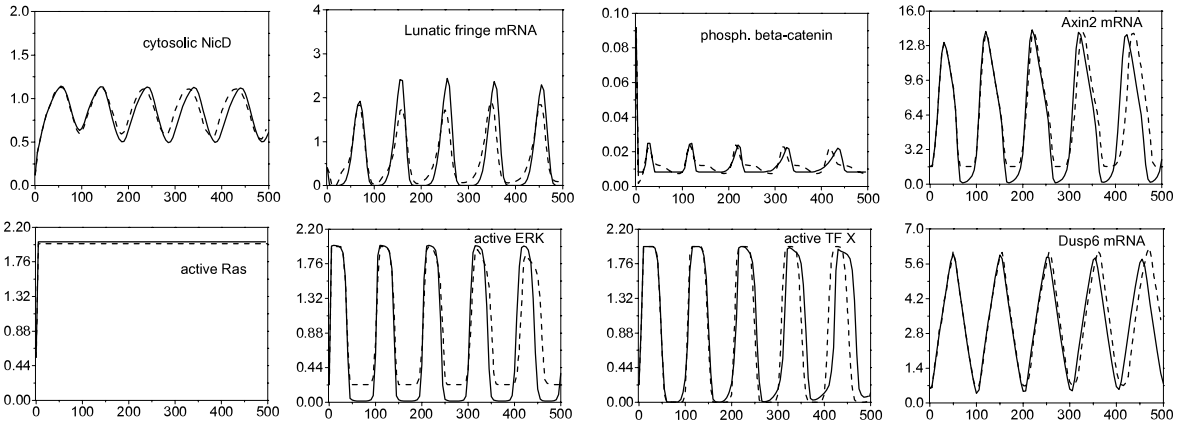

Fig. 13. Simulation results of segmentation clock pathway. Solid lines represent nominal profiles and dash lines represent BDM simulation profiles.

5.2.1. Probabilistic inference

To generate stable nominal profiles, we averaged 10^4 trajectories according to the prior. The nominal profiles were then compared with the BDM-simulation profiles computed from the FF inference results. The comparison results are shown in Fig. 13, which shows a good fit between them. In terms of running time, a single execution of FF inference required 0.01 s while generating a stable nominal profile took 407.3 s.

5.2.2. Parameter estimation

We next tested the performance of the BDM-based parameter estimation method. We synthesized population based experimental time series data for 8 (out of 22) proteins {Notch protein, nuclear NicD, Lunatic fringe mRNA, Axin2 mRNA, active ERK, Dusp6 mRNA, Dusp6 protein, cytosolic NicD}, measured at the time points {400, 410, 420, 430, 440, 450, 460, 470, 480, 490} (min) based on the prior knowledge about initial conditions and parameters (see Supplementary Tables 6 and 7). We averaged 10^4 random trajectories generated by sampling initial states and rate constants, and then added observation noise with variance 5% to the simulated values. The data of 6 proteins were reserved for training the parameters and the rest data were used for testing the quality of the estimated parameter values.

For the 40 parameters which had been designated to be unknown, the BDM based implementation of Hooke & Jeeves algorithm was applied. The results can be found in Supplementary Table 8. As shown in Fig. 14 (a) and (b), the BDM-simulation profiles generated using the estimated parameters obtained (with the match to training data as shown) has good agreement with the test data.

We then compared the efficiency and quality of our results with the ODE based optimization algorithms: LM, GA, SRES, and PSO introduced in previous subsection. The results are summarized in Fig. 14(c) suggesting again that our method achieves a good balance between accuracy and performance and the cost of constructing the BDM representation gets rapidly amortized.

