

A Simple Method for Inferring Strengths of Protein-Protein Interactions

Morihiro Hayashida*

morihiro@kuicr.kyoto-u.ac.jp

Nobuhisa Ueda

ueda@kuicr.kyoto-u.ac.jp

Tatsuya Akutsu

takutsu@kuicr.kyoto-u.ac.jp

Bioinformatics Center, Institute for Chemical Research, Kyoto University, Gokasho, Uji 611-0011, Japan

Abstract

Various computational methods have been proposed for inference of protein-protein interactions since protein-protein interaction plays an essential role in many cellular processes. One of well-studied approaches is to infer protein-protein interactions based on domain-domain interactions. To extend this approach, we proposed a method called LPNM to infer ratios of interactions, which outperformed other existing methods in terms of error of predicted ratios. However, since the LPNM method is based on the linear programming approach, it may require a large amount of time to infer interactions for a large data set.

In this paper, we propose a simple method to infer the ratios of protein-protein interactions based on the association method by Sprinzak et al. In an experiment with a data set of protein-protein interactions in yeast, it runs more than 150 times as fast as the LPNM method, and achieves almost the same accuracy.

On implementing algorithms for the inference problem, it is essential to understand how difficult the problem is. Even though various methods for the problem have been already proposed, it has not been analyzed rigorously from a computational point of view. We hence define a problem to maximize correctly classified examples, and prove the problem is MAX SNP-hard, which also means the problem is NP-hard.

Keywords: protein-protein interaction, association method, domain-domain interaction, MAX SNP-hard

1 Introduction

As sequences for the whole genome of a considerable number of organisms have become available, many researchers have paid attention to understanding functions of genes and proteins. Information about protein-protein interaction is indispensable for understanding protein functions since protein-protein interaction plays a fundamental role in many cellular processes such as regulation of transcription and translation, signal transduction, and recognition of foreign molecules. To understand protein-protein interaction comprehensively, large-scale two-hybrid systems have been developed for analyzing protein-protein interactions in *Saccharomyces cerevisiae* (budding yeast) [10, 11, 21], and many unknown interactions were observed in experiments with the systems. Unfortunately, two independent experiments in [10, 11] and in [21] do not share many common interactions between proteins. The results suggest that there remains much to be done for understanding protein-protein interactions. One way to overcome this issue is a computational method for inferring protein-protein interactions from various kinds of data.

*To whom correspondence should be addressed.

Several diverse computational approaches have been proposed for inference of protein-protein interactions so far. For example, the gene fusion/Rosetta stone method finds pairs of proteins each of which putatively interact if each of them is encoded separately as a distinct gene in an organism, and they are fused in another organism [6, 16]. An algorithm was proposed that predicts interactions between proteins with multiple source of data such as proteins evolved in a correlated fashion and correlated messenger RNA expression patterns [17]. A probabilistic model was proposed to form a network of protein-protein interactions based on probabilities of interactions (attractions and repulsions) between domains [7]. Another probabilistic model called the hierarchical class model was proposed to predict protein-protein interactions from information on protein classes [15]. Support vector machine [3] was applied to inference of protein-protein interactions [2]. A prediction method called MULTIPROSPECTOR was proposed based on a threading algorithm, and is able to identify the residues that participate directly in an interaction between proteins [14].

Recently, some methods have been proposed for inferring domain-domain interactions (and/or signature-signature interactions) from protein-protein interaction data. Information of domain-domain interaction is favorable not only for more detailed understanding of protein-protein interactions but also for inferring protein-protein interactions: a protein pair is expected to interact if a domain in a protein interacts with a domain in the other protein. From this observation, Sprinzak and Margalit proposed a method called the association method for estimating the score for each domain pair [20]. One extension of the association method considers that interactions between domain combinations also contribute to the interactions between proteins containing the domains [12]. A probabilistic model and an algorithm were proposed to estimate parameters (probabilities of interacting two domains) based on an EM (expectation-maximization) algorithm [5]. Note that these methods consider only whether any protein pair interacts or not.

On the other hand, a data set of ratios of interaction between two proteins becomes accessible since multiple experiments were performed recently for the same protein pairs in practice [10, 11, 21], and the ratios can be more informative than binary data (whether two proteins interact or not). A data set of the ratios is hereafter called a *numerical data set* for notational convenience. We introduced a concept of *strength* of protein-protein interactions, which corresponded to the ratio, and proposed a linear programming-based method (LPNM method) for inference of strengths of protein-protein interactions [8]. It outperformed existing methods, that is, the EM method [5] and the association method when they were applied to a numerical data set. However, since the LPNM method uses linear programming to infer the strengths, it may require a large amount of time for a large data set.

We therefore propose a new method for numerical data set which runs much faster than the LPNM method. It is an extension of the association method, which infers whether two proteins interact or not based on domain-domain interactions. It runs more than 150 times as fast as the LPNM method, and achieves almost the same accuracy in an experiment with a data set of protein-protein interactions in yeast.

On implementing algorithms for the inference problem, it is essential to understand how difficult the problem is. Even though various methods have been already proposed, all of them can be seen as heuristics from an algorithmic point of view. This suggests that it should be difficult in nature to minimize the errors of predicted strengths of protein-protein interactions. To verify the difficulty of inferring the strengths, as the first step, we prove in this paper that the problem to maximize correctly classified examples of protein-protein interactions is MAX SNP-hard. Although protein-protein interactions have already been formalized with the Markov random fields [4, 13], on which it is NP-hard to find the likelihood, our formalization of the problem and its proof take different approaches. This new formalization may shed light on detailed computational analysis of protein-protein interactions. Note that, if a problem is MAX SNP-hard, it is also NP-hard. For details of MAX SNP, see [19, 22].

