In *Proc. Int. Conf. on Research in Computational Molecular Biology (RECOMB), 2010*

Incremental Signaling Pathway Modeling by Data Integration

Geoffrey Koh[1⋆], David Hsu[2], and P. S. Thiagarajan[2]

[1] Bioprocessing Technology Institute, Singapore, 117456, Singapore
[2] National University of Singapore, Singapore, 117417, Singapore

Abstract. Constructing quantitative dynamic models of signaling pathways is an important task for computational systems biology. Pathway model construction is often an inherently *incremental* process, with new pathway players and interactions continuously being discovered and additional experimental data being generated. Here we focus on the problem of performing model parameter estimation incrementally by integrating new experimental data into an existing model. A probabilistic graphical model known as the *factor graph* is used to represent pathway parameter estimates. By exploiting the network structure of a pathway, a factor graph compactly encodes many parameter estimates of varying quality as a probability distribution. When new data arrives, the parameter estimates are refined efficiently by applying a probabilistic inference algorithm known as *belief propagation* to the factor graph. A key advantage of our approach is that the factor graph model contains enough information about the old data, and uses only new data to refine the parameter estimates without requiring explicit access to the old data. To test this approach, we applied it to the Akt-MAPK pathways, which regulate the apoptotic process and are among the most actively studied signaling pathways. The results show that our new approach can obtain parameter estimates that fit the data well and refine them incrementally when new data arrives.

1 Introduction

To fully understand complex biological pathways, we must uncover not only the constituent elements—genes, proteins, and other molecular species—and their interactions, but also the *dynamics*, *i.e.*, the evolution of these interactions over time. One important goal of computational systems biology is to build quantitative models of pathway dynamics [1, 2]. These models should not only capture our understanding of the underlying mechanisms, but also predict behaviors yet to be observed experimentally. A key challenge is to address the inherently incremental nature of the model construction process, as new pathway players and interactions are discovered and additional experimental data are generated. In this work, we address the problem of incrementally constructing pathway models as new data becomes available.

A signaling pathway is a network of biochemical reactions. To build a model, we need both the network structure and the parameters. Structure modeling captures the interdependencies among the molecular species, based on the reactions producing and consuming them. Parameter modeling determines the kinetic rate constants, initial conditions, *etc.* that govern the biochemical reactions. Here, we focus on parameter modeling, also called parameter estimation.

⋆ The work was done when G. Koh was a PhD student at the National University of Singapore.

Parameter estimation for large signaling pathways is a well-known difficult problem, due to the need to search a high-dimensional parameter space and the lack of accurate data. Conventional parameter estimation algorithms fit an estimate of the parameters with all available experimental data and produce a single best estimate of the parameters (see [3] for a survey). When new data arrives, the entire procedure must be repeated afresh, in order to fit both the new and the old data well. This simplistic approach of recomputing the parameter estimate is undesirable. It does not take advantage of the earlier estimates. Furthermore, it may be not even be feasible, if the old data is not easily accessible. Often, many parts of the current model are obtained from external sources. For these "imported" parts, we have the estimated parameter values, but are unlikely to have access to the data used to produce these estimates. Hence we need a modeling approach that encodes the information from the old data compactly in the model itself and furthermore can *integrate* new data into an existing model to refine it.

We propose to use a probabilistic graphical model known as the *factor graph* [4] to represent pathway parameter estimates. We view a factor graph as a representation of a probability function $p(k_1, k_2, \dots)$ over the parameters $k_1, k_2, \dots$. A particular estimate of parameter values has high probability if it fits well with experimental data according to a suitable error measure. A factor graph represents many parameter estimates of varying quality, encoded as a probability function, rather than a single best estimate based on the existing data. A large pathway model typically involves many parameters. As a result, $p(k_1, k_2, \dots)$ is a high-dimensional function, which is expensive to compute and store. A key advantage of the factor graph model is that it exploits the *network structure* of a pathway to factor $p(k_1, k_2, \dots)$ as a product of lower-dimensional functions. This drastically reduces the complexity of representing $p(k_1, k_2, \dots)$ and allows parameter estimates to be refined efficiently.

To incorporate new data, we add new nodes to a factor graph and apply a probabilistic inference technique known as *belief propagation* (see [5] for a survey) to refine the parameter estimates represented by $p(k_1, k_2, \dots)$. Belief propagation reconciles the local constraints encoded in the new and the old factor graph nodes and ensures that they are globally consistent.

To test our approach, we applied it to the Akt-MAPK pathways. The kinase Akt plays an important role in regulating cellular functions, including, in particular, apoptosis, and has been identified as a major factor in several types of cancer. We created multiple data sets through simulation and introduced them one at a time into the factor graph model. The results show that our approach can obtain estimates that fit the data well and refine them incrementally when new data becomes available.

2 Background

2.1 Modeling Pathway Dynamics

The dynamics of a signaling pathway is often modeled as a system of nonlinear ordinary differential equations (ODEs):

(a) $x_1 + x_2 \underset{k_2}{\overset{k_1}{\rightleftharpoons}} x_3$

(b) $f_1 : \dot{x}_1 = k_2 x_3 - k_1 x_1 x_2$
$f_2 : \dot{x}_2 = k_2 x_3 - k_1 x_1 x_2$
$f_3 : \dot{x}_3 = k_1 x_1 x_2 - k_2 x_3$

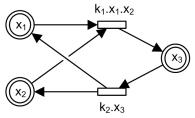
(c) 

Fig. 1. (a) A reaction, in which two substrates x_1 and x_2 bind reversibly to form a complex x_3. The speed of the forward and backward reactions depends on the kinetic rate constants k_1 and k_2, respectively. (b) The corresponding system of ODEs. (c) The HFPN model. The places are drawn as circles, and the transitions, as rectangles.

$$\begin{aligned} \dot{x}_1(t) &= f_1(x_1(t), x_2(t), \ldots; k_1, k_2, \ldots) \\ \dot{x}_2(t) &= f_2(x_1(t), x_2(t), \ldots; k_1, k_2, \ldots) \\ &\vdots \end{aligned}, \tag{1}$$

where $x_i(t)$ denotes the concentration level of molecular species i at time t and $\dot{x}_i(t)$ denotes the corresponding rate of change. Each function f_i, usually nonlinear, encodes the kinetics of the reactions that produce or consume x_i. The reactions are typically modeled with the mass action law or Michaelis-Menten kinetics [6], and we assume that the functions $f_1, f_2, \ldots$ are given. The kinetic rate constants $k_1, k_2, \ldots$ are parameter that govern the speed of reactions. See Fig. 1 for an example.