5.2.3. Global sensitivity analysis

The global sensitivities of the parameters were computed using a BDM based MPSA method and are shown in Fig. 15. Specifically, the reactions involved in the degradation of Dusp6 mRNA (k_{34}), the transcription of Dusp6 gene induced by TF X (k_{33}) and the transcription of the Axin2 gene induced by factor TF X (k_{18}) have the highest sensitivities, indicating that these reactions affect the system's behavior most directly. Since all these reactions are present in the FGF pathway, we

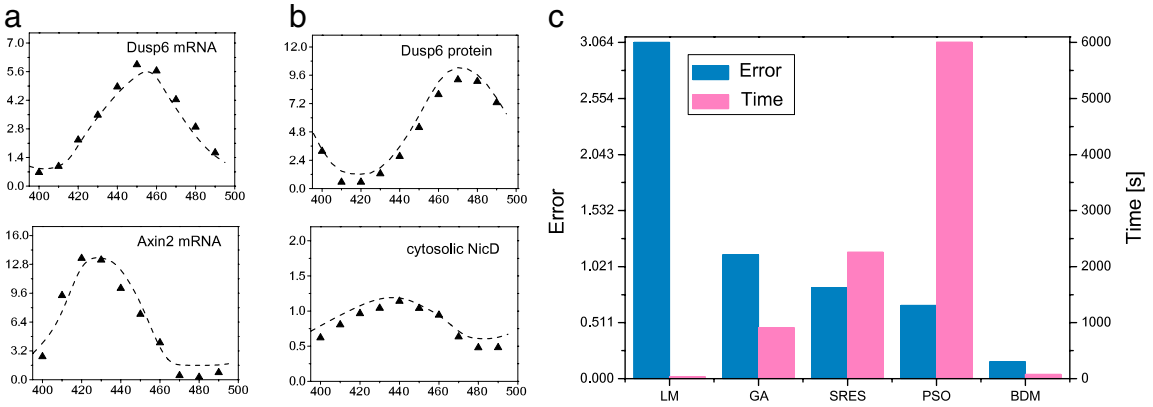

Fig. 14. Parameter estimation results. (a) BDM-simulation profiles vs. training data. (b) BDM-simulation profiles vs. test data. (c) Performance comparison of our parameter estimation method (BDM) and 4 other methods.

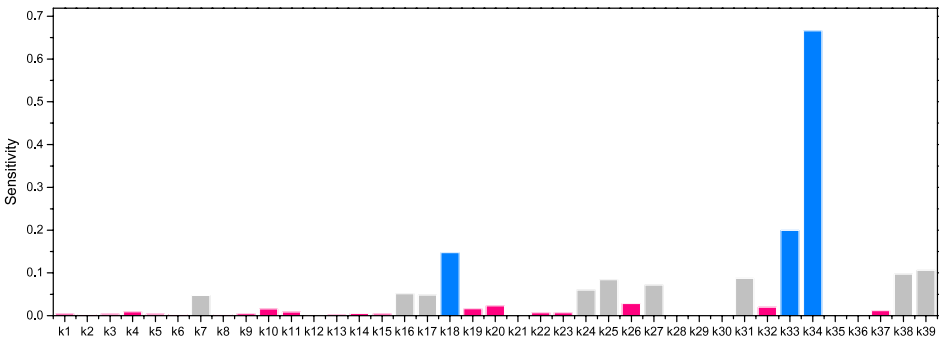

Fig. 15. Parameter sensitivities.

hypothesize that FGF pathway is the key regulatory mechanism that drives and synchronizes the oscillatory gene expression of the segmentation clock.

The total running time of the ODE based MPSA method was about 81.25 h, while the MPSA method based on the BDM required only 3.25 h.

6. Discussion

We have proposed a probabilistic approximation scheme for signaling pathway dynamics specified as a system of ODEs. Assuming a discretization and an initial distribution, it consists of pre-computing and storing a representative sample of trajectories induced by the system of ODEs. We use a dynamic Bayesian network representation, called the Bayesian Dynamics Model, to compactly represent these trajectories by exploiting the pathway structure. Basically, the underlying graph of the BDM captures the dependencies of the variables on other variables and rate constants as defined by the system of ODEs. Due to the probabilistic graphical representation, a variety of analysis questions concerning the pathway dynamics traditionally addressed using Monte Carlo simulations can be converted to Bayesian inference and solved more efficiently. Using the FF algorithm for doing basic Bayesian inference, we have adapted standard parameter estimation and sensitivity analysis algorithms to the BDM setting.