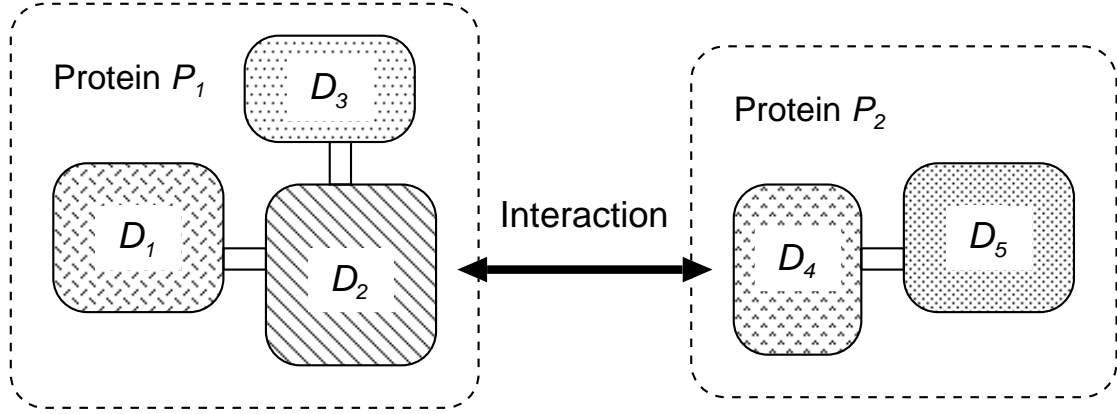


Figure 1: Inference of protein-protein interactions through domain-domain interactions. In this case, we infer that proteins P_1 and P_2 interact with each other since domains D_2 and D_4 interact with each other.

2 Preliminaries

We briefly review the probabilistic model of protein-protein interactions [5] which we will follow in this paper, and methods to infer interactions between proteins [20, 8]. We call this probabilistic model the *original probabilistic model* for reference.

2.1 Probabilistic Model of Protein-Protein Interactions

Let P_1, \dots, P_N be proteins. We also use P_i to denote a set of domains in P_i . Let D_1, \dots, D_M be domains in proteins P_1, \dots, P_N . For notational convenience, P_{ij} and D_{mn} represent the protein pair (P_i, P_j) and the domain pair (D_m, D_n) respectively. Let \mathcal{P} be a multi set of protein pairs P_{ij} . We also use P_{ij} to denote a set of domain pairs between P_i and P_j (i.e., $P_{ij} = \{D_{mn} | D_m \in P_i, D_n \in P_j\}$).

In this probabilistic model, an interaction between P_i and P_j (one between D_m and D_n) is represented as a random variable $P_{ij}(D_{mn})$. P_{ij} takes 1 if P_i and P_j interact with each other, otherwise $P_{ij} = 0$. In the same manner, $D_{mn} = 1$ if D_m and D_n interact with each other, otherwise $D_{mn} = 0$. This probabilistic model assumes that domain-domain interactions are independent and two proteins interact if and only if at least one domain pairs from the two proteins interact (see Figure 1). Under this assumption, the probability that P_i and P_j interact with each other is given by

$$\Pr(P_{ij} = 1) = 1 - \prod_{D_{mn} \in P_{ij}} (1 - \lambda_{mn}),$$

where λ_{mn} denotes the probability that D_m and D_n interact with each other (i.e., $\lambda_{mn} = \Pr(D_{mn} = 1)$).

2.2 Association Method

The association method [20] is based on binary interaction data, and assigns a score to each domain pair (D_m, D_n) . Let N_{mn} be the number of protein pairs (in the training data set) containing domain pairs (D_m, D_n) . Let I_{mn} be the number of interacting protein pairs (in the training data set) containing domain pairs (D_m, D_n) . The score (probability of interactions) for (D_m, D_n) is defined as

$$ASSOC(D_m, D_n) = \frac{I_{mn}}{N_{mn}}.$$

2.3 LPNM: LP-based Method for Numerical Interaction Data

The LPNM method [8] transforms the original probabilistic model into a set of linear constraints, and finds parameters related to interactions between domains by minimizing an objective function. One notable aspect of the LPNM method is that interactions between proteins are represented as ratios in a numerical data set.

We set ρ_{ij} to be the ratio of interactions between proteins P_i and P_j in a series of experiments, that is,

$$\rho_{ij} = \frac{N(P_{ij})}{Z},$$

where $N(P_{ij})$ is the number of times that the interaction between proteins P_i and P_j is observed in the experiments, and Z is the total number of the experiments.

Since ρ_{ij} is the ratio of interactions between P_i and P_j , we consider here to minimize the difference between $\Pr(P_{ij} = 1)$ and ρ_{ij} , in other words, between the probability of observing an interaction in the original probabilistic model and the ratio of the interactions observed in the experiments.

When $\Pr(P_{ij} = 1)$ and ρ_{ij} are equivalent, the following holds:

$$\sum_{D_{mn} \in P_{ij}} \ln(1 - \lambda_{mn}) = \ln(1 - \rho_{ij}).$$

For each P_{ij} and any $D_{mn} \in P_{ij}$, we can rewrite this equation with auxiliary variables γ_{mn} and β_{ij} :

$$\sum_{D_{mn} \in P_{ij}} \gamma_{mn} = \beta_{ij},$$

where $\gamma_{mn} = \ln(1 - \lambda_{mn})$ and $\beta_{ij} = \ln(1 - \rho_{ij})$. If we have γ_{mn} for any m and n satisfying the above equations, we can obtain parameters for domain-domain interactions consistent with a numerical interaction data set.

