Composing Globally Consistent Pathway Parameter Estimates Through Belief Propagation

Geoffrey Koh[1], Lisa Tucker-Kellogg[2], David Hsu[1,2], and P.S. Thiagarajan[1,2]

[1] Graduate School for Integrative Sciences and Engineering,
National University of Singapore
g0306427@nus.edu.sg
[2] Department of Computer Science, National University of Singapore
{tucker,dyhsu,thiagu}@comp.nus.edu.sg

Abstract. Parameter estimation of large bio-pathway models is an important and difficult problem. To reduce the prohibitive computational cost, one approach is to decompose a large model into components and estimate their parameters separately. However, the decomposed components often share common parts that may have conflicting parameter estimates, as they are computed independently within each component. In this paper, we propose to use a probabilistic inference technique called *belief propagation* to reconcile these independent estimates in a principled manner and compute new estimates that are globally consistent and fit well with data. An important advantage of our approach in practice is that it naturally handles incomplete or noisy data. Preliminary results based on synthetic data show promising performance in terms of both accuracy and efficiency.

1 Introduction

Quantitative modeling of the dynamics of bio-pathways - gene regulatory networks, metabolic pathways and signaling pathways - can play a vital role in understanding fundamental intra- and inter-cellular processes. Abstractly, a bio-pathway can be viewed as a network of biochemical reactions modeled as a simultaneous system of differential equations. In practice, the values of many of the rate parameters governing these reactions (equations) are often unknown. Hence techniques for estimating the values of these unknown parameters are of considerable importance.

To perform parameter estimation of large bio-pathway models, we face two major challenges: (i) a high-dimensional search space, due to a large number of unknown parameters and (ii) insufficient and noisy data. In our earlier work (Koh *et al.*, 2006), we have proposed a decompositional approach to address the first challenge. To decompose a large model into smaller components, we exploit the structure of a large pathway model and the distribution of locations where experimental data is available within the pathway. We then estimate the parameters for each component separately. The decompositional approach dramatically

R. Giancarlo and S. Hannenhalli (Eds.): WABI 2007, LNBI 4645, pp. 420–430, 2007.

improves computational efficiency; however, the decomposed pathway components often share common parts that may have conflicting parameter estimates, as they are computed independently within each individual component. A key question, which is the focus of this work, is then how to reconcile these different estimates in a principled manner.

Our main idea is to use a probabilistic inference technique called *belief propagation* (Yedidia *et al.*, 2003). We model the value of an unknown parameter as a probability distribution, commonly called a *belief* in this context. We then propagate and update the beliefs of the unknown parameters within the pathway model. This way, local beliefs arising from different pathway components are collated to construct globally consistent beliefs for all the parameters. Another important advantage of our approach is that it naturally handles incomplete or noisy data and helps to address the second challenge described earlier.

We have implemented a discretized version of our belief propagation algorithm and performed initial tests. Simulation results using the estimated parameter values show good correlation with (synthetic) experimental data. Furthermore, based on the performance of belief propagation in many other applications (e.g. Felzenszwalb and Huttenlocher, 2006, Ihler *et al.*, 2004, Friedman, 2004), we expect that our algorithm will scale up well with the pathway size.

The rest of this paper is organized as follows. In Section 2, we review background material and related work. In Section 3, we describe a probabilistic graphical model called the *Factor Graph* and show how it can be used to represent a bio-pathway. We then explain the details of our parameter estimation algorithm. It applies belief propagation on a Factor Graph model of a pathway in order to reconcile the parameter estimates for the individual components and yield globally consistent parameter estimates for the entire pathway. In Section 4, we present simulation results on the performance of our algorithm. In the final section, we conclude and discuss the prospects for future work.

2 Background

We first briefly review the background on bio-pathway modeling and then the parameter estimation problem. Here and in the rest of the paper, we focus on signaling pathways in eukaryotic cells.

