

# Belief Propagation Estimation of Protein and Domain Interactions Using the Sum–Product Algorithm

Faruck Morcos, *Member, IEEE*, Marcin Sikora, *Member, IEEE*, Mark S. Alber, Dale Kaiser, and Jesús A. Izaguirre

**Abstract**—In this paper, a novel framework is presented to estimate protein–protein interactions (PPIs) and domain–domain interactions (DDIs) based on a belief propagation estimation method that efficiently computes interaction probabilities. Experimental interactions, domain architecture, and gene ontology (GO) annotations are used to create a factor graph representation of the joint probability distribution of pairwise protein and domain interactions. Bound structures are used as *a priori* evidence of domain interactions. These structures come from experiments documented in iPfam. The probability distribution contained in the factor graph is then efficiently marginalized with a message passing algorithm called the sum–product algorithm (SPA). This method is compared against two other approaches: maximum-likelihood estimation (MLE) and maximum specificity set cover (MSSC). SPA performs better for simulated scenarios and for inferring high-quality PPI data of *Saccharomyces cerevisiae*. This framework can be used to predict potential protein and domain interactions at a genome wide scale and for any organism with identified protein–domain architectures.

**Index Terms**—Belief propagation, Bayesian networks, domain–domain interactions (DDIs), protein networks, protein–protein interactions (PPIs), PPI inference, *Saccharomyces cerevisiae*, sum–product algorithm (SPA).

## I. INTRODUCTION

**P**ROTEINS perform most of the vital functions in the living cell. The range of processes in which they are involved is vast, including signal transduction, DNA replication, immune system, cell development, and differentiation. During the past century, an important research effort was devoted to elucidation

of structure, function and characterization of individual genes and their protein products. However, one way to understand the complexity of the living cell and disease related pathways is by studying the network of interactions of its basic components [1]. Recent advances in experimental techniques and computational methods are helping to unravel as well as understand the web of protein–protein interactions (PPIs) in living organisms. Large quantities of data often coming from high throughput experiments are being used to produce large networks of interactions and study their structure [2]–[4]. This higher level view of networks provides global properties of the biological system and helps understand patterns of behavior that might be used to predict gaps in the incomplete interaction map of an organism. Even though the advance of experimental techniques has been remarkable, the complexity of experimentally identifying all protein interactions for complete organisms remains to be a challenge. Furthermore, high-throughput experimental techniques are prone to errors that increase the difficulty of this task. Hence, computational techniques to cope with such levels of complexity and uncertainty are needed in the pursuit of complete and correct maps of protein interactions in living organisms. Techniques from information theory might be of importance when solving these types of problems found in molecular biology.

In this work, we estimate the plausibility of PPI and domain–domain interactions (DDIs) through a framework that combines protein interaction data (database of interacting protein (DIP) [5]), structural information from the protein data bank (PDB) [6] or iPfam [7], domain architecture of proteins (Pfam [8]), and functional information about domains and proteins (gene ontology (GO) [9]). Statistical estimation of protein interactions based on the domain conformation of proteins and interaction data has been shown to be promising. Recent algorithms are based on the fact that proteins are composed by independent folding polypeptides subunits called domains. Organisms, in general, have multiple-domain and single-domain proteins in their proteomes. Eukaryotes tend to have a larger proportion of multiple-domain proteins than prokaryotes [10]. Some of these independent units are found, with remarkable aminoacid sequence similarity, within and across species. Furthermore, these domain families have regions, where physical interactions take place, that tend to be more conserved than protein sequences outside these domains [11]. Domain interaction networks have a rich correlation between functional similarity and network proximity. This means that domains that are separated by fewer links or hops in a DDI network tend to perform more similar functions in the cell [12], a useful fact for estimating PPI. Physical protein interactions are mediated

Manuscript received May 05, 2009. Current version published February 24, 2010. The work of F. Morcos and J. A. Izaguirre was supported in part by the National Science Foundation (NSF) under Grants DBI-0450067 and CCF-0622940. The work of M. S. Alber was supported in part by the National Science Foundation (NSF) under Grant DMS 0719895. The material in this paper was presented in part at the Information Theory and Application Workshop, San Diego, CA, January 2007.

F. Morcos and J. A. Izaguirre are with the Department of Computer Science and Engineering, University of Notre Dame, Notre Dame, IN 46545 USA (e-mail: amorcosg@nd.edu; izaguirr@nd.edu).

M. Sikora was with the Department of Electrical Engineering, University of Notre Dame, Notre Dame, IN 46545 USA. He is now with Life Technologies, Foster City, CA 94404 USA (e-mail: msikora@ieee.org).

M. S. Alber is with the Department of Mathematics, University of Notre Dame, Notre Dame, IN 46545 USA (e-mail: malber@nd.edu).

D. Kaiser is with the Departments of Biochemistry and Developmental Biology, Kaiser Laboratory, School of Medicine, Stanford University, Stanford, CA 94305-5307 USA (e-mail: adkaiser@stanford.edu).

Communicated by O. Milenkovic, Associate Guest Editor for Special Issue on Molecular Biology and Neuroscience.

Color versions of Figures 1, 2, and 7–12 in this paper are available online at <http://ieeexplore.ieee.org>.

Digital Object Identifier 10.1109/TIT.2009.2037051

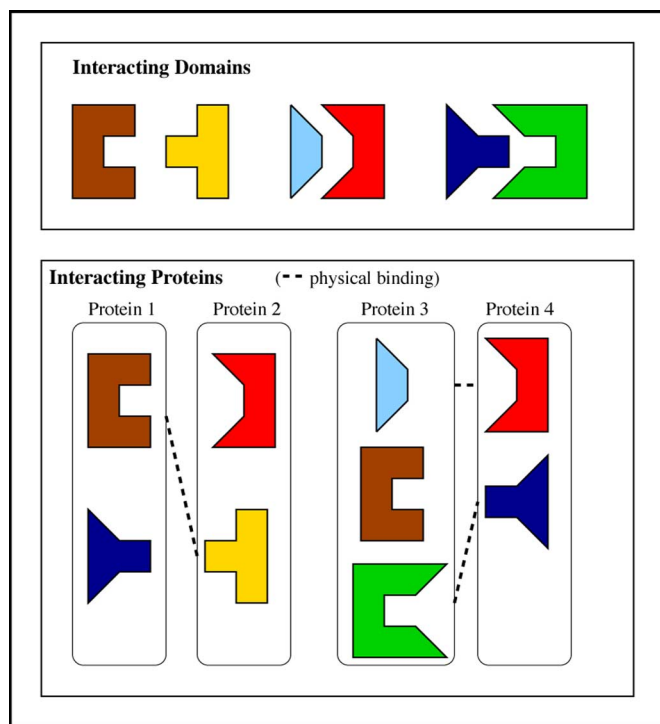


Fig. 1. Multiple-domain architecture of proteins. Using the multiple-domain architecture of proteins and the knowledge of interacting domains, it is possible to estimate a likelihood of physical PPIs.

by interactions between domains and protein motifs (short aminoacid sequences that do not fold independently) [13]. In this context, if two proteins interact, we assume there is an interaction mode involving at least two domains. We do not consider domain–motif interactions, which are also feasible but are not considered in the scope of this investigation. Many proteins share their conserved domain composition as well as their domain physical interactions. This is depicted in Fig. 1. As more experimental evidence is gathered indicating the interaction of certain domain pair in several protein pairs, the higher the likelihood this domain pair interacts in a different protein pair containing such domains. These observations led to the development of statistical predictors using different methods, such as maximum-likelihood estimation (MLE) using expectation maximization (EM) [14], support vector machines [15], and set cover approaches [16]. These algorithms learn from interaction data, such as yeast two-hybrid (Y2H) assays, and assign high scores to domain pairs that occur significantly in experimentally observed interactions. Others treat this problem using model learning by incorporating domain fusions, coexpression, and GO information [17]. More general genomic data integration using Bayesian approaches has helped the discovery of interaction networks. Myers *et al.* [18], [19] integrated Kyoto Encyclopedia of Genes and Genomes (KEGG) pathways, sets of biological process GO terms and Munich Information Center for Protein Sequences (MIPS) protein complexes to reconstruct interaction networks in *Saccharomyces cerevisiae*.

We propose a new protein and domain interaction estimation framework (see Fig. 2), which combines data from biological databases including experimental PPI and DDI, domain architecture (i.e., which domains are contained in each protein), GO

terms, and structural domain information into a factor graph representation of the joint probability distribution of pairwise protein and domain interactions. A marginalization of this graph takes place to obtain estimates of protein and domain interactions. The output set of this training stage, a collection of domain and protein pair estimates derived from the experimental input set, is then used to infer the probabilities for new protein pairs which have domains for which we computed an interaction probability. Validation can be further performed with experimental techniques for physical protein interactions. This algorithm is positioned in the class of estimation algorithms that use domain families and experimental evidence of protein and domain physical interactions. Examples of algorithms that aim to solve the same estimation problem are MLE as in [14] and the maximum specificity set cover (MSSC) described in [16]. An important additional feature of our method is that it recomputes PPI/DDI estimates of the input experimental data set simultaneously providing a method to correct errors introduced in the experimental assays. This error-correcting feature might have applications not only for PPI/DDI inference but also for denoising experimental data. In contrast, algorithms like the one developed by Burger *et al.* [20], use aminoacid sequence information in a Bayesian framework to estimate probabilities of interaction without using any experimental information. Another class of protein–protein docking algorithms deal with 3-D coordinates of proteins and aim to find interaction hotspots or geometric configurations that approximate *in vivo* interactions. Examples of these type of algorithms are discussed in [21] and [22]. Our framework belongs to an intermediate category that aims to optimize both specificity and sensitivity in PPI/DDI prediction.

Our approach to predict domain and protein interactions *in silico* has several advantages. First, it allows us to do predictions at a proteome scale, being able to consider all the proteins available with domain annotations. This feature is shared with other algorithms that use domain annotations of proteins like those described in [14], [17], and [16]. Second, as part of a belief propagation framework, evidence of experiment, domain, and structural data as well as GO data of protein pairs can be combined to increase the quality of our predictions. As databases gather more information our algorithm performs better. Finally, this inference framework estimates false positives and false negatives in the experimentally measured PPI, and provides assessment of the quality of the data.

