

Bayesian Inference of Protein and Domain Interactions Using The Sum-Product Algorithm

Marcin Sikora*, Faruck Morcos[†], Daniel J. Costello, Jr.*, and Jesús A. Izaguirre[†]

*Department of Electrical Engineering, University of Notre Dame, Notre Dame, IN 46556

Email: msikora@ieee.org, costello.2@nd.edu

[†]Department of Computer Science and Engineering, University of Notre Dame, Notre Dame, IN 46556

Email: amorcosg@nd.edu, izaguirr@cse.nd.edu

Abstract—In order to fully understand the functions of proteins in living organisms we must study their interactions and construct accurate interaction maps. Each protein can be composed of one or several peptide chains called domains and each protein interaction can be seen as a consequence of an underlying interaction of two domains, one from each protein. Since high-throughput methods of measuring protein interactions, such as yeast two-hybrid assay, have high error rates, the structural similarities between proteins can be exploited to detect some of the experimental errors and predict new, unmeasured interactions. In this paper we solve the problem of Bayesian inference of protein and domain interactions by computing their likelihoods conditioned on the measurement results. We formulate the task of calculating these conditional likelihoods as a functional marginalization problem, where the multivariate function to be marginalized naturally factors into simpler local functions, and demonstrate how this equivalent problem can be solved using the sum-product algorithm. We note that such task is structurally similar to the decoding the low density parity check codes using the message passing algorithm. The robustness and accuracy of our approach is demonstrated by predicting protein and domain interactions on both real and artificial measurement data.

I. INTRODUCTION

The progress in genomic technology has made it possible to map entire genomes of numerous organisms, with new species being examined on a steady basis. As the number of known genes increases, it is the goal of the modern systems biology to understand the function and the interrelation between proteins encoded within them [1]–[4]. Of special interest is establishing protein interaction maps that ultimately allow us to build a complete interactome for a given organism. Reaching this goal is crucial to gain full understanding of the living processes in the cell.

Interactions, in which proteins transiently or stably bind to each other, can be detected by several experimental methods. Since the typical number of proteins in mammals and plants is between 20 and 40 thousand, hundreds of millions of protein pairs need to be examined for a potential interaction. Unfortunately, testing techniques that can be automated and performed in high throughput setting produce large number of measurement errors, while the most accurate techniques are too complex, time consuming, and costly to be practical

for mapping the entire interaction networks. For example, the high-throughput yeast two-hybrid assay [5], [6] is estimated to have a 0.67 false negative and 0.0007 false positive rate [7]. Clearly, any data processing techniques that can extrapolate from existing measurements to unmeasured protein pairs and deal with noisy results are of great practical interest.

The most promising approaches are based on the observation that similar proteins generally interact with the same partners. A particularly useful tool, which we will call an independent domain model (IDM) throughout this paper, models the fact that proteins are generally composed of smaller, independently folding peptide modules called *domains* [7], [8]. Any interaction between two proteins is in fact a consequence of an underlying interaction of two domains, one from each protein. Since most domains appear as part of more than one protein, detecting an interaction between one protein pair can be used as an evidence in inferring interactions of other pairs sharing same domain pairs. Additionally, the knowledge of domain interactions can also be used to gain insight into the physical character or mechanism of protein interactions.

The practical determination of domain composition of a given protein has been studied thoroughly. Since domains fold independently, it is sufficient to check whether the amino acid sequence representing a given domain is a substring of the protein sequence. Databases of both protein sequences and domain sequences are readily available on the Internet. Processed lists of proteins and their component domains are also available [9], [10].

Several techniques that use protein interaction measurements to determine protein and domain interactions have been presented in the literature, such as set cover techniques [11] and variants of Maximum Likelihood estimation (MLE) [7], [8]. These methods first search for domain pairs that are likely to interact and compute their likelihood score, and then use this information to obtain the likelihoods of protein pair interactions.

In [11], the authors interpret protein pairs as elements and domain pairs as sets, with a set containing an element if the two domains are parts of the two proteins. The search for interacting domain pairs is then treated as a problem of covering protein pairs that were measured as interacting with sets. Proposed optimality criteria include minimum set cover, minimum exact set cover, and maximum specificity set cover

¹This work was supported in part by NSF grants DBI-0450067 and CCF-0622940.

(MSSC). The domain interaction score is then computed as a ratio of the number of measured elements in the set to the size of the set for domain pairs in the cover set and zero for all other domain pairs.

In [7], the domain pairs are assumed to interact randomly, and each domain pair is characterized by its probability of interaction. These probabilities are then estimated using the MLE method, i.e., the values that best explain the observed interactions are found. The authors search for the MLE solution using the Expectation Maximization algorithm [12].

In the present work, we propose to use the IDM and experimental data to infer protein and domain interactions using Bayesian inference, and use the proper conditional likelihood of interaction as a natural measure of the prediction confidence. We demonstrate how this conditional likelihood for measured protein pairs, new protein pairs and domain pairs can be computed using the Sum-Product Algorithm (SPA) [13], an iterative algorithm for computing marginal values of multivariate functions.

The necessary notation is introduced in Section 2. Section 3 states the problem of computing the conditional likelihood of protein and domain interaction and formulates the SPA. Section 4 demonstrates prediction capabilities of our technique on different scenarios and in real data and Section 5 draws conclusions.