These equations, however, do not always hold. It is hence reasonable to try to minimize the summation of the differences

$$\sum_{P_{ij} \in \mathcal{P}} \left| \sum_{D_{mn} \in P_{ij}} \gamma_{mn} - \beta_{ij} \right|.$$

We therefore use the following linear program to minimize the sum:

$$\begin{aligned} & \text{minimize} && \sum_{P_{ij} \in \mathcal{P}} \alpha_{ij} \\ & \text{subject to} && \left\{ \begin{array}{l} \sum_{D_{mn} \in P_{ij}} \gamma_{mn} - \beta_{ij} \leq \alpha_{ij} \\ \beta_{ij} - \sum_{D_{mn} \in P_{ij}} \gamma_{mn} \leq \alpha_{ij} \end{array} \right. \quad (\text{for all } P_{ij}) \\ & && \gamma_{mn} \leq 0 \quad (\text{for all } \gamma_{mn}) \\ & && \alpha_{ij} \geq 0 \quad (\text{for all } \alpha_{ij}) \end{aligned}$$

where each β_{ij} is a non-positive constant obtained from experiments of interactions of the protein pair (P_i, P_j) .

3 Simple Method for Numerical Interaction Data

We propose a new simple method for inferring the strength of protein-protein interactions, which we call the ASNM method. This method is derived by extending the association method [20] (for binary

interaction data) into one for numerical interaction data. The association method uses the number of interacting protein pairs (I_{mn}) to infer the score (probability of interaction) for (D_m, D_n) . In the ASNM method, we use the summation of the strengths (ρ_{ij}) of interaction between P_i and P_j instead of I_{mn} , where the protein pair (P_i, P_j) includes the target domain pair (D_m, D_n) . We then define the score $ASNM(D_m, D_n)$ for (D_m, D_n) as

$$ASNM(D_m, D_n) = \frac{1}{N_{mn}} \sum_{P_{ij}: D_{mn} \in P_{ij}} \rho_{ij}.$$

Recall that N_{mn} is the number of protein pairs containing domain pairs (D_m, D_n) . If the ratio ρ_{ij} for each protein pair (P_i, P_j) always takes either 0 or 1, $ASNM(D_m, D_n)$ becomes equivalent to the score $ASSOC(D_m, D_n)$ in the association method because it holds that $I_{mn} = \sum_{P_{ij}: D_{mn} \in P_{ij}} \rho_{ij}$.

4 Experimental Results

4.1 Data and Implementation

We compared the ASNM method with the LPNM method and the association method. For the training and test data of strengths of protein-protein interactions, we used the full data of Ito’s Yeast Interacting Proteins (YIP) database [10, 11]. The YIP database provides numerical interaction data for pairs of proteins in terms of IST (Interaction Sequence Tag pairs).

For each protein in the database, we obtained its sequence data from the Swissprot/TrEMBL database [1]. In order to derive domains from the sequences, we used InterProScan (version 3.1) [24] as in [12, 20]. Though InterProScan identified not only protein domains but also protein signatures such as functional sites and sequence motives, we used all domains and signatures because the signatures may also play an important role in protein-protein interaction. As in [12, 20], InterPro signatures in the same parent-child relationship were also merged into a single signature. The sequence and signature pairs we used in the experiment are available from <http://sunflower.kuicr.kyoto-u.ac.jp/~morihiro/protint/supplement.html>.

We used `lp_solve` (version 4.0) on Linux for solving linear programs. The experiments were mostly performed on a PC cluster with 8 Xeon 2.8 GHz processors, where only one CPU was used in all experiments.

We evaluated the methods by root mean squared error (RMSE) between the predicted probability $\Pr(P_{ij} = 1)$ and the observed ratio ρ_{ij} from the YIP database. Precisely, for a set \mathcal{P} of protein pairs,

$$RMSE = \sqrt{\frac{1}{|\mathcal{P}|} \sum_{P_{ij} \in \mathcal{P}} (\Pr(P_{ij} = 1) - \rho_{ij})^2}.$$

We used 1,586 interacting pairs of proteins, and employed five-fold cross validation. In addition, we estimated the time complexity of each method by measuring the elapsed time.

4.2 Results

Table 1 shows the root mean squared errors for training and test data sets and the average elapsed time for training data sets using ASNM, LPNM and the association method. Since we employed five-fold cross validation and split the data set into five blocks, the k -th row means that the k -th block among five blocks of the data is used as a test data set.

We see from the table that the errors of both ASNM and LPNM for test data sets are quite similar, and much smaller than the association method. The average error of ASNM for test data sets are slightly worse than that of LPNM, and that of LPNM for training data sets are quite smaller than that of ASNM. This suggests that LPNM may overfit in this case. Errors for training data sets are smaller

Table 1: Root mean squared errors and average training elapsed time for numerical interaction data.