We have demonstrated the applicability of our techniques with the help of the EGF-NGF signaling pathway and segmentation clock pathway. The BDM approximation successfully captured the dynamics of the two pathways so that queries about cell population behavior can be quickly answered. We showed that with our model the unknown rate constants can be efficiently estimated from noisy experimental data. We also gained insights about the pathway dynamics by identifying critical parameters in signal transduction via rapid global sensitivity analysis. At the end of performing these analysis tasks we had easily regained the initial computational investment made to construct the BDM model.

Apart from its computational efficiency, it is worth noting that the BDM model is a more realistic model for recording the current state of knowledge about a bio-pathway. In particular, the probabilistic and intervals-based estimates it returns will better match the noisy and incremental fashion in which experimental data is usually generated.

Turning to future work, a crucial ingredient in the construction of the BDM is the family of sample trajectories that are needed to get a good approximation. Our error analysis can provide the lower bound of the sample size only for the individual

states and transitions. More work is needed to develop sampling schemes using which good lower bounds for global error estimates can be achieved.

We need to apply our method to a variety of pathway models. We are currently doing so in collaboration with biologists in the settings of Complement pathways, Apoptosis/Autophage pathways and DNA damage/repair pathways. Further, it will be useful to augment the ODE model with some discrete features but this should be easy to achieve. A more challenging issue is to abstract the BDM representation to an input–output transducer so that one can efficiently model networks of pathways and inter-cellular interactions models. It is also important to develop formal verification techniques based on the BDM representation. In this context, we note that the FF algorithm can compute the marginal probabilities of the discretized values of variables at specific time points. Hence it will be appropriate to develop probabilistic bounded model checking methods for the BDM model and we are beginning to pursue this. Finally, our approximation technique might have a wider applicability. A rich class of dynamical systems can be captured via ODEs and in a variety of situations it may be appropriate and useful to abstract their behaviors as dynamic Bayesian networks as we have done here.