II. NOTATION AND ASSUMPTIONS

In order to precisely state the problem and develop our solution we first need to define the necessary concepts and variables. The main objects in our problem are protein pairs and domain pairs, the interaction of which we measure and predict. We denote the set of such protein pairs as \mathcal{A} and domain pairs as \mathcal{B} . For each protein pair $(i, j) \in \mathcal{A}$ we will write $\mathcal{B}_{i,j} \subset \mathcal{B}$ to denote the set of domain pairs (x, y) such that domain x is present in protein i and domain y is present in protein j . Analogously, $\mathcal{A}_{x,y} \subset \mathcal{A}$ is a set of protein pairs that contain the domain pair (x, y) .

For each protein pair $(i, j) \in \mathcal{A}$ we define an interaction indicator $A_{i,j}$ such that $A_{i,j} = 1$ denotes a hypothesis that proteins i and j interact and $A_{i,j} = 0$ is the opposite hypothesis. In an analogous way we define an interaction indicator $B_{x,y}$ for each domain pair (x, y) in \mathcal{B} . Also, for each $(i, j) \in \mathcal{A}$ we will use $M_{i,j}$ to describe the results of interaction measurements performed on this pair. In general, $M_{i,j}$ can include zero, one, or more measurements.

The indicators $A_{i,j}$, $B_{x,y}$, and $M_{i,j}$ for all protein and domain pairs in \mathcal{A} and \mathcal{B} are grouped into collections \mathbf{A} , \mathbf{B} , and \mathbf{M} , respectively. Furthermore, $\mathbf{B}_{i,j}$ denotes the collection of all domain pairs $B_{x,y}$, such that $(x, y) \in \mathcal{B}_{i,j}$. For each of the above indicators and collections written in capital letters we will use lower case letter variables as their values. For example, $\mathbf{A} = \mathbf{a}$ will refer to the hypothesis that all protein pairs interact according to configuration \mathbf{a} and the probability or likelihood of such hypothesis will be denoted by $P_{\mathbf{A}}(\mathbf{a})$. Whenever the indicators are clear from the context, we will omit the subscript, simply writing $P(\mathbf{a})$.

Throughout the paper we will apply a compact notation for marginalizing sums of multivariate functions introduced in [13]. For example, for $f(x, y, z, q)$ we will write

$$\sum_{\sim\{x\}} f(x, y, z, q) = \sum_{y \in \mathcal{Y}} \sum_{z \in \mathcal{Z}} \sum_{q \in \mathcal{Q}} f(x, y, z, q),$$

where $\sim\{x\}$ indicates that the summation takes place over the complete domains \mathcal{Y} , \mathcal{Z} , \mathcal{Q} of all arguments to f , except x . Sums with $\sim\{\}$ are taken over all arguments.

III. BAYESIAN INFERENCE AND THE SUM-PRODUCT ALGORITHM

A. The Sum-Product Algorithm

Bayesian inference of domain and protein interactions involves computing $P_{A_{i,j}|\mathbf{M}}(1|\mathbf{m})$ and $P_{B_{x,y}|\mathbf{M}}(1|\mathbf{m})$, the likelihoods of interaction given the available measurements. These likelihoods, besides directly measuring our confidence in declaring certain domain or protein pair as interacting, can be used to compute any optimal Bayesian estimate of interaction. In fact, all such estimates reduce to a simple thresholding operation on the value of $P_{A_{i,j}|\mathbf{M}}(1|\mathbf{m})$ or $P_{B_{x,y}|\mathbf{M}}(1|\mathbf{m})$, with protein or domain pair declared as interacting if the likelihood exceeds the threshold and declared noninteracting otherwise.

By applying the Bayes formula we obtain

$$P_{A_{i,j}|\mathbf{M}}(1|\mathbf{m}) = \frac{\sum_{\sim\{a_{i,j}\}} P(\mathbf{a}, \mathbf{b}, \mathbf{m}) \Big|_{a_{i,j}=1}}{\sum_{\sim\{\}} P(\mathbf{a}, \mathbf{b}, \mathbf{m})}, \quad (1)$$

$$P_{B_{x,y}|\mathbf{M}}(1|\mathbf{m}) = \frac{\sum_{\sim\{b_{x,y}\}} P(\mathbf{a}, \mathbf{b}, \mathbf{m}) \Big|_{b_{x,y}=1}}{\sum_{\sim\{\}} P(\mathbf{a}, \mathbf{b}, \mathbf{m})}, \quad (2)$$

where the sums do not marginalize \mathbf{m} , a collection of known constants. The direct computation of protein and domain interaction likelihoods according to formulas (1) and (2) is in most cases prohibitively complex, since the number of summands that need to be evaluated is exponential in the number of protein and domain pairs involved. Instead, in this paper we will use the Sum-Product Algorithm (SPA) [13], an iterative algorithm for computing marginals of multivariate functions. The algorithm can be applied to functions which can be decomposed into products of simpler “local” functions in lower number of variables. The particular function to be marginalized in our case is $P(\mathbf{a}, \mathbf{b}, \mathbf{m})$, which naturally factors into $P(\mathbf{b})P(\mathbf{a}|\mathbf{b})P(\mathbf{m}|\mathbf{a})$. Each of these factors can be further

decomposed into

$$P(\mathbf{b}) = \prod_{(x,y) \in \mathcal{B}} P(b_{x,y}), \quad (3)$$

$$P(\mathbf{a}|\mathbf{b}) = \prod_{(i,j) \in \mathcal{A}} P(a_{i,j}|\mathbf{b}_{i,j}), \quad (4)$$

$$P(\mathbf{m}|\mathbf{a}) = \prod_{(i,j) \in \mathcal{A}} P(m_{i,j}|a_{i,j}). \quad (5)$$