		ASNM		LPNM		ASSOC	
		Train	Test	Train	Test	Train	Test
Error	1st	0.0365687	0.0408624	0.0103880	0.0312939	0.452380	0.315208
	2nd	0.0381153	0.0480632	0.0145225	0.0329882	0.455613	0.308925
	3rd	0.0429533	0.0471907	0.0143729	0.0347589	0.455444	0.290413
	4th	0.0397846	0.0356935	0.0141168	0.0282775	0.453617	0.241639
	5th	0.0424590	0.0306575	0.0140418	0.0266282	0.467038	0.227669
Average		0.0399762	0.0404935	0.0134884	0.0307893	0.456818	0.276771
Time	(sec)	0.0077122	-	1.203068	-	0.0088252	-

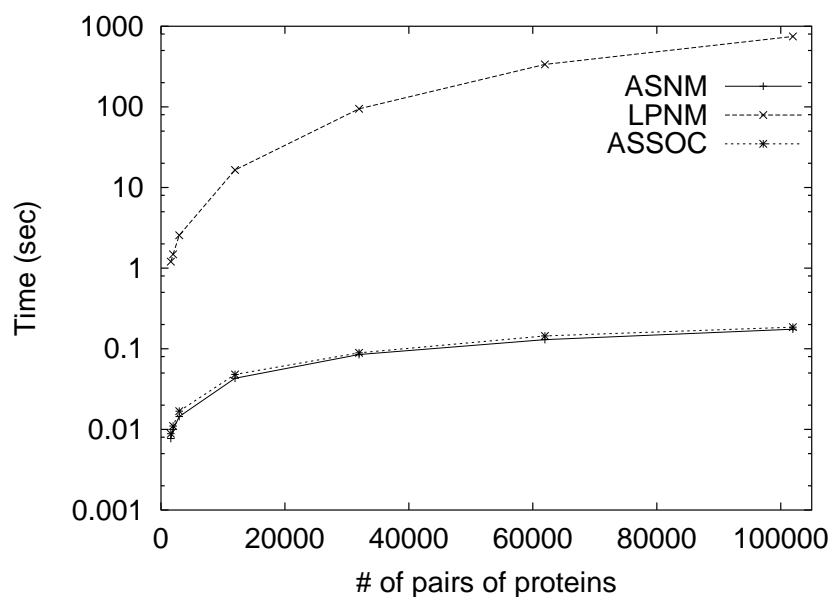


Figure 2: Elapsed time (log scale) for training in ASNM, LPNM and the association method. X-axis shows the number of input data sets, pairs of proteins. Y-axis shows the logarithm of elapsed time.

than those for test data sets generally. However, the errors of the association method for training data sets are larger than those for test data sets. It is considered because the errors are calculated for their strengths, the association method uses just binary data (whether or not each pair of proteins interacts), and does not use those strengths.

In the average time, it is seen that both ASNM and the association method are much faster than the LPNM method. Figure 2 shows elapsed time of training for ASNM, LPNM and the association method. It is seen from the figure that the elapsed times of ASNM and the association method are much smaller than that of LPNM, and that of LPNM increases more steeply than those of ASNM and the association method when the number of input data sets increases.

To see the distributions of errors for the methods, we plotted the number of proteins according to the error between the ratio in the data set and predicted one in Figure 3. It shows the average frequencies of probability errors of protein-protein interactions for the test data during the cross validation by ASNM, LPNM and the association method respectively. It is seen also from this figure that the ASNM method performs similarly as the LPNM method. Their errors were distributed around zero, whereas the errors of the association method were distributed more widely. It is reasonable

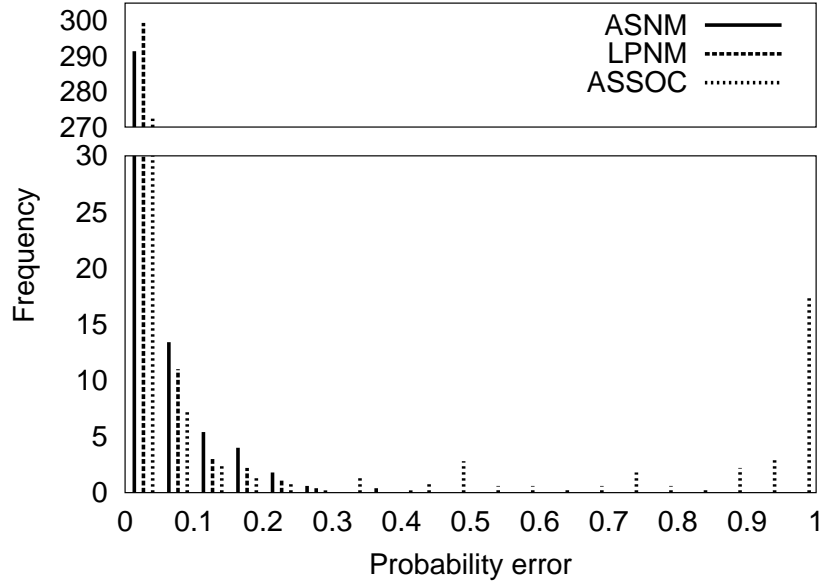


Figure 3: Distributions of probability errors for ASNM, LPNM and the association method. Y-axis shows the number of interacting protein pairs for which the errors (between the predicted probabilities and the observed probabilities) are within the specified range. The average numbers over 5 test data sets are shown. We omit the range of frequencies between 30 and 270.

because the association method uses either 0 or 1 as probabilities of interactions instead of strengths ρ_{ij} in the ASNM method.

5 Hardness of Inferring Protein-Protein Interactions

In the previous section, we proposed a practical method of inferring numerical interaction data. In this section, we consider complexity of an inference problem for *binary* interaction data. First, we define the problem as a maximization problem.