References

[1] K.S. Brown, C.C. Hill, G. A. Calero, K. H. Lee, J. P. Sethna, R. A. Cerione, The statistical mechanics of complex signaling networks: nerve growth factor signaling, Physical Biology 1 (2004) 184–195.
[2] A. Goldbeter, O. Pourquie, Modeling the segmentation clock as a network of coupled oscillations in the notch, wnt and fgf signaling pathways, Journal of Theoretical Biology 252 (2008) 574–585.
[3] B.B. Aldridge, J.M. Burke, D.A. Lauffenburger, P.K. Sorger, Physicochemical modelling of cell signalling pathways, Nature Cell Biology 8 (2006) 1195–1203.
[4] K.P. Murphy, Dynamic Bayesian networks: representation, inference and learning, Ph.D. Thesis, University of California, Berkeley, 2002.
[5] H. Matsuno, Y. Tanaka, H. Aoshima, A. Doi, M. Matsui, S. Miyano, Biopathways representation and simulation on hybrid functional Petri net, In Silico Biology 3 (3) (2003) 389–404.
[6] M. Antoniotti, A. Policriti, N. Ugel, B. Mishra, XS-systems: extended s-systems and algebraic differential automata for modeling cellular behavior, in: S. Sahni, V.K. Prasanna, U. Shukla (Eds.), HiPC, in: Lecture Notes in Computer Science, vol. 2552, Springer, 2002, pp. 431–442.
[7] H. de Jong, M. Page, Search for steady states of piecewise-linear differential equation models of genetic regulatory networks, IEEE/ACM Transactions on Computational Biology and Bioinformatics 5 (2) (2008) 208–223.
[8] R. Ghosh, C. Tomlin, Symbolic reachable set computation of piecewise affine hybrid automata and its application to biological modelling: delta-notch protein signalling, Systems Biology 1 (1) (2004) 170–183.
[9] M. Calder, S. Gilmore, J. Hillston, Modelling the influence of RKIP on the ERK signalling pathway using the stochastic process algebra PEPA, Transactions on Computational Systems Biology VII 4230 (2006) 1–23.
[10] M. Calder, V. Vyshemirsky, D. Gilbert, R. Orton, Analysis of signalling pathways using continuous time Markov chains, Transactions on Computational Systems Biology VI 4220 (2006) 44–67.
[11] F. Ciocchetta, A. Degasperi, J. Hillston, M. Calder, Some investigations concerning the CTMC and the ODE model derived from Bio-PEPA, Electronic Notes in Theoretical Computer Science 229 (2009) 145–163.
[12] J. Hillston, A Compositional Approach to Performance Modelling, University Press, 1996.
[13] M.Z. Kwiatkowska, G. Norman, D. Parker, PRISM: probabilistic symbolic model checker, in: T. Field, P.G. Harrison, J.T. Bradley, U. Harder (Eds.), Computer Performance Evaluation/TOOLS, in: Lecture Notes in Computer Science, vol. 2324, Springer, 2002, pp. 200–204.
[14] U. Nodelman, C.R. Shelton, D. Koller, Continuous time Bayesian networks, in: Proceedings of the 18th Conference in Uncertainty in Artificial Intelligence, UAI'02, 2002, pp. 378–387.
[15] S.J. Russell, P. Norvig, Artificial Intelligence: A Modern Approach, 3rd Edition, Prentice Hall, 2003.
[16] C. Langmead, S. Jha, E. Clarke, Temporal logics as query languages for dynamic Bayesian networks: Application to D. Melanogaster embryo development, Tech. rep., Carnegie Mellon University, 2006.
[17] E.M. Clarke, J.R. Faeder, C.J. Langmead, L.A. Harris, S.K. Jha, A. Legay, Statistical model checking in BioLab: applications to the automated analysis of T-Cell receptor signaling pathway, in: M. Heiner, A.M. Uhrmacher (Eds.), CMSB, in: Lecture Notes in Computer Science, vol. 5307, Springer, 2008, pp. 231–250.
[18] J. Heath, M. Kwiatkowska, G. Norman, D. Parker, O. Tymchyshyn, Probabilistic model checking of complex biological pathways, Theoretical Computer Science 319 (3) (2008) 239–257.
[19] N. Geisweiller, J. Hillston, M. Stenico, Relating continuous and discrete PEPA models of signalling pathways, Theoretical Computer Science 404 (2) (2008) 97–111.
[20] K.P. Murphy, Y. Weiss, The factored frontier algorithm for approximate inference in DBNs, in: Proceedings of the 17th Conference in Uncertainty in Artificial Intelligence, San Francisco, CA, USA, 2001, pp. 378–385.
[21] B. Liu, P.S. Thiagarajan, D. Hsu, Probabilistic approximations of signaling pathway dynamics, in: P. Degano, R. Gorrieri (Eds.), CMSB, in: Lecture Notes in Computer Science, vol. 5688, Springer, 2009, pp. 251–265.
[22] Supplementary Materials, http://www.comp.nus.edu.sg/~rpsysbio/tcs10.
[23] M.W. Hirsch, S. Smale, R.L. Devaney, Differential Equations, Dynamical Systems and In Introduction to Chaos, 2nd Edition, Elsevier, 2004.
[24] H. Ammann, Ordinary Differential Equations: An Introduction to Nonlinear Analysis, Walter de Gruyter, 1990.
[25] R. Durrett, Probability: Theory and Examples, Duxbury Press, 2004.
[26] J. Feldman, Review of measurable functions, University of British Columbia.
[27] V. Bryant, Metric Spaces: Iteration and Application, Cambridge University Press, 1985.
[28] E. Klipp, R. Herwig, A. Kowald, C. Wierling, H. Lehrach, Systems Biology in Practice: Concepts, in: Implementation and Application, Wiley-VCH, 2005.
[29] J.R. Norris, Markov Chains, Cambridge University Press, 1997.
[30] L. Stryer, Biochemistry, W. H. Freeman, New York, 1988.
[31] L.M. Nunez, On the relationship between temporal Bayes networks and Markov chains, Master's thesis, Brown University, 1989.
[32] J.R. Banga, Optimization in computational systems biology, BMC Systems Biology 2 (47) (2008) 1–7.
[33] R.N. Gutenkunst, J.J. Waterfall, F.P. Casey, K.S. Brown, C.R. Myers, J.P. Sethna, Universally sloppy parameter sensitivities in systems biology, PLoS Computational Biology 3 (10) (2007) 189.
[34] R. Hooke, T.A. Jeeves, "Direct search" solution of numerical and statistical problems, Journal of the Association for Computing Machinery 8 (1961) 212–229.
[35] N.A. van Riel, Dynamic modelling and analysis of biochemical networks: mechanism-based models and model-based experiments, Briefings in Bioinformatics 7 (4) (2006) 364–374.
[36] K.H. Cho, S.Y. Shin, W. Kolch, O. Wolkenhauer, Experimental design in systems biology, based on parameter sensitivity analysis using a Monte Carlo method: a case study for the TNFα-mediated NF-κB signal transduction pathway, Simulation 79 (12) (2003) 726–739.
[37] D.J. Sheskin, Handbook of Parametric and Nonparametric Statistical Procedures, 3rd Edition, Chapman & Hall/CRC, 2004.