## II. SUM–PRODUCT ALGORITHM

For every protein–protein and domain–domain pair, an interaction is regarded as a Boolean feature (present or absent). We establish the likelihood of presence of such interactions by Bayesian inference, i.e., by computing their probabilities conditioned on the experimental data and the domain composition of the proteins. These posterior probabilities are obtained from the marginals of the joint probability distribution function of all the PPI, DDI, and experiment outcomes derived from the underlying statistical model. We achieve this using the sum–product algorithm (SPA) [23], an efficient technique for computing marginal values of multivariate functions that factor into products of simpler functions in fewer variables. The SPA uses a graph representation of this factorization, called the *factor graph*, and

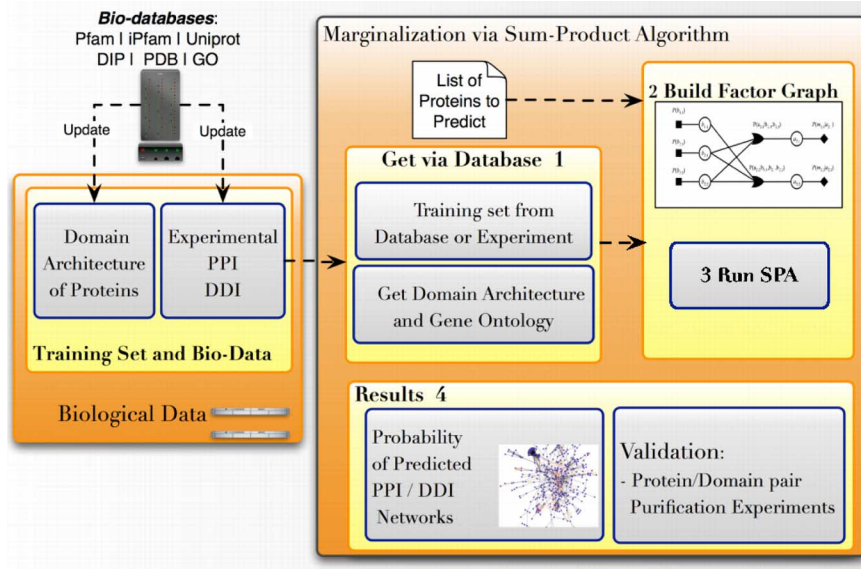


Fig. 2. Architecture of our protein–domain interaction estimation methodology. Information extracted from online databases and experimental data (step 1) is represented by a factor graph (step 2) and processed by the SPA algorithm (step 3). Predicted PPI/DDI networks are further validated with experimental evidence (step 4).

performs its computations by iteratively exchanging messages along the edges of the graph. The SPA has been successfully applied to estimation problems in communication systems and signal processing [24]–[27], combining good accuracy and low computational complexity.

#### A. General Form of the SPA

In its most general statement, the SPA solves the following problem. Given a function  $g(x_1, x_2, \dots) = g(\mathbf{x})$  that factors into

$$g(\mathbf{x}) = \prod_j f_j(\mathbf{x}_j)$$

where  $\mathbf{x}_j$  is a subset of variables in  $\mathbf{x}$ , SPA finds the marginal functions

$$g_i(x_i) = \sum_{\sim\{x_i\}} g(\mathbf{x})$$

where  $\sim\{x_i\}$  denotes marginalization of the function over all its arguments, except  $x_i$ . The computations performed by SPA involve calculation of messages along the edges of the factor graph, in which a variable node  $x_i$  is connected by an edge to a function node  $f_j$  if  $x_i$  is an argument of  $f_j$ . The messages, denoted as  $\mu_{f_j \rightarrow x_i}(x_i)$  and  $\mu_{x_i \rightarrow f_j}(x_i)$ , are real-valued functions in the variable from or to which the message is passed. For the variable node  $x_i$ , the outgoing messages are computed as

$$\mu_{x_i \rightarrow f_j}(x_i) = \prod_{k \neq j} \mu_{f_k \rightarrow x_i}(x_i) \quad (1)$$

and for the function node  $f_j$  as

$$\mu_{f_j \rightarrow x_i}(x_i) = \sum_{\sim\{x_i\}} \left( f_j(\mathbf{x}_j) \prod_{k \neq i} \mu_{x_k \rightarrow f_j}(x_k) \right). \quad (2)$$

The products computed in (1) and (2) only include messages arriving at this node from all remaining edges (except  $j$  and  $i$ ,

respectively) and marginalization  $\sim\{x_i\}$  in (2) only includes variables adjacent to  $f_j$  (except  $x_i$ ).

The messages are iteratively recomputed until convergence (or until the number of iterations exceeds a preset number), at which point the marginal functions are obtained as

$$g_i(x_i) = \prod_j \mu_{f_j \rightarrow x_i}(x_i). \quad (3)$$

The entire procedure yields an exact answer if the factor graph contains no loops, otherwise it is only an approximation.

#### B. Log-Ratio Form of the SPA

For marginalization problems involving probability functions of binary variables, SPA can be usually simplified. When all variables are binary, the messages (1) and (2) are pairs of numbers, e.g.,  $(\mu_{x_i \rightarrow f_j}(0), \mu_{x_i \rightarrow f_j}(1))$ . Additionally, probability functions often need only be known up to a multiplicative constant, in which case only a ratio  $\mu_{x_i \rightarrow f_j}(0)/\mu_{x_i \rightarrow f_j}(1)$  suffices to be passed. Finally, numerical stability of this iterative algorithm can benefit from operating in logarithmic domain.

In the log-ratio formulation, variable node formula (1) is replaced by

$$\log \frac{\mu_{x_i \rightarrow f_j}(0)}{\mu_{x_i \rightarrow f_j}(1)} = \sum_{k \neq j} \log \frac{\mu_{f_k \rightarrow x_i}(0)}{\mu_{f_k \rightarrow x_i}(1)}. \quad (4)$$

For (2), a general form cannot be given and the specific properties of  $f_j(\mathbf{x}_j)$  ultimately determine whether log-ratio version can be obtained for a given problem. Finally, if  $g_i(x_i)$  is indeed a probability distribution, then (3) can be written as

$$g_i(x_i = 1) = \left( 1 + \exp \sum_j \log \frac{\mu_{f_j \rightarrow x_i}(0)}{\mu_{f_j \rightarrow x_i}(1)} \right)^{-1} \quad (5)$$

leveraging the fact that  $g_i(x_i = 0) + g_i(x_i = 1) = 1$ . In our PPI and DDI inference problem, we adopt this latter formulation.

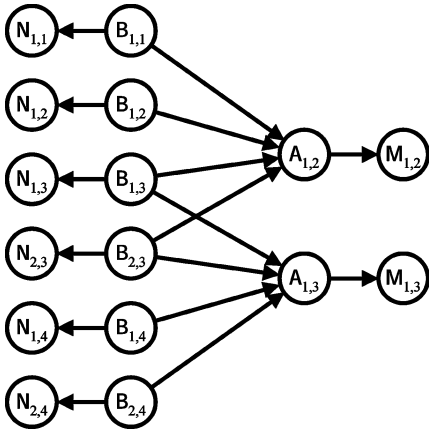


Fig. 3. Bayesian network representing a factorization of the joint probability distribution  $P(\mathbf{A}, \mathbf{B}, \mathbf{M}, \mathbf{N})$ , where  $\mathbf{A}$  and  $\mathbf{B}$  are the collection of hypotheses that proteins and domains interact, respectively. Variables  $\mathbf{M}$  and  $\mathbf{N}$  describe collections of results of interaction measurements or experimental evidence performed on protein pairs and domain pairs, respectively.

In the very first iteration, all messages  $\log \mu_{x_i \rightarrow f_j}(0) / \mu_{x_i \rightarrow f_j}(1)$  are initialized to zero.

### III. PPI AND DDI INFERENCE

#### A. Statistical Model

The statistical model adopted in this paper to describe the relations between the PPI, the DDI, and the experiment outcomes is a highly modular Bayesian network depicted in Fig. 3. We denote by  $A_{i,j} = 1$  a hypothesis that proteins  $i$  and  $j$  interact, by  $B_{x,y} = 1$  a hypothesis that domains  $x$  and  $y$  interact, and use  $M_{i,j}$  and  $N_{x,y}$  to describe the results of interaction measurements or experimental evidence performed on the protein pair  $(i, j)$  and domain pair  $(x, y)$ , respectively. The Bayesian network implies the following factorization of the joint probability function:

$$P(\mathbf{A}, \mathbf{B}, \mathbf{M}, \mathbf{N}) = \prod_{(x,y)} P(N_{x,y}|B_{x,y})P(B_{x,y}) \times \prod_{(i,j)} P(M_{i,j}|A_{i,j})P(A_{i,j}|\{B_{x,y}\}_{(x,y) \in \mathcal{B}_{i,j}}) \quad (6)$$

where  $\mathcal{B}_{i,j}$  is the set of domain pairs, such that one domain is present in protein  $i$  and the other in  $j$ . The terms  $P(M_{i,j}|A_{i,j})$  and  $P(N_{x,y}|B_{x,y})$  model the statistical properties of experiments that produced measurements indicative of PPI and DDI, and serve as the main information input points of the model. The term  $P(B_{x,y})$  represents the *a priori* probability of a DDI and is set for all  $x$  and  $y$  to an estimated probability that two randomly selected domains would interact. The central part of our model  $P(A_{i,j}|\{B_{x,y}\}_{(x,y) \in \mathcal{B}_{i,j}})$  is described through deterministic relations of the form

$$A_{i,j} = \bigvee_{(x,y) \in \mathcal{B}_{i,j}} B_{x,y} \quad (7)$$

stating that a protein pair interacts if and only if at least one of its domain pairs interacts.

