Theoretical Computer Science 412 (2011) 2840–2853

Contents lists available at ScienceDirect

Theoretical Computer Science

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

Component-based construction of bio-pathway models: The parameter estimation problem

Geoffrey Koh[a], David Hsu[b], P.S. Thiagarajan[b,*]

[a] Bioprocessing Technology Institute, Agency for Science, Technology and Research, 20 Biopolis Way, #06-01 Centros, Singapore 138668, Singapore

[b] Department of Computer Science, National University of Singapore, Singapore

ARTICLE INFO

Keywords:
Biochemical networks
ODEs
Components
Factor graphs
Belief propagation
Decomposition
Composition

ABSTRACT

Constructing and analyzing large biological pathway models is a significant challenge. We propose a general approach that exploits the structure of a pathway to identify pathway components, constructs the component models, and finally assembles the component models into a global pathway model. Specifically, we apply this approach to pathway parameter estimation, a main step in pathway model construction. A large biological pathway often involves many unknown parameters and the resulting high-dimensional search space poses a major computational difficulty. By exploiting the structure of a pathway and the distribution of available experimental data over the pathway, we decompose a pathway into components and perform parameter estimation for each component. However, some parameters may belong to multiple components. Independent parameter estimates from different components may be in conflict for such parameters. To reconcile these conflicts, we represent each component as a *factor graph*, a standard probabilistic graphical model. We then combine the resulting factor graphs and use a probabilistic inference technique called *belief propagation* to obtain the maximally likely parameter values that are globally consistent. We validate our approach on a synthetic pathway model based on the Akt-MAPK signaling pathways. The results indicate that the approach can potentially scale up to large pathway models.

1. Introduction

Biological pathways are inherently complex. To understand the functioning of these pathways, we not only need to identify the constituent elements and their interactions, but also how their dynamics evolve over time [1,2]. A common approach is to view the pathway as a network of bio-chemical reactions and model them as a system of ordinary differential equations (ODEs). One then attempts to analyze this system of ODEs to gain insights into the pathway dynamics. This approach can be extended to include discrete aspects as well [3].

However, this model construction method must overcome several barriers. Firstly, the values of the rate constants of all the bio-chemical reactions and the initial concentration levels of the reactants are needed to complete the model. Unfortunately only a few of these parameters will be known or can be measured experimentally. The rest must be estimated by fitting the model to experimental data. The amount of such data will be limited and noisy. Consequently, parameter estimation for large bio-pathways is a difficult problem and applying a decomposition based technique is an

[*] Corresponding author.
E-mail addresses: geoffrey_koh@bti.a-star.edu.sg (G. Koh), 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.2011.02.013

attractive option. This however raises the issue of integration. Different components of a model may share parts whose dynamics vary according to the components they belong to. One will have to reconcile these differences to obtain a globally consistent model of dynamics. Secondly, model construction is often an ongoing process. As new experimental data arrives, it will have to be incorporated into a new model. Instead of having to start from scratch each time, one would like to have techniques by which the existing model is updated to yield the new model. Finally, one may also want to integrate a number of existing models constructed independently by adding cross-talks or feedback loops. These considerations lead to basic questions as to what a component is and what algorithmic techniques are needed to compose component models.

Our main goal here is to explore these issues in the context of parameter estimation. Specifically, we decompose a pathway model into components to achieve dimension reduction of the search space for parameter values. We then carry out parameter estimation for the individual components. We then represent the estimates for each component as a *factor graph*, a well-known probabilistic graphical model and apply a probabilistic inference technique called *belief propagation* [4] to reconcile the different estimates for the parameters shared by different components to obtain a globally consistent set of estimated values. We wish to emphasize that our technique is *orthogonal* to the specific method used for solving the parameter estimation problem for the individual components. Further, although we focus on systems of ODEs here, it should be possible to extend our method to related system models such as Hybrid Functional Petri Nets [3].

The present paper extends our previous work [5,6] in several ways. The decomposition based estimation technique in [5] was based on a component notion which depended on the structure of the pathway model as well as the distribution of experimental data. Here instead, we define a purely structural notion of components. Different subsets of components can constitute the decomposition of the model. We exploit the distribution of experimental data to guide the choice of components to form the decomposition. In addition, the method that we present here yields a canonical decomposition. In contrast, choosing one decomposition from the several possible ones was based on pragmatic considerations in [5]. More crucially, the problem of reconciling conflicting estimates of parameters belonging to more than one component was resolved in an *ad hoc* and local manner in [5] by arbitrarily designating one of the estimated values as the "true" one.

Later, in [6] we identified belief propagation as a way to integrate conflicting parameter estimates of the parameters shared by different components. However we explored this in a setting where the components were allowed to be related to each other only in restricted ways. Here we remove such restrictions.

Parameter estimation commonly involves searching a high-dimensional space for a suitable vector of parameter values, which, when plugged into the model, yields a behavior that matches the given experimental data well. Hence, the problem is often formulated as a nonlinear optimization problem with differential-algebraic constraints. Several algorithms have been designed in this setting to estimate the parameters given experimental data, and it continues to be investigated [7–10]. Interestingly, the formal verification technique called model checking is also being deployed to attack this problem [11,12]. However, the estimated parameters are required to be consistent only relative to the dynamic properties captured by the specified temporal logic formulas. As pointed out above, our decomposition–estimation–integration technique can be combined with these various approaches.

The rest of this paper is organized as follows. In the next section we introduce the ODE-based representation of a bio-chemical network. We then formulate the graphical model called RNGs (Reaction Network Graphs) to capture the coupling of the variables induced by the ODEs systems. In the subsequent section we present our decomposition method. In Section 4 we introduce factor graphs. We show how a factor graph can be constructed for each of the components present in a decomposition. We also describe how globally consistent parameter estimates can be obtained by piecing together the factor graphs corresponding to the individual components and applying belief propagation. In Section 5 we use our technique to carry out the parameter estimation of a simplified model of well-studied Akt–MAPK pathway. Our main aim in this case study is to show how belief propagation based integration will play out in a realistic setting. In the final section, we summarize our results and discuss prospects for future work.

2. The problem setting

We begin with a system of ODEs as the basic system model and introduce Reaction Network Graphs (RNGs) to capture the structure of the underlying pathway. Our notion of components and the decomposition procedure will be developed using RNGs.

We shall assume a bio-pathway is presented as a network of bio-chemical reactions which in turn is modeled as a system of Ordinary Differential Equations (ODEs). There will be one equation corresponding to each molecular species describing how the concentration of this species is increased (decreased) by each reaction that produces (consumes) this species. As a simple example consider the enzyme–catalyzed reaction scheme in (1). In this system, enzyme E binds reversibly to substrate S to form the intermediate complex ES. Within the complex, enzyme E then converts S into product P before releasing it. The parameters k, k', and k'' are the rate constants that govern the speed of the reactions.

$$E + S \underset{k'}{\overset{k}{\rightleftharpoons}} ES \overset{k''}{\rightarrow} E + P \tag{1}$$

The corresponding system of ODEs will be:

$$\frac{dS}{dt} = -k \cdot S \cdot E + k' \cdot ES$$
$$\frac{dE}{dt} = -k \cdot S \cdot E + (k' + k'') \cdot ES$$
$$\frac{dES}{dt} = k \cdot S \cdot E - (k' + k'') \cdot ES$$
$$\frac{dP}{dt} = k'' \cdot ES$$

Above and elsewhere we will abuse notation for convenience and use E, ES etc. to denote both the name of a molecular species and the real-valued variable denoting its concentration as a function of time. For typical bio-pathways, many of the rate constants associated with the reactions will be unknown; they will have to be estimated using available experimental data. As mentioned in the introduction, several parameter estimation methods are available for ODE-based models. The basic strategy in all these methods may be viewed as a systematic search through the parameter value space. For instance, in evolutionary strategies one maintains a small population of values for the parameters being estimated. After scoring them according to their fitness to experimental data, one eliminates a few unfit ones and introduces new members to the population – systematically or otherwise – by "mutating" the values of some of the members in the current population [13,14].

Formally, the parameter estimation problem can be formulated as an optimization problem with differential-algebraic constraints. It boils down to minimizing an objective function that measures fitness to data:

$$J(\mathbf{k}) = \sum_{i,j} w_{ij}(x_i(t_j; \mathbf{k}) - \tilde{x}_{ij})^2 \tag{2}$$

subject to

$$\dot{\mathbf{x}} = F(\mathbf{x}, \mathbf{k}) \tag{3}$$
$$h(\mathbf{x}, t) > 0 \tag{4}$$
$$\mathbf{k}^L \leq \mathbf{k} \leq \mathbf{k}^U \tag{5}$$

$\tilde{x}_{ij}$ denotes the measured concentration level of molecular species x_i at time t_j. $x_i(t_j; \mathbf{k})$ is the corresponding value generated by the model using a particular parameter combination vector $\mathbf{k}$. The nonlinear function F describes the system dynamics. h is the constraints on the concentration levels for each molecule species $x \in \mathbf{x}$ while $\mathbf{k}^L$ and $\mathbf{k}^U$ are the lower and upper bound constraints imposed on the parameters. The variable w_{ij} is the weight that is used to normalize the contributions of each term to the objective function.