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

$$\dot{x}_i = f_i(\mathbf{x}(t), \mathbf{p}) \tag{1}$$

where $\dot{x}_i$ denotes the rate of change of the concentration level for the species x_i (typically a protein). The vector $\mathbf{x}(t)$ denotes the concentration levels of the various species at time t while $\mathbf{p}$ is the set of parameters, many of which will be unknown and have to be estimated. The nonlinear function f_i encodes the rates of the reactions that produce or consume x_i. Without loss of generality, we restrict our discussion and examples to mass action kinetics where the rate of reaction is proportional to the concentration of the participating molecules.

In this setting, a typical reaction can be written as Equation 2. The substrates x_1 and x_2 bind reversibly to form the complex x_3 at a rate that is affected by the forward and reverse rate constants k_1 and k_2.

$$x_1 + x_2 \underset{k_2}{\overset{k_1}{\rightleftharpoons}} x_3 \tag{2}$$

The system of equations that describe this reaction is

$$\begin{aligned}
\dot{x}_1 &= k_2 x_3 - k_1 x_1 x_2 \\
\dot{x}_2 &= k_2 x_3 - k_1 x_1 x_2 \\
\dot{x}_3 &= k_1 x_1 x_2 - k_2 x_3
\end{aligned}$$

Clearly, the correctness of a pathway model crucially depends on the values of the parameters. Determining these parameters through wet-lab experiments consumes significant time and cost, and is sometimes impossible. Hence, one must resort to computational techniques to estimate their values, based on available experimental data.

Parameter estimation can be viewed as an optimization problem with differential-algebraic constraints (e.g. Kikuchi *et al.*, 2003, Moles *et al.*, 2003). The algebraic constraints result from the input data, which consists of experimentally measured gene expression levels or protein concentration levels at selected discrete time points. The differential constraints come from the ODEs that govern the biochemical reactions in the pathway. The problem is to estimate the pathway parameters (initial conditions and kinetic rate constants) and all the unknown gene expression levels and protein concentration levels so as to fit the experimental data as best as possible according to a suitable error measure (see Koh *et al.*, 2006 for details).

Many approaches are available to solve this optimization problem, including standard local descent and evolutionary strategies. Each algorithm has its own merits and limitations (Mendes and Kell, 1998, Moles *et al.*, 2003). A common characteristic of these techniques is that they consider the entire pathway and all its parameters simultaneously during estimation. In general, this leads to a combinatorial explosion in terms of the dimensionality of the parameter space being searched. Therefore, the methods do not scale well for large pathway models having many unknown parameters. Furthermore, the existing methods provide point estimates as solutions, which is unrealistic.

In our previous work, we have tackled the high-dimensionality barrier by exploiting the structure of the pathway to break it down into smaller components so that parameter estimation can be done by parts. The identification of a component is achieved by back-tracing from the observed molecules until there are no more molecules to include in the component, or when we encounter molecules that are already observed at all time points, possibly due to prior simulations. As a key consequence, the resulting component can be *simulated* on its own. We have shown that this approach can produce reasonable estimates using much less time compared to global methods. However, the components defined by our method are not necessarily disjoint. When two components overlap, the parameter values of the common portion are fixed to be those associated with one of

the components. This is one limitation which has not been explored previously, especially when experimental data is available about the unshared parts in both the components.

In this work we propose to use belief propagation to address this critical difficulty. Belief propagation is a message passing protocol that operates on probabilistic graphical models such as Bayesian Networks and Markov Random Fields (Pearl 1988). Belief propagation has been applied in systems biology for the analysis of gene regulatory networks and for inferring network structure. See, for example, Yeang and Jaakkola, 2003 and Friedman, 2004. In Gat-Viks *et al.*, 2005, it was used to develop a method that incorporates prior biological knowledge and learn a refined model with improved fit to experimental data.

Interestingly, as we show in this paper, belief propagation can be used to integrate the estimates provided by the different components that share portions of the pathway such that they are globally consistent.