#### B. Inference Using SPA

The goal of PPI and DDI inference is the calculation of  $P(A_{i,j} = 1|\mathbf{M}, \mathbf{N})$  and  $P(B_{x,y} = 1|\mathbf{M}, \mathbf{N})$ , the posterior

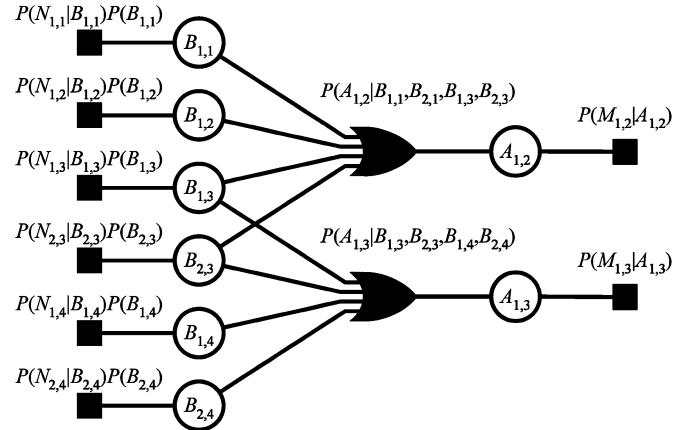


Fig. 4. Factor graph representation of the joint probability distribution  $P(\mathbf{A}, \mathbf{B}, \mathbf{M}, \mathbf{N})$ . Circle nodes represent variables of unknown value, while solid nodes represent factors of the joint probability function.

probabilities of interaction given the available measurements and experimental evidence. Using the Bayes formula, these probabilities can be obtained from marginal functions of the joint probability distribution  $P(\mathbf{A}, \mathbf{B}, \mathbf{M}, \mathbf{N})$

$$P(A_{i,j} = 1|\mathbf{M}, \mathbf{N}) = \frac{\left[ \sum_{\sim\{A_{i,j}, \mathbf{M}, \mathbf{N}\}} P(\mathbf{A}, \mathbf{B}, \mathbf{M}, \mathbf{N}) \right]_{A_{i,j}=1}}{\sum_{\sim\{\mathbf{M}, \mathbf{N}\}} P(\mathbf{A}, \mathbf{B}, \mathbf{M}, \mathbf{N})} \quad (8)$$

$$P(B_{x,y}=1|\mathbf{M}, \mathbf{N}) = \frac{\left[ \sum_{\sim\{B_{x,y}, \mathbf{M}, \mathbf{N}\}} P(\mathbf{A}, \mathbf{B}, \mathbf{M}, \mathbf{N}) \right]_{B_{x,y}=1}}{\sum_{\sim\{\mathbf{M}, \mathbf{N}\}} P(\mathbf{A}, \mathbf{B}, \mathbf{M}, \mathbf{N})}. \quad (9)$$

We can now employ the SPA to calculate the necessary marginals of  $P(\mathbf{A}, \mathbf{B}, \mathbf{M}, \mathbf{N})$ , based on the factorization (6). The factor graph illustrating the complete factorization is presented in Fig. 4. Note that  $M_{i,j}$  and  $N_{x,y}$  are not included in the graph, since these variables have known, fixed values at the time of the inference. The SPA iteratively recomputes messages along each edge in the graph according to equations specific to the type of the node from which the message originates.

There are three unique types of nodes in the factor graph of our model, the variable nodes, the measurement function nodes, and the constraint function nodes, all of which are shown in Fig. 5. We denote  $\alpha_n^v$  and  $\beta_n^v$  to be the log-ratio messages, respectively, entering and leaving node  $v$  along its  $n$ th edge. The formulas for  $\beta_n^v$  in terms of  $\alpha^v$ 's for each of the three node types are given in Table I. The formula for variable nodes is identical to (4). The formulas for the measurement function nodes, due to the fact that they have only a single edge, is trivially derived from (2). Only the expression for the constraint function nodes requires more thorough derivation in the subsequent section. Table I also contains the formulas for the *a posteriori* probabilities of PPI  $P(A_{i,j} = 1|\mathbf{M}, \mathbf{N})$  and DDI  $P(B_{x,y} = 1|\mathbf{M}, \mathbf{N})$ , which are the final result of the Bayesian inference task, analogous to (5).

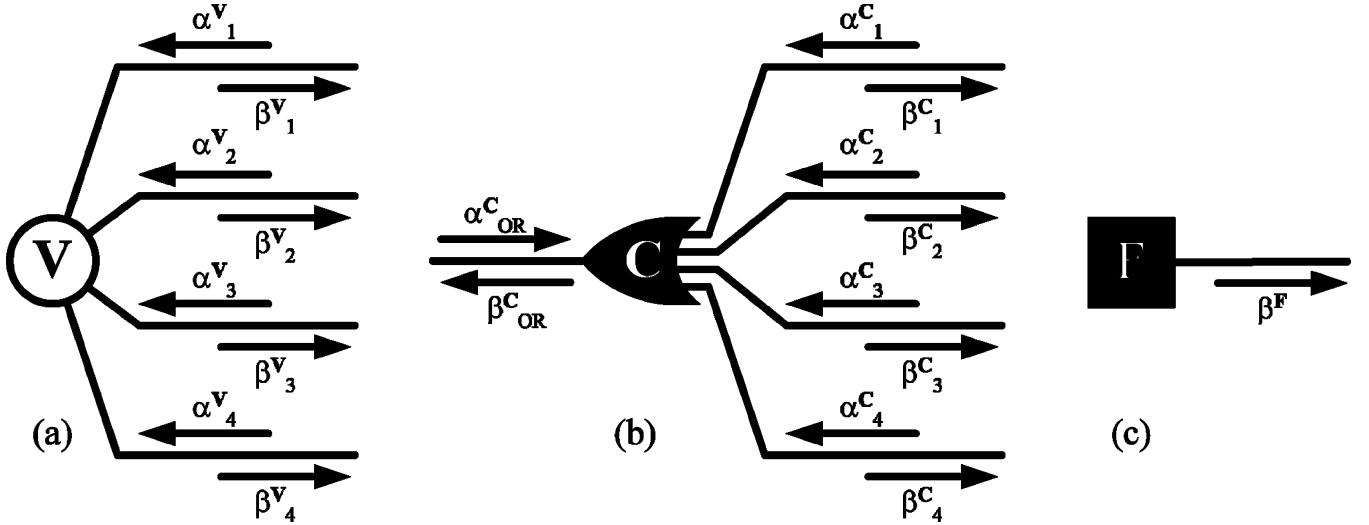


Fig. 5. Messages computed by the SPA at (a) the variable nodes, (b) constraint function nodes, and (c) measurement function nodes. Outgoing messages  $\beta$  are computed from all incoming messages  $\alpha$ , except the one coming from the same edge.

TABLE I  
SPA MESSAGE EQUATIONS FOR THE PPI/DDI INFERENCE PROBLEM

Operation	Formula
Variable node	$\beta_n^v = \sum_{j \neq n} \alpha_j^v$
Protein pair measurement node	$\beta^{A_{i,j}} = \log \frac{P(M_{i,j} A_{i,j}=0)}{P(M_{i,j} A_{i,j}=1)}$
Domain pair measurement node	$\beta^{B_{x,y}} = \log \frac{P(N_{x,y} B_{x,y}=0)P(B_{x,y}=0)}{P(N_{x,y} B_{x,y}=1)P(B_{x,y}=1)}$
Constraint node	See (11) and (12).
PPI probability	$P(A_{i,j}=1 M, N) = \left(1 + \prod_{k=1}^{K_{A_{i,j}}} e^{\alpha_k^{A_{i,j}}}\right)^{-1}$
DDI probability	$P(B_{x,y}=1 M, N) = \left(1 + \prod_{k=1}^{K_{B_{x,y}}} e^{\alpha_k^{B_{x,y}}}\right)^{-1}$

### C. Constraint Function Nodes

The constraint function nodes, illustrated in Fig. 5(b), implement the logical disjunctions of the form  $v_{OR} = \bigvee v_n$ . The corresponding function to be inserted into the formula (2) is

$$f^c(v_{OR}, v_1, \dots, v_{K^c}) = \begin{cases} 1, & \text{if } v_{OR} = \bigvee v_n \\ 0, & \text{otherwise.} \end{cases} \quad (10)$$

Whenever the above function equals zero, it removes the corresponding summands from (2). After some rearrangements, (2) for the edges leading to  $v_n$  (i.e., all edges except the OR edge) takes the form

$$\begin{aligned} \mu_{c \rightarrow v_n}(0) &= \mu_{v_{OR} \rightarrow c}(1) \prod_{k \neq n} (\mu_{v_k \rightarrow c}(0) + \mu_{v_k \rightarrow c}(1)) \\ &\quad + (\mu_{v_{OR} \rightarrow c}(0) - \mu_{v_{OR} \rightarrow c}(1)) \prod_{k \neq n} \mu_{v_k \rightarrow c}(0) \\ \mu_{c \rightarrow v_n}(1) &= \mu_{v_{OR} \rightarrow c}(1) \prod_{k \neq n} (\mu_{v_k \rightarrow c}(0) + \mu_{v_k \rightarrow c}(1)). \end{aligned}$$

For the OR edges, the message equations take the form

$$\begin{aligned} \mu_{c \rightarrow v_{OR}}(0) &= \prod_k \mu_{v_k \rightarrow c}(0) \\ \mu_{c \rightarrow v_{OR}}(1) &= \prod_k (\mu_{v_k \rightarrow c}(0) + \mu_{v_k \rightarrow c}(1)) - \prod_k \mu_{v_k \rightarrow c}(0). \end{aligned}$$

These equations can be further converted to the log-ratio notation

$$\begin{aligned} \beta_n^c &= \log \left( 1 + (e^{\alpha_{OR}^c} - 1) \prod_{k \neq n} (1 + e^{-\alpha_k^c})^{-1} \right) \\ \beta_{OR}^c &= -\log \left( \prod_k (1 + e^{-\alpha_k^c}) - 1 \right). \end{aligned}$$