Unfortunately, the size of the search space will be exponential in the number of unknown parameters. For typical pathway models, this space can hence become prohibitively large with many local minima. This motivates the dimension reduction technique based on decompositions that we wish to develop. The starting point is a directed graph that captures the dependencies between variables.

2.1. Reaction network graphs

To define and depict the directed graph representing the dependencies of variables, we shall adopt the notations used for depicting the *structure* of Petri nets with test (read) arcs [3]. However, the notion of markings and the firing rule associated with the dynamics of Petri nets will play no role here. Hence to avoid confusion and to emphasize the purely graph-theoretic nature of our representation, we shall use the name RNGs (Reaction Network Graphs). A RNG is a directed bipartite graph with two types of nodes and arcs. Place nodes will correspond to the molecular species (typically, proteins) and transition nodes will correspond to the bio-chemical reactions in the pathway. The two types of arcs are used to denote whether a species participates as a normal reactant (governed by mass law kinetics) or as an enzyme (governed by Michaelis–Menten kinetics) in a reaction. Formally, a RNG is a structure $G = (\Pi, \Gamma, F)$, where

- Π is the set of place nodes with p ranging over Π.
- Γ is the set of transition nodes with γ ranging over Γ.
- $F \subseteq (\Pi \times \Gamma \times \{0, 1\}) \cup (\Gamma \times \Pi)$ is the flow relation connecting the nodes in a bipartite way using two types of arcs.

The place-to-transition arc $(p, \gamma, 1)$ is said to be a *normal* arc while $(p, \gamma, 0)$ is said to be a *test* arc. Every transition-to-place arc is deemed to be normal. A weight function can be added to the arcs but we will not make use of this feature in this paper.

The RNG of the enzyme reaction system of (1) is shown in Fig. 1(a). Since mass action is used in this model for all the reactions, there are no test arcs. As usual, one can apply quasi-steady-state approximation to the reversible pair of reactions and collapse the system into one Michaelis–Menten equation $V_{max}[S]/(K_M + [S])$, with $V_{max} = k''[E]$ and $K_M = (k' + k'')/k$.

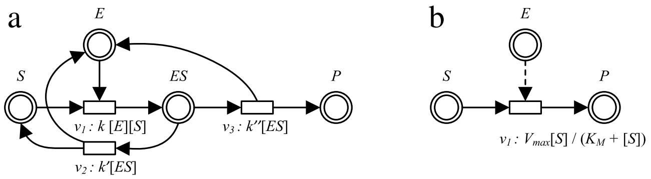


Fig. 1. (a) The RNG of the enzyme–catalyzed reaction system. Applying the quasi-steady-state approximation, the RNG of the simplified model will be as in (b).

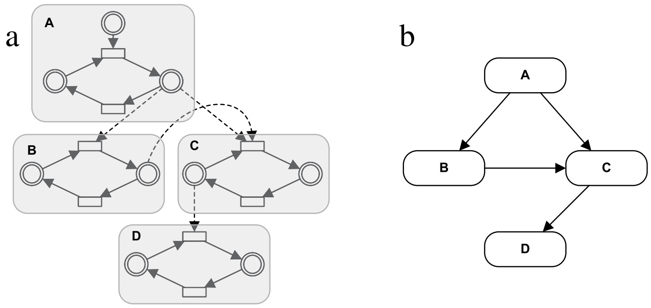


Fig. 2. (a) The minimal components of a RNG. (b) The component graph of this RNG.

The corresponding RNG is shown in Fig. 1(b). As illustrated, we will use solid lines to denote normal arcs and dotted lines to indicate test arcs.

3. Pathway decomposition

We start with the notion of a component which depends only on the structure of the pathway as captured by its RNG. As explained later, based on the distribution of experimental data over the pathway, certain components will be singled out to serve as sub-models in our decomposition procedure.

In what follows, we implicitly assume a system of ODEs modeling a bio-chemical network and the associated RNG to be $G = (\Pi, \Gamma, F)$. A *component* of G is a non-empty subset C of $\Pi \cup \Gamma \cup F$ which satisfies the following closure conditions. As before, we let γ range over Γ and p range over Π.

- Suppose $\gamma \in C$. If $(\gamma, p) \in F$ then $p \in C$ and $(\gamma, p) \in C$. Next if $(p, \gamma, 1) \in F$ then $p \in C$ and $(p, \gamma, 1) \in C$. Finally, if $(p, \gamma, 0) \in F$ then $(p, \gamma, 0) \in C$.
- Now suppose $p \in C$. If $(\gamma, p) \in F$ then $\gamma \in C$ and $(\gamma, p) \in C$. Next if $(p, \gamma, 1) \in F$ then $\gamma \in C$ and $(p, \gamma, 1) \in C$. Finally, if $(p, \gamma, 0) \in F$ then $(p, \gamma, 0) \in C$.

Thus the key idea of a component is to exploit test arcs to break the model down into smaller parts. Specifically, for a test arc of the form $(p, \gamma, 0)$ that belongs to the component C, one could have that γ (p) belongs to C but not p (γ).

Intuitively, if $\gamma \in C$ but $p \notin C$ then such an arc may serve as an input to C from a component to which p belongs. If we have sufficient time series information about the concentration level of p then one may simulate C without having to simulate the component that p belongs to as well. If on the other hand $p \in C$ and $\gamma \notin C$, then one may simulate C without considering the component to which γ belongs.

Clearly the union of components is a component. C is a *minimal* component iff it is a component and no proper subset of C is also a component. The nodes enclosed in the portion A constitutes a minimal component of the RNG shown Fig. 2(a).

In order to estimate the parameters for a component, one should be able to simulate the component with a candidate set of values for the unknown parameters in the component, compare the simulation data with experimental data and evaluate the quality of the candidate set. This strategy is feasible only if there is experimental data available for some of the species in the component. To identify such components we shall use the notions of a *component graph* and *blocks*.

Let $G = (\Pi, \Gamma, F)$ be a RNG. Then its component graph is denoted as CG_G and it is a directed graph $CG_G = (\mathcal{C}, E)$ given by:

- $\mathcal{C}$ is the set of minimal components of G.
- $E \subseteq \mathcal{C} \times \mathcal{C}$ satisfies:
  Suppose $C, C' \in \mathcal{C}$ and there exists $(p, \gamma, 0) \in F$ such that $p \in C - C'$, $\gamma \in C' - C$ and $(p, \gamma, 0) \in C \cap C'$. Then $(C, C') \in E$.

The component graph of the RNG in Fig. 2(a) is shown in Fig. 2(b).

The maximal strongly connected components of CG_G will be called blocks and are defined as follows. Let $\equiv \subseteq \mathcal{C} \times \mathcal{C}$ be the relation given by:

- $C \equiv C'$ iff $(C, C') \in E^\star$ and $(C', C) \in E^\star$ where $E^\star$ is the reflexive transitive closure of the edge relation E of CG_G.

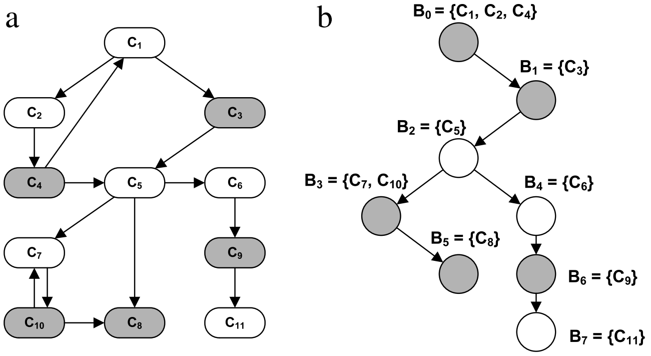


Fig. 3. (a) An example of a component graph of an RNG G (where G is not shown) and (b) the Hasse diagram of the blocks-poset of G.

Clearly $\equiv$ is an equivalence relation. In what follows, we shall denote the $\equiv$-equivalence containing the minimal component C as $[C]$. Further, a *block* is such an equivalence class of minimal components. Clearly every block is also a component. Next we define the relation $\sqsubset \subseteq \mathcal{C}/\equiv\ \times \mathcal{C}/\equiv$ as:

- $[C] \sqsubset [C']$ iff $(C, C') \in E^\star$ and $(C', C) \notin E^\star$.