3 Parameter Estimation by Belief Propagation

Belief propagation provides a principled way of reconciling inconsistent estimates that arise from decomposition. The reasons for these inconsistencies are twofold: (i) the multimodal solution landscape, which may lead to the algorithms converging to different solutions, and (ii) noisy and sparse data sets being used for the various components and pathways.

Our method consists of the following steps. We first decompose the pathway model into smaller components using the method presented in Koh *et al.*, 2006 (See Section 2 for a short summary). In the second step, we convert each component into a probabilistic graphical model, which in our current setting is the Factor Graph (Kschischang *et al.*, 2001). The parameters are given a probability distribution (belief) over the interval between their upper and lower bounds. The initial beliefs of the unknown parameters are uniformly distributed. However, one can also assign them non-uniform distributions to reflect any prior knowledge about their values. Functional dependencies between the parameters are captured by building *compatibility functions* of each factor node via sampling. The next step is to compose the Factor Graphs together to form a larger Factor Graph. Finally, we use belief propagation to reconcile the local estimates, so as to generate globally consistent estimates for all the parameters. In addition, we may refine the estimates through local descent on the entire pathway.

3.1 Factor Graph

A Factor Graph is an undirected bipartite graph consisting of *factor nodes* and *variable nodes*. In the present setting, where each equation in the system of ODEs is of the form $\dot{x}_i = f_i(\mathbf{x}(t), \mathbf{p})$, we will have one factor node for each such f_i. For convenience, we will denote this factor node as F_i. The variable nodes then denote the unknown parameters. An edge exists between the factor node F_i and

the parameter k iff k appears in f_i or some x_j (representing the concentration level of the jth molecular species) appears in f_i and k is an argument of f_j.

An example of a simple pathway model, its system of equations and the associated Factor Graph is shown in Figure 1. In this Factor Graph, there is an edge from factor node F_2 to each of the variable nodes k_1, k_2, k_3 and k_4. This is because k_3 and k_4 are arguments of f_2 while k_1 and k_2 directly affect the rate of change of x_1, which is also an argument of f_2.

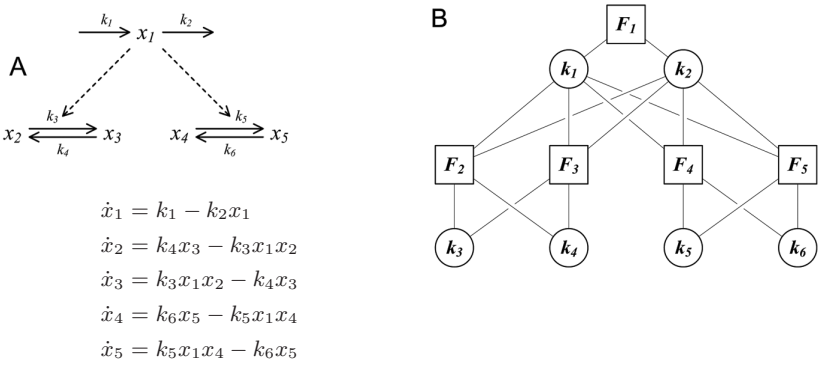


Fig. 1. (A) Simple pathway model with its system of ODEs (B) The Factor Graph representation of the pathway. The round nodes are variable nodes and the square nodes are factor nodes.