The above formulas can be used directly, but the numerical stability will suffer from repeated exponentiations and logarithms. This problem can be relieved by employing the well-behaved  $\max^*$  operation, defined for two arguments as

$$\begin{aligned} \max^*(a, b) &= \log(e^a + e^b) \\ &= \max(a, b) + \log(1 + e^{-|a-b|}) \end{aligned}$$

and obtained by successive applications for multiple arguments. Together with the substitution

$$\prod_k (1 + e^{-\alpha_k^c}) - 1 = \sum_k (e^{-\alpha_k^c} \prod_{k' > k} (1 + e^{-\alpha_{k'}^c}))$$

we can rewrite the message expressions as

$$\beta_n^c = \max^* \left[ \max_{\substack{k=1 \\ k \neq n}}^{K^c} \left( -\alpha_k^c + \sum_{\substack{k'=k+1 \\ k' \neq n}}^{K^c} \max^*(0, -\alpha_{k'}^c) \right), \alpha_{OR}^c \right] + \sum_{\substack{k=1 \\ k \neq n}}^{K^c} \max^*(0, -\alpha_k^c) \quad (11)$$

$$\beta_{OR}^c = -\max_{k=1}^{K^c} \left( -\alpha_k^c + \sum_{k'=k+1}^{K^c} \max^*(0, -\alpha_{k'}^c) \right). \quad (12)$$

### D. Common Subexpression Elimination

The accuracy of SPA is influenced by the presence of cycles in the factor graph. If there were no cycles at all, the algorithm

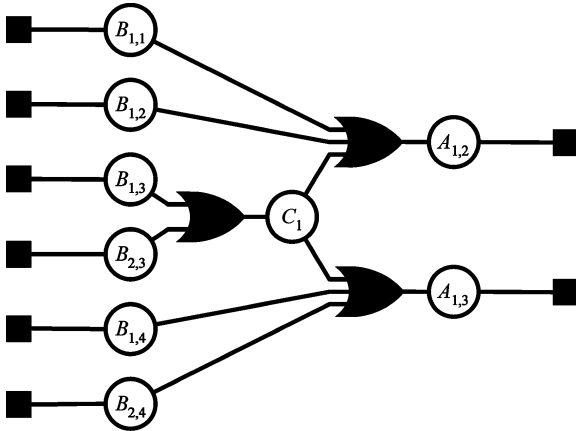


Fig. 6. Factor graph of the joint distribution function after subexpression elimination. At the cost of introducing an additional variable node and constraint node, the length four cycle has been eliminated, improving SPA accuracy.

would solve the inference problem exactly. The more and the shorter the cycles, the more the result of SPA deviates from the true posterior probabilities. Particularly harmful are the cycles of length four, found when two protein pairs share the same two domain pairs. Fortunately, all such short cycles can be removed by a subexpression elimination procedure, analogous to that proposed in [28]. This procedure manipulates the set of (7), replacing reoccurring subsets of variables by additional, miscellaneous terms. For example, the equations defining the graph of Fig. 4

$$\begin{cases} A_{1,2} = B_{1,1} \vee B_{1,2} \vee B_{1,3} \vee B_{2,3} \\ A_{1,3} = B_{1,4} \vee B_{2,4} \vee B_{1,3} \vee B_{2,3} \end{cases}$$

can be rewritten as

$$\begin{cases} A_{1,2} = B_{1,1} \vee B_{1,2} \vee C_1 \\ A_{1,3} = B_{1,4} \vee B_{2,4} \vee C_1 \\ C_1 = B_{1,3} \vee B_{2,3} \end{cases}$$

yielding the cycle-free graph presented in Fig. 6. Despite having additional variable and constraint nodes, such graphs can be processed by the same equations from Table I as the original graph, while improving accuracy of the inference.

We implemented this common subexpression elimination procedure in all the graphs and inference tasks discussed in this work. This produced improved performance in PPI/DDI predictions at an expense of an initial elimination phase that did not have any significant impact on the complexity of the complete inference task.

#### IV. PARAMETERS AND DATA SETS FOR THE SPA

##### A. GO Distance

The GO project [9] is a standardization initiative to provide a controlled and structured vocabulary to annotate gene products and their functions. In general, a gene product can be mapped to vocabulary terms associated with three broad categories: biological processes, cellular components, and molecular functions.

These terms are shared across proteins and are also independent from species. One of the metrics we use as an indirect measure of the possibility of a protein interaction is called GO distance. The GO distance evaluates how similar a given protein pair is in terms of shared vocabulary elements found in the GO annotation. This measure assigns a quantitative value to this similarity and it is expressed as

$$GO_{\text{dist}} = \frac{|GO_{p1} \cup GO_{p2}| - |GO_{p1} \cap GO_{p2}|}{|GO_{p1} \cup GO_{p2}| + |GO_{p1} \cap GO_{p2}|}, \quad (13)$$

where  $GO_{pi}$  represents the set of GO vocabulary terms annotated for proteins  $i = 1, 2$  and  $|\cdot|$  represents the cardinality of the union or intersection set. If two proteins with GO annotation do not share any GO term then their distance is 1; on the other hand, if a pair of proteins share many of their terms, then the distance become smaller indicating that the proteins are related in at least one of the GO categories. SPA uses experimentally determined positive interaction pairs to create a factor graph representation of the relationship between protein pairs and their domains, however, these positive interaction pairs only account for a fraction of all the possible protein pair combinations. It is also important to include in the factor graph such protein pairs for which we know the interaction has low probability. Since the set of known noninteracting protein pairs is scarce, we use the GO distance metric as a guideline to select negative protein pairs of interactions by picking those pairs for which the GO distance is maximum. This provides the algorithm with a set of potential low probability interactions used for interaction estimation rather than just picking protein pairs at random.

##### B. Data Sets

Several data sources were used in this study to extract relevant information for the PPI inference. Protein information was obtained from Uniprot online database [29]. Domain composition of each protein was extracted from Pfam version 22 [8]. PPI data were obtained from two main sources: DIP [5], where most of the interaction pairs were obtained using Y2H assays and IntAct [30], which contains a higher number of interaction pairs with different levels of reliability. We use a high-quality binary data set of protein interactions in *Saccharomyces cerevisiae*, compiled by Yu *et al.* [31] for performance evaluation and validation of our predictions. The algorithm also uses domain interaction data as input. These data were retrieved from the iPfam database [7]. In this database, each domain pair has a 3-D structure in PDB format, assigned to it, providing strong evidence of physical interaction. This study used 6081 DDIs from iPfam, having a tag to a PDB in which the domains are found in complex. These domain pairs were assigned a high *a priori* probability ( $P(N_{x,y}^{\text{ipfam}} = 1 | B_{x,y} = 1) > 0.9$ ) since most of these structures come from crystallography experiments, an indication of higher interaction confidence. A value larger than 0.9 was chosen, however, it does not have any statistical meaning other than an indication of a high score. Three-dimensional structures of domains were extracted from the protein data bank [6]. The data sets used are summarized in Table II.

TABLE II  
DATA SETS USED BY OUR INFERENCE IN SILICO METHODOLOGY

Data Source	Description
Uniprot	Protein general information and sequence
DIP Database and literature (May 2007)	33,234 protein interactions
IntAct	150,876 protein interactions
Yeast Golden Set (Vidal)	2,581 protein interactions
Pfam version 22	Domain composition of proteins
iPfam	6,081 Domain-domain interactions with complex
PDB	Protein and Domain three dimensional structure

### C. Input Parameters

Table III summarizes the parameters required by the SPA. The choice of parameters is based primarily on the criterion of performance optimization. Therefore, values of false positive and false negative rates, GO distance threshold, and initial probabilities yielded an improvement in overall performance curves of cross validation.

## V. PERFORMANCE OF THE SPA

To assess the performance of this model for PPI and DDI inference, we compare it against two other inference algorithms which perform estimation using the same input data and are also based on the idea of domain modularity. These algorithms are MLE approach with EM for PPI/DDI estimation [14] and the MSSC approach for PPI/DDI inference [16]. Performance evaluation is done at two levels. First, we use artificial interactions generated using real proteins with known domain architectures, and second, we infer protein pairs on a high-confidence set of real protein interactions in *Saccharomyces cerevisiae*.

### A. Performance on Artificial Data

Using artificial interaction data sets provides the advantage of a controlled environment to verify the performance of our model as well as other aspects that are harder to validate on real data like error correction in experiments and prediction of domain interactions. The basic setup for our simulated data sets involves three parts. First, a set of real proteins is obtained from Uniprot and their domain architecture is extracted using the Pfam database. Let this protein set be called  $\mathcal{P}_{\text{sim}}$ . Once the set of all domains  $\mathcal{D}_{\text{sim}}$  is known, then a user-defined DDI probability parameter is used to pick which domain pairs we designate as interacting. These domain pairs will induce protein interactions in the initial set of proteins. Hence, we know in advance which proteins and domain interact. The second part is the simulation of an experiment to test if a protein pair interacts. Wet lab experiments to detect protein interaction pairs have inherent errors expressed in terms of false positive ( $f_p$ ) and false negative ( $f_n$ ) rates. We use these rates as parameters to add noise to our known protein interactions to simulate the errors seen in the laboratory. Finally, we select a set of 981 protein pairs, derived from the individual protein set  $\mathcal{P}_{\text{sim}}$ , that are connected by their domain pairs as shown in Fig. 4. This network of protein pairs and

TABLE III  
SPA PARAMETERS USED WHEN COMPUTING PPI/DDI PROBABILITIES

Parameters	Description
$\beta_{M=1}^{A_{i,j}} = \log \frac{P(M_{i,j}=1 A_{i,j}=0)}{P(M_{i,j}=1 A_{i,j}=1)}$	Initial log-likelihood $\log(f_p) - \log(1 - f_n)$
$\beta_{M=0}^{A_{i,j}} = \log \frac{P(M_{i,j}=0 A_{i,j}=0)}{P(M_{i,j}=0 A_{i,j}=1)}$	Initial log-likelihood $\log(1 - f_p) - \log(f_n)$
$P(B_{x,y}=1)$	Uniform <i>a priori</i> probability
$P(N_{x,y}^{i,p} f_{am} = 1   B_{x,y} = 1)$	iPfam <i>a priori</i> probability
GO distance threshold	Threshold to consider negative interaction
SPA iterations	Message passing iterations
$f_p$	Prediction false positive rate
$f_n$	Prediction false negative rate

domain pairs is processed by the SPA, MSSC, and MLE algorithms and receiving operating characteristic (ROC) curves are obtained to compare performance.

Fig. 7 shows the topology of this network and a closeup of a region in this graph. Square nodes represent domain pairs and diamond-shaped nodes are protein pairs.