It is easy to see that $\sqsubset$ is a strict partial ordering relation. As usual, we shall denote its reflexive closure as $\sqsubseteq$. Our decomposition procedure will systematically group together the blocks in the poset $(\mathcal{C}/\equiv, \sqsubseteq)$. In what follows, we will refer to this poset as the *blocks-poset* of the given RNG. In Fig. 3(a), we show a component graph (the RNG it is derived from is not shown). Its blocks and the Hasse diagram of the resulting blocks-poset is shown in Fig. 3(b). For the moment, the reader should ignore the gray shading of some of the minimal components and blocks.

We can now specify our scheme for grouping together the blocks into components based on the distribution of experimental data. To start with, we introduce a coloring scheme on the place nodes. For a place node in the RNG, if there is experimental data available, we color it *gray*. Otherwise it is colored *white*. If a minimal component contains at least one gray place then it is assigned the color gray. Otherwise it is colored white. It is also colored gray if all the parameter values associated with this minimal component are known. Finally, a block is colored gray if at least one of the minimal components contained in it is colored gray. The notion of a colored blocks-poset is now defined in the obvious way.

We now wish to group the blocks into components for doing parameter estimation. The basic idea is to exploit the experimental data available in the gray blocks to constrain the guessed parameter values in the white blocks. For instance consider the blocks B_2 and B_3 in Fig. 3(b). If we group B_2 and B_3 together, then since B_2 influences the dynamics of B_3, the experimental data available concerning the dynamics of the gray block B_3 can be used to constrain the guessed values of the unknown parameters in B_2. On the other hand, grouping together the gray block B_6 and the white block B_7 will not help in constraining the parameter values of B_7 using the experimental data available for B_6 since B_7 does not influence the dynamics of B_6.

For stating our decomposition algorithm, it will be convenient to introduce the following notation. Suppose $(\mathcal{B}, \sqsubseteq)$ is a blocks-poset and $B, B' \in \mathcal{B}$ with $B \sqsubseteq B'$. Then $[B, B') = \{B'' \mid B \sqsubseteq B'' \sqsubset B'\}$.

The Decomposition Algorithm

```
Input         : A colored blocks-poset $(\mathcal{B}, \sqsubseteq)$ with $\mathcal{B}_w$ as its set of white blocks and $\mathcal{B}_g$ as its set of gray blocks.
Initialization: $i = 0$ and $ACT_0 = \mathcal{B}_g$ and $\mathcal{D} = \emptyset$
1 while $ACT_i \neq \emptyset$ do
2     Pick a $\sqsubseteq$-minimal element $B$ in $ACT_i$. Set $B_i^g = B$ and $\Delta_i = \{B_i^g\} \cup X_i$ where $X_i$ is the subset of $\mathcal{B}$ given by: $B' \in X_i$
      iff $B' \sqsubset B_i^g$ and $[B', B_i^g) \subseteq \mathcal{B}_w$.
3     $ACT_{i+1} = ACT_i - \{B_i^g\}$ and $\mathcal{D} = \mathcal{D} \cup \{\Delta_i\}$
4     $i = i + 1$;
5 end
6 Output: $\mathcal{D}$
7 Quit
```

In each ACT_i there will be at most $l = |\mathcal{B}_g|$ blocks. Further, the cardinality of ACT_{i+1} will be one less than that of ACT_i. Thus the algorithm will terminate after executing each statement at most l times.

In step (2) of the algorithm a $\sqsubseteq$-minimal element from ACT_i is picked non-deterministically. We wish to assert that $\mathcal{D}$, the final output of the algorithm, will be insensitive to this non-determinism. To see this, we call the sequence $\sigma = B_0B_1 \ldots B_{k-1}$ to be a ($\sqsubseteq$-respecting) linearization of $\mathcal{B}_g$ iff every element of $\mathcal{B}_g$ appears exactly once in σ. Furthermore, for every i, j in

$\{0, 1, \dots, l-1\}$, if $B_i \sqsubseteq B_j$ then $i \le j$. We note that every sequence $B_0^g B_1^g \dots B_{l-1}^g$ generated by the algorithm will be a linearization of $\mathcal{B}_g$. Moreover, every linearization can be generated by the algorithm by making suitable choices in step (2).

Next we define the binary relation R over the linearizations of $\mathcal{B}_g$ as follows:

$\sigma R \sigma'$ iff there exists sequences τ_1 and τ_2 and blocks B and B' such that $\sigma = \tau_1 B B' \tau_2$ and $\sigma' = \tau_1 B' B \tau_2$.

Now suppose $\sigma = B_0 B_1 \dots B_i B_{i+1} B_{i+2} \dots B_{l-1}$ and $\sigma' = B_0 B_1^g \dots B_{i+1}^g B_i^g B_{i+2} \dots B_{l-1}$ are two linearizations generated by the algorithm so that $\sigma R \sigma'$. From the construction of the algorithm, along any execution, for each j, the set of blocks Δ_j will contain exactly one gray block namely B_j^g and all other blocks will be white and lie strictly below B_j^g. Hence if we set $\mathcal{B}_1 = \Delta_i$ and $\mathcal{B}_2 = \Delta_{i+1}$ along the execution generating σ and similarly set $\mathcal{B}'_1 = \Delta_i$ and $\mathcal{B}'_2 = \Delta_{i+1}$, then it is easy to verify that $\mathcal{B}_1 = \mathcal{B}'_2$ and $\mathcal{B}_2 = \mathcal{B}'_1$. Clearly, for any two linearizations σ and σ', we will have $\sigma R^m \sigma'$ for some $m \in \{0, 1, \dots, l-1\}$ (with R^0 viewed as the identity relation). Hence by an easy induction on m one can establish that the output of the algorithm will be insensitive to the non-determinism in step (2) and hence will be unique.

In Fig. 3(b) we show the iterative construction of $\mathcal{D}$ when the algorithm chooses the linearization $B_0 B_1 B_2 B_6 B_5$. The output of the algorithm will be $\{\{B_0\}, \{B_1\}, \{B_3, B_2\}, \{B_6, B_4, B_2\}, \{B_5\}\}$. It is easy to check that the output will be the same if the algorithm chooses the linearization $B_0 B_1 B_6 B_3 B_5$.

Each Δ_i produced by the algorithm will be a component of the RNG that the colored blocks-poset is based on. More precisely if we collect together all the minimal components that appear in the blocks of Δ_i and then collect together all the nodes and arcs of these minimal components we will end up with a component. The idea should be clear and we will not formalize it here. Instead we will loosely term Δ_i itself to be a component. In this light $\mathcal{D} = \{\Delta_0, \Delta_1, \dots, \Delta_{k-1}\}$ is the set of components yielded by our decomposition procedure.

We will do parameter estimation of these components in the order suggested by their indices. Once a component's parameters have been estimated, then this component can be simulated to yield the enzyme profiles to be used by the components that lie downstream of it. For instance, for the linearization $B_0 B_1 B_2 B_6 B_5$, we first estimate the parameters of B_0. Now suppose there is just one edge $(p, \gamma, 0)$ with p in B_0 and γ in B_1. Then we simulate B_0 with the estimated parameters to generate time series profile for the enzyme associated with p and then use this to do the parameter estimation for B_1. We note that instead of following a strict sequential order, the parameter estimation for Δ_2 can be followed by that of Δ_4 while estimating the parameters of Δ_3 in parallel.

As illustrated by Fig. 3(b), a white colored block (consider B_2) can appear in more than one component. Consequently, conflicting estimates for the parameters of such a block can be produced by the parameter estimation procedures executed for the components in which this shared white block appears. This calls for a technique to reconcile these conflicting estimates in a globally consistent manner.

We also note that a white colored block may not get included in any component during the decomposition procedure such as B_7 in Fig. 3(b). For such blocks one must seek fresh experimental data or work with the full range of values that can be assumed by the unknown parameters belonging to the block. The number of components that the decomposition procedure yields and the number of unknown parameters they contain is a measure of the dimension reduction that is being achieved. Clearly, this is determined by the structure of the RNG and the availability and distribution of experimental data.

As for the complexity of our decomposition procedure, we note that the RNG of an ODE system can be derived in time linear in the number of equations, say n and the number of reactions, say, m. With $n \cdot m = k$, the size of the RNG will be $O(k)$. For each node in the *RNG* one can run a backward tracing algorithm to determine the minimal component it is contained in and this will run in time $O(k)$ and in $O(k^2)$ time one can construct the component graph. Then using Tarjan's algorithm [15] for computing the strongly connected components of a directed graph, the blocks-poset can be computed in time $O(\hat{k})$ where $\hat{k} = O(k)$ is the size of the component graph. This shows that the time complexity of the decomposition algorithm is at most $O(\hat{k}^2)$.