Following the definition of the original model [5], we introduce a parameter Θ as a threshold for predicting protein-protein interactions. With Θ , we predict protein-protein interactions by the following rule:

$$P_i \text{ and } P_j \text{ interact} \iff \Pr(P_{ij} = 1) = 1 - \prod_{D_{mn} \in P_{ij}} (1 - \lambda_{mn}) \geq \Theta.$$

Problem 1 (MAX PPI) Let \mathcal{P}_{pos} and \mathcal{P}_{neg} be a multi set of interacting protein pairs (P_i, P_j) and a multi set of non-interacting protein pairs respectively. Let λ_{mn} denote the probability that domains D_m and D_n interact. We consider two types of inequalities,

$$\begin{aligned} \prod_{D_{mn} \in P_{ij}} (1 - \lambda_{mn}) &\leq 1 - \Theta && \text{if a protein pair } (P_i, P_j) \text{ is in } \mathcal{P}_{\text{pos}}, \\ \prod_{D_{mn} \in P_{ij}} (1 - \lambda_{mn}) &> 1 - \Theta && \text{if a protein pair } (P_i, P_j) \text{ is in } \mathcal{P}_{\text{neg}}. \end{aligned}$$

Given \mathcal{P}_{pos} , \mathcal{P}_{neg} , and sets P_1, \dots, P_N of domains, find the parameters λ_{mn} and Θ to maximize the number of the inequalities that are satisfied.

We will show that MAX PPI is in the class of intractable problems called MAX SNP-hard. For a MAX SNP-hard problem, it is impossible to approximate with an arbitrary ratio in polynomial time unless $P=NP$. For proving the hardness of MAX PPI, it is sufficient to show that there is an L -reduction from MAX 2SAT- B [19], known as a MAX SNP-complete problem, to MAX PPI. The L -reduction is a very restricted form of transformation, and preserves approximability [19].

Problem 2 (MAX 2SAT- B) Consider a set I of m clauses C_1, \dots, C_m where each clause contains up to two literals, over a set of Boolean variables x_1, \dots, x_n . Given I , find a truth assignment that maximizes the number of clauses evaluated true.

Definition 1 (L-reduction) Let Π and Π' be two optimization problems. We say that Π L -reduces to Π' if there are two polynomial-time algorithms f, g , and constants $\alpha, \beta > 0$ such that for each instance I of Π :

- (a) Algorithm f produces an instance $I' = f(I)$ of Π' , such that the optima of I and I' , $OPT(I)$ and $OPT(I')$, respectively, satisfy $OPT(I') \leq \alpha OPT(I)$.
- (b) Given any solution of I' with cost c' , algorithm g produces a solution of I with cost c such that $|c - OPT(I)| \leq \beta |c' - OPT(I')|$.

The following theorem concerning a property of MAX SNP class takes a key role in the proof of the hardness of MAX PPI.

Theorem 1 ([19]) Every problem in MAX SNP can be approximated within some fixed ratio.

Theorem 2 MAX PPI is in MAX SNP-hard.

Proof In order to show there is an L -reduction from MAX-2SAT- B to MAX PPI, we will construct algorithms f and g satisfying the conditions of L -reductions.

Consider a fixed instance I of MAX 2SAT- B , that is, $I = C_1 \wedge \dots \wedge C_m$, where $C_k = l_{k,1} \vee l_{k,2}$, and each of $l_{k,1}$ and $l_{k,2}$ denotes a literal, i.e., one of variables x_1, \dots, x_n and their negations $\bar{x}_1, \dots, \bar{x}_n$. For this I , we prepare a set \mathcal{P} of proteins s.t. $\mathcal{P} = \{P_1, \dots, P_{m+n}, P_{m+n+1}\}$, and the algorithm f generates, as an instance I' of MAX PPI,

$$\begin{aligned} \mathcal{P}_{\text{pos}} &= \{(P_1, P_{m+n+1}), (P_2, P_{m+n+1}), \dots, (P_m, P_{m+n+1})\}, \\ \mathcal{P}_{\text{neg}} &= \left\{ \underbrace{(P_{m+1}, P_{m+n+1}), \dots, (P_{m+1}, P_{m+n+1})}_{2B \text{ times}}, \underbrace{(P_{m+2}, P_{m+n+1}), \dots, (P_{m+2}, P_{m+n+1})}_{2B \text{ times}}, \dots, \right. \\ &\quad \left. \underbrace{(P_{m+n}, P_{m+n+1}), \dots, (P_{m+n}, P_{m+n+1})}_{2B \text{ times}} \right\}, \\ P_k &= \begin{cases} \{D_{l_{k,1}}, D_{l_{k,2}}\} & 1 \leq k \leq m, \\ \{D_{x_{k-m}}, D_{\bar{x}_{k-m}}\} & m+1 \leq k \leq m+n, \\ \{D_\alpha\} & k = m+n+1. \end{cases} \end{aligned}$$

From \mathcal{P}_{pos} and \mathcal{P}_{neg} , we have the following inequalities:

$$\begin{aligned} (1 - \lambda_{l_{k,1}\alpha})(1 - \lambda_{l_{k,2}\alpha}) &\leq 1 - \Theta \quad \text{if } (P_k, P_{m+n+1}) \in \mathcal{P}_{\text{pos}}, \\ (1 - \lambda_{x_k\alpha})(1 - \lambda_{\bar{x}_k\alpha}) &> 1 - \Theta \quad \text{if } (P_{m+k}, P_{m+n+1}) \in \mathcal{P}_{\text{neg}}. \end{aligned}$$

Note that we have $2B$ inequalities $(1 - \lambda_{x_k\alpha})(1 - \lambda_{\bar{x}_k\alpha}) > 1 - \Theta$ for each protein pair (P_{m+k}, P_{m+n+1}) in \mathcal{P}_{neg} .