Since $a_{i,j}$ is a deterministic function of the interaction variables $\mathbf{b}_{i,j}$, the probability $P(a_{i,j}|\mathbf{b}_{i,j})$ takes a form of a simple predicate function,

$$P(a_{i,j}|\mathbf{b}_{i,j}) = \begin{cases} 1 & \text{if } a_{i,j} = 0 \text{ and } \forall b_{x,y} \in \mathbf{b}_{i,j} b_{x,y} = 0, \\ 1 & \text{if } a_{i,j} = 1 \text{ and } \exists b_{x,y} \in \mathbf{b}_{i,j} b_{x,y} = 1, \\ 0 & \text{otherwise.} \end{cases} \quad (6)$$

The decomposition of $P(\mathbf{a}, \mathbf{b}, \mathbf{m})$ can be illustrated using the *factor graph* presented in Fig. 1. The variable nodes $a_{i,j}$ and $b_{x,y}$ are represented by circles, while factors in (3), (4), and (5) are shown as squares, OR-gate symbols, and diamonds, respectively. In a factor graph, a variable node is connected by an edge to a function node if the variable is an argument to the function. Note that $m_{i,j}$ are not included in the graph, since these variables are not being marginalized.

The SPA computes marginal functions of $P(\mathbf{a}, \mathbf{b}, \mathbf{m})$ by passing messages between variable and function nodes. A message entering or departing a variable node is itself a function of this variable. The general formula for the messages and the motivation for the message passing operation of SPA can be found in [13]. When applied to our factor graph, the distinct message types, shown in Fig. 2a, are computed as follows.

$$\alpha_{i,j}(a_{i,j}) = P(m_{i,j}|a_{i,j}), \quad (7)$$

$$\beta_{x,y}(b_{x,y}) = P(b_{x,y}), \quad (8)$$

$$\begin{aligned} \gamma_{i,j}^{x,y}(b_{x,y}) &= \sum_{\sim\{b_{x,y}\}} P(a_{i,j}|\mathbf{b}_{i,j}) \alpha_{i,j}(a_{i,j}) \\ &\quad \cdot \prod_{\substack{(x',y') \in \mathcal{B}_{i,j} \\ (x',y') \neq (x,y)}} \delta_{i,j}^{x',y'}(b_{x',y'}), \quad (9) \end{aligned}$$

$$\delta_{i,j}^{x,y}(b_{x,y}) = \beta_{x,y}(b_{x,y}) \prod_{\substack{(i',j') \in \mathcal{A}_{x,y} \\ (i',j') \neq (i,j)}} \gamma_{i',j'}^{x,y}(b_{x,y}), \quad (10)$$

$$\epsilon_{i,j}(a_{i,j}) = \sum_{\sim\{a_{i,j}\}} P(a_{i,j}|\mathbf{b}_{i,j}) \prod_{(x,y) \in \mathcal{B}_{i,j}} \delta_{i,j}^{x,y}(b_{x,y}). \quad (11)$$

The marginal functions (1) and (2) can be then expressed as a product of messages arriving at the node of the variable not being marginalized over,

$$\sum_{\sim\{a_{i,j}\}} P(\mathbf{a}, \mathbf{b}, \mathbf{m}) = \epsilon_{i,j}(a_{i,j}) \alpha_{i,j}(a_{i,j}), \quad (12)$$

$$\sum_{\sim\{b_{x,y}\}} P(\mathbf{a}, \mathbf{b}, \mathbf{m}) = \beta_{x,y}(b_{x,y}) \prod_{(i,j) \in \mathcal{A}_{x,y}} \gamma_{i,j}^{x,y}(b_{x,y}). \quad (13)$$

If the factor graph of $P(\mathbf{a}, \mathbf{b}, \mathbf{m})$ is free of cycles, it is possible to order the above computations in such way that every message can be computed from values obtained in earlier steps. In such case, formulas (7)-(13) yield the exact marginal functions. If, however, the factor graph contains cycles, some functions $\gamma_{i,j}^{x,y}$ and $\delta_{i,j}^{x,y}$ must be set to appropriate initial values, after which computations (9) and (10) are performed iteratively. In such case, the final steps of (12) and (13) are only approximations. Although this iterative processing is not guaranteed to converge to the exact solution, applications of iterative SPA to estimation problems in communication theory have shown its good performance [14].

In the procedure described by equations (7)-(13), the messages are functions of a single binary variable. This effectively means that the actual message would have to consist of two numbers. For example, computing $\delta_{i,j}^{x,y}(b_{x,y})$ corresponds to calculating both $\delta_{i,j}^{x,y}(0)$ and $\delta_{i,j}^{x,y}(1)$. However, for the purpose of computing (1) and (2) it is sufficient to only track the ratios of these numbers. In particular, the algorithm (7)-(10) can be rewritten as follows

$$\tilde{\alpha}_{i,j} = \frac{\alpha_{i,j}(0)}{\alpha_{i,j}(1)} = \frac{P_{M_{i,j}|A_{i,j}}(m_{i,j}|0)}{P_{M_{i,j}|A_{i,j}}(m_{i,j}|1)}, \quad (14)$$

$$\tilde{\beta}_{x,y} = \frac{\beta_{x,y}(0)}{\beta_{x,y}(1)} = \frac{P_{B_{x,y}}(0)}{P_{B_{x,y}}(1)}, \quad (15)$$