Using the vector notation, we can rewrite (1) more concisely as $\dot{\mathbf{x}}(t) = \mathbf{f}(\mathbf{x}(t); \mathbf{k})$, where $\mathbf{x}(t) = (x_1(t), x_2(t), \ldots)$, $\dot{\mathbf{x}}(t) = (\dot{x}_1(t), \dot{x}_2(t), \ldots)$, and $\mathbf{k} = (k_1, k_2, \ldots)$. Finally, we also need to specify the initial concentration levels $\mathbf{x}(0) = \mathbf{x}_0$.

The system of ODEs in (1) can be represented as a hybrid functional Petri net (HFPN) [7], which makes pathway structure explicit. A HFPN is a directed bipartite graph consisting of two types of nodes: *places* and *transitions*. In our case, places represent molecular species, and transitions represent reactions. The places and transitions are connected by *arcs* to indicate the flow of reactants and products. For an enzyme-catalyzed reaction, a *read arc*, shown pictorially as a dashed arc, connects an enzyme place to a catalyzed transition. It indicates that the enzyme influences, but is not consumed by the reaction. See [7] for more details on the HFPN model.

2.2 Parameter Modeling

An important step in building a pathway model is to determine the pathway parameters, which include kinetic rate constants and initial concentration levels of molecular species. Here we mainly deal with unknown kinetic rate constants, but the basic idea applies to unknown initial concentration levels as well.

Experimental determination of parameter values *in vitro* may not be possible or prohibitively expensive. A more practical approach is to estimate the parameter values based on experimental data. Suppose that we are given a set D of experimental data $\{\tilde{x}_{ij}\}$, where $\tilde{x}_{ij}$ is the experimentally measured concentration level of molecular species i at time T_j. The goal is to determine the values of the unknown parameters $\mathbf{k}$ so that the resulting pathway dynamics, *i.e.*, the evolution of molecular concentration levels over time, fits experimental data well. Mathematically, our goal consists of minimizing an objective function measuring the error in fit to data:

$$J(\mathbf{k}|D) = \sum_{i \in M} \sum_{j} (x_i(T_j; \mathbf{k}) - \tilde{x}_{ij})^2, \tag{2}$$

where M denotes the set of experimentally measured molecular species, and $x_i(t;\mathbf{k}), i = 1,2,\ldots$ are the solution to the system of ODEs in (1) with parameters $\mathbf{k}$. Typically we obtain $x_i(t;\mathbf{k}), i = 1,2,\ldots$ by simulating (1), using a numerical method. We can generalize $J(\mathbf{k}|D)$ by multiplying each term in (2) by a weight w_{ij} to favor the fit to data for some species at certain time over others. In the following, however, we use (2) to simplify the presentation. For multiple data sets $D_1, D_2, \ldots, D_n$, we simply sum up the error due to each data set and denote the total error by $J(\mathbf{k}|D_1, D_2, \ldots, D_n)$.

Standard estimation algorithms traverse the space of all parameter values and search for an optimal set of values with the best fit with D. A major challenge is that the size of the parameter space grows exponentially with the number of unknown parameters. Many different search strategies have been proposed to overcome this challenge, including local strategies (such as gradient descent) and global strategies (such as simulated annealing and evolutionary algorithms). See [3] for a survey, as well as [8, 9]. However, almost all current algorithms aim to find a single best parameter estimate based on the data available. This is inadequate for incremental pathway modeling: the single estimate cannot be easily improved when new data arrives. We propose instead to use a factor graph to represent a probability distribution that encodes multiple parameter estimates. Using this representation, we can refine the estimates systematically by adjusting their probabilities when new data becomes available.

Yoshida *et al.* adopts a similar probabilistic, data-driven view of parameter estimation [8], but their method assumes that *all* the data is available and is not geared towards incremental modeling. Factor graphs have been used to model biological systems [10], but the main goal there is to study the functional correlations among the molecular species in the pathway rather than the dynamics. An early use of belief propagation in computational biology is to predict protein secondary structure assignment [11].

3 Incremental Pathway Parameter Modeling

3.1 Overview

Often, experimental data are obtained in an incremental fashion. As a new data set D_n arrives at some time T_n with $T_1 < T_2 < T_3 < \cdots$, we want to incorporate D_n and compute a new estimate of the parameters $\mathbf{k}$. A simplistic approach would be to use all the data available up to time T_n, $\bigcup_{i=1}^{n} D_n$, and recompute the estimate of $\mathbf{k}$ from scratch. The error in fit to data is then given by $J(\mathbf{k}|D_1, D_2, \ldots, D_n)$. This approach, however, may be infeasible, because experimental data are generated by different research groups at different times. While the estimated parameter values may be published and accessible, the data used to produce these estimates is usually not. Recomputing the parameter estimate is also inefficient, as it does not take advantage of earlier estimates.

We would like to compute an estimate of $\mathbf{k}$ at time T_n using only D_n and the estimates obtained from the earlier data $\bigcup_{i=1}^{n-1} D_i$. To do so, we encode a set of estimates of $\mathbf{k}$ as a probability function

$$p(\mathbf{k}|D) = (1/\lambda)\exp(-J(\mathbf{k}|D)), \tag{3}$$

where D is a given data set, λ is a normalizing constant ensuring that $\int p(\mathbf{k}|D)\, d\mathbf{k} = 1$, and $J(\mathbf{k}|D)$ measures the error in fit to data, as defined in (2). The probability function $p(\mathbf{k}|D)$ encodes a set of parameter estimates, with large $p(\mathbf{k}|D)$ value indicating small

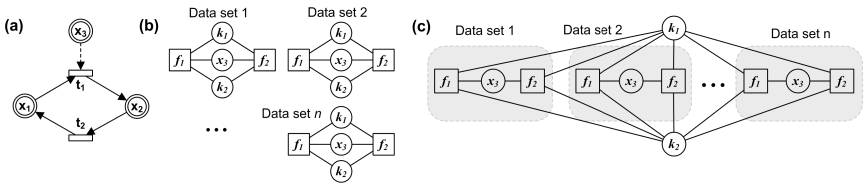


Fig. 2. (a) The HFPN model of an enzyme-mediated reversible reaction. (b) A factor graph S_n is constructed for each data set D_n. (c) The factor graphs are merged by fusing their common variable nodes representing unknown parameters.