At present, we have not implemented the decomposition procedure. However, it should be an easy task since obtaining the RNG from a system of ODEs can be done symbolically and the remaining steps consist of simple graph algorithms.

4. Pathway integration and parameter reconciliation

4.1. Overview

For convenience, assume in what follows that parameter estimates are available for each of the individual components C_i for $i = 1, 2, \dots, n$. The key task now is to reconcile the potentially conflicting parameter estimates to obtain globally consistent ones which will lead to a single integrated model. Let $\mathbf{k}$ denote the set of all parameters. Let $\mathbf{k}_i$ be the subset of parameters for component C_i, and D_i be the experimental data used in estimating $\mathbf{k}_i$. The error in fit to data for component C_i is then $J(\mathbf{k}_i|D_i)$. To reconcile the estimates for all the components, we need to compute an estimate of $\mathbf{k} = \bigcup_{i=1}^{n} \mathbf{k}_i$ with minimum $J(\mathbf{k}|D_1, D_2, \dots, D_n)$. Computing the estimate by a global method will be computationally expensive and hence we want to compute an estimate of $\mathbf{k}$ based on the estimates of $\mathbf{k}_i$, $i = 1, 2, \dots, n$.

To do so, we encode a set of estimates of $\mathbf{k}_i$ as a probability function

$$p(\mathbf{k}_i|D_i) = (1/\lambda) \exp(-J(\mathbf{k}_i|D_i)), \tag{6}$$

where λ is a normalizing constant ensuring that $\int p(\mathbf{k}_i|D_i)\, d\mathbf{k}_i = 1$. A large value of $p(\mathbf{k}_i|D_i)$ indicates small error in fit to the data set D_i. In other words, we view $p(\mathbf{k}_i|D_i)$ as a probabilistic weight on $\mathbf{k}_i$, expressing preferences over $\mathbf{k}_i$ values due to the

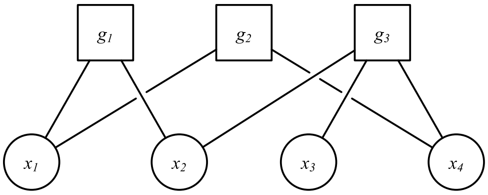


Fig. 4. The factor model for the function $g(x_1, x_2, x_3, x_4) = g_1(x_1, x_2) \cdot g_2(x_1, x_4) \cdot g_3(x_2, x_3, x_4)$.

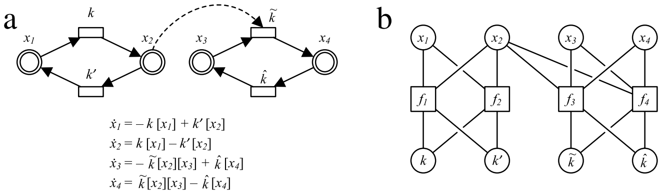

a
k
$\widetilde{k}$
x_1
x_2
x_3
x_4
k'
$\hat{k}$
$\dot{x}_1 = -k[x_1] + k'[x_2]$
$\dot{x}_2 = k[x_1] - k'[x_2]$
$\dot{x}_3 = -\widetilde{k}[x_2][x_3] + \hat{k}[x_4]$
$\dot{x}_4 = \widetilde{k}[x_2][x_3] - \hat{k}[x_4]$
b
x_1
x_2
x_3
x_4
f_1
f_2
f_3
f_4
k
k'
$\widetilde{k}$
$\hat{k}$

Fig. 5. (a) A simple signaling cascade and its ODEs. (b) The factor graph representation. The variable nodes in gray – x_1, x_3, and x_4 – can be eliminated.

constraints from the data set D_i. Now our basic idea is to represent the probability function $p(\mathbf{k}_i|D_i)$ for each component C_i as a *factor graph*, a well-known probabilistic graphical model. Next, we connect these factor graphs together according to the structural relationships between the components. Finally, we apply *belief propagation* to reconcile the potentially conflicting parameter estimates from the various components. We start with a short review of factor graphs.

4.2. Factor graphs

Suppose that a high-dimensional function $g(\mathbf{z})$ can be factored as a product of lower-dimensional functions: $g(\mathbf{z}) = \prod_i g_i(\mathbf{z}_i)$, where $\mathbf{z} = (z_1, z_2, \ldots)$ is a set of variables and each $\mathbf{z}_i$ is a (small) subset of variables in $\mathbf{z}$. A factor graph for $g(\mathbf{z})$ is an undirected bipartite graph consisting of two types of nodes: *factor* nodes and *variable* nodes. Each factor $g_i(\mathbf{z}_i)$ has a corresponding factor node in G, and each variable z_j has a corresponding variable node in G. There is an undirected edge between the factor node for $g_i(\mathbf{z}_i)$ and the variable node for z_j if $z_j \in \mathbf{z}_i$, *i.e.*, z_j is a variable of the function $g_i(\mathbf{z}_i)$. An example is shown in Fig. 4.

A variable node for z_j contains a probability distribution over the values of z_j. A factor node for $g_i(\mathbf{z}_i)$ specifies the dependencies among the variables in $\mathbf{z}_i$ and expresses preferences over their values due to some constraints. In pathway parameter estimation, the main variables are the parameters, and the constraints arise from the ODEs in which the parameters appear. For example, consider the reaction shown in Fig. 5. Suppose that data are available for $x_1(t), x_2(t), x_3(t)$ at all times t, but the rate constants k_1 and k_2 are unknown. Then, each of the three equations in the system of ODEs for the reactions imposes a constraint on the unknowns k_1 and k_2 at all times t. Those combinations of k_1 and k_2 values that satisfy the constraints are favored. In general, each equation in an ODE model represents a local constraint on the parameters involved in the equation, and each such constraint results in a factor node.

4.3. The factor graph structure

We now consider a specific component C and construct a factor graph P for its unknown parameters $\mathbf{k}$. We have dropped the component index i to simplify the notation. Each component contains a set of place nodes and ODEs that describe how the concentration levels of these molecular species change over time. For each ODE f_i associated with the component, we create a factor node $\nu(f_i)$ in P. We also create a variable node $\nu(k_j)$ for each parameter k_j and a variable node $\nu(x_j)$ for each molecular concentration level x_j, if k_j or x_j is involved in f_i. We also insert an edge that connects a factor node for f_i and a variable node for k_j or x_j. An example is shown in Fig. 5.

Our main goal is to capture the dependencies among the parameters. We can eliminate many of the variable nodes representing molecular concentration levels and thus simplify P. However, we can eliminate a variable node only if it does not represent the concentration level of an enzyme. The reason is that although enzymes are not consumed in catalytic reactions, their concentration levels influence the reactions. In general, eliminating a variable node corresponding to an enzyme results in the loss of dependency between the reaction producing the enzyme and the reaction catalyzed by the enzyme. To see this, consider again the example in Fig. 5. If we eliminate the variable nodes for x_1, x_3 and x_4, which are not enzymes, the dependencies among k_1, k_2, k_3, and k_4 remain intact. However, if we eliminate the variable node x_2, an enzyme, the factor graph breaks into two disconnected components. There is no constraint that connects k_1 and k_2 with k_3 and k_4, implying that k_1 and k_2 are independent of k_3 and k_4. This is clearly not the case.

To summarize, the structure of a factor graph – the variable nodes, the factor nodes, and the edges – is constructed from the ODEs that model a pathway component. Each factor captures the dependencies among the parameters involved in a particular equation.

4.4. The compatibility functions

To complete the construction of the factor graph P, we need to associate a *compatibility function*, with each factor node $\nu(f_i)$. The compatibility function for $\nu(f_i)$ is given by

$$g_i(\mathbf{k}_i, \mathbf{x}_i(t)) = \exp(-E_i(\mathbf{k}_i, \mathbf{x}_i(t))), \tag{7}$$

where $\mathbf{k}_i$ and $\mathbf{x}_i(t)$ are respectively the set of parameters and the set of molecular concentration levels corresponding to the variables nodes connected to $\nu(f_i)$. Note the distinction between x_i, which denotes the concentration level of species i, and $\mathbf{x}_i$. The function $E_i(\mathbf{k}_i, \mathbf{x}_i(t))$ consists of two terms:

$$E_i(\mathbf{k}_i, \mathbf{x}_i(t)) = E_{i,1}(\mathbf{k}_i) + E_{i,2}(\mathbf{k}_i, \mathbf{x}_i(t)). \tag{8}$$

The first term $E_{i,1}(\mathbf{k}_i)$ measures the fit to data for a particular choice of values for the parameters in $\mathbf{k}_i$. The second term $E_{i,2}(\mathbf{k}_i, \mathbf{x}_i(t))$ measures whether the values for $\mathbf{k}_i$ are consistent with those for $\mathbf{x}_i(t)$.