$$\tilde{\gamma}_{i,j}^{x,y} = \frac{\gamma_{i,j}^{x,y}(0)}{\gamma_{i,j}^{x,y}(1)} = 1 + (\tilde{\alpha}_{i,j} - 1) \prod_{\substack{(x',y') \\ \neq (x,y)}} \frac{\tilde{\delta}_{i,j}^{x',y'}}{\tilde{\delta}_{i,j}^{x',y'} + 1}, \quad (16)$$

$$\tilde{\delta}_{i,j}^{x,y} = \frac{\delta_{i,j}^{x,y}(0)}{\delta_{i,j}^{x,y}(1)} = \tilde{\beta}_{x,y} \prod_{\substack{(i',j') \\ \neq (i,j)}} \tilde{\gamma}_{i',j'}^{x,y}, \quad (17)$$

where each message is just a single number. Furthermore, (1) and (2) can be expressed as

$$P_{A_{i,j}|\mathbf{M}}(1|\mathbf{m}) = \left(1 + \tilde{\alpha}_{i,j} \prod_{(x,y)} ((\tilde{\delta}_{i,j}^{x,y})^{-1} + 1)\right)^{-1} \quad (18)$$

$$P_{b_{x,y}|\mathbf{M}}(1|\mathbf{m}) = \left(1 + \tilde{\beta}_{x,y} \prod_{(i,j)} \tilde{\gamma}_{i,j}^{x,y}\right)^{-1}. \quad (19)$$

Equations (16)-(19) take advantage of the structure of (6) and the calculation (11) is done within (18). The complete algorithm performs the following steps.

- 1) Initialization: compute (14) and (15), and initialize $\tilde{\delta}_{i,j}^{x,y}$,
- 2) Iterative processing: iteratively compute (16) and (17),
- 3) Final processing: evaluate (18) and (19).

B. Input to the algorithm

The input information to our sum-product algorithm is provided through the parameters $\tilde{\alpha}_{i,j}$ and $\tilde{\beta}_{x,y}$. In general, for any protein pair (i, j) of interest, we will have the results of zero, one, or several experiments testing for their interaction. In case of K experiments, with statistically independent false positive and false negative rates $f_{p,i,j}^{(k)}$ and $f_{n,i,j}^{(k)}$, $k = 1, \dots, K$,

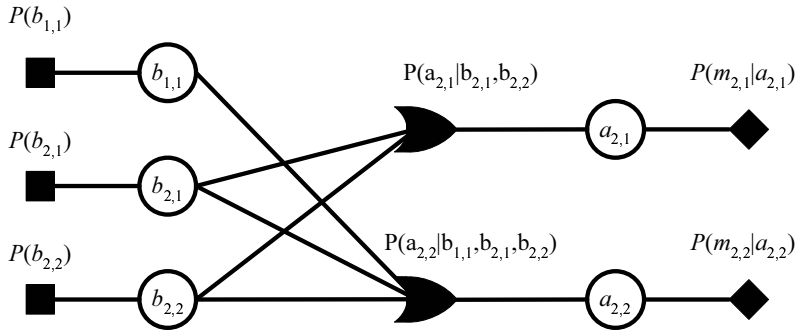


Fig. 1. Factor graph representation of the joint probability distribution function P_{ABM} . The variable nodes $a_{i,j}$ and $b_{x,y}$ are represented by circles, while factors in (3), (4), and (5) are shown as squares, OR-gate symbols, and diamonds, respectively.

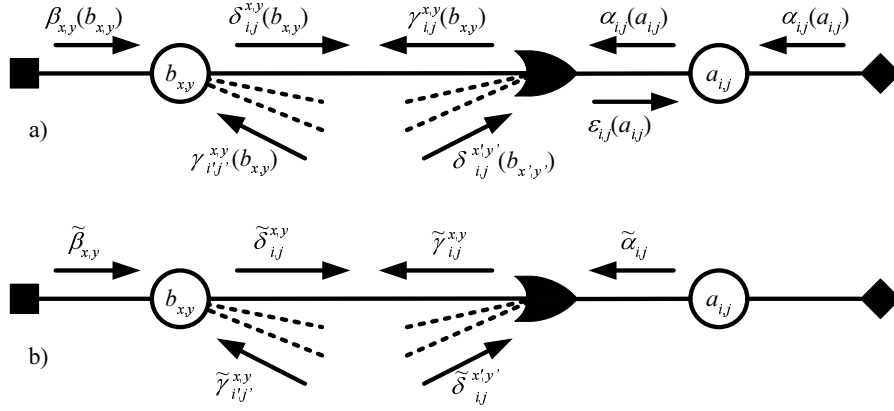


Fig. 2. Messages computed by the Sum-Product Algorithm a) before and b) after the simplification (14)-(19).

respectively, the value of the parameter $\tilde{\alpha}_{i,j}$ can be expressed as

$$\tilde{\alpha}_{i,j} = \prod_{k=1}^{K_a} \frac{P_{M_{i,j}^{(k)}|A_{i,j}}(m_{i,j}^{(k)}|0)}{P_{M_{i,j}^{(k)}|A_{i,j}}(m_{i,j}^{(k)}|1)}, \quad (20)$$

where

$$\begin{aligned} P_{M_{i,j}^{(k)}|A_{i,j}}(1|1) &= 1 - f_{n,i,j}^{(k)}, \\ P_{M_{i,j}^{(k)}|A_{i,j}}(0|1) &= f_{n,i,j}^{(k)}, \\ P_{M_{i,j}^{(k)}|A_{i,j}}(1|0) &= f_{p,i,j}^{(k)}, \\ P_{M_{i,j}^{(k)}|A_{i,j}}(0|0) &= 1 - f_{p,i,j}^{(k)}. \end{aligned}$$

With appropriate statistical models, the parameter $\tilde{\alpha}_{i,j}$ can also be calculated for measurements with correlated errors and for other evidence that exhibits statistical dependence on interaction of proteins (i, j) , but not on other protein pairs.