Suppose that we have optimal values of λ_{mn} and Θ . To compare between the optima $OPT(I)$ and $OPT(I')$, we assign a Boolean auxiliary variable θ_{l_i} for each literal l_i as follows:

$$\theta_{l_i} = \begin{cases} \text{true} & \text{if } 1 - \lambda_{l_i\alpha} \leq \sqrt{1 - \Theta}, \\ \text{false} & \text{otherwise.} \end{cases}$$

The followings hold for $k = 1, \dots, m$ and $k' = 1, \dots, n$ from the definition of θ_{l_k} :

$$\begin{aligned} \theta_{l_{k,1}} \vee \theta_{l_{k,2}} & \text{ if } (1 - \lambda_{l_{k,1}\alpha})(1 - \lambda_{l_{k,2}\alpha}) \leq 1 - \Theta, \\ \theta_{x_{k'}} \wedge \theta_{\bar{x}_{k'}} & \text{ if } (1 - \lambda_{x_{k'}\alpha})(1 - \lambda_{\bar{x}_{k'}\alpha}) > 1 - \Theta. \end{aligned}$$

For each literal l_i , the optimum solution of I' must satisfy $2B$ inequalities generated from \mathcal{P}_{neg} because $\lambda_{l_i\alpha}$ s appear at most B times in inequalities from \mathcal{P}_{pos} . It follows that either θ_{x_i} or $\theta_{\bar{x}_i}$ true. Then, m clauses which consist of θ_{x_i} s from \mathcal{P}_{pos} in I' become equivalent to m clauses of I . Therefore,

$$OPT(I') = OPT(I) + 2Bn, \quad (1)$$

From the Theorem 1, $OPT(I)$ has at least a constant fraction of m such that $OPT(I) \geq m/\alpha_1$. Since the number of variables is less than twice the number of clauses, it holds that $n \leq 2m$ and $n \leq 2\alpha_1 OPT(I)$. Substituting n of the equation (1) into this, we have

$$OPT(I') \leq (1 + 4B\alpha_1)OPT(I).$$

It follows that f satisfies the condition (a) in the definition of L -reduction with the constant $\alpha = 1 + 4B\alpha_1$.

Next, we show that the algorithm g we will construct satisfies the condition (b) in L -reduction. Recall that, given the solution of I' with cost c' , the function g has to produce the solution of I with cost c . In this case, given solutions of I and I' , costs of I and I' correspond to the number of clauses satisfied and the number of inequalities that hold, respectively. We can obtain truth assignments of θ_{x_i} s from the solution of I' in the previous manner. g assigns either true or false to each x_i on the basis of both assignments of θ_{x_i} and $\theta_{\bar{x}_i}$. And for each i of x_i and θ_{x_i} , we evaluate difference of costs denoting Δ_i when we replace θ_{x_i} and $\theta_{\bar{x}_i}$ with x_i and \bar{x}_i respectively. $c - c' = \sum_{i=1}^n \Delta_i$.

- (i) When $\theta_{x_i} = \text{true}$ and $\theta_{\bar{x}_i} = \text{true}$, g assigns true to x_i (and false to \bar{x}_i). The cost c then decreases by at most the number of appearances of \bar{x}_i , which is less than or equal to B . This results in that $\Delta_i \geq -B$.
- (ii) When $\theta_{x_i} = \text{true}$ and $\theta_{\bar{x}_i} = \text{false}$, g assigns true to x_i . Since, for each i , I' contains $2B$ inequalities that hold of the form $(1 - \lambda_{x_i\alpha})(1 - \lambda_{\bar{x}_i\alpha}) > 1 - \Theta$, then $\Delta_i = -2B$.
- (iii) When $\theta_{x_i} = \text{false}$ and $\theta_{\bar{x}_i} = \text{true}$, g assigns false to x_i . Similarly, $\Delta_i = -2B$.
- (iv) When $\theta_{x_i} = \text{false}$ and $\theta_{\bar{x}_i} = \text{false}$, g assigns true to x_i . Then, c increases by at most the appearances of x_i . It follows that $\Delta_i \geq 0$.

Therefore,

$$\begin{aligned} c - c' &= \sum_{i=1}^n \Delta_i \geq -2Bn \\ \Leftrightarrow c &\geq c' - 2Bn \\ \Leftrightarrow c - OPT(I') &\geq c' - 2Bn - OPT(I') \\ \Leftrightarrow c - OPT(I) &\geq c' - OPT(I') && \text{(from the equation (1))} \\ \Leftrightarrow |c - OPT(I)| &\leq |c' - OPT(I')| && \text{(because } c - OPT(I) \leq 0) \end{aligned}$$

The condition (b) of the L -reduction is satisfied with $\beta = 1$. In consequence of the properties of f and g , MAX PPI is MAX SNP-hard. \blacksquare

Theorem 2 implies directly that MAX PPI is NP-hard.