1) *Error Correction in Simulated Experiments*: One additional feature of these methods for protein and domain pair inference is the capacity of identifying errors in the experimental interaction data they use for predictions. Instead of predicting potential interactions on new pairs of proteins, we can assign an interaction probability to protein pairs already known to interact. Since there are inherent errors on experiments, our inference framework can calculate probability estimates for the input data and correct initial experimental estimates. The input set for this prediction mode is the subset of protein pairs in  $\mathcal{P}_{\text{sim}}$  that were designated as interacting by the simulated noisy experiments. The factor graph representation of this set includes both protein and domain pairs (see Fig. 7) and in the first run of SPA, estimates for both DDI and the initial experimental PPI are calculated. Hence, initial experimental protein pairs get a new value for their interaction probability. This reassessment leads to error correction in initial experimental outcomes for protein pairs. We evaluated the performance of this feature using our artificial data set and the data of our noisy protein pair experiments for different values of false positive ( $f_p$ ) and false negative ( $f_n$ ) rates. The ROC curves are shown in Fig. 8. These curves were calculated by estimating PPI probabilities and then comparing against the real prior interactions known before adding errors to the graph. A DDI *a priori* probability of 0.05 was used; for this case, there was no structural evidence for domains pairs, and 200 interactions were averaged to get these curves per estimation algorithm.

The ROC curves in Fig. 8 show the effect of error rates on the experimental data and how at relatively low values for  $f_p$  and  $f_n$  rates these algorithms can actually do a good job correcting errors in the experiment. It can also be seen that high  $f_p$  values have a higher impact on error correction making it more difficult to achieve. This is due to the fact that there is a higher number of negative interactions and spotting false negatives is easier



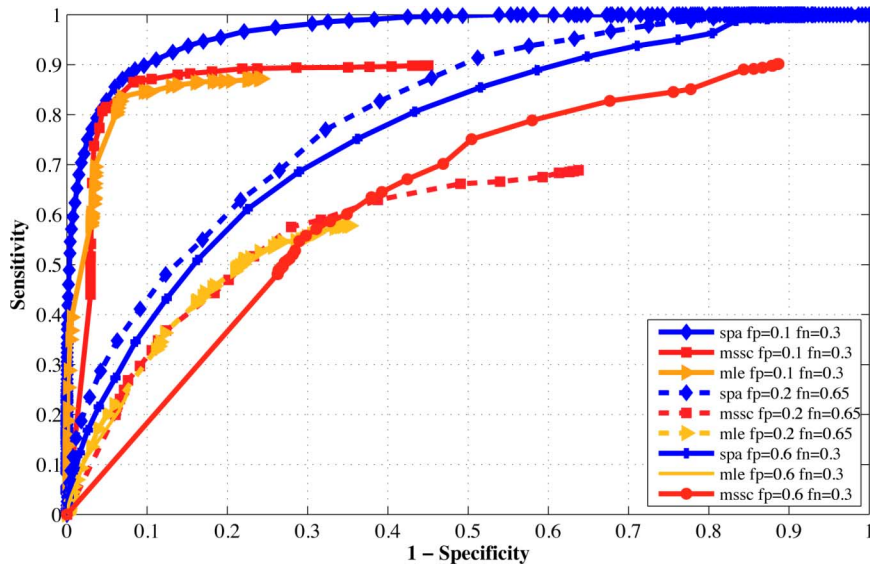


Fig. 8. Error correction in PPI experiments. PPI/DDI inference algorithms can be used to identify errors on noisy experimental data. Error-correction capabilities of SPA, MSSC, and MLE are contrasted in terms of ROC curve.

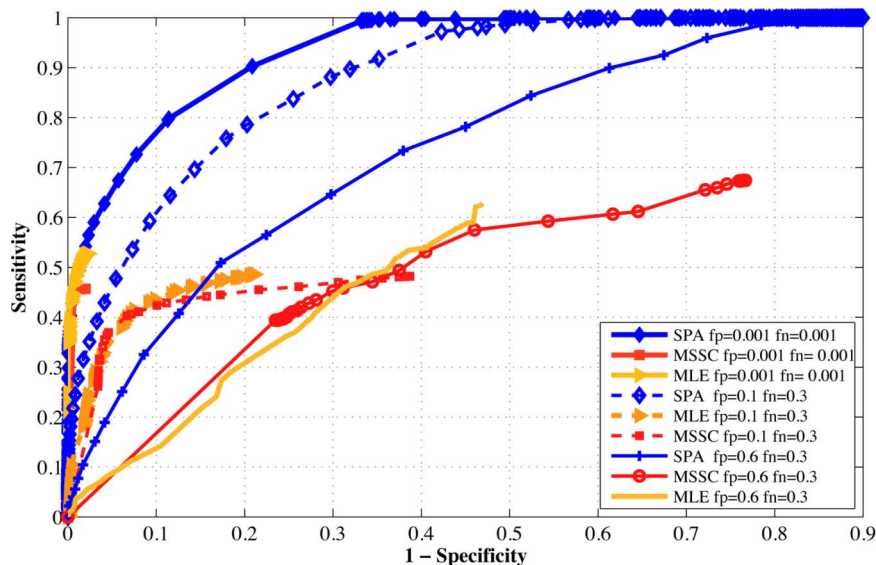


Fig. 9. Simulated PPI prediction. ROC performance of SPA, MSSC, and MLE for PPI inference task. Performance is shown for different reliabilities of noisy experimental data.

pairs are calculated using the DDI and PPI estimates from the training mode. Again 200 iterations are averaged to obtain the curves shown in Fig. 9. Three increasing error rate scenarios are studied for the SPA, MSSC, and MLE algorithms.

As depicted in Fig. 9, it can be seen that for low values of error rates, the performance of the algorithm is very good. Also, for low error rates, the performance of the three algorithms is comparable. However, as the error rates increase, a very clear performance gap is shown in favor of SPA specially when high error rates ( $f_p = 0.6$  and  $f_n = 0.3$ ) are applied to the simulated experiments. Although these values seem high, they are actually close to observed error rates in PPI experiments [31]–[33]. A similar, yet less pronounced, trend can be observed when we do predictions on domain interactions, where we follow the same cross-validation parameters and error rates as with PPI prediction. The results are depicted in Fig. 10.

### B. Estimation in Experimental Data Sets: Yeast Golden Set

Experiments with artificial data helped us test the performance of SPA under known conditions and model assumptions. However, in order to validate the predictions made by our approach, real interaction data must be used and predicted. Furthermore, real PPI data itself are not enough; given the wide variety of experiments and conditions, it is hard to evaluate the quality of most experimental data sets. To avoid potential biases, we use a recently compiled set of protein interactions in yeast (*Saccharomyces cerevisiae*). This set of high-quality binary interaction was obtained by Yu *et al.* [31] and uses the Y2H technique to obtain pairwise interactions. Although high-throughput techniques like Y2H are known to contain high rates of false positives and negatives, this study focused on creating a golden set of PPI that were confirmed by several

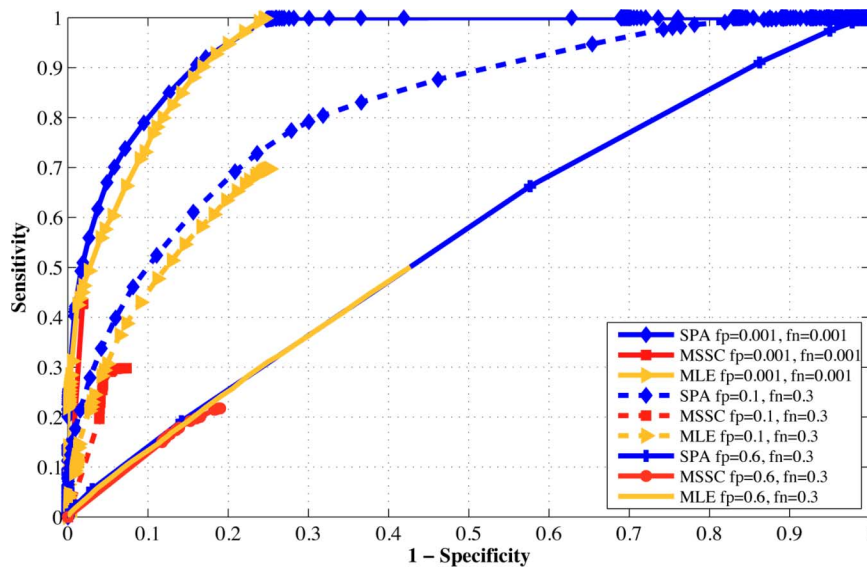


Fig. 10. Simulated DDI prediction. DDI prediction performance for SPA, MSSC, and MLE approaches using the receiver operating curve metric. Different reliabilities of experimental PPI data are used as input.

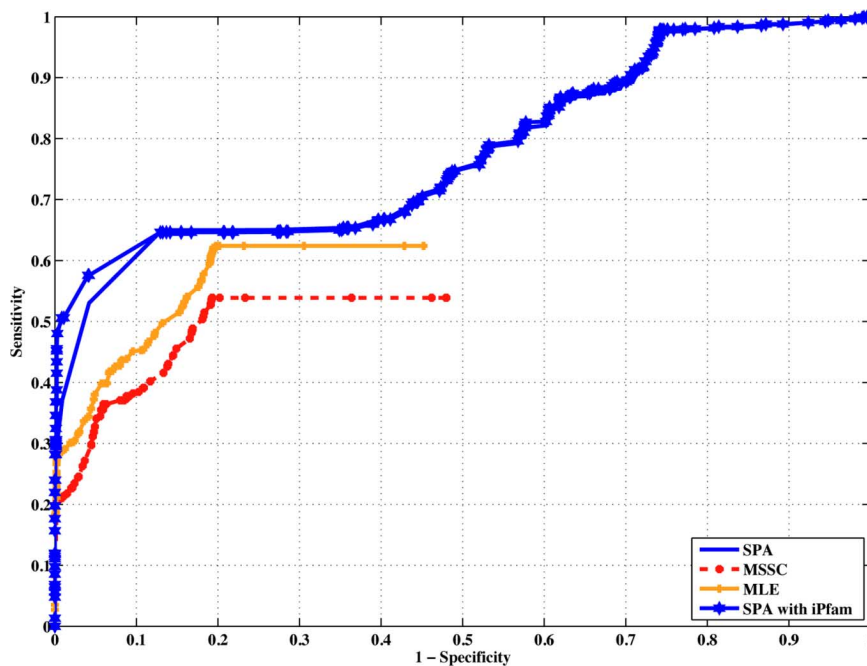


Fig. 11. Performance assessment of SPA in YGS data set. This curve was obtained using a training set of 283 (80%) protein pairs with a testing set of 70 protein pairs. There were 159 domain pairs and a GO threshold of 0.9 was used to select negative interaction pairs. A total of 1000 iterations were done to accumulate statistics.

experimental and statistical measures as well as by other experimental results [2], [34]. We use this high-confidence set with 1799 proteins and 2581 protein interactions that we were able to map to Uniprot accession numbers.