When no direct measurements are available for a protein pair (i, j) , the value of $\tilde{\alpha}_{i,j}$ is simply set to 1. This corresponds to the case of protein pair, for which we would like to determine the likelihood of interaction based on measurements of other pairs. In such case, inspecting the formula (16) reveals that all messages $\tilde{\gamma}_{i,j}^{x,y}$ for this protein pair are equal 1 independently of all incoming messages $\tilde{\delta}_{i,j}^{x,y}$. As a consequence, these incoming $\tilde{\delta}_{i,j}^{x,y}$ need not be computed during the iterative

evaluation of (16) and (17), and only need to be computed right before evaluating (18). Combined with the fact that, from the point of formula (17), the value $\tilde{\gamma}_{i,j}^{x,y} = 1$ is neutral, protein pairs with no direct measurements can be completely omitted in the iterative part of the SPA.

The parameter $\tilde{\beta}_{x,y}$ represents the a priori likelihood of interaction between domains (x, y) , i.e., the likelihood of interaction before the measurements \mathbf{m} are observed. If the average probability of randomly picking a domain pair that interacts is denoted by ρ , then $\tilde{\beta}_{x,y}$ can be simply set to $(1 - \rho)/\rho$. Also, any additional evidence indicative of the interaction between domains (x, y) that was obtained independently of \mathbf{m} can be provided to the algorithm through $\tilde{\beta}_{x,y}$.

C. Structure of the factor graph and the performance of SPA

A message arriving at a protein or domain pair node $a_{i,j}$ or $b_{x,y}$ through one of the adjacent edges, carries information about the interaction likelihood of this pair based on all measurements reachable through this edge. The SPA obtains the complete interaction likelihood for this pair by combining all arriving messages, treating them as independent sources of information. This approach is certainly correct if the factor graphs contains no cycles, since each edge leaving certain variable node connects it to a disjoint subgraph, and the incoming messages are based on disjoint sets of measurements. If the cycles are present, the solution generated by the algorithm

can deviate from the exact answer, with shorter cycles causing larger deviations.

The shortest possible cycles in the factor graphs are cycles of length 4. These 4-cycles arise when two protein pairs are both connected to the same two domain pairs. A significant number of 4-cycles is introduced to the graph if two or more protein pairs have an identical set of domain pairs, to which they are connected. Fortunately, this severe case can be easily eliminated by declaring such protein pairs as equivalent, and representing them by only a single variable node in the graph. If each protein pair has been independently measured, all these measurements are combined according to (20) as described in the previous subsection.

Another common source of 4-cycles are domain pairs that are connected to identical set of protein pairs. In such case, it is also possible to represent these domain pairs with a single variable node denoting the interaction of at least one of the domains, although some additional preprocessing and post-processing must be performed by the algorithm. If such domain pairs are denoted as $b_{x,y}^{(k)}$, $k = 1, \dots, K_b$, then $\tilde{\beta}_{x,y}$ for the joint node is obtained from

$$\tilde{\beta}_{x,y} = \frac{1 - \prod_{k=1}^{K_b} P_{B_{x,y}^{(k)}}(1)}{\prod_{k=1}^{K_b} P_{B_{x,y}^{(k)}}(1)}, \quad (21)$$

and the conditional likelihood of interaction is computed according to

$$P_{b_{x,y}^{(k)} | \mathbf{M}}(1 | \mathbf{m}) = \frac{P_{B_{x,y}^{(k)}}(1)}{1 + \left(\prod_{(i,j)} \tilde{\gamma}_{i,j}^{x,y} - 1 \right) \prod_{k'=1}^{K_b} (1 - P_{B_{x,y}^{(k')}}(1))}. \quad (22)$$

IV. PREDICTION ACCURACY ON SIMULATED DATA

In order to verify the performance of the algorithm under controlled conditions, we developed an *in silico* framework where, based on the IDM, we generated artificial domain-domain interactions (DDI) and as well as matching protein-protein interactions (PPI). A domain interaction rate was assigned and protein interaction measurements were simulated by adding noise to the PPI with a fixed rate of false positives (f_p) and false negative (f_n) measurements. This environment let us calculate a quantitative measure of the estimation performance of several DDI and PPI prediction algorithms. This is specially important when evaluating performance of DDI predictors, since there are no standard methods to test DDI in the laboratory. Although the interaction patterns in these simulations are random, it is important to mention that proteins and their respective domain conformations were extracted from real biological data, specifically from the Pfam database [9].

A. Prediction of Domain-Domain Interactions

The performance of our proposed algorithm was compared with two current methods of DDI estimation mentioned in Section I: MLE and MSSC. The results of this comparison are

TABLE I
INPUT PARAMETERS OF DDI/PPI PREDICTION ALGORITHMS.