Incidentally, MAX PPI can be similar to a known NP-complete problem called the induction of oblique decision trees (IODT) [9, 18] when we take another look at MAX PPI from a different point of view. By taking logarithms of both sides of inequalities in MAX PPI, we have the following linear inequalities,

$$\begin{aligned} \sum_{D_{mn} \in P_{ij}} \gamma_{mn} &\leq \beta && \text{if } (P_i, P_j) \in \mathcal{P}_{\text{pos}}, \\ \sum_{D_{mn} \in P_{ij}} \gamma_{mn} &> \beta && \text{if } (P_i, P_j) \in \mathcal{P}_{\text{neg}}, \end{aligned}$$

where $\gamma_{mn} = \ln(1 - \lambda_{mn})$ and $\beta = \ln(1 - \Theta)$. Let M' denote the number of all domain pairs, and M' is set to $M(M+1)/2$. For each protein pair (P_i, P_j) , we construct a vector $\mathbf{v}_{ij} (\in R^{M'})$ such that each element of \mathbf{v}_{ij} corresponding to a domain pair (D_m, D_n) is defined as

$$\mathbf{v}_{ij}^{(mn)} = \begin{cases} 1 & \text{if } D_{mn} \in P_{ij}, \\ 0 & \text{otherwise.} \end{cases}$$

In this setting, MAX PPI is equivalent to a problem to find a hyperplane (γ, β) which splits examples (vectors) into positive and negative ones such that the number of misclassified examples with γ is minimized. The optimal hyperplane (γ, β) is described as

$$\gamma \cdot \mathbf{v} = \beta, \quad (2)$$

where $\gamma (\in R^{M'})$ is a vector with γ_{mn} and $\mathbf{v} \in R^{M'}$.

On the other hand, IODT [9, 18] is defined in a similar manner, but the objective function e (called the sum-minority measure) to be minimized is different. As we have seen, a hyperplane divides a set of examples into two subsets, which we call X_1 and X_2 , respectively. For brevity, let the number of examples in \mathcal{P}_{pos} (\mathcal{P}_{neg}) in X_1 be u_1 (v_1), and the number of examples in \mathcal{P}_{pos} (\mathcal{P}_{neg}) in X_2 be u_2 (v_2). Then, IODT is defined as, given a positive point set \mathcal{P}_{pos} , a negative point set \mathcal{P}_{neg} and a value k , finding a hyperplane (γ, β) in eq. (2) such that $e \leq k$, where $e = \min(u_1, v_1) + \min(u_2, v_2)$. It is known that the problem of determining if there is a hyperplane (γ, β) that satisfies $e \leq k$ is NP-complete [9].

There are two major differences between MAX PPI and IODT. One is constraints on parameters in MAX PPI. That is, coefficients $\gamma_{mn} = \ln(1 - \lambda_{mn}) \leq 0$ and $\beta = \ln(1 - \Theta) \leq 0$ of the hyperplane in MAX PPI can take only non-positive values. By these constraints, the intractabilities of MAX PPI may differ from that of IODT. Recall that MAX PPI is MAX SNP-hard as well as NP-hard, and IODT is NP-complete.

The other difference is the scores of the objective functions. The objective function of IODT may result in assigning the same label to all examples. We can suppose that $\gamma \cdot \mathbf{v}_{ij} \leq \beta$ holds for any example \mathbf{v}_{ij} in X_1 without loss of generality. For simplicity, we consider here two dimensional space and only one hyperplane (line) that splits examples for IODT. In Figure 4, each of two lines (1) and (2) splits seven positive and two negative examples. Let e_i (s_i) be the score of the objective function of IODT (MAX PPI) with line (i). Recall that IODT uses the sum-minority measure $e = \min(u_1, v_1) + \min(u_2, v_2)$ and MAX PPI uses the sum $s = v_1 + u_2$ as the objective functions. We then obtain $e_1 = \min\{4, 0\} + \min\{3, 2\} = 2$, $e_2 = \min\{4, 1\} + \min\{3, 1\} = 2$, $s_1 = 4 + 2$, and $s_2 = 4 + 1$. In this example, IODT can choose one of the two lines. Moreover, the score of the sum-minority measure is always 2 with any line in Figure 4, and IODT assigns positive labels to all examples with some of the lines like lines (1) and (2), in other words, these lines do not contribute to the classification. However, MAX PPI always chooses line (1) with the maximum score $s_1 = 6$ among the two lines.

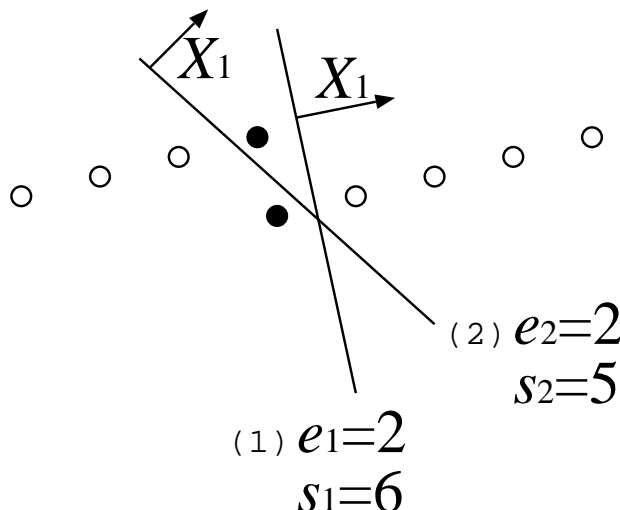


Figure 4: Possible hyperplanes (lines) that split examples. The open circles belong to class \mathcal{P}_{pos} and the filled ones belong to class \mathcal{P}_{neg} .

6 Conclusion

We have proposed a method called the ASNM method for inferring strengths of protein-protein interactions based on the association method. We have validated its performance by comparing the proposed method with the association method for binary interaction data and the LP-based method. The ASNM method ran much faster than the LPNM method and it performed almost as accurately as the LPNM method. In addition, the ASNM method did not overfit on the training data sets like the LPNM method in the experiment.

We have defined a problem of maximizing the classification accuracy for interacting or non-interacting pairs of proteins, and have proved that the problem is MAX SNP-hard. It is left as future work to understand how difficult it is to minimize errors of strengths of protein-protein interactions.