3.2 Compatibility Function

With each factor node F_i, we associate a compatibility function ψ_i over the parameters that are connected to it. This compatibility function is defined by a joint probability distribution between the values of the parameters. This distribution has to be built up by *sampling* the parameter space and scoring the samples using the following objective function

$$J(\mathbf{p}_i) = \left(\sum_{k \in \mathbf{x}_{obs}, j} (x_{kj}^{\text{sim}}(\mathbf{p}) - x_{kj}^{\text{exp}})^2 / w_{kj}^2 \right)^{1/2} \tag{3}$$

where x_{kj}^{exp} is the jth experimental data point of variable x_k and $x_{kj}^{\text{sim}}(\mathbf{p})$ is the corresponding predicted value generated by simulating the ODEs using the sampled values of the parameters $\mathbf{p}$. The set $\mathbf{p}_i$ consists of the parameters that are connected to the factor node F_i. Finally w_{kj} is the weight, typically the maximal value of x_k, used to normalize its contribution to the objective function. Within a component, we distinguish between *local* score and *global* score. A local score is obtained by applying the objective function on a single observed variable while a global score is obtained by using all the observed variables. For a factor node whose corresponding variable x_i is observed, its compatibility function will be derived using the local score, i.e. $\mathbf{x}_{obs} = \{x_i\}$. Otherwise the global score will be used.

The scores are then converted into probabilities by the following equation

$$\psi_i(\mathbf{p}_i) = \frac{1}{z} e^{-\frac{\mu_i J(\mathbf{p}_i)}{\max J(\mathbf{p}_i)}} \tag{4}$$

where z is a normalizing constant to ensure that the probabilities sum up to 1, and μ_i is a scaling factor, which is usually set to a value between 10 and 20, depending on the accuracy of the experimental data. These distributions capture the dependencies between the parameters, and they are immutable.

3.3 Composing the Factor Graph for the Entire Pathway

Suppose that we have possibly overlapping components after pathway decomposition. Each of them is converted into a Factor Graph as described above and sampled separately to build their respective compatibility functions. To reconcile the beliefs of their parameters, we compose a single Factor Graph for the entire pathway by fusing together their common *variable nodes* (e.g. Figure 3B). These variable nodes will allow information from one Factor Graph to propagate to the other.

3.4 Message Passing and Updating

Having constructed the joint probability distributions between the parameters and recomposed the Factor Graphs, we can now update the beliefs of their parameters. This is achieved by forming messages and passing them between the nodes of the Factor Graph. As the messages are being propagated, they cause the beliefs of the receiving parameters to be updated.

Since there are two types of nodes, there will be two types of messages. Denoting the message from node n_i to n_j as $m_{ij}(n_j)$, a message from a factor node n_f to a variable node n_v is defined as:

$$m_{fv}(n_v) = \underset{i \in N(n_f)/n_v}{\arg\max}\ \psi_f \prod_{i \in N(n_f)/n_v} m_{if}(n_f) \tag{5}$$

and a message from a variable node n_v to a factor node n_f is then:

$$m_{vf}(n_f) = \alpha_v \phi(k_v) \prod_{j \in N(n_v)/n_f} m_{jv}(n_v) \tag{6}$$

where $N(n)$ denotes the set of nodes neighboring n, α_v is a normalizing constant for n_v and k_v is the parameter that is represented by n_v. $\phi(k_v)$ denotes the current belief of k_v and it is updated after receiving messages from all its neighboring factor nodes.

The scheme described above is also known as the max-product algorithm. Applied to a probabilistic graphical model, it computes the *maximum a posteriori* (MAP) probabilities of the variable nodes. On loop free graphs, this algorithm converges to a unique probability distribution. Further, the assignment based on this distribution yields the most probable values for the nodes, which can

then be reported as the estimated values. Although the Factor Graphs induced by bio-pathways are seldom loop free, it has been shown that applying belief propagation on such *loopy* graphs will still yield a distribution that gives a *neighborhood maximum* and that for some graphs, this neighborhood can be exponentially large (Weiss and Freeman, 2001). We have implemented this loopy belief propagation algorithm (Murphy *et al.*, 1999) on parameters that have been discretized. In this algorithm, the messages will be generated and propagated throughout the Factor Graph until the beliefs of the parameters have converged, or a pre-determined number of iterations have been executed.