We do predictions on this set in two different ways. For the first case, we use the yeast golden set (YGS) as both training and testing sets. Approaches based on the domain model need domain information about the protein pairs for which we do inference. This is because only through shared domains messages from measurements can propagate in the factor graph. In other words, we aim to predict interactions only in factor graphs (see Fig. 4) with more than one protein pair connected via their domain architecture. Given the nature of the YGS, the

factor graphs that are formed consist of many isolated small factor graphs. To compare the performance of our algorithm, we picked the largest of those factor graphs containing 353 protein pairs, 159 connected domain pairs, and 565 edges. Positive interactions are extracted from the experiments and negative interactions are selected from the remaining pairings that have a GO distance of at least 0.9 (see Section IV-A).

Fig. 11 shows the performance of SPA compared to MSSC and MLE when doing predictions on this set. Cross validation was performed using 80% of the protein pairs as training set and the remaining 20% as testing set. To get statistical consistency, 1000 iterations of these predictions were averaged to get these curves. Fig. 11 shows that SPA performs better than MLE

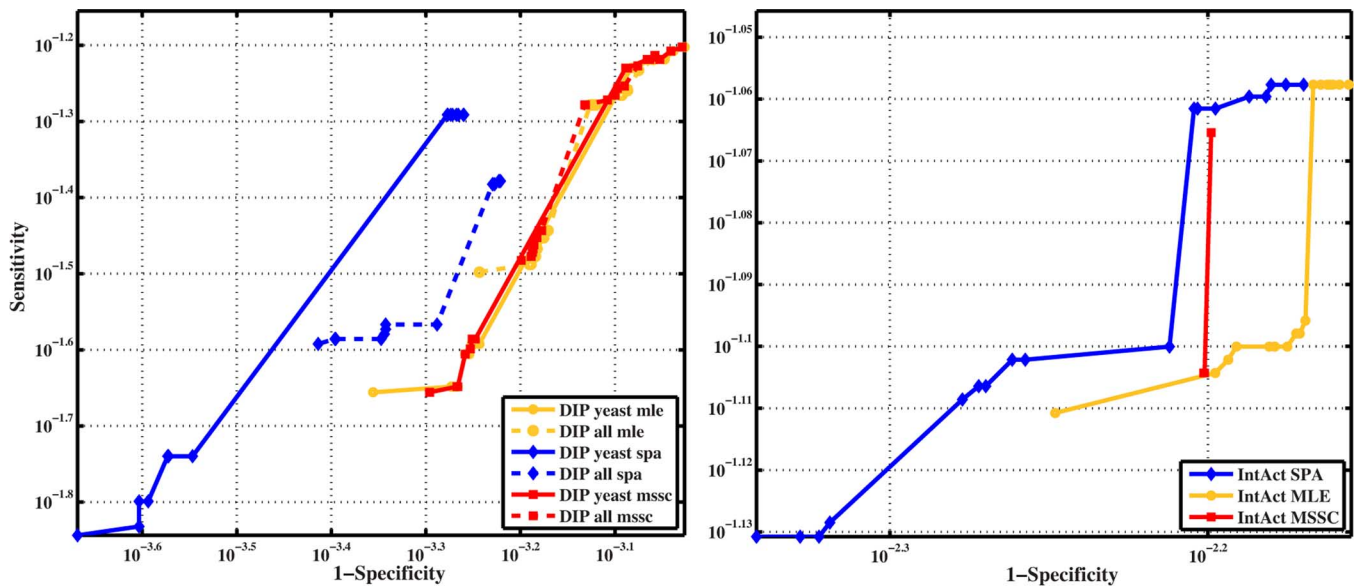


Fig. 12. Coverage. ROC curves of PPI predictions in the YGS of Yu *et al.* [31] using different input data from DIP (only yeast proteins), DIP (all species), and IntAct (all species).

and MSSC. We also analyzed the effect of using iPfam data as input data in our algorithm. Our iPfam data set consists of 6081 known domain interaction data with known 3-D structures. The inclusion of this data set as *a priori* DDI data has a positive impact on the predictive behavior (see Fig. 11) of our proposed methodology.

The second approach to assess estimation of PPI on the YGS is to use other sources of PPI data (low and high quality) to predict all the interactions in the golden set. Here, we do not consider connectedness of the factor graphs, but rather test how predictive is the use of external data sets to infer protein pairs in the golden set. We are investigating the coverage of different data sources in our testing set. We use three types of training data sets, excluding the YGS in all of them: 1) DIP yeast data set (DipY) which contains 12 670 PPI known for yeast in the DIP database; 2) DIP all species which has 33 234 known PPI for many organisms in the DIP database; and 3) IntAct all species with 150 876 known PPI for different species and coming from different experiments. To deal with the issue of protein pair specificity, i.e., proteins pairs with known pairs of interacting domains that do not interact *in vivo*, we use protein localization information from the GO database. We restrict our predictions to only those pairs that are known to be colocalized in the cell. Fig. 12 shows the ROC curves (in logarithmic scale) when trying to predict all protein pairs in YGS.

This plot shows very high values of specificity for all prediction algorithms, SPA being the one that performs better; however, the value of sensitivity is very small. This coverage is explained by the fact that the input data sets used do not have enough shared domains with protein pairs in the YGS to take advantage of the belief propagation of protein and domain interaction information. We analyzed the factor graphs that emerge from the input data and the YGS, and identified 1295 factor graphs containing only YGS pairs. This means that 1295 protein pairs in the golden set did not share any domain information

with other proteins in the training set that could enable them to do predictions. This lack of connectivity explains the sensitivity values observed in Fig. 12. For the particular case of the YGS, the domain pair connectivity was not enough to allow a higher coverage of their protein interactions. However, this is not the case for many other unexplored data sets for which more connectivity allows better coverage. Fig. 12 also shows the effect on coverage and specificity using different types of data sets. As expected, the predictions using the largest data set (IntAct) have a better coverage with a tradeoff of specificity. For the case of the smallest data set using only yeast interactions (DipY), we obtained higher values of specificity and sensitivity compared to using the complete DIP dataset. From the predictions obtained using only yeast proteins in DIP, which had the highest value of specificity, we were able to compile a high score set of 85 protein pairs (see the Appendix) that are not found in the YGS. We hypothesize that these might be potential novel interactions that were not identified by experimental methods but still have high probability values and are colocalized in the cell. Further, low-throughput experiments could help us prove if these interactions are indeed novel interactions.

## VI. CYTOPROPHET

We developed a Cytoscape [35] plug-in for PPI and DDI inference. It contains an implementation of the SPA algorithm as described in this work. This software tool called *Cytrophphet* [36] is accessible via the Cytoscape website ([www.cytoscape.org](http://www.cytoscape.org)) or in the Cytrophphet project website ([cytoprophphet.cse.nd.edu](http://cytoprophphet.cse.nd.edu)). This site contains instructions for installation, use, and sample sessions.

## VII. DISCUSSION

We present a new framework to perform PPI and DDI inference based on the efficient marginalization of the protein pair and domain pair joint probability distribution using the SPA. We show that the use of SPA for PPI and DDI estimation has

TABLE IV  
PREDICTED YEAST PROTEIN PAIRS NOT FOUND IN THE YGS. PROTEIN NAMES LISTED WITH ITS UNIPROT ACCESSION ID

Protein 1	Protein 2	Protein 1	Protein 2	Protein 1	Protein 2	Protein 1	Protein 2	Protein 1	Protein 2
P39533	P39533	P33309	P33309	P07991	P32835	P39954	P39954	P53378	P53378
P39714	P39945	P13382	P13382	P05150	P05150	P24280	P24280	P27796	P27796
P39714	P39714	Q04409	Q04409	P15873	P15873	P15303	P20606	P04786	P04786
P32451	P32451	P32771	P32771	P04046	P36973	P15303	P15303	Q08919	Q08919
P07258	P07258	P39676	P39676	P04046	P32895	P40482	P20606	P29509	P29509
P33322	P33322	P00359	P00359	P04046	P38620	P40482	P40482	P38816	P25372
P24868	P24868	Q00055	Q00055	P04046	P38689	P53953	P20606	P38816	P38816
P24869	P24869	P41911	P41911	P04046	P38063	P53953	P40482	P28274	P28274
P24870	P24868	P41921	P41921	P04046	Q12265	P53953	P53953	P40343	P40343
P24870	P24870	P32769	P32769	P38972	P38972	P38810	P20606	P53039	P53039
Q06440	Q06440	Q02821	Q02821	P26793	P32836	P38810	P40482	P53093	P53093
P33307	P33307	P11986	P11986	P26793	P26793	P38810	P53953	P53108	P53108
P53550	P53550	P52489	P52489	P40348	P40348	P38810	P38810	P47008	P47008
P07262	P07262	Q12446	Q12446	P38629	P38629	P52286	P52286	Q04869	P04387
P39708	P39708	P38998	P38998	P40339	P40348	P37297	P37297	Q04869	Q04869
P09624	P09624	P22855	P22855	P40339	P40339	P38714	P38714	Q12746	Q12746
P04819	P04819	P22133	P22133	Q02555	Q02555	P02557	P02557	P53729	P53729

predictive performance advantages over other algorithms that estimate protein interactions based on domain composition of proteins. In theory, MLE and maximum *a posteriori* (MAP) algorithms have comparable performance, however, since SPA is an approximate solution to the MAP problem and EM approximates the MLE problem, the relation between the exact solutions need not necessarily translate to the relation between the approximations. SPA has two additional advantages over EM solutions: it could benefit from the subexpression elimination (which was incompatible with the EM formulation in [14]), and it could generate more informative posterior probabilities for individual calls (advantageous in terms of ROC).