We calculate $E_{i,1}(\mathbf{k}_i)$ based on the global effect of $\mathbf{k}_i$ on the fit to data for the molecular species that are experimentally measured:

$$E_{i,1}(\mathbf{k}_i) = \min_{\mathbf{k}\backslash\mathbf{k}_i} \sum_{m\in M} \sum_j (x_m(T_j; \mathbf{k}) - \tilde{x}_{mj})^2, \tag{9}$$

where $\mathbf{k}\backslash\mathbf{k}_i$ denotes the set of parameters in $\mathbf{k}$, but not in $\mathbf{k}_i$, M denotes the set of all species that are measured experimentally, $x_m(t; \mathbf{k})$ is the concentration level of species m at time t, obtained by simulating the system of ODEs with parameters $\mathbf{k}$, and finally $\tilde{x}_{mj}$ is the experimental concentration level of species m at time T_j.

Finally, we observe that though our decomposition procedure yields a unique result the process of converting a component model into a factor graph involves sampling and simulations. However, for a fixed discretization, sampling method and number of samples, the differences between different runs – in terms of the factor graphs that are constructed – will be small. Clearly the underlying structures will be identical. Further, in case we start with point estimates of the various parameters in the ODE model, the resulting factor graphs will be identical even in terms of the belief states associated with the variable nodes under the convention that each interval containing the estimated value of a parameter is assigned the probability 1 and all other intervals are assigned the probability 0. However, if we start with a component model which has parameter estimates in terms of prior probability distributions, then the resulting factor graphs will differ from each other slightly in terms of the belief states. The integration step achieved through belief propagation will not introduce any additional differences.

The second term $E_{i,2}(\mathbf{k}_i, \mathbf{x}_i(t))$ measures the consistency between the parameter values $\mathbf{k}_i$ and concentration levels $\mathbf{x}_i(t)$: $\mathbf{k}_i$ and $\mathbf{x}_i(t)$ are *consistent* if $\mathbf{x}_i(t)$ can be obtained by simulating the system of ODEs with parameter values $\mathbf{k}_i$ and some suitable choice of values for parameters in $\mathbf{k}\backslash\mathbf{k}_i$. The function $E_{i,2}(\mathbf{k}_i, \mathbf{x}_i(t))$ takes binary values. If $\mathbf{k}_i$ and $\mathbf{x}_i(t)$ are consistent, $E_{i,2}(\mathbf{k}_i, \mathbf{x}_i(t)) = 0$; otherwise, $E_{i,2}(\mathbf{k}_i, \mathbf{x}_i(t)) = +\infty$. This way, $\mathbf{k}_i$ values that are inconsistent with the dynamics defined by the ODEs are filtered out, regardless of their agreement with experimental data according to $E_{i,1}(\mathbf{k}_i)$.

With our definition of compatibility functions, the factor graph P encodes exactly the function

$$g(\mathbf{k}, \mathbf{x}(t)) = \frac{1}{\lambda} \prod_i g_i(\mathbf{k}_i, \mathbf{x}_i(t)) = \frac{1}{\lambda} \exp\Big(-\sum_i E_i(\mathbf{k}_i, \mathbf{x}_i(t))\Big), \tag{10}$$

where $\mathbf{k} = \bigcup_i \mathbf{k}_i$, $\mathbf{x} = \bigcup_i \mathbf{x}_i$, and λ is a normalizing constant ensuring that $g(\mathbf{k}, \mathbf{x}(t))$ represents a well-defined probability function. The function $g(\mathbf{k}, \mathbf{x}(t))$ has the same extremal values as $J(\mathbf{k})$ and $p(\mathbf{k})$ [16]. It can be shown that if the parameter values $\mathbf{k}^*$ minimize $J(\mathbf{k})$, then $\mathbf{k}^*$ maximize $p(\mathbf{k})$, and $\mathbf{k}^*$ and concentration levels $\mathbf{x}(t; \mathbf{k}^*)$ maximize $g(\mathbf{k}, \mathbf{x}(t))$, where $\mathbf{x}(t; \mathbf{k}^*)$ is the molecular concentration levels obtained by simulating the ODE model with parameter values $\mathbf{k}^*$. This result implies that to minimize $J(\mathbf{k})$ or maximize $p(\mathbf{k})$, we may equivalently maximize $g(\mathbf{k}, \mathbf{x}(t))$.

The compatibility functions defined above measure the fit to data globally over all experimentally measured molecular species. As a heuristic for improving efficiency, we introduce a variant which measures the fit to data locally as well. The definition of $E_{i,1}(\mathbf{k}_i)$ then depends on whether the concentration level x_i of molecular species i is measured experimentally. If it is, we calculate $E_{i,1}(\mathbf{k}_i)$ locally using only the data for x_i:

$$E_{i,1}(\mathbf{k}_i) = \min_{\mathbf{k}\backslash\mathbf{k}_i} \sum_j (x_i(T_j; \mathbf{k}) - \tilde{x}_{ij})^2. \tag{11}$$

If x_i is not measured experimentally, we calculate $E_{i,1}(\mathbf{k}_i)$ globally using Eq. (9). Intuitively, calculating the fit to data locally strengthens the local constraints and makes belief propagation more greedy. This turns out to be helpful in the experiments presented in the next section. However this is only a heuristic without any theoretical guarantees.

We now discuss how to represent and compute the compatibility functions $g_i(\mathbf{k}_i, \mathbf{x}_i(t))$. First, the parameter values and the concentration levels are discretized into a finite of set of intervals. Both the probability distributions for variable nodes and the compatibility functions for factor nodes are represented using this discretization. This is common practice for factor graphs used in conjunction with belief propagation [4] and it is not a severe limitation here, since the experimentally measured concentration levels for proteins in a signaling pathway often have very limited accuracy. Furthermore, once belief

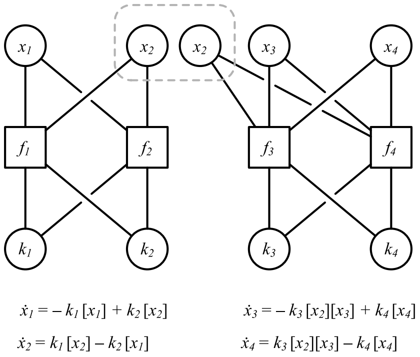

x_1
x_2
x_2
x_3
x_4
f_1
f_2
f_3
f_4
k_1
k_2
k_3
k_4
$\dot{x}_1 = -k_1[x_1] + k_2[x_2]$
$\dot{x}_2 = k_1[x_2] - k_2[x_1]$
$\dot{x}_3 = -k_3[x_2][x_3] + k_4[x_4]$
$\dot{x}_4 = k_3[x_2][x_3] - k_4[x_4]$

Fig. 6. Merging two component factor graphs.

propagation gives the best parameter estimate up to the resolution of the discretization, we can further refine the estimate by performing a local search, thus mitigating the effect of discretization. One advantage of the discrete representation is that the resulting factor graph can represent arbitrary probability distributions, up to the resolution of the discretization. There is no need to assume a particular parametric form of the distribution.

Next, to compute $g_i(\mathbf{k}_i, \mathbf{x}_i(t))$, we need to perform the minimization required in (9) or (11). We first sample a representative set of parameter values. We may use pseudo-random or the more efficient quasi-random sampling methods [17] to sample parameter values uniformly at random. We may also use any optimization methods, such as evolutionary algorithms [13], to sample the parameter values in a biased, but likely more efficient manner. We then perform the minimization over the set of sampled values.

4.5. Pathway composition

We build a factor graph P_i for each component C_i. Each component factor graph P_i encodes a probability function expressing preferences over the values of the parameters $\mathbf{k}_i$ contained in P_i. To account for the dependency among the parameters from different components, we merge the component factor graphs by fusing their common variable nodes (see Fig. 6 for an example) and construct a single factor graph P.

The combined factor graph P represents a probability function $p(\mathbf{k}|D)$ over all the parameters $\mathbf{k} = \bigcup_{i=1}^{n} \mathbf{k}_i$. We apply belief propagation (BP) over P to reconcile the different preferences over parameter values from each component. In other words, we apply BP to minimize $\mathbf{k}^*$ of $J(\mathbf{k}|D_1, D_2, \dots, D_n)$, or equivalently maximize $g(\mathbf{k}, \mathbf{x}(t))$ represented by P.