4 Simulation Results

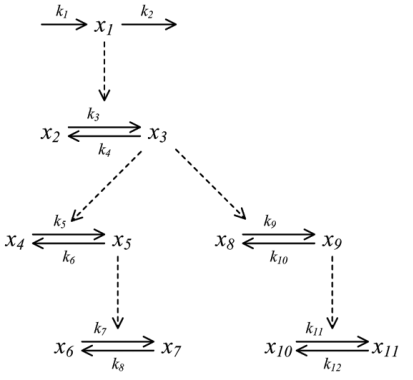
Our parameter estimation algorithm is implemented in C++. It works in the discrete domain by dividing each dimension of the parameter space into finite partitions. Belief propagation will then give us the most likely domains, or the *MAP* partitions. We fine-tune the estimates by applying the Levenberg-Marquardt algorithm (Gill *et al.*, 1982), starting from the mid-points of the MAP partitions.

4.1 Case Study: Branching Signaling Pathway

To test our approach, we have applied it to a synthetic pathway model that exhibits branching, a typical feature found in signaling pathways. We constructed a branched pathway with 11 molecules and with "nominal values" for the kinetic rate constants falling in the interval $[0.0, 1.0]$. Simulation of this model yielded the synthetic data we use in lieu of experimental data, and then the nominal rate constants were set aside. We allowed synthetic time series data at 20 discrete time points to be made available for 5 of the molecules - x_1, x_5, x_7, x_9 and x_{11}. There remain 12 unknown parameters to be estimated. This is a reasonable approximation of the parameter estimation problem in actual settings.

Using our decompositional approach of (Koh *et al.*, 2006), the pathway can be broken down into multiple overlapping components. We will consider components C1 and C2, which consist of the variable sets $\{x_1, x_2, x_3, x_4, x_5, x_6, x_7\}$ and $\{x_1, x_2, x_3, x_8, x_9, x_{10}, x_{11}\}$ respectively (Figure 3A). We first estimate the parameters by sampling the components separately, and then we reconcile their values by belief propagation on the combined Factor Graph (Figure 3B). As an alternative, we also apply our previous method by estimating the parameters for C1 followed by the remaining ones in C2 (Figure 3C - Scheme S1) and vice versa (Figure 3C - Scheme S2). We compare the efficiency and quality of our results with other optimization algorithms - Levenberg-Marquardt (LM), Evolutionary Strategies with Stochastic Ranking (SRES) and Genetic Algorithm (GA).

All simulations are performed on an Intel Pentium M processor with 1 GB memory. Fine-tuning of the estimates, and the running of other optimization techniques are done using the open source software COPASI (Hoops *et al.*, 2006). We score the resulting parameters obtained from all the algorithms using the weighted sum-of-squares difference between the experimental data and the corresponding simulation profile. The results of the simulations are summarized in Table 1.



$$
\begin{aligned}
\dot{x}_1 &= k_1 - k_2 x_1 \\
\dot{x}_2 &= k_4 x_3 - k_3 x_1 x_2 \\
\dot{x}_3 &= k_3 x_1 x_2 - k_4 x_3 \\
\dot{x}_4 &= k_6 x_5 - k_5 x_3 x_4 \\
\dot{x}_5 &= k_5 x_3 x_4 - k_6 x_5 \\
\dot{x}_6 &= k_8 x_7 - k_7 x_5 x_6 \\
\dot{x}_7 &= k_7 x_5 x_6 - k_8 x_7 \\
\dot{x}_8 &= k_{10} x_9 - k_9 x_3 x_8 \\
\dot{x}_9 &= k_9 x_3 x_8 - k_{10} x_9 \\
\dot{x}_{10} &= k_{12} x_{11} - k_{11} x_{10} x_9 \\
\dot{x}_{11} &= k_{11} x_{10} x_9 - k_{12} x_{11}
\end{aligned}
$$

Fig. 2. Schema of the pathway model and the system of ODEs that defines it. The dashed arrows in the schema represent enzyme-catalyzed reactions where the enzymes are not consumed by the reactions.