error in fit to the data set D. In other words, we view $p(\mathbf{k}|D)$ as a probabilistic weight on $\mathbf{k}$, expressing preferences over $\mathbf{k}$ values due to the constraints from the data set D. Now suppose that $p(\mathbf{k}|D_1, D_2, \ldots, D_{n-1})$ represents the parameter estimates at time T_{n-1}. When a new data set D_n arrives at T_n, we use D_n to update the probabilistic weights on the estimates encoded by $p(\mathbf{k}; D_1, D_2, \ldots, D_{n-1})$ and obtain a new probability function $p(\mathbf{k}; D_1, D_2, \ldots, D_{n-1}, D_n)$. This is similar to Bayesian update, except that $p(\mathbf{k}|D)$ is basically a weight on $\mathbf{k}$ that depends on the error in fit to data $J(\mathbf{k}|D)$ and does not in itself have any real statistical meanings.

This incremental approach would be beneficial only if we can store and update $p(\mathbf{k}|D)$ efficiently. For a large pathway model with many unknown parameters, $p(\mathbf{k}|D)$ is a high-dimensional global function over the entire parameter space. However, each species in a typical signaling pathway interacts with only a small number of other species (see Fig. 5 for an example). We can exploit this insight on the *network structure* of a pathway to approximately factor the high-dimensional function $p(\mathbf{k}|D)$ into a product of lower-dimensional functions, and represent this factored probability function as a *factor graph* [4]. When combined with belief propagation (see Section 4), this representation helps us to find the best parameter estimates efficiently. Furthermore, it enables us to store and update $p(\mathbf{k}|D)$ efficiently in an incremental fashion.

Let S_n be a factor graph representing $p(\mathbf{k}|D_n)$, which, as mentioned earlier, represents preferences over $\mathbf{k}$ values due to the constraints from the data set D_n. In our incremental approach to parameter modeling, we compute a sequence of factor graphs $K_n, n = 1, 2, \ldots$, where $K_1 = S_1$ and K_n for $n \geq 2$ is obtained by merging S_n into K_{n-1} See Fig. 2 for an illustration. The merging process uses belief propagation to combine the preferences on $\mathbf{k}$ values represented by K_{n-1} with those represented by S_n. This results in new preferences represented by K_n.

We are ready to present the factor graph model for $p(\mathbf{k}|D)$. We begin with a brief introduction to factor graphs. We then describe how to construct a factor graph S_n, given a data set D_n and how to merge $S_1, S_2, \ldots, S_n$ incrementally to build K_n.

3.2 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

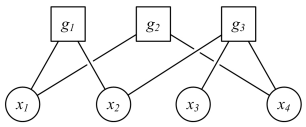


Fig. 3. The factor model for the function $g(x_1,x_2,x_3,x_4) = g_1(x_1,x_2)\cdot g_2(x_1,x_4)\cdot g_3(x_2,x_3,x_4)$.

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. 3.

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 modeling, the main variables are the parameters, and the constraints arise from the ODEs in which a parameter appears. For example, consider the reaction shown in Fig. 1. 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. The resulting factor graph represents the probability function $p(\mathbf{k}|D)$ as a product of factors, each involving only a small number of unknown parameters.

3.3 The Factor Graph Structure

Given a data set D, we now construct the factor graph S for the parameters of a system of ODEs modeling a biological pathway. For each equation $\dot{x}_i = f_i(\mathbf{x};\mathbf{k})$ in (1), we create a factor node $\nu(f_i)$ in S. 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. We insert an edge that connects a factor node for f_i and a variable node for k_j (or x_j), if k_j (or x_j) is involved in f_i. An example is shown in Fig. 4.

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 S. 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. 4. 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,

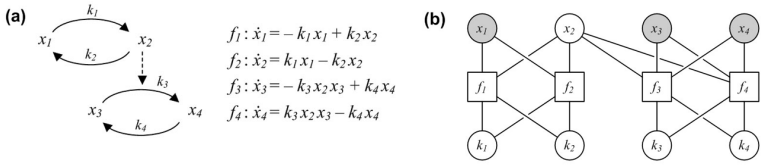


Fig. 4. (a) A simple signaling cascade and its ODEs. (b) The factor graph representation. The variable nodes in gray—x_1, x_3, and x_4—can be eliminated.

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 signaling pathway. Each factor captures the dependencies among the parameters involved in a particular equation.

3.4 The Compatibility Functions

To complete the construction of the factor graph S, we need to associate a factor, also called a *compatibility function*, with each factor node $\nu(f_i)$ and decomposes $p(\mathbf{k}|D)$ as a product of these compatibility functions. Although all compatibility functions depend on D, we drop the explicit mention of D in this section to simplify the notation. It is understood that compatibility functions are defined with respect to a given data set D. The compatibility function for $\nu(f_i)$ is given by

$$g_i(\mathbf{k}_i, \mathbf{x}_i(t)) = \exp(-E_i(\mathbf{k}_i, \mathbf{x}_i(t))), \tag{4}$$

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:

$$E_i(\mathbf{k}_i, \mathbf{x}_i(t)) = E_{i,1}(\mathbf{k}_i) + E_{i,2}(\mathbf{k}_i, \mathbf{x}_i(t)). \tag{5}$$

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:

$$E_{i,1}(\mathbf{k}_i) = \min_{\mathbf{k} \setminus \mathbf{k}_i} \sum_{m \in M} \sum_j (x_m(T_j; \mathbf{k}) - \tilde{x}_{mj})^2, \tag{6}$$

where $\mathbf{k} \setminus \mathbf{k}_i$ 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 in (1) with parameters $\mathbf{k}$, and finally $\tilde{x}_{mj}$ is the experimental concentration level of species m at time T_j.

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 in (1) with parameter values $\mathbf{k}_i$ and some suitable choice of values for parameters in $\mathbf{k} \setminus \mathbf{k}_i$. 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 S encodes exactly the function

$$g(\mathbf{k}, \mathbf{x}(t)) = \frac{1}{\lambda} \prod_i g_i(\mathbf{k}_i, \mathbf{x}_i(t)) = \frac{1}{\lambda} \exp\Big(-\sum_i E_i(\mathbf{k}_i, \mathbf{x}_i(t))\Big), \tag{7}$$

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 $p(\mathbf{k})$:

Theorem 1. *The following statements are equivalent:*

1. *The parameter values* $\mathbf{k}^*$ *minimize* $J(\mathbf{k})$.
2. *The parameter values* $\mathbf{k}^*$ *maximize* $p(\mathbf{k})$.
3. *The parameter values* $\mathbf{k}^*$ *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 in (1) with parameter values* $\mathbf{k}^*$.