We give only a quick overview of BP here (see [4] for a comprehensive account). Let *FG* be a factor graph representing a factored non-negative function $g(\mathbf{z}) = g(z_1, z_2, \dots) = \prod_i g_i(\mathbf{z}_i)$, where $\mathbf{z}_i$ is the subset of variables involved in the factor $g_i(\mathbf{z}_i)$. After normalization, $g(\mathbf{z})$ can be considered as a probability function. Each variable node $\nu(z_j)$ of *FG* is initialized with a probability distribution $\pi_0(z_j)$ – commonly called a *belief* – over the values of z_j. A preferred z_j value has higher probability. The initial distribution $\pi_0(z_j)$ represents our prior knowledge on the value of z_j. If there is no prior information on z_j, we set its initial distribution to be uniform. After initialization, a variable node $\nu(z_j)$ sends its belief $\pi(z_j)$ as a message to each adjacent factor node $\nu(g_i)$. Upon receiving the messages from the adjacent variable nodes, a factor node $\nu(g_i)$ combines them with its own compatibility function $g_i(\mathbf{z}_i)$ and creates a new message, which is sent to each variable node $\nu(z_j)$ adjacent to $\nu(g_i)$. The belief at $\nu(z_j)$ is then updated so that z_j values satisfying the compatibility function $g_i(\mathbf{z}_i)$ will have their probabilities increased. The order in which to send the messages must follow a suitable protocol, and the messages stop when a termination condition is met.

When BP terminates, the variable nodes take on beliefs favoring values that satisfy well the local constraints represented by the compatibility functions in the factor nodes. If a factor graph contains no cycles, BP converges to the *global* maximum of the function the factor graph represents [18]. In practice, a factor graph modeling a complex system often contains cycles. So convergence is not guaranteed, and one needs to terminate the algorithm using heuristic criteria. Nevertheless, BP on general factor graphs often generates good results in diverse applications [19,20]. One reason is that BP is in essence a dynamic programming algorithm, which performs a more global search than strategies such as gradient descent, and is less likely to get stuck in local maxima.

We now apply BP to a factor graph representing the function $g(\mathbf{k}, \mathbf{x}(t))$ in (10). Each compatibility function $g_i(\mathbf{k}_i, \mathbf{x}_i(t))$ in the factor graph encodes two types of constraints: $E_{i,1}(\mathbf{k}_i)$ measures the fit to data, and $E_{i,2}(\mathbf{k}_i, \mathbf{x}_i(t))$ measures the consistency between $\mathbf{k}_i$ and $\mathbf{x}_i(t)$ with respect to the dynamics defined by the ODEs. BP will favor $\mathbf{k}$ and $\mathbf{x}$ values that satisfy all the constraints well.

To end this section, we wish to point out that although our decomposition procedure will return a unique result, the integration step may return slightly different factor graphs for different runs. This is due to the fact that the procedure for converting an ODEs based model into a factor graph involves discretization, sampling and simulations to compute the belief states to be associated with the variable nodes. Since the sampling procedure will typically compute different seeds, the belief states of the variables in the component factor graphs may have slightly differing belief states for different runs.

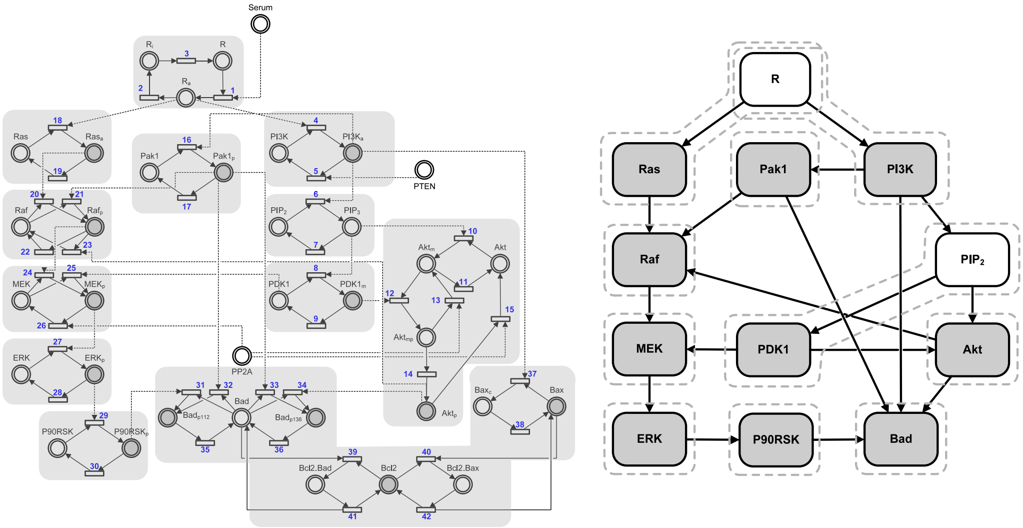


Fig. 7. (a) The RNG model of the pathway showing the components (gray boxes). Places with experimental data are colored gray. (b) The component graph of the pathway together with the grouping of components shown dashed boxes forming the decomposition.

Consequently, after performing belief propagation to integrate the component factor graphs, the final belief states may also be different for different runs. As expected, in our case study we found these deviations to be minor.

5. Case study — Akt and MAPK signaling pathway

We applied our approach to the parameter estimation problem for a synthetic Akt–MAPK signaling pathway. The Akt and MAPK signaling pathways play major roles in apoptosis and cell proliferation respectively. Both signaling pathways can be activated by a wide range of extra-cellular stimuli. Disruptions to their signaling mechanisms can lead to the onset of several diseases, such as cancer and Alzheimer's [21]. In the recent years, cross-talks between some of the signaling pathways have been recognized to play an important regulatory role. Specifically, the cross-talk between the Akt and MAPK signaling pathways has been attracting considerable interest [22,23].

Since our main aim was to test out the belief propagation based integration method, we extracted a synthetic and simplified model for our case study. The resulting model consists of 36 molecule species and 42 unknown rate parameters. It is a mass-conserved model where, aside from PTEN, the total concentration levels of all the molecules were set at 5 nM. The *nominal values* of the parameters were set within the interval [0.0, 1.0]. We assumed that 13 of the molecule species to be observed at chosen discrete time points. This data was generated by simulating our model using the nominal parameter values; to be used *in lieu* of actual experimental data. The pathway diagram, the rate equations and the initial conditions are shown in Fig. 7(a), Table 1 and Table 2 respectively.

We first decomposed our model into 12 minimal components as shown in Fig. 7. We then converted them into their factor graph representation. Both these steps were done manually. As shown, there are 2 overlapping components.

We discretized each parameter value range into 10 equal bins. Hence, belief propagation will return a probability distribution over the bins, using which we can pick the *maximum a posteriori* (MAP) bin as the one that best fits the experimental data. We also discretized the time series data by dividing the time domain into 10 equal bins and the concentration domain into 5 equal bins for each molecular species. Since there can be 5^{10} possible traces, we hash a trace running through this discretized space into an integer index.

For the sake of convenience, we carried out the parameter estimation procedure for each component itself using the discretization and belief propagation. As a result, for each component, the parameter estimates were obtained in the form of maximal likelihood estimates for the (assumed) unknown parameters belonging to the component.

To reduce the space required for our algorithm, we exploited the partial order on the components. The profiles of the *upstream* components were cached and used during the sampling of *downstream* components. This way, the probability distribution of the variable nodes representing enzymes could be sampled from *observed* traces, instead of all potential traces.

Our algorithm is implemented in C++. All simulations were performed on an Intel Core 2 Duo 1.66 GHz processor with 1 Gb memory. Each component was simulated 10,000 times using parameter values randomly sampled from their allowable ranges to build the joint distribution tables in the factor nodes. For components that require concentration profiles as inputs, the profiles were picked from a cache of previously observed traces. Upon composing all the factor graphs together, belief propagation was then performed to obtain the final beliefs of the discretized parameters. The open source tool COPASI [24] was then used for the fine-tuning of the parameter estimates using the Levenberg–Marquardt [7] algorithm while starting from the mid-points of the MAP bins for the parameters.

Table 1
Rate equations of the various reactions in the pathway. [X] denotes the concentration of the species X.