4.2 Results and Discussion

The measure of quality for this set of results is not the closeness of the estimated parameters to the nominal ones. Rather, it is the *score*, which is the weighted sum-of-squares difference between the simulated concentration profiles (generated using those parameters) and the pseudo-experimental data sets.

We can see from the results that belief propagation out-performs all the other algorithms both in terms of efficiency (requiring 10440 evaluations, where each evaluation is a complete numerical simulation of a single component) as well as quality (with the lowest score of 0.0119). Note that due to the discretized nature of our implementation, we are not able to get the best estimates simply by using belief propagation. Instead, we are able to provide a starting point within the vicinity of a good solution, which we can locate using the LM algorithm. Without this starting point, it is not surprising to see that the LM algorithm, starting from the midpoint of the parameter space performs the worst with an astonishingly high score of 269.061 even though it requires only 177 evaluations to converge. Clearly, this is an indication of the algorithm getting trapped in a local minimum.

It is interesting to note that the decompositional schemes (S1 and S2) provide better results than SRES, GA and LM. This is largely due to the lower dimensionality of the components and thus of their search.

However, the quality varies depending on the order in which the components C1 and C2 are considered. This variation is a result of prematurely fixing the estimates of the parameters in one component when there could be better solutions when taking into account the entire pathway. The issues of ordering and choosing the components were largely left unaddressed in our previous work but belief propagation fills this gap nicely by allowing each of the components to be estimated for separately, and later combining their estimates by its message passing scheme.

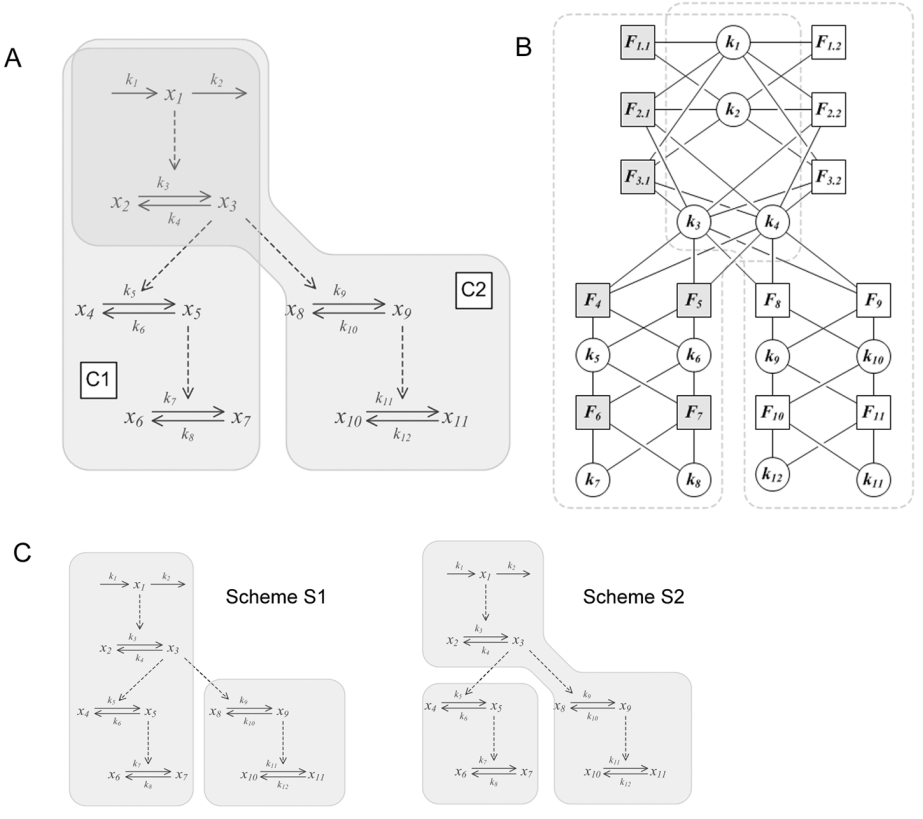


Fig. 3. (A) Overlapping components C1 and C2 (B) Factor graph induced by C1 and C2, with the left Factor Graph corresponding to C1 and the right one, C2. They are composed together via the common variable nodes k_1, k_2, k_3 and k_4 (C) The two estimation schemes S1 and S2.