The proof is given in Appendix A. This result implies that that to minimize $J(\mathbf{k})$ or maximize $p(\mathbf{k})$, we may equivalently maximize $g(\mathbf{k}, \mathbf{x}(t))$. Why do we want to do so? The reason is that although $g(\mathbf{k}, \mathbf{x}(t))$ is also a high-dimensional function, it is factored as a product of lower-dimensional functions represented by the factor graph S. We can maximize it effectively using belief propagation (Section 4), when searching for a parameter estimate with the best fit to data.

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:

$$E_{i,1}(\mathbf{k}_i) = \min_{\mathbf{k} \setminus \mathbf{k}_i} \sum_j (x_i(T_j;\mathbf{k}) - \tilde{x}_{ij})^2. \tag{8}$$

If x_i is not measured experimentally, we calculate $E_{i,1}(\mathbf{k}_i)$ globally using (6). Intuitively, calculating the fit to data locally strengthens the local constraints and makes belief propagation (Section 4) more greedy. This turns out to be helpful in our experiments (Section 4). However, it does not have the theoretical guarantee stated in Theorem 1.

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 [5]. It is not a severe limitation here, as the experimentally measured concentration levels for proteins in a signaling pathway often have very limited accuracy. Furthermore, once belief 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. More details regarding this can be found in Section 5. 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 (6) or (8). For this, we sample a representative set of parameter values and perform the minimization over the set of sampled values. This would be expensive computationally if performed on the space of all parameters. We need sophisticated sampling methods

such as Latin square sampling [12] to reduce the computational cost. Whenever possible, we also decompose a pathway model into components (Section 3.5). Sampling is performed only within a pathway component, which usually contains a small subset of parameters. This keeps the computational cost low.

3.5 Pathway Decomposition

For computational efficiency, we decompose a pathway into components. Each component usually contains only a small subset of unknown parameters. We build a factor graph S' for each component independently, assuming that the component is unaffected by the other components. Each component factor graph S' encodes a probability function expressing preferences over the values of the parameters contained in S'. To account for the dependency among the parameters from different components, we merge the component factor graphs and apply belief propagation (Section 4) to reconcile the different preferences over parameter values from each component. We do not have space here to describe this somewhat elaborate procedure. The details can be found in [13]. See Fig. 5 for an example of a decomposed pathway model.

3.6 Data Integration

Suppose that a sequence of data sets $D_1, D_2, \ldots$ arriving at time $T_1, T_2, \ldots$. Let K_n denote the factor graph for $p(\mathbf{k}|D_1, D_2, \ldots, D_n)$. We want to build K_n incrementally by integrating the data sets one at a time. At the nth stage, we first apply the procedure described above to construct a factor graph S_n for D_n. To construct K_n, we merge S_n with K_{n-1} by fusing their common variable nodes. Specifically, if a node of S_n represents the same unknown parameter as a node of K_{n-1}, they are merged as a single node in K_n. The edges are rearranged accordingly. Other nodes of S_n and K_{n-1} remain the same in K_n. See Fig. 2 for an illustration. It is important to note that although K_n takes into account all the data $\bigcup_{i=1}^{n} D_i$, the construction of K_n requires only D_n. Information from the earlier data sets $\bigcup_{i=1}^{n-1} D_i$ is encoded in K_{n-1}. Intuitively each new data set D_n adds a "slice" to our final factor graph K_n. So the size of K_n grows linearly with n.

We now turn to the important step of belief propagation, which reconciles the local constraints encoded by K_{n-1} and S_n.

4 Finding the Best Parameter Estimate

Theorem 1 shows that to find the minimum $\mathbf{k}^*$ of $J(\mathbf{k}|D_1, D_2, \ldots, D_n)$, we can equivalently maximize $g(\mathbf{k}, \mathbf{x}(t))$ represented by the factor graph K_n. We compute the maximum by applying a standard belief propagation (BP) algorithm called the max-product algorithm to K_n.

We give only a quick overview of BP here. See [4, 5] for comprehensive tutorials. Let G be a factor graph representing a factored non-negative function $g(\mathbf{z}) = g(z_1, z_2, \ldots) = \prod_i g_i(\mathbf{z}_i)$, 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 a probability function. Each variable node $v(z_j)$ of G 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 $g_i(\mathbf{z}_i)$ well 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 G contains no cycles, BP converges to the *global* maximum of the function that G represents [14]. 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 has generated good results in diverse applications [15, 16]. 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 apply BP to a factor graph representing the function $g(\mathbf{k},\mathbf{x}(t))$ in (7). 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 in (1). BP favors $\mathbf{k}$ and $\mathbf{x}$ values that satisfy these constraints well. It is also important to remember that when BP terminates, the variable nodes of the factor graph contain not only the parameter values with the best fit to existing data, but also alternative parameter values of varying quality weighted by the probabilities. These alternatives will become useful when new data arrives.

We run BP on each incrementally constructed factor graph K_n. For $n = 1$, the variable nodes of K_1 are initialized with the uniform probability distribution. For $n \geq 2$, the variable nodes of K_n are initialized with beliefs resulting from BP at the previous stage. Recall that K_n is obtained by merging K_{n-1} with a factor graph slice S_n representing the new data set D_n (Section 3.6). So BP has the effect of reconciling the constraints due to the new data (encoded in S_n) with those due to the earlier data (encoded in K_{n-1}) and favoring those parameter values with good fit to both the new and the old data.

5 Results

We tested our approach on the Akt-MAPK signaling pathways. The kinase Akt is a major factor in many types of cancer. The Akt pathway is one of the most actively studied kinase pathways, as it plays a key role in essential cellular functions, including cell survival, differentiation, *etc.* [17]. The Akt pathway interacts with several other pathways while performing its functions, in particular, the MAPK pathway.