[38] Z. Zi, K.H. Cho, M.H. Sung, X. Xia, J. Zheng, Z. Sun, In silico identification of the key components and steps in ifn-c induced jak-stat signaling pathway, FEBS Letters 579 (5) (2005) 1101–1108.
[39] M. McKay, R. Beckman, W. Conover, A comparison of three methods for selecting values of input variables in the analysis of output from a computer code, Technometrics 42 (1) (2000) 55–61.
[40] B.N. Kholodenko, Untangling the signalling wires, Nature Cell Biology 9 (3) (2007) 247–249.
[41] N. Le Novere, B. Bornstein, A. Broicher, M. Courtot, M. Donizelli, H. Dharuri, L. Li, H. Sauro, M. Schilstra, B. Shapiro, J. Snoep, M. Hucka, BioModels database: a free, centralized database of curated, published, quantitative kinetic models of biochemical and cellular systems, Nucleic Acids Research 34 (2006) D689–D691.
[42] K. Levenberg, A method for the solution of certain nonlinear problems in least squares, Quart. Appl. Math. (2) (1994) 164–168.
[43] T. Back, D. Fogel, Z. Michalewicz, Handbook of Evolutionary Computation, Oxford University Press, 1997.
[44] T. Runarsson, X. Yao, Stochastic ranking for constrained evolutionary optimization, IEEE Transactions on Evolutionary Computation 4 (2000) 284–294.
[45] J. Kennedy, R. Eberhart, Particle Swarm Optimization, in: Proceedings of the Fourth IEEE International Conference on Neural Networks, Perth, Australia, 1995, pp. 1942–1948.
[46] S. Hoops, S. Sahle, R. Gauges, C. Lee, J. Pahle, N. Simus, M. Singhal, L. Xu, P. Mendes, U. Kummer, COPASI - a COmplex PAthway SImulator, Bioinformatics 22 (24) (2006) 3067–3074.
[47] C.S. Babu, S. Yoon, H.-S. Nam, Y. Yoo, Simulation and sensitivity analysis of phosphorylation of EGFR signal transduction pathway in PC12 cell model, IEE Systems Biology 1 (2) (2004) 213–221.
[48] M.-L. Dequeant, E. Glynn, K. Gaudenz, M. Wahl, J. Chen, A. Musheqian, O. Pourquie, A complex oscillating network of signaling genes underlies the mouse segmentation clock, Science 314 (2006) 1595–1598.