Besides requiring fewer model evaluations, the additional runtime incurred for message passing itself is quadratic in the size of the pathway. The size of the joint probability distribution table of the factor node F_i, on the other hand, is exponential in the number of variable nodes it is connected to. Hence, it is the degree of the Factor Graph, rather than the pathway size, that will be the limiting factor for the performance of belief propagation. It will be interesting to determine if there is a reasonable upper bound on the connectivity of the factor nodes for bio-pathways.

5 Conclusion

Parameter estimation of large bio-pathway models is an important, but difficult problem. To reduce the prohibitive computational cost, we take a decompositional approach that consists of three main steps conceptually: (i) divide a large

Table 1. Comparison of belief propagation (BP/LM) with the original decompositional approach (S1 and S2) and other optimization techniques (LM, SRES, GA). The zeroth, first and second order rate constants are given in $nM.s^{-1}$, s^{-1} and $nM^{-1}.s^{-1}$ respectively. Search parameters are specific to their individual algorithms for them to run: **Iter** = Maximum number of iterations; **Tol** = Tolerance; **Gen** = Number of generations; **Pop** = Population size. An evaluation is a complete numerical simulation of the ODEs using current parameter estimates. The **score** for the parameters is the weighted sum-of-squares difference between the experimental data and the corresponding simulation profile generated using those parameters.

Parameters	**Nominal Values**	**BP/LM**	**S1**	**S2**	**LM**	**GA**	**SRES**
k_1	0.625	0.6368	0.6915	0.4635	0.3991	0.3035	0.5031
k_2	0.228	0.2317	0.2609	0.1311	0.7912	$1.2e^{-138}$	0.1629
k_3	0.112	0.4007	0.0739	0.0565	0.3236	0.3997	0.1020
k_4	0.96	0.1917	0.3264	0.7327	1	0.6247	0.6063
k_5	0.579	0.1089	0.9317	1	0.5592	0.2361	0.6859
k_6	0.312	0.2269	0.8134	0.3240	0.3779	0.3815	0.4444
k_7	0.628	0.6235	0.6349	0.7405	0.5877	0.6680	0.8878
k_8	0.104	0.0983	0.1657	0.1224	0.3032	0.0481	0.5638
k_9	0.04	0.0075	0.0456	0.1184	0.3310	0.0192	0.0710
k_{10}	0.286	0.1836	0.5523	0.8324	0.7980	0.4360	0.8602
k_{11}	0.624	0.6317	0.6551	0.7041	0.2847	0.4132	0.5773
k_{12}	0.88	0.8896	0.9468	0.8388	0.9402	0.5318	0.7079
Search Parameters	–	**Iter**: 200 **Tol**: $1e^{-5}$	**Gen**: 200 **Pop** : 20	**Gen**: 200 **Pop** : 20	**Iter**: 200 **Tol**: $1e^{-5}$	**Gen**: 400 **Pop**: 40	**Gen**: 400 **Pop**: 40
Score	–	**0.0119**	**0.4138**	**2.3498**	**269.061**	**4.75115**	**2.5103**
Evaluations	–	10440	47820	47820	177	15258	95814

pathway model into components, (ii) compute the estimates for the parameters in each component separately, and (iii) combine the parameter estimates for all the components into a globally consistent one. We have shown that belief propagation is an effective method for the critical last step. It takes into account all the local constraints represented as beliefs and reconciles them in a principled manner by exploiting the pathway structure. An additional advantage of this method is that it handles incomplete or noisy data well. Preliminary results based on simulation show that our new parameter estimation algorithm, which applies belief propagation followed by local descent, substantially outperforms existing alternatives based on local descent alone or evolutionary strategies in both accuracy and efficiency.