In our earlier work [18], we performed parameter estimation on a combined model of the Akt-MAPK pathways using experimental data and studied the crosstalk between them. In the present setting, due to the lack of sufficient number of experimental data sets, we used synthetic data. The Akt-MAPK model used in our case study contains 36 molecular species and 42 unknown parameters. See Fig. 5. A larger figure along

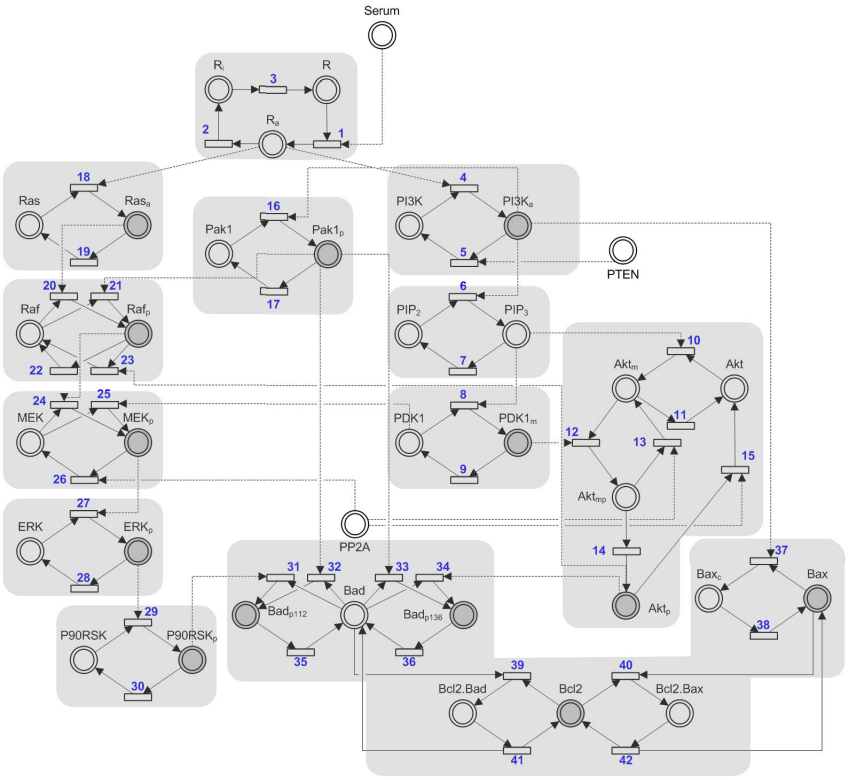


Fig. 5. The HFPN model of the Akt-MAPK pathways. A place node in the model is shaded in gray if data is available for the corresponding molecular species. The light gray boxes indicate the components obtained through pathway decomposition.

with model parameter values is available at `http://www.comp.nus.edu.sg/~rpsysbio/recomb2010`. We generated six data sets by simulating the model under different knockdown conditions, in which the initial concentration level of each of six molecular species—Akt, PDK1, PP2A, PTEN, and the cell receptor—is reduced. Each data set contains concentration levels of 13 molecular species at 50 time points.

We normalized the value of each parameter to a range between 0 and 1 and divided the range into 10 equally-sized intervals. Due to the discretization, belief propagation produces the best parameter intervals rather than exact values. As a post-processing step, we apply the Levenberg-Marquardt algorithm [19], starting from the mid-points of the best parameter intervals obtained from belief propagation. This gives us the final parameter estimate that minimizes the error in fit to data.

As mentioned in Section 1, a key goal of our work is to address the issue of not having access to all the data at the same time. In our test, we introduced the six data sets one at a time, in an arbitrary, but fixed order. At each stage, to perform parameter estimation, we used only one data set along the factor graph model from the earlier estimation; all other data sets were kept away. Since sampling is used during the factor graph modeling (Section 3.4), we repeated each test 10 times.

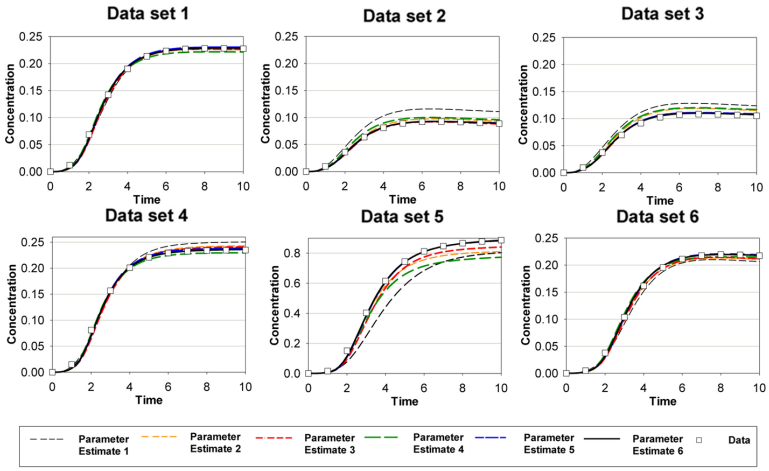


Fig. 7. The change in concentration level over time for Bad_{p136} under six knockdown conditions. The six curves in each plot correspond to the six different parameter estimates as more data sets are integrated. Data points are shown every 5 time steps to avoid cluttering the plots.

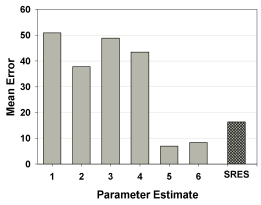


Fig. 6. The error in fit to data, as six data sets were introduced one at a time. The darker bar indicates the error of the parameter estimate obtained by SRES, using all six data sets.

Fig. 6 shows the mean error in fit to data over the 10 runs in each stage. To examine the benefits of using multiple data sets for parameter estimation, the error is measured according to (2) using all six data sets. The plot shows that as more data are used, the error generally decreases, as expected. Fig. 7 shows the concentration level of Bad (Bcl2 antagonist of cell death), an important downstream protein in the pathway. Each plot shows how the concentration level changes over time under one of the six knockdown conditions. Figs. 6 and 7 indicate that the fit to data improves, as more data sets are introduced to refine the parameter estimate. For example, parameter estimate 1 causes substantial error in fit to data set 2, 3, and 5, while parameter estimate 6, after integrating all data, fits well with all data sets. The results confirm that our approach can integrate new data and improve the parameter estimates effectively.