Parameter	DDI Prediction (Figure 4)	PPI prediction (Figure 5)	Error Detection (Figure 6)	Cross Validation (Figure 7)
Training Set	4 / 7 / 10	3000	6000	70%
Testing Set	2	1000	-	30%
f_p	1e-4	7e-4	7e-4	7e-4
f_n	1e-4	0.65	0.65	0.65
DDI prob.	0.05	0.05	0.05	0.05
Graph Size	4 / 7 / 10	random	random	random
Iterations	1000	1000	200	10
Total Number of Proteins: 12,158				
Total Number of Domains: 14,314				

presented as the specificity vs. sensitivity curves in Figure 4. Specificity is defined as $Sp = \frac{|P \cap T^c|}{P}$ and sensitivity as $S_n = \frac{|P \cap T|}{T}$, where P is the set of positive predictions and T is the testing set of interactions we want to estimate. The plots represent an average over 1000 independently simulated graphs. Our simulation framework let us investigate how the algorithms respond to different parameters and, most importantly, what is their behavior when these parameters are not exactly known by the prediction algorithm. Table I contains a summary of the parameters used with our algorithm using experimental interaction data and simulated data.

Figure 4 shows the performance for a class of factor graphs in Figure 1 with a size expressed by the number of connected protein pairs through domain pair nodes. If the number of protein pairs in this subgraph is only one, then the MLE and SPA produce the same obvious result. Since this single protein pair subgraph is common (see Figure 3 for the distribution of graph sizes in Pfam), then it is more interesting to present results of larger graph sizes by removing the effect of predicting in the single protein pair subgraphs and focusing in specific graph sizes.

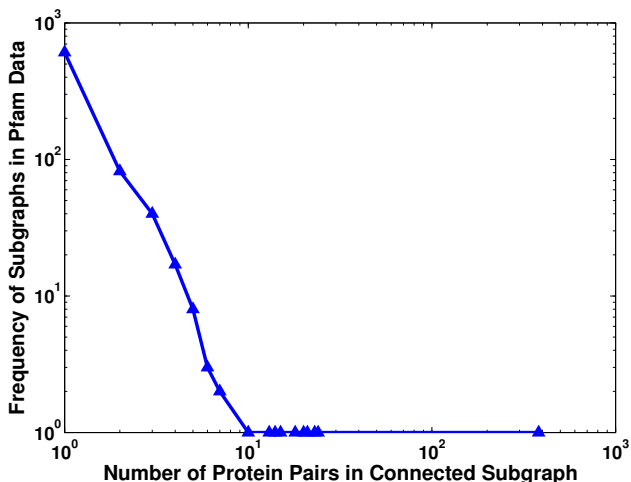


Fig. 3. Frequency distribution of the size of connected protein pairs subgraphs.

Figure 4 shows how the prediction of domain-domain in-

teractions can be improved by constructing a factor graph and calculating the marginal probabilities with the sum-product algorithm. We observe that for the case of a subgraphs, chosen according to the distribution in Figure 3, with 4,7 and 10, protein pairs connected through domain pairs, the prediction accuracy of the SPA exceeds that of MLE and MSSC for all the values in the specificity vs. sensitivity curve. Since higher specificity and sensitivity indicates more accurate prediction, a shift to the right represents a better performance of the algorithm.

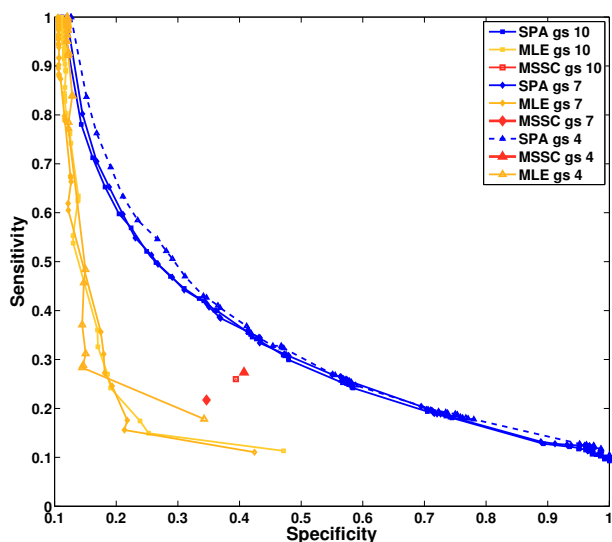


Fig. 4. Sp vs. Sn performance of prediction algorithms when estimating DDI for subgraphs of size 4, 7, 10.

The previous performance metric shows how the algorithms performed under the assumption that the prediction algorithm in fact had the information of the real domain interaction probability, however, in practice this is hard to estimate.

B. Prediction of Protein-Protein Interactions

The framework that was discussed in the previous section can be used to assess the quality of our algorithm in predicting PPI. Again, the same parameters that were discussed in the past section, affect the outcome of the PPI prediction.

Figure 5 presents specificity vs. sensitivity curves for the case when there exists a fixed rate of f_p and f_n rates that affected the measurement.

We see that our proposed method behaves better against the noise affecting the measurements, for the particular case of Figure 5, the rates are: $f_n = 0.65$ and $f_p = 0.001$, selected to resemble the estimated experimental values in [7]. It can be observed that the curve of the SPA, shown in blue, always shows a better compromise in the specificity/sensitivity points compared to the set covering method and MLE using expectation maximization. This provides evidence that Bayesian inference using the SPA provides a stronger predictor under the presence of noisy PPI measurements.