These advantages become more evident when the quality of experimental input data is exposed to a higher rate of errors. This is an important feature specially in real case scenarios where experiments are prone to high false positive and false negative rates. We use real proteins and their domain architecture to construct a network of related protein interactions. The performance of SPA was tested with artificially generated DDI and simulated PPI experiments under controlled parameters. We show that the use of SPA outperforms MLE and MSSC not only for interaction inference but for error correction of experimentally obtained interactions as well. Thus, SPA for PPI/DDI inference can also be used by experimentalists to “clean” interaction data.

For the case of real protein interaction data, we estimate PPI on a high-quality set of binary interactions in yeast, which allows us to confirm the predictive performance advantages of SPA over MLE and MSSC on real data. The use of structural domain interaction data from iPfam as *a priori* information has a positive effect on estimation accuracy. In addition, we identify that the use of a more extensive interaction data set, like IntAct, helps to increase coverage with still a high level of specificity. On the other hand, our results show that indiscriminate use of interaction data from all organisms has detrimental effect on performance. This is illustrated by the better results when using only interactions from yeast to predict on the golden set from the same organism, compared to using interactions from all organisms in DIP.

The coverage of algorithms based on domain architecture is relatively low in our experiments. Domain architecture of

proteins is relatively sparse; however, as more domains and their interactions are identified, this coverage should improve. Nonetheless, these algorithms predict highly accurate interactions, which are useful to experimental efforts to unravel the interaction maps of organisms. This methodology is general and can be applied to any organism where annotation of their protein and domains exists and can also take advantage of crystal structures of complete domains and proteins when available. Fortunately, the present increase in availability of these data would allow us to do these types of studies for many organisms and systems.

## VIII. FUTURE WORK

A model where domain interactions lead to protein interactions in combination with a powerful estimation algorithm yields good accuracy in molecular interaction inference. Conserved domains vary their specificity across different protein families, for instance, in the case of kinases [37]. Protein regions without an identified conserved domain might have an important role in PPI. The model presented here could be extended and upgraded to include such biological insights about domain differences across proteins as well as nonconserved regions. For example, disordered regions are particularly involved in transient interactions, such as those in signaling networks [38]. Identifying these regions in addition to the knowledge of conserved domains could help improve this framework by exploiting the information about transient interactions that these disordered regions could provide. As part of our future work, we will use this methodology to study organisms and systems where the knowledge of PPI is limited. One concrete example is a pathway of motility reversals that allow the gram-negative bacterium *Myxococcus xanthus* to swarm [39]. PPI estimation could provide evidence of the role that motility proteins play in *M. xanthus* reversals. This could help develop testable biological hypotheses that would increase our understanding of bacterial motility.

## APPENDIX

### PREDICTED INTERACTIONS NOT FOUND IN YGS

See Table IV.

## ACKNOWLEDGMENT

The authors would like to thank C. Lamanna for his help compiling several of the interaction data sets used in this work and his work on Cytrophet.

## REFERENCES

- [1] A.-L. Barabási and Z. N. Oltvai, "Network biology: Understanding the cell's functional organization," *Nature Rev. Genetics*, vol. 5, no. 2, pp. 101–113, Feb. 2004 [Online]. Available: <http://dx.doi.org/10.1038/nrg1272>
- [2] P. Uetz, L. Giot, G. Cagney, T. A. Mansfield, R. S. Judson, J. R. Knight, D. Lockshon, V. Narayan, M. Srinivasan, P. Pochart, A. Qureshi-Emili, Y. Li, B. Godwin, D. Conover, T. Kalbfleisch, G. Vijayadmodar, M. Yang, M. Johnston, S. Fields, and J. M. Rothberg, "A comprehensive analysis of protein-protein interactions in *Saccharomyces cerevisiae*," *Nature*, vol. 403, no. 6770, pp. 623–627, Feb. 2000 [Online]. Available: <http://dx.doi.org/10.1038/35001009>
- [3] S. Suthram, T. Sittler, and T. Ideker, "The plasmodium protein network diverges from those of other eukaryotes," *Nature*, vol. 438, no. 7064, pp. 108–112, Nov. 2005 [Online]. Available: <http://dx.doi.org/10.1038/nature04135>
- [4] S. Bandyopadhyay, R. Sharan, and T. Ideker, "Systematic identification of functional orthologs based on protein network comparison," *Genome Res.*, vol. 16, no. 3, pp. 428–435, Mar. 2006 [Online]. Available: <http://dx.doi.org/10.1101/gr.4526006>
- [5] I. Xenarios, L. Salwinski, X. J. Duan, P. Higney, S.-M. Kim, and D. Eisenberg, "DIP, the database of interacting proteins: A research tool for studying cellular networks of protein interactions," *Nucleic Acids Res.*, vol. 30, no. 1, pp. 303–305, Jan. 2002.
- [6] N. Deshpande, K. J. Address, W. F. Bluhm, J. C. Merino-Ott, W. Townsend-Merino, Q. Zhang, C. Knezevich, L. Xie, L. Chen, Z. Feng, R. K. Green, J. L. Flippen-Anderson, J. Westbrook, H. M. Berman, and P. E. Bourne, "The RCSB protein data bank: A redesigned query system and relational database based on the mmCIF schema," *Nucleic Acids Res.*, vol. 33, no. Database issue, pp. D233–D237, Jan. 2005 [Online]. Available: <http://dx.doi.org/10.1093/nar/gki057>
- [7] R. D. Finn, M. Marshall, and A. Bateman, "iPfam: Visualization of protein-protein interactions in PDB at domain and amino acid resolutions," *Bioinformatics*, vol. 21, no. 3, pp. 410–412, Feb. 2005 [Online]. Available: <http://dx.doi.org/10.1093/bioinformatics/bti011>
- [8] A. Bateman, L. Coin, R. Durbin, R. D. Finn, V. Hollich, S. Griffiths-Jones, A. Khanna, M. Marshall, S. Moxon, E. L. L. Sonnhammer, D. J. Studholme, C. Yeats, and S. R. Eddy, "The Pfam protein families database," *Nucleic Acids Res.*, vol. 32, no. Database issue, pp. D138–D141, Jan. 2004 [Online]. Available: <http://dx.doi.org/10.1093/nar/gkh121>
- [9] M. Ashburner, C. A. Ball, J. A. Blake, D. Botstein, H. Butler, J. M. Cherry, A. P. Davis, K. Dolinski, S. S. Dwight, J. T. Eppig, M. A. Harris, D. P. Hill, L. Issel-Tarver, A. Kasarskis, S. Lewis, J. C. Matese, J. E. Richardson, M. Ringwald, G. M. Rubin, and G. Sherlock, "Gene ontology: Tool for the unification of biology. The gene ontology consortium," *Nature Genetics*, vol. 25, no. 1, pp. 25–29, May 2000.
- [10] D. Ekman, A. K. Björklund, J. Frey-Skött, and A. Elofsson, "Multi-domain proteins in the three kingdoms of life: Orphan domains and other unassigned regions," *J. Mol. Biol.*, vol. 348, no. 1, pp. 231–243, Apr. 2005 [Online]. Available: <http://dx.doi.org/10.1016/j.jmb.2005.02.007>
- [11] Z. Itzhaki, E. Akiva, Y. Altuvia, and H. Margalit, "Evolutionary conservation of domain-domain interactions," *Genome Biol.*, vol. 7, no. 12, pp. R125–R125, 2006 [Online]. Available: <http://dx.doi.org/10.1186/gb-2006-7-12-r125>
- [12] J. Pandey, M. Koyutürk, S. Subramaniam, and A. Grama, "Functional coherence in domain interaction networks," *Bioinformatics*, vol. 24, no. 16, pp. i28–i34, Aug. 2008 [Online]. Available: <http://dx.doi.org/10.1093/bioinformatics/btn296>
- [13] T. Pawson and P. Nash, "Assembly of cell regulatory systems through protein interaction domains," *Science*, vol. 300, no. 5618, pp. 445–452, Apr. 2003 [Online]. Available: <http://dx.doi.org/10.1126/science.1083653>
- [14] M. Deng, S. Mehta, F. Sun, and T. Chen, "Inferring domain-domain interactions from protein-protein interactions," *Genome Res.*, vol. 12, no. 10, pp. 1540–1548, Oct. 2002 [Online]. Available: <http://dx.doi.org/10.1101/gr.153002>
- [15] J. Ye, C. Kulikowski, and I. Muchnik, "Protein-protein interaction prediction based on sequence data by support vector machine with probability assignment," in *Proc. IEEE Symp. Comput. Intell. Bioinf. Comput. Biol.*, Nov. 14–15, 2005, pp. 1–7.
- [16] C. Huang, F. Morcos, S. P. Kanaan, S. Wuchty, D. Z. Chen, and J. A. Izaguirre, "Predicting protein-protein interactions from protein domains using a set cover approach," *IEEE/ACM Trans. Comput. Biol. Bioinf.*, vol. 4, no. 1, pp. 78–87, 2007 [Online]. Available: <http://dx.doi.org/10.1109/TCBB.2007.1001>
- [17] H. Wang, E. Segal, A. Ben-Hur, Q.-R. Li, M. Vidal, and D. Koller, "In-Site: A computational method for identifying protein-protein interaction binding sites on a proteome-wide scale," *Genome Biol.*, vol. 8, no. 9, pp. R192–R192, Sep. 2007 [Online]. Available: <http://dx.doi.org/10.1186/gb-2007-8-9-r192>
- [18] C. L. Myers, D. Robson, A. Wible, M. A. Hibbs, C. Chiriack, C. L. Theesfeld, K. Dolinski, and O. G. Troyanskaya, "Discovery of biological networks from diverse functional genomic data," *Genome Biol.*, vol. 6, no. 13, pp. R114–R114, 2005 [Online]. Available: <http://dx.doi.org/10.1186/gb-2005-6-13-r114>
- [19] C. L. Myers and O. G. Troyanskaya, "Context-sensitive data integration and prediction of biological networks," *Bioinformatics*, vol. 23, no. 17, pp. 2322–2330, Sep. 2007 [Online]. Available: <http://dx.doi.org/10.1093/bioinformatics/btm332>
- [20] L. Burger and E. v. Nimwegen, "Accurate prediction of protein-protein interactions from sequence alignments using a Bayesian method," *Mol. Syst. Biol.*, vol. 4, pp. 165–165, 2008 [Online]. Available: <http://dx.doi.org/10.1038/msb4100203>
- [21] R. Chen, L. Li, and Z. Weng, "ZDOCK: An initial-stage protein-docking algorithm," *Proteins*, vol. 52, no. 1, pp. 80–87, Jul. 2003 [Online]. Available: <http://dx.doi.org/10.1002/prot.10389>
- [22] X. Li, O. Keskin, B. Ma, R. Nussinov, and J. Liang, "Protein-protein interactions: Hot spots and structurally conserved residues often locate in complemented pockets that pre-organized in the unbound states: Implications for docking," *J. Mol. Biol.*, vol. 344, no. 3, pp. 781–795, Nov. 2004 [Online]. Available: <http://dx.doi.org/10.1016/j.jmb.2004.09.051>
- [23] F. Kschischang, B. Frey, and H.-A. Loeliger, "Factor graphs and the sum-product algorithm," *IEEE Trans. Inf. Theory*, vol. 47, no. 2, pp. 498–519, Feb. 2001.
- [24] B. L. Ng, J. S. Evans, S. V. Hanly, and D. Aktas, "Distributed downlink beamforming with cooperative base stations," *IEEE Trans. Inf. Theory*, vol. 54, no. 12, pp. 5491–5499, Dec. 2008.
- [25] A. Kavcic, X. Ma, and M. Mitzenmacher, "Binary intersymbol interference channels: Gallager codes, density evolution, and code performance bounds," *IEEE Trans. Inf. Theory*, vol. 49, no. 7, pp. 1636–1652, Jul. 2003.
- [26] X.-Y. Hu, E. Eleftheriou, D.-M. Arnold, and A. Dholakia, "Efficient implementations of the sum-product algorithm for decoding LDPC codes," in *Proc. IEEE Global Telecommun. Conf.*, 2001, vol. 2, pp. 1036–1036E, DOI: 10.1109/GLOCOM.2001.965575.
- [27] M. Chiang, "Distributed network control through sum product algorithm on graphs," in *Proc. IEEE Global Telecommun. Conf.*, Nov. 17–21, 2002, vol. 3, pp. 2395–2399.
- [28] S. Sankaranarayanan and B. Vasic, "Iterative decoding of linear block codes: A parity-check orthogonalization approach," *IEEE Trans. Inf. Theory*, vol. 51, no. 9, pp. 3347–3353, Sep. 2005.
- [29] U. Consortium, "The universal protein resource (UniProt)," *Nucleic Acids Res.*, vol. 35, no. Database issue, pp. D193–D197, Jan. 2007.
- [30] S. Kerrien, Y. Alam-Faruque, B. Aranda, I. Bancarz, A. Bridge, C. Derow, E. Dummer, M. Feuermann, A. Friedrichsen, R. Huntley, C. Kohler, J. Khadake, C. Leroy, A. Liban, C. Lieftink, L. Montecchi-Palazzi, S. Orchard, J. Risse, K. Robbe, B. Roechert, D. Thorneycroft, Y. Zhang, and R. A. Hermjakob, "Intact-open source resource for molecular interaction data," *Nucleic Acids Res.*, vol. 35, no. Database issue, pp. D561–D565, Jan. 2007 [Online]. Available: <http://dx.doi.org/10.1093/nar/gkl958>
- [31] H. Yu, P. Braun, M. A. Yildirim, I. Lemmens, K. Venkatesan, J. Sahalie, T. Hirozane-Kishikawa, F. Gebreab, N. Li, N. Simonis, T. Hao, J.-F. Rual, A. Dricot, A. Vazquez, R. R. Murray, C. Simon, L. Tardivo, S. Tam, N. Svrikapa, C. Fan, A.-S. de Smet, A. Motyl, M. E. Hudson, J. Park, X. Xin, M. E. Cusick, T. Moore, C. Boone, M. Snyder, F. P. Roth, A.-L. Barabási, J. Tavernier, D. E. Hill, and M. Vidal, "High-quality binary protein interaction map of the yeast interactome network," *Science*, vol. 322, no. 5898, pp. 104–110, Oct. 2008 [Online]. Available: <http://dx.doi.org/10.1126/science.1158684>
- [32] C. M. Deane, L. Salwinski, I. Xenarios, and D. Eisenberg, "Protein interactions: Two methods for assessment of the reliability of high throughput observations," *Mol. Cell Proteomics*, vol. 1, no. 5, pp. 349–356, May 2002.
- [33] A. M. Edwards, B. Kus, R. Jansen, D. Greenbaum, J. Greenblatt, and M. Gerstein, "Bridging structural biology and genomics: Assessing protein interaction data with known complexes," *Trends Genetics*, vol. 18, no. 10, pp. 529–536, Oct. 2002.
- [34] T. Ito, T. Chiba, R. Ozawa, M. Yoshida, M. Hattori, and Y. Sakaki, "A comprehensive two-hybrid analysis to explore the yeast protein interactome," *Proc. Nat. Acad. Sci. USA*, vol. 98, no. 8, pp. 4569–4574, Apr. 2001 [Online]. Available: <http://dx.doi.org/10.1073/pnas.061034498>
- [35] P. Shannon, A. Markiel, O. Ozier, N. S. Baliga, J. T. Wang, D. Ramage, N. Amin, B. Schwikowski, and T. Ideker, "Cytoscape: A software environment for integrated models of biomolecular interaction networks," *Genome Res.*, vol. 13, no. 11, pp. 2498–2504, Nov. 2003 [Online]. Available: <http://dx.doi.org/10.1101/gr.1239303>