Next, we compared our results with that from COPASI [20], a well-established pathway simulator with parameter estimation functions. COPASI contains several methods for parameter estimation. We used SRES, which is the best based our experiences. We ran SRES for an extended duration (10 hours), using all *six data sets*. After integrating enough data sets, our approach of incremental parameter modeling obtained comparable and better estimates (Fig. 6). The results suggest that our incremental approach through data integration does not sacrifice parameter estimation accuracy, compared with global estimation methods that require access to all the data sets at once.

To test the robustness of our approach, we considered four additional knockdown conditions by combining the knockdown conditions specified earlier. We generated four new data sets under these additional conditions and computed the error in fit to data for

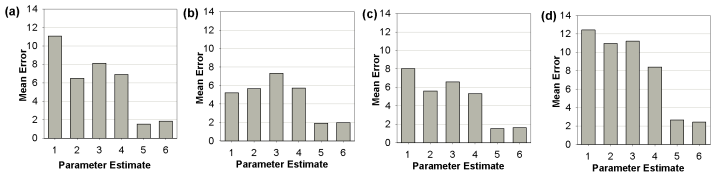


Fig. 8. The error in fit to data under combination of knockdown conditions.

the six parameter estimates obtained earlier (Fig. 8). We did not recompute the parameter estimates using the additional data, as the purpose here is to check the robustness of the estimates obtained earlier under new conditions. The results indicate a trend similar to that shown in Fig. 6.

As more data sets are integrated, we expect that the uncertainty of parameter estimates decreases. Fig. 9 shows the change in the standard deviations of some estimated parameters as the number of data sets increases. There is a general decrease in the standard deviations for all estimated parameters, indicating that data integration is effective for reducing the uncertainty of estimated parameters. For some parameters, such as the ones shown in Fig. 9, the uncertainty is very low, after six data sets were integrated. However, for some other parameters, including $k_7, k_9, k_{10}, k_{11}, k_{12}, k_{13}, k_{21}, k_{22}, k_{32}, k_{37}$, k_{40}, k_{41}, k_{42}, the uncertainty remains large. The graphical model of the pathway (Fig. 5) reveals that such parameters are mostly associated with molecular species that are either (i) involved in several reactions, *e.g.*, Akt_m, Raf, Bad, or (ii) have insufficient data to constrain their values, *e.g.*, PIP_3, Akt_m. This observation suggests that biological pathways are less sensitive to parameter variations around molecular species involved in more than one set of production-consumption reactions. The uncertainty level in parameter estimates can also provide guidance to biologists in the subsequent design of their experiments to further constrain important pathway parameters.

6 Conclusion

Pathway model construction is often an incremental process, as new experiments lead to discoveries of additional players and interactions in a pathway. This paper presents a data integration approach to incremental pathway parameter modeling. We use the factor graph as a probabilistic model of pathway parameter estimates. It enables us to refine the parameter estimates in a principled and efficient manner, when new data becomes available. A main benefit of our approach is that the factor graph model compactly encodes the information from old data in itself and uses only new data to refine the parameter estimates. It eliminates the unrealistic requirement of having access to all data, both old and new, in order to improve the parameter estimates.

Several aspects of our approach require further investigation. So far, we have only tested it with unknown kinetic rate constants as parameters. Our approach can also deal with unknown initial molecular concentration levels by treating them as parameters, but we are yet to implement and test our approach to handle this variant. We also need to test this method on multiple signaling pathway models using real experimental data.

An important underlying idea of our approach is to *compose* factor graph models. The current work exploits temporal composition by merging successive slices of factor

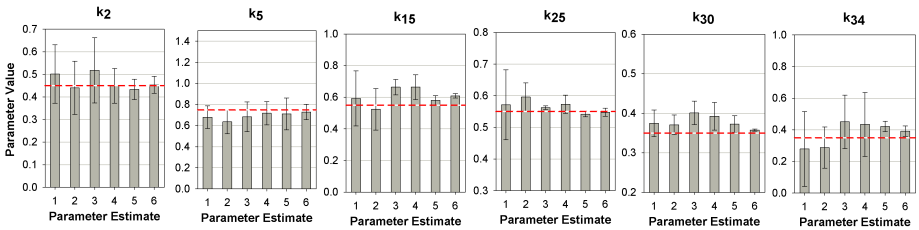


Fig. 9. The mean values and the standard deviations of the estimated parameters over 10 runs. The bars indicate the mean values of estimated parameters. The error bars indicate the standard deviations. The dashed lines indicate the nominal parameter values.

graphs representing new data sets. This allows us to integrate new data and refine model parameters. We can go one and exploit spatial composition. When new experiments suggest additional components of a pathway or interacting pathways, we may compose the models for these components and pathways to form a single model. Spatial composition allows us to expand a model and incorporate missing players and interactions. The pathway decomposition technique described briefly in Section 3.5 in fact constitutes a special case of spatial composition, but more work is needed to explore spatial composition methods. Together temporal and spatial compositions create a modeling framework that supports model refinement and expansion systematically.

Acknowledgments. This work is supported in part by AcRF grant R-252-000-350-112 from the Ministry of Education, Singapore.

References

1. Bhalla, U.S., Iyengar, R.: Emergent properties of networks of biological signaling pathways. Science **283** (1999) 381–387
2. Aldridge, B.B., Burke, J.M., Lauffenburger, D.A., Sorger, P.K.: Physicochemical modelling of cell signalling pathways. Nature Cell Biology **8**(11) (2006) 1195–1203
3. Moles, C.G., Mendes, P., Banga, J.R.: Parameter estimation in biochemical pathways: A comparison of global optimization methods. Genome Research **13**(11) (2003) 2467–2474
4. Kschischange, F., Frey, B., Loeliger, H.: Factor graphs and the sum-product algorithm. IEEE Trans. on Information Theory **42**(2) (2001) 498–519
5. Koller, D., Friedman, N.: Probabilistic Graphical Models: Principles and Techniques. The MIT Press (2009)
6. Klipp, E., et al.: Systems Biology in Practice. Wiley-VCH (2005)
7. Matsuno, H., Tanaka, Y., Aoshima, H., Doi, A., Matsui, M., Miyano, S.: Biopathways representation and simulation on hybrid functional Petri net. In Silico Biology **3**(3) (2003) 389–404
8. Yoshida, R., Nagasaki, M., Yamaguchi, R., Imoto, S., Miyano, S., Higuchi, T.: Bayesian learning of biological pathways on genomic data assimilation. Bioinformatics **24**(22) (2008) 2592–2601
9. Purvis, J., Radhakrishnan, R., Diamond, S.: Steady-state kinetic modeling constrains cellular resting states and dynamic behavior. PLoS Computational Biology **5**(3) (2009)
10. Gat-Viks, I., Tanay, A., Raijman, D., Shamir, R.: The factor graph network model for biological systems. In: Proc. Int. Conf. on Research in Computational Molecular Biology (RECOMB). (2005)