No	Rate equation	No	Rate equation
1	k_1[R]	22	k_{22}[Raf$_p$]
2	k_2[R$_a$]	23	k_{23}[Akt$_p$][Raf$_p$]
3	k_3[R$_i$]	24	k_{24}[Raf$_p$][MEK]
4	k_4[R$_a$][PI3K]	25	k_{25}[PDK1][MEK]
5	k_5[PTEN][PI3K$_a$]	26	k_{26}[MEK$_p$][PP2A]
6	k_6[PI3K$_a$][PIP$_2$]	27	k_{27}[MEK$_p$][ERK]
7	k_7[PIP$_3$]	28	k_{28}[ERK$_p$]
8	k_8[PIP$_3$][PDK1]	29	k_{29}[ERK$_p$][P90RSK]
9	k_9[PDK1$_m$	30	k_{30}[P90RSK$_p$]
10	k_{10}[PIP$_3$][Akt]	31	k_{31}[P90RSK$_p$][Bad]
11	k_{11}[Akt$_m$]	32	k_{32}[Pak1$_p$][Bad]
12	k_{12}[PDK1$_m$][Akt$_m$]	33	k_{33}[Pak1$_p$][Bad]
13	k_{13}[PP2A][Akt$_{mp}$]	34	k_{34}[Akt$_p$][Bad]
14	k_{14}[Akt$_{mp}$]	35	k_{35}[Bad$_{p112}$]
15	k_{15}[PP2A][Akt$_p$]	36	k_{36}[Bad$_{p136}$]
16	k_{16}[PI3K$_a$][Pak1]	37	k_{37}[PI3K$_a$][Bax]
17	k_{17}[Pak1$_p$]	38	k_{38}[Bax$_c$]
18	k_{18}[R$_a$][Ras]	39	k_{39}[Bad][Bcl2]
19	k_{19}[Ras$_a$]	40	k_{40}[Bax][Bcl2]
20	k_{20}[Ras$_a$][Raf]	41	k_{41}[Bcl2.Bad]
21	k_{21}[Pak1$_p$][Raf]	42	k_{42}[Bcl2.Bax]

Table 2
The set of initial conditions. Values are given in nM.

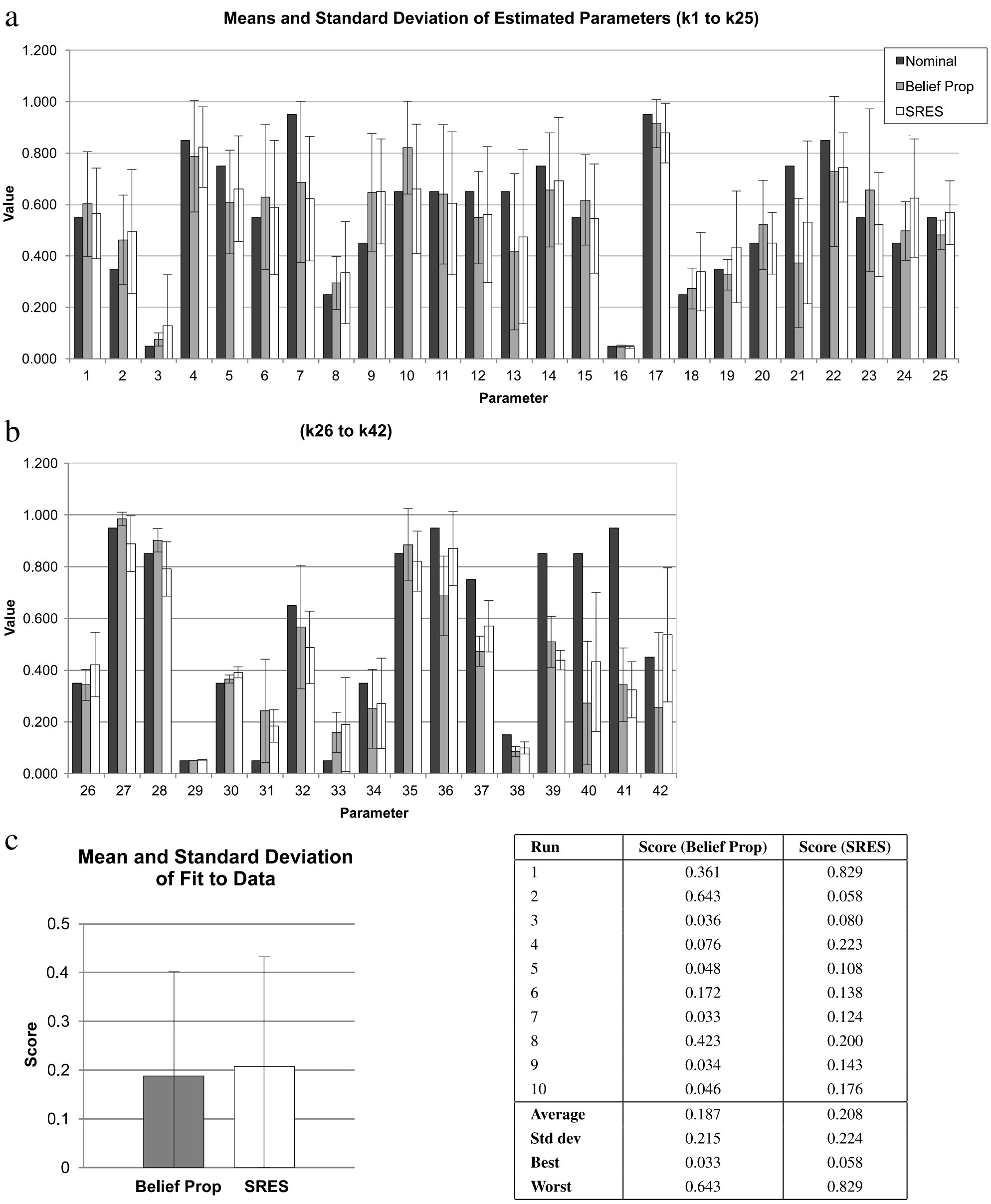
Molecule species	Initial condition
Serum	**1.0**
PP2A	**5.0**
R	**5.0**
PI3K	**5.0**
PIP$_2$	**5.0**
PDK1	**5.0**
Akt	**5.0**
Pak1	**5.0**
Ras	**5.0**
Raf	**5.0**
MEK	**5.0**
ERK	**5.0**
P90RSK	**5.0**
Bad	**5.0**
Bax$_c$	**5.0**
Bcl2	**5.0**
PTEN	**1.0**

We ran our algorithm 10 times to iron out the deviations caused by the sampling procedure. For each parameter, we computed its mean and standard deviation (of its belief state) to assess its relative performance. For further assessing the quality of our method, we also used the COPASI tool to do a *global* parameter estimation for the same set of experimental data, initial conditions and unknown parameters. We ran the global algorithm too 10 times and took the means of the estimated parameter values. The results are shown in Fig. 8(a) and (b). The overall time required for our method was approximately 922 s.

We compared the fitness to data of our simulation profiles (after plugging in the estimated parameters to the model) with the synthetic experimental data using the weighted cost function specified by Eq. (2) in Section 2 and as implemented in COPASI. The means and standard deviations, as well as the details of the scores are shown in Fig. 8(c). From the results, we can see that the performance of our method with mean score of 0.187 is comparable to that of SRES (Stochastic Rankings based Evolutionary Strategy) with a mean score of 0.208. The best score for our method is 0.033 while the worst score is 0.643 while for the global method, the corresponding numbers are 0.058 and 0.829 respectively.

Looking into the estimates of individual parameters, we observe that aside from the parameters, (k_7, k_9, k_{21}, k_{31}, k_{36}, k_{37}, k_{39}, k_{40}, k_{41}), most other estimated values are comparable with the nominal ones. One possible reason for the few mismatches is that the amount of data is insufficient to constrain the values of those parameters. Further, ours is a simplified model and adding further details may improve the quality of the estimates.

Finally, the profiles and their fit to data for some of the molecule species are shown in Fig. 9.



Run	Score (Belief Prop)	Score (SRES)
1	0.361	0.829
2	0.643	0.058
3	0.036	0.080
4	0.076	0.223
5	0.048	0.108
6	0.172	0.138
7	0.033	0.124
8	0.423	0.200
9	0.034	0.143
10	0.046	0.176
Average	0.187	0.208
Std dev	0.215	0.224
Best	0.033	0.058
Worst	0.643	0.829

Fig. 8. (a), (b) Bar graphs showing the means of the estimated parameters for SRES (white), our method (gray) and nominal values (dark gray). The standard deviation is denoted by the error bars. (c) Bar graph showing the mean fit to data and its standard deviation for SRES (white) and our method (gray) for the 10 runs. The table shows the scores of the individual runs.

In summary, our approach yields good quality estimates given limited experimental data. We note that the *uncertainty* in the estimated parameter values is represented in our method in the form of belief states. Consequently, using them as *a priori* distributions one can improve these estimates via belief estimation when new experimental data becomes available.

6. Conclusion

In this work, we have shown how a decomposition–estimation–integration based approach can be adopted to help solve the parameter estimation problem for bio-pathways models. We have focused here mainly on models based on ODEs. Our decomposition procedure relies on the structure of the pathway as captured by its RNG. The components resulting from the decomposition are then obtained in a canonical fashion by exploiting the distribution of experimental data. The second step then is to carry out parameter estimation for the individual components using any of the available methods. In the third step, we convert each of the estimated components into a factor graph, glue together the resulting set of factor graphs and then apply belief propagation to obtain a globally consistent set of parameter estimates.