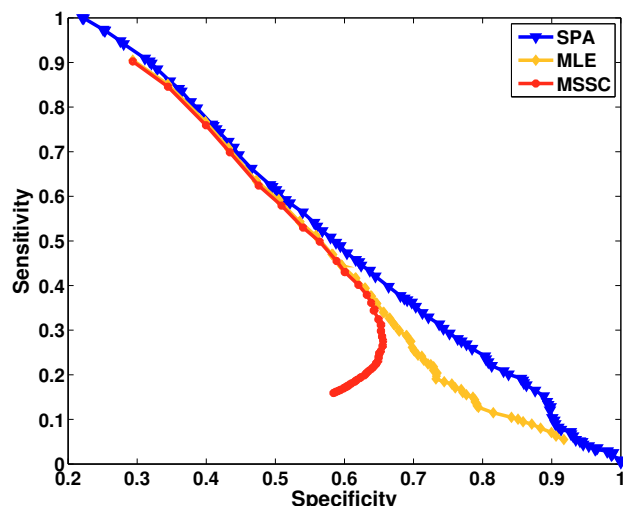


Fig. 5. Performance of predicting algorithms when estimating PPI under the presence of noisy measurements.

C. Correction of false positive and false negative measurements

We can also evaluate the capability of the PPI prediction methods to detect which measurements were in fact false positives or false negatives. This can be done by predicting interactions among protein pairs that were used as the training set in our algorithms. The results are compared with the original PPI interactions before adding noise to the measurements. Figure 6 shows the error correction capabilities of the three algorithms: MSSC, MLE and SPA.

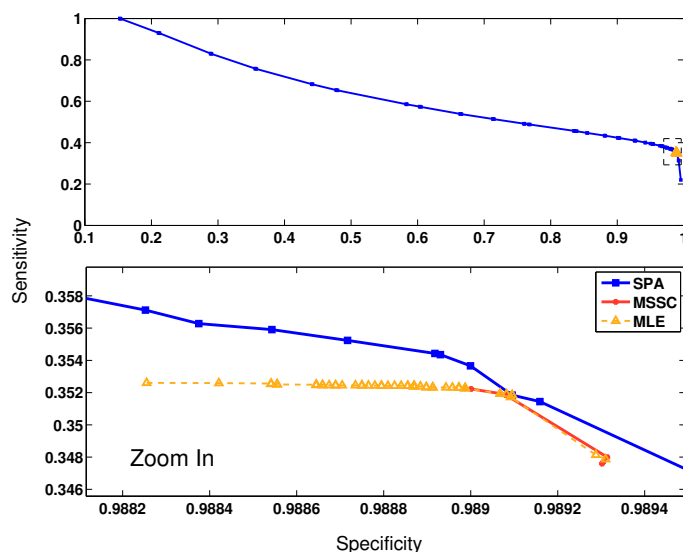


Fig. 6. False positive and false negative detection capabilities of MLE, MSSC and SPA algorithms.

The results show how Bayesian inference using the SPA is more effective detecting errors that occur in the experimental assay, a feature of the algorithm that helps to improve the quality of the input data and improve prediction of DDI and PPI.

D. Cross Validation on real interaction data

In addition to the simulation environment, we applied our estimation method to a set of real measurements of interacting protein pairs. The domain structure for the proteins in the interacting data set was retrieved from the Pfam database version 20 [9]. The experimental PPI measurements were obtained from the Database of Interacting Proteins (DIP) [10], which contains results from laboratory experiments (including high throughput methods) to detect protein interactions. In order to test our algorithm, a standard method of cross-validation was applied. Here, 30 percent of the measured interactions is used as testing set of the algorithm and the remaining disjoint set is used as training set, then the performance curves are calculated iteratively to get sufficient statistics for the performance assessment. Figure 7 shows the Sp vs Sn curves for the SPA compared to MLE and MSSC.

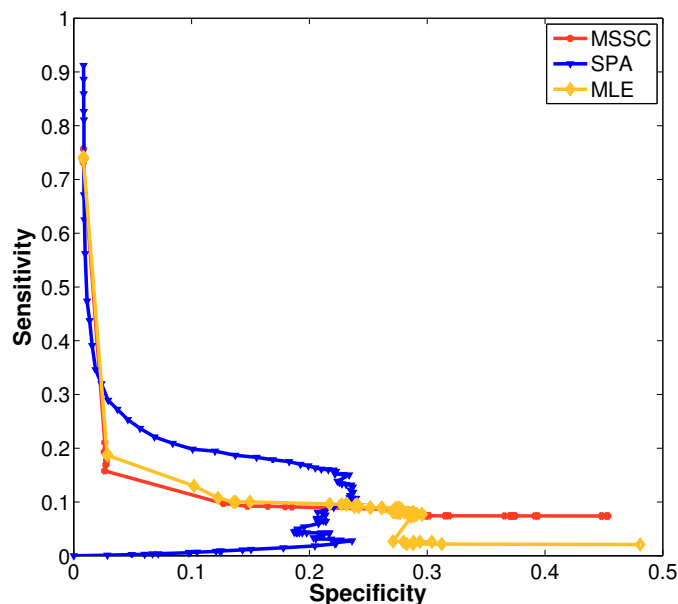


Fig. 7. Performance of predicting algorithms in experimental interaction data from DIP.

Again, we observe how SPA provides a better trade-off between specificity and sensitivity on real data. The fact that the measurements are noisy for these real data, in addition to the uncertainty of the real values of the false positive and negative rates affect the overall performance of the prediction algorithm. This lack of information affects the three presented algorithms uniformly.

V. CONCLUSIONS

We have presented a new method to predict domain-domain and protein-protein interactions based on the concept of Bayesian inference and implemented via the Sum-Product Algorithm. The contributions of this paper are twofold: We provide a new representation of prediction DDI and PPI based on factor graphs, and a framework to efficiently and accurately predict DDI and PPI based on a message passing algorithm. This framework allows to build a probabilistic DDI network and predict new potential PPI interactions based on that information. In addition, the presented method is able to detect false positives and false negatives that are common in the experimental assays. The present methodology can be used to predict, analyze and understand domain and protein interaction networks in different organisms. This knowledge has important implications in the understanding of the dynamic behavior of molecular interactions in the cell.