We are currently working on several extensions of our algorithm to improve the reliability and efficiency of belief propagation. We also plan to test the algorithm on larger pathway models with many components and feedback loops. More importantly, along with our collaborator, we are applying the algorithm to study the Akt-MAPK pathway (Koh *et al.*, 2006) using real experimental data.

References

[Felzenszwalb and Huttenlocher, 2006]Felzenszwalb, P.F., Huttenlocher, D.P.: Efficient belief propagation for early vision. International J. of Comp. Vision 70(1), 41–54 (2006)

[Friedman, 2004]Friedman, N.: Inferring cellular networks using probabilistic graphical models. Science 303, 799–805 (2004)

[Gill *et al.*, 1982]Gill, P.E., Murray, W., Wright, M.H.: Practical Optimization. Academic Press, London (1982)

[Gat-Viks *et al.*, 2005]Gat-Viks, I., Tanay, A., Raijman, D., Shamir, R.: The factor graph network model for biological systems. In: Proc. of the 9th Int. Conf. on Res. in Comp. Mol. Biol., pp. 31–48 (2005)

[Hoops *et al.*, 2006]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), 3067–3074 (2006)

[Ihler *et al.*, 2004]Ihler, A.T., Fisher, J.W., Moses, R.L., Willsky, A.S.: Nonparametric belief propagation for self-calibration in sensor networks. In: Proc. of the 2004 Int. Conf. on Inf. Proc. in Sensor Networks, pp. 225–233 (2004)

[Kikuchi *et al.*, 2003]Kikuchi, S., Tominaga, D., Arita, M., Takahashi, K., Tomita, M.: Dynamic modeling of genetic networks using genetic algorithm and S-system. Bioinformatics 19(5), 643–650 (2003)

[Koh *et al.*, 2006]Koh, G., Teong, H.F.C., Clement, M.V., Hsu, D., Thiagarajan, P.S.: A decompositional approach to parameter estimation in pathway modeling: a case study of the Akt and MAPK pathways and their crosstalk. Bioinformatics 22(14), e271–e280 (2006)

[Kschischang *et al.*, 2001]Kschischang, F.R., Frey, B.J., Loeliger, H.A.: Factor graphs and the sum-product algorithm. IEEE Trans. on Information Th. 47(2), 498–519 (2001)

[Mendes and Kell, 1998]Mendes, P., Kell, D.B.: Non-linear optimization of biochemical pathways: applications to metabolic engineering and parameter estimation. Bioinformatics 14(10), 869–883 (1998)

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

[Murphy *et al.*, 1999]Murphy, K.P., Weiss, Y., Jordan, M.I.: Loopy belief propagation for approximate inference: An empirical study. In: Proc. of the 15th Ann. Conf. on Uncertainty in AI, pp. 467–475 (1999)

[Pearl 1988]Pearl, J.: Probabilistic reasoning in intelligent systems: networks of plausible inference, 2nd edn. Morgan Kaufmann Publishers, Inc., San Francisco (1988)

[Weiss and Freeman, 2001]Weiss, Y., Freeman, W.T.: On the optimality of solutions of the max-product belief propagation algorithm in arbitrary graphs. IEEE Trans. on Information Th. 47(2), 736–744 (2001)

[Yeang and Jaakkola, 2003]Yeang, C.H., Jaakkola, T.: Physical network models and multi-source data integration. In: Proc. of the 7th Int. Conf. on Res. in Comp. Mol. Biol. pp. 312–321 (2003)

[Yedidia *et al.*, 2003]Yedidia, J.S., Freeman, W.T., Weiss, Y.: Understanding belief propagation and its generalizations, Exploring Artificial Intelligence in the new Millenium, pp. 239–269 (2003)