When the structure of the pathways admits good decompositions (many low-dimensional components, each having a small number of unknown parameters and sufficient experimental data) then the individual estimation tasks will entail

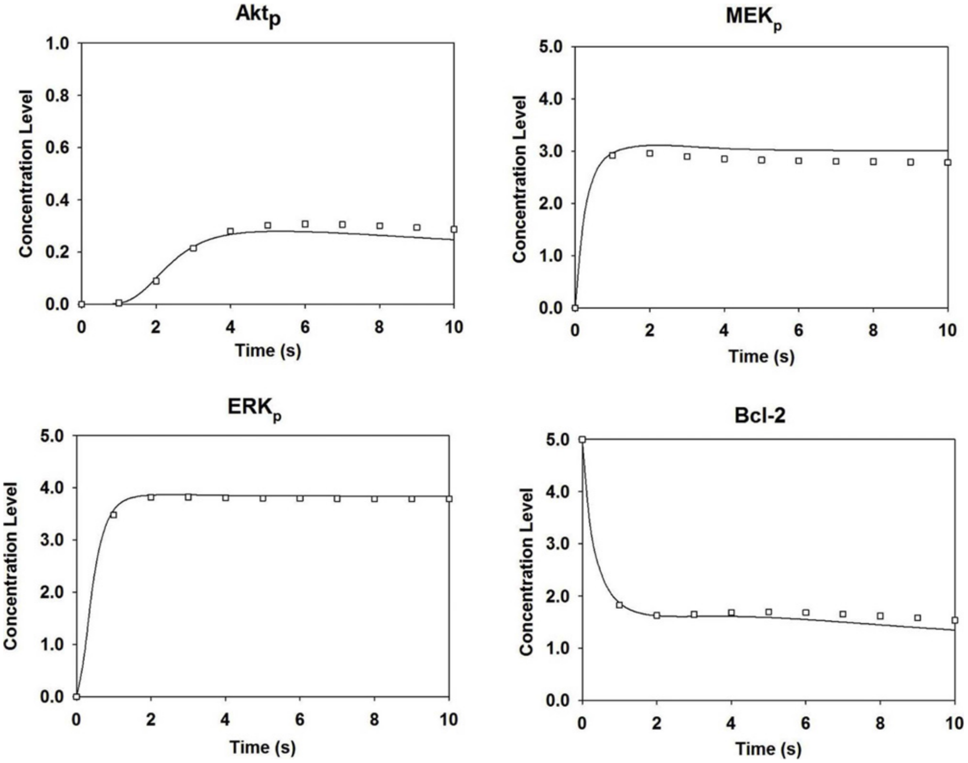


Fig. 9. The fit to data for some of the molecule species, i.e. Akt$_p$, MEK$_p$, ERK$_p$ and Bcl2. The boxes represent experimental data points.

searching through small parameter spaces. The factor graph representation maintains parameter estimates in terms of probability distributions rather than point values and this could be a more realistic and robust method of maintaining the current state of knowledge of the pathway model. It also eases the task of updating the model as additional information becomes available as we have recently shown [16]. In this context, we also note that model refinement and extensions may be easier to perform in our setting since the component graph can be used to localize the changes that need to be made to the model when it is extended. The components that are not affected can be identified and their beliefs can be maintained when doing parameter estimation for the extended model. We also note that we can analyze the blocks-poset to determine where it will be beneficial to generate additional experimental data.

In terms of future work, it will be important to explore methods for improving the efficiency and scalability of our approach. We need to carry out larger case studies involving "live" models and experimental data. In addition, it will be interesting to explore the possibility of using (probabilistic) model checking methods as a substitute for carrying out belief propagation in the integration step.

References

[1] U.S. Bhalla, R. Iyengar, Emergent properties of networks of biological signaling pathways, Science 283 (5400) (1999) 381–387.

[2] E. Klipp, W. Liebermeister, Mathematical modeling of intracellular signaling pathways, BMC Neurosci. 7 (suppl 1) (2006) S10.

[3] H. Matsuno, Y. Tanaka, H. Aoshima, A. Doi, M. Matsui, S. Miyano, Biopathways representation and simulation on hybrid functional petri net, In Silico Biol. 3 (3) (2003) 389–404.

[4] D. Koller, N. Friedman, Probabilistic Graphical Models: Principles and Techniques, MIT Press, Cambridge, 2009.

[5] G. Koh, H.F.C. Teong, M.-V. Clément, D. Hsu, P. Thiagarajan, A decompositional approach to parameter estimation in pathway modeling, Bioinformatics 22 (14) (2006) e271–e280.

[6] G. Koh, L. Tucker-Kellogg, D. Hsu, P.S. Thiagarajan, Composing globally consistent pathway parameter estimates through belief propagation, in: WABI'07: Proceedings of the 7th International Workshop on Algorithms in Bioinformatics, Springer-Verlag, Berlin, Heidelberg, 2007, pp. 420–430.

[7] C.G. Moles, P. Mendes, J.R. Banga, Parameter estimation in biochemical pathways: a comparison of global optimization methods, Genome Res. 13 (11) (2003) 2467–2474.

[8] I.-C. Chou, H. Martens, E.O. Voit, Parameter estimation in biochemical systems models with alternating regression, Theoret. Biol. Med. Model. 3 (25) (2006) 1–11.

[9] H. Rohn, B. Ibrahim, T. Lenser, T. Hinze, P. Dittrich, Enhancing parameter estimation of biochemical networks by exponentially scaled search steps, in: EvoBIO'08: Proceedings of the 6th European Conference on Evolutionary Computation, Machine Learning and Data Mining in Bioinformatics, Springer-Verlag, Berlin, Heidelberg, 2008, pp. 177–187.

[10] O.R. Gonzalez, C. Küper, K. Jung, P.C. Naval, E. Mendoza, Parameter estimation using simulated annealing for s-system models of biochemical networks, Bioinformatics 23 (4) (2007) 480–486.

[11] R. Donaldson, D. Gilbert, A model checking approach to the parameter estimation of biochemical pathways, in: CMSB'08: Proceedings of the 6th International Conference on Computational Methods in Systems Biology, Springer-Verlag, Berlin, Heidelberg, 2008, pp. 269–287.

[12] A. Rizk, G. Batt, F. Fages, S. Soliman, On a continuous degree of satisfaction of temporal logic formulae with applications to systems biology, in: CMSB'08: Proceedings of the 6th International Conference on Computational Methods in Systems Biology, Springer-Verlag, Berlin, Heidelberg, 2008, pp. 251–268.

[13] H. Beyer, H. Schwefel, Evolution strategies — a comprehensive introduction, Nat. Comput. 1 (2002) 3–52.

[14] T.P. Runarsson, X. Yao, Stochastic ranking for constrained evolutionary optimization, IEEE Trans. Evol. Comput. 4 (3) (2000) 284–294.

[15] R. Tarjan, Depth-first search and linear graph algorithms, SIAM J. Comput. 1 (2) (1972) 146–170.

[16] G. Koh, D. Hsu, P.S. Thiagarajan, Incremental signaling pathway modeling by data integration, in: Res. in Comput. Mol. Biol, RECOMB, in: LNCS, vol. 6044, Springer-Verlag, 2010, pp. 281–296.

[17] M. Kalos, P. Whitlock, Monte Carlo Methods, vol. 1, John Wiley & Sons, New York, 1986.

[18] J. Pearl, Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference, Morgen Kaufmann, New York, 1988.

[19] R. McEliece, J. MacKay, J. Cheng, Turbo decoding as an instance of pearl's belief propagation algorithm, IEEE J. Sel. Areas Commun. 16 (1998) 140–152.

[20] P. Felzenszwalb, D. Huttenlocher, Efficient belief propagation for early vision, Int. J. Comput. Vis. 70 (2004) 41–54.

[21] A. Rickle, N. Bogdanovic, I. Volkman, B. Winblad, R. Ravid, R.F. Cowburn, Akt activity in alzheimer's disease and other neurodegenerative disorders, Neuroreport 15 (6) (2004) 955–959.

[22] S. Hardy, P.N. Robillard, Phenomenological and molecular-level petri net modeling and simulation of long-term potentiation, Biosystems 82 (1) (2005) 26–38.

[23] D. Langford, R. Hurford, M. Hashimoto, M. Digicaylioglu, E. Masliah, Signalling crosstalk in fgf2-mediated protection of endothelial cells from hiv-gp120, BMC Neurosci. 6 (2005) 8.

[24] 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.