- [36] F. Morcos, C. Lamanna, M. Sikora, and J. Izaguirre, "Cytrophet: A cytoscape plug-in for protein and domain interaction networks inference," *Bioinformatics*, vol. 24, no. 19, pp. 2265–2266, Oct. 2008 [Online]. Available: <http://dx.doi.org/10.1093/bioinformatics/btn380>
- [37] K. Fujii, G. Zhu, Y. Liu, J. Hallam, L. Chen, J. Herrero, and S. Shaw, "Kinase peptide specificity: Improved determination and relevance to protein phosphorylation," *Proc. Nat. Acad. Sci. USA*, vol. 101, no. 38, pp. 13744–13749, Sep. 2004 [Online]. Available: <http://dx.doi.org/10.1073/pnas.0401881101>
- [38] A. K. Dunker, J. D. Lawson, C. J. Brown, R. M. Williams, P. Romero, J. S. Oh, C. J. Oldfield, A. M. Campen, C. M. Ratliff, K. W. Hipps, J. Ausio, M. S. Nissen, R. Reeves, C. Kang, C. R. Kissinger, R. W. Bailey, M. D. Griswold, W. Chiu, E. C. Garner, and Z. Obradovic, "Intrinsically disordered protein," *J. Mol. Graph. Model*, vol. 19, no. 1, pp. 26–59, 2001.
- [39] Y. Wu, A. D. Kaiser, Y. Jiang, and M. S. Alber, "Periodic reversal of direction allows myxobacteria to swarm," *Proc. Nat. Acad. Sci. USA*, vol. 106, no. 4, pp. 1222–1227, Jan. 2009 [Online]. Available: <http://dx.doi.org/10.1073/pnas.0811662106>

**Faruck Morcos** (S'99–M'07) received the B.S. degree in electronics and communications engineering from ITESM, Monterrey, Mexico, in 2001 and the M.Sc. degree in communications engineering from the Technische Universität München, Munich, Germany, in 2004. Currently, he is working towards the Ph.D. degree at the Department of Computer Science and Engineering, University of Notre Dame, Notre Dame, IN.

His current research interests include systems biology, network science, and applications of information theory and signal processing in molecular biology.

**Marcin Sikora** (S'99–M'07) received the M.S. degree in computer science from Silesian Technical University, Gliwice, Poland, in 2000, the M.S. degree in communications engineering from the Technische Universität München, Munich, Germany, in 2002, and the Ph.D. degree in electrical engineering from University of Notre Dame, Notre Dame, IN, in 2008.

Since 2008 he has been with Life Technologies, Foster City, CA. His research interests are in the areas of error correcting codes, information theory, and bioinformatics.

**Mark S. Alber** received the Ph.D. degree in mathematics from the University of Pennsylvania, College Station, in 1990.

Currently, he is the Vincent J. Duncan Family Professor of Applied Mathematics and Concurrent Professor of Physics at the University of Notre Dame, Notre Dame, IN, where he also serves as Director of the Center for the Study of Biocomplexity. His research interests are in mathematical and computational biology.

**Dale Kaiser** received the B.S. degree in liberal science from Purdue University, West Lafayette, IN, in 1950 and the Ph.D. degree in biology and chemistry from the California Institute of Technology, Pasadena, in 1955.

Currently, he is Professor of Biochemistry and of Developmental Biology at Stanford University, Stanford, CA. He has been Professor at Stanford University since 1959. His research interests include genetic regulation in organism development, cellular signaling, and multicellular coordination. He has worked extensively in several aspects of *Myxococcus xanthus* fruiting body development, sporulation, and aggregation.

Dr. Kaiser received the 1981 Waterford Prize for Basic Medical Research, the 1992 Thomas Hunt Morgan Award given by the Genetics Society of America, and in 2005 he was appointed Wilson Professor of Biochemistry (Emeritus) at Stanford University.

**Jesús A. Izaguirre** received the Ph.D. degree in computer science from the University of Illinois at Urbana-Champaign, Urbana, in 1999.

Currently, he is an Associate Professor of Computer Science and Engineering at the University of Notre Dame, Notre Dame, IN. His current research is on efficient methods in chemistry and biology, particularly molecular dynamics, Monte Carlo methods, cellular automata, and analysis of biological networks. He is also interested in the portable implementation of high performance software for scientific computing.

Dr. Izaguirre received a CAREER Award of the National Science Foundation in 2001, and a BP Foundation Outstanding Teacher of the Year Award in 2005.