11. Delcher, A., Kasif, S., Goldberg, H., Hsu, W.: Protein secondary structure modelling with probabilistic networks. In: Proc. Int. Conf. on Intelligent Systems & Molecular Biology. (1993) 109–117
12. Kalos, M., Whitlock, P.: Monte Carlo Methods. Volume 1. John Wiley & Sons, New York (1986)
13. Koh, G.: Pathway Models Decomposition and Composition Techniques for Parameter Estimation. PhD thesis, Graduate School of Integrative Sciences, National University of Singapore, Singapore (2008)
14. Pearl, J.: Probabilistic reasoning in intelligent systems: networks of plausible inference. Morgan Kaufmann (1988)
15. Felzenszwalb, P.F., Huttenlocher, D.P.: Efficient belief propagation for early vision. Int. J. Computer Vision **70**(1) (2004) 41–54
16. McEliece, R.J., MacKay, D.J., Cheng, J.F.: Turbo decoding as an instance of Pearl's "Belief Propagation" algorithm. IEEE J. on Selected Areas in Communications **16**(2) (1998) 140–152
17. Brazil, D.P., Yang, Z.Z., Hemmings, B.A.: Advances in protein kinase B signalling: AKTion on multiple fronts. Trends in Biochemical Sciences **29**(5) (2004) 233–242
18. Koh, G., Teong, H.F.C., Clément, M.V., Hsu, D., Thiagarajan, P.: A decompositional approach to parameter estimation in pathway modeling: a case study of the Akt and MAPK pathways and their crosstalk. Bioinformatics **22**(14) (2006) e271–e280
19. Gill, P., Murray, W., Wright, M. In: Practical Optimization. Academic Press (1982)
20. Hoops, S., Sahle, S., Gauges, R., Lee, C., Pahle, J., Simus, N., Singhal, M., Xu, L., Mendes, P., Kummer, U.: COPASI—a COmplex PAthway SImulator. Bioinformatics **22**(24) (2006) 3067–3074

A Proof of Theorem 1

Proof. Since $p(\mathbf{k}) = (1/\lambda)\exp(-J(\mathbf{k}))$ and the exponential function is monotonic, the equivalence between statements 1 and 2 clearly holds.

We now prove the equivalence between statements 1 and 3. Define $E(\mathbf{k},\mathbf{x}(t)) = \sum_i E_i(\mathbf{k}_i,\mathbf{x}_i(t))$. Since we want to minimize $E(\mathbf{k},\mathbf{x}(t))$, we are only interested in the case when $E(\mathbf{k},\mathbf{x}(t))$ is finite. The function $E(\mathbf{k},\mathbf{x}(t))$ is finite if and only if $\mathbf{k}_i$ and $\mathbf{x}_i$ are consistent for all i. Let $\mathbf{x}(t;\mathbf{k})$ denote the concentration levels consistent with the parameters $\mathbf{k}$. In this case, $E_{i,2}(\mathbf{k}_i,\mathbf{x}_\mathbf{i}(t)) = 0$ for all i. Using this and (6), we then get

$$\begin{aligned}\min_{\mathbf{k}} E(\mathbf{k},\mathbf{x}(t;\mathbf{k})) &= \min_{\mathbf{k}}\Big(\sum_i E_{i,1}(\mathbf{k}_i,\mathbf{x}_i(t))\Big)\\ &= \min_{\mathbf{k}}\Big(\sum_i \min_{\mathbf{k}\setminus\mathbf{k}_i}\sum_{m\in M}\sum_j (x_m(t_j;\mathbf{k}) - \tilde{x}_{mj})^2\Big)\\ &= \min_{\mathbf{k}}\Big(\sum_i \min_{\mathbf{k}\setminus\mathbf{k}_i} J(\mathbf{k})\Big) \qquad (9)\end{aligned}$$

Note that $\min_{\mathbf{k}\setminus\mathbf{k}_i} J(\mathbf{k})$ is a function of $\mathbf{k}_i$. If $\mathbf{k}^*$ minimizes $J(\mathbf{k})$, then $\mathbf{k}_i^*$ minimizes $\min_{\mathbf{k}\setminus\mathbf{k}_i} J(\mathbf{k})$ for all i. It then follows from (9) that

$$\min_{\mathbf{k}} E(\mathbf{k},\mathbf{x}(t;\mathbf{k})) = \sum_i \min_{\mathbf{k}} J(\mathbf{k}).$$

Since $g(\mathbf{k},\mathbf{x}(t)) = (1/\lambda)\exp(-E(\mathbf{k},\mathbf{x}(t)))$, the conclusion follows. □

B HFPN Model of the Akt-MAPK Signaling Pathways

The figure below shows the HFPN model of the Akt-MAPK signaling pathways. This pathway model contains 36 molecular species and 42 unknown kinetic rate constants. The equations associated with each transition are shown in Table 1. The nominal values of the rate constants fall within the interval [0.0, 1.0]. The molecular concentrations of the various species fall within the interval [0.0, 5.0]. The nominal values of the rate constants and the initial concentration levels of the molecular species are given in Tables 2 and 3.