Acknowledgments

The authors express our deep appreciation to the anonymous reviewers. This work is supported in part by a Grand-in-Aid for Scientific Research on Priority Areas (C) for “Genome Information Science” from the Ministry of Education, Science, Sports, and Culture of Japan.

References

- [1] Bairoch, A. and Apweiler, R., The SWISS-PROT protein sequence database and its supplement TrEMBL in 2000, *Nucleic Acids Res.*, 28:45–48, 2000.
- [2] Bock, R.J. and Gough, A.D., Predicting protein-protein interactions from primary structure, *Bioinformatics*, 17:455–460, 2001.
- [3] Cortes, C. and Vapnik, V., Support-vector networks, *Machine Learning*, 20:273–297, 1995.
- [4] Deng, M., Chen, T., and Sun, F., An integrated probabilistic model for functional prediction of proteins, *Proc. 7th Annual International Conf. Computational Biology*, 95–103, 2003.

- [5] Deng, M., Mehta, S., Sun, F., and Chen, T., Inferring domain-domain interactions from protein-protein interactions, *Genome Research*, 12:1540–1548, 2002.
- [6] Enright, J.A., Iliopoulos, I., Kyrpides, C.N., and Ouzounis, A.C., Protein interaction maps for complete genomes based on gene fusion events, *Nature*, 402:86–90, 1999.
- [7] Gomez, M.G., Lo, H.S., and Rzhetsky, A., Probabilistic prediction of unknown metabolic and signal-transduction networks, *Genetics*, 159:1291–1298, 2001.
- [8] Hayashida, M., Ueda, N., and Akutsu, T., Inferring strengths of protein-protein interactions from experimental data using linear programming, *Bioinformatics*, 19(2):58–65, 2003.
- [9] Heath, D., Kasif, S., and Salzberg, S., Induction of oblique decision trees, *Proc. 13th International Joint Conf. Artificial Intelligence*, 1002–1007, 1993.
- [10] Ito, T., Chiba, T., Ozawa, R., Yoshida, M., Hattori, M., and Sakaki, Y., A comprehensive two-hybrid analysis to explore the yeast protein interactome, *Proc. Natl. Acad. Sci.*, 98:4569–4574, 2001.
- [11] Ito, T., Tashiro, K., Muta, S., Ozawa, R., Chiba, T., Nishizawa, M., Yamamoto, K., Kuhara, S., and Sakaki, Y., Towards a protein-protein interaction map of the budding yeast: a comprehensive system to examine two-hybrid interactions in all possible combinations between the yeast proteins, *Proc. Natl. Acad. Sci.*, 97:1143–1147, 2000.
- [12] Kim, K.W., Park, J., and Suh, K.J., Large scale statistical prediction of protein-protein interaction by potentially interacting domain (PID) pair, *Genome Informatics*, 13:42–50, 2002.
- [13] Letovsky, S. and Kasif, S., Predicting protein function from protein/protein interaction data: a probabilistic approach, *Bioinformatics*, 19(1):197–204, 2003.
- [14] Lu, L., Arakaki, A.K., Lu, H., and Skolnick, J., Multimeric threading-based prediction of protein-protein interactions on a genomic scale: Application to the *Saccharomyces cerevisiae* Proteome, *Genome Research*, 13:1146–1154, 2003.
- [15] Mamitsuka, H., Efficient mining from heterogeneous data sets for predicting protein-protein interactions, *Proc. 14th International Workshop Database and Expert Systems*, 32–36, 2003.
- [16] Marcotte, M.E., Pellegrini, M., Ng, H., Rice, D.W., Yeates, O.T., and Eisenberg, D., Detecting protein function and protein-protein interactions from genome sequences, *Science*, 285:751–753, 1999.
- [17] Marcotte, M.E., Pellegrini, M., Thompson, J.M., Yeates, O.T., and Eisenberg, D., A combined algorithm for genome-wide prediction of protein function, *Nature*, 402:83–86, 1999.
- [18] Murthy, K., S., Kasif, S., and Salzberg, S., A system for induction of oblique decision trees, *J. Art. Int. Res.*, 2:1–32, 1994.
- [19] Papadimitriou, C.H. and Yannakakis M., Optimization, approximation, and complexity classes, *J. Comp. Sys. Sci.*, 43:425–440, 1991.
- [20] Sprinzak, E. and Margalit, H., Correlated sequence-signatures as markets of protein-protein interaction. *J. Mol. Biol.*, 311:681–692, 2001.

- [21] Uetz, P., Giot, L., Cagney, G., Mansfield, T.A., Judson, S.R., Knight, R.J., Lockshon, D., Narayan, V., Srinivasan, M., Pochart, P., Qureshi-Emili, A., Li, Y., Godwin, B., Conover, D., Kalbfleisch, T., Vijayadamodar, G., Yang, M., Johnston, M., Fields, S., and Rothberg, M.J., A comprehensive analysis of protein-protein interactions in *Saccharomyces cerevisiae*, *Nature*, 403:623–627, 2000.
- [22] Vazirani, V.V., *Approximation algorithms*, Springer-Verlag, 1998.
- [23] Xenarios, I., Salwinski, L., Duan, X.J., Higney, P., Kim, S., and Eisenberg, D., DIP: The database of interacting proteins. A research tool for studying cellular networks of protein interactions, *Nucleic Acids Res.*, 30:303–305, 2002.
- [24] Zdobnov, M.E. and Apweiler, R., InterProScan - an integration platform for the signature-recognition methods in InterPro, *Bioinformatics*, 17:847–848, 2001.