REFERENCES

- [1] D. Eisenberg, E. M. Marcotte, I. Xenarios, and T. O. Yeates, "Protein function in the post-genomic era," *Nature*, vol. 405, pp. 823–826, 2000.
- [2] S. Li, C. M. Armstrong, N. Bertin, H. Ge, S. Milstein, M. Boxem, P.-O. Vidalain, J.-D. J. Han, A. Chesneau, T. Hao, D. S. Goldberg, N. Li, M. Martinez, J.-F. Rual, P. Lamesch, L. Xu, M. Tewari, S. L. Wong, L. V. Zhang, G. F. Berriz, L. Jacotot, P. Vaglio, J. Reboul, T. Hirozane-Kishikawa, Q. Li, H. W. Gabel, A. Elewa, B. Baumgartner, D. J. Rose, H. Yu, S. Bosak, R. Sequerra, A. Fraser, S. E. Mango, W. M. Saxton, S. Strome, S. van den Heuvel, F. Piano, J. Vandenhaute, C. Sardet, M. Gerstein, L. Doucette-Stamm, K. C. Gunsalus, J. W. Harper, M. E. Cusick, F. P. Roth, D. E. Hill, and M. Vidal, "A Map of the Interactome Network of the Metazoan *C. elegans*," *Science*, vol. 303, no. 5657, pp. 540–543, 2004.
- [3] L. Giot, J. S. Bader, C. Brouwer, A. Chaudhuri, B. Kuang, Y. Li, Y. L. Hao, C. E. Ooi, B. Godwin, E. Vitols, G. Vijayadamar, P. Pochart, H. Machineni, M. Welsh, Y. Kong, B. Zerhusen, R. Malcolm, Z. Varrone, A. Collis, M. Minto, S. Burgess, L. McDaniel, E. Stimpson, F. Sardet, J. Williams, K. Neurath, N. Ioime, M. Agee, E. Voss, K. Furtak, R. Renzulli, N. Aanensen, S. Carrolla, E. Bickelhaupt, Y. Lazovatsky, A. DaSilva, J. Zhong, C. A. Stanyon, J. Finley, R. L., K. P. White, M. Braverman, T. Jarvie, S. Gold, M. Leach, J. Knight, R. A. Shinkets, M. P. McKenna, J. Chant, and J. M. Rothberg, "A Protein Interaction Map of *Drosophila melanogaster*," *Science*, vol. 302, no. 5651, pp. 1727–1736, 2003.
- [4] J.-F. Rual, K. Venkatesan, T. Hao, T. Hirozane-Kishikawa, A. Dricot, N. Li, G. F. Berriz, F. D. Gibbons, M. Dreze, N. Ayivi-Guedehoussou, N. Klitgord, C. Simon, M. Boxem, S. Milstein, J. Rosenberg, D. S. Goldberg, L. V. Zhang, S. L. Wong, G. Franklin, S. Li, J. S. Albala, J. Lim, C. Fraughton, E. Llamasas, S. Cevik, C. Bex, P. Lamesch, R. S. Sikorski, J. Vandenhaute, H. Y. Zoghbi, A. Smolyar, S. Bosak, R. Sequerra, L. Doucette-Stamm, M. E. Cusick, D. E. Hill, F. P. Roth, and M. Vidal, "Towards a proteome-scale map of the human protein-protein interaction network," *Nature*, vol. 437, pp. 1173–1178, 2005.
- [5] 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. Vijayadamar, M. Yang, M. Johnston, S. Fields, and J. M. Rothberg, "A comprehensive analysis of protein-protein interactions in *saccharomyces cerevisiae*," *Nature*, vol. 403, pp. 623–627, 2000.
- [6] T. Ito, T. Chiba, R. Ozawa, M. Yoshida, M. Hattori, and Y. Sakaki, "A comprehensive two-hybrid analysis to explore the yeast protein interactome," *PNAS*, vol. 98, no. 8, pp. 4569–4574, 2001.
- [7] M. Deng, S. Mehta, F. Sun, and T. Chen, "Inferring domain-domain interactions from protein-protein interactions," *Genome Res*, vol. 12, pp. 1540–1548, Oct 2002.
- [8] R. Riley, C. Lee, C. Sabatti, and D. Eisenberg, "Inferring protein domain interactions from databases of interacting proteins," *Genome Biol*, vol. 6, no. 10, p. R89, 2005.

- [9] 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, pp. D138–D141, Jan 2004.
- [10] 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, pp. 303–305, Jan 2002.
- [11] 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 Transactions on Computational Biology and Bioinformatics (in print)*, vol. 4, no. 1, pp. 1–10, 2007.
- [12] T. Moon, "The expectation-maximization algorithm," *Signal Processing Magazine, IEEE*, vol. 13, pp. 47–60, Nov. 1996.
- [13] F. Kschischang, B. Frey, and H.-A. Loeliger, "Factor graphs and the sum-product algorithm," *Information Theory, IEEE Transactions on*, vol. 47, pp. 498–519, Feb 2001.
- [14] S.-Y. Chung, T. J. Richardson, and R. L. Urbanke, "Analysis of sum-product decoding of low-density parity-check codes using a gaussian approximation," *Information Theory, IEEE Transactions on*, vol. 47, pp. 657–670, Feb 2001.