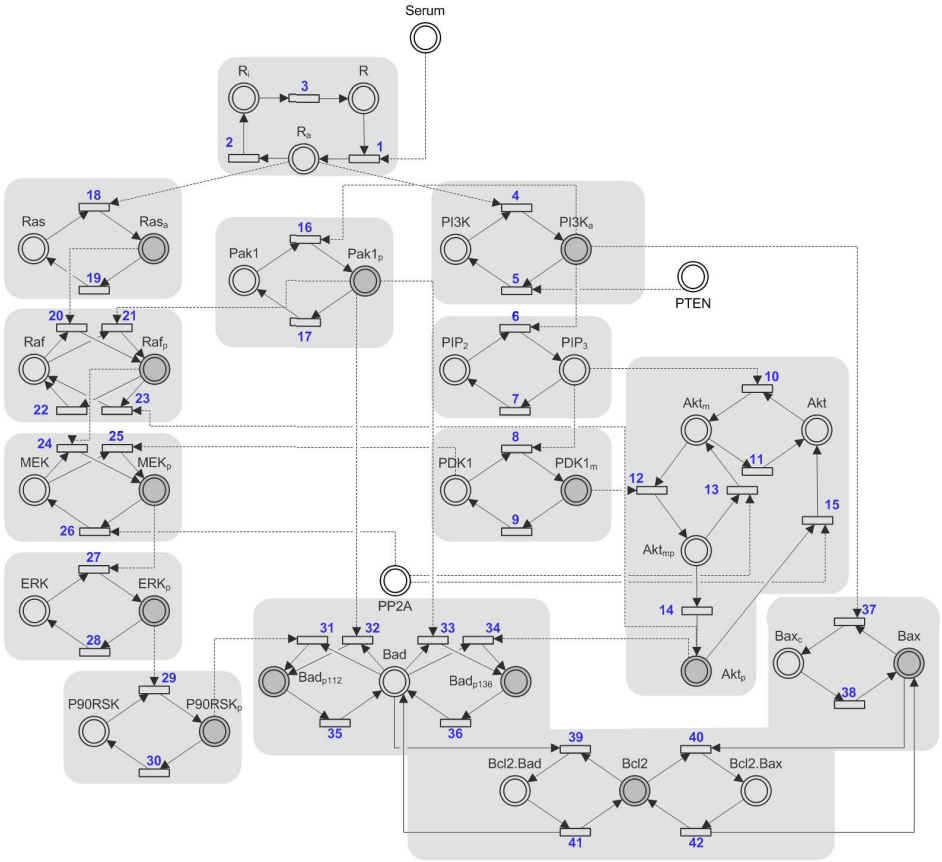


Fig. 1. The HFPN model of the Akt-MAPK pathways. A place node in the model is shaded in gray if data is available for the corresponding molecular species. The light gray boxes indicate the components obtained through pathway decomposition.

Table 1. Equations for the reactions in the pathway. The molecular species are denoted in "[]".

No	Rate Equation	No	Rate Equation	No	Rate Equation	No	Rate Equation
1	$k_1[R]$	12	$k_{12}[PDK1_m][Akt_m]$	23	$k_{23}[Akt_p][Raf_p]$	34	$k_{34}[Akt_p][Bad]$
2	$k_2[R_a]$	13	$k_{13}[PP2A][Akt_{mp}]$	24	$k_{24}[Raf_p][MEK]$	35	$k_{35}[Bad_{p112}]$
3	$k_3[R_i]$	14	$k_{14}[Akt_{mp}]$	25	$k_{25}[PDK1][MEK]$	36	$k_{36}[Bad_{p136}]$
4	$k_4[R_a][PI3K]$	15	$k_{15}[PP2A][Akt_p]$	26	$k_{26}[MEK_p][PP2A]$	37	$k_{37}[PI3K_a][Bax]$
5	$k_5[PTEN][PI3K_a]$	16	$k_{16}[PI3K_a][Pak1]$	27	$k_{27}[MEK_p][ERK]$	38	$k_{38}[Bax_c]$
6	$k_6[PI3K_a][PIP_2]$	17	$k_{17}[Pak1_p]$	28	$k_{28}[ERK_p]$	39	$k_{39}[Bad][Bcl2]$
7	$k_7[PIP_3]$	18	$k_{18}[R_a][Ras]$	29	$k_{29}[ERK_p][P90RSK]$	40	$k_{40}[Bax][Bcl2]$
8	$k_8[PIP_3][PDK1]$	19	$k_{19}[Ras_a]$	30	$k_{30}[P90RSK_p]$	41	$k_{41}[Bcl2.Bad]$
9	$k_9[PDK1_m$	20	$k_{20}[Ras_a][Raf]$	31	$k_{31}[P90RSK_p][Bad]$	42	$k_{42}[Bcl2.Bax]$
10	$k_{10}[PIP_3][Akt]$	21	$k_{21}[Pak1_p][Raf]$	32	$k_{32}[Pak1_p][Bad]$		
11	$k_{11}[Akt_m]$	22	$k_{22}[Raf_p]$	33	$k_{33}[Pak1_p][Bad]$		

Table 2. Nominal values for the unknown kinetic rate constants.

Parameter	Value	Parameter	Value	Parameter	Value	Parameter	Value	Parameter	Value
k_1	0.55	k_{11}	0.65	k_{21}	0.75	k_{31}	0.05	k_{41}	0.95
k_2	0.35	k_{12}	0.65	k_{22}	0.85	k_{32}	0.65	k_{42}	0.45
k_3	0.05	k_{13}	0.65	k_{23}	0.55	k_{33}	0.05		
k_4	0.85	k_{14}	0.75	k_{24}	0.45	k_{34}	0.35		
k_5	0.75	k_{15}	0.55	k_{25}	0.55	k_{35}	0.85		
k_6	0.55	k_{16}	0.05	k_{26}	0.35	k_{36}	0.95		
k_7	0.95	k_{17}	0.95	k_{27}	0.95	k_{37}	0.75		
k_8	0.25	k_{18}	0.25	k_{28}	0.85	k_{38}	0.15		
k_9	0.45	k_{19}	0.35	k_{29}	0.05	k_{39}	0.85		
k_{10}	0.65	k_{20}	0.45	k_{30}	0.35	k_{40}	0.85		

Table 3. Initial concentration levels of the molecular species.

Species	Concentration	Species	Concentration	Species	Concentration	Species	Concentration
Serum	1.0	$PDK1_m$	0.0	Raf_p	0.0	Bax	0.0
PP2A	5.0	Akt	5.0	MEK	5.0	Bax_c	5.0
R	5.0	Akt_m	0.0	MEK_p	0.0	Bcl2	5.0
R_a	0.0	Akt_{mp}	0.0	ERK	5.0	Bcl2.Bad	0.0
R_i	0.0	Akt_p	0.0	ERK_p	0.0	Bcl2.Bax	0.0
PI3K	5.0	Pak1	5.0	P90RSK	5.0	PTEN	1.0
$PI3K_a$	0.0	$Pak1_p$	0.0	$P90RSK_p$	0.0		
PIP_2	5.0	Ras	5.0	Bad	5.0		
PIP_3	0.0	Ras_a	0.0	Bad_{p112}	0.0		
PDK1	5.0	Raf	5.0	Bad_{p136}	0.0		