BIOINFORMATICS

Vol. 19 Suppl. 2 2003, pages ii58–ii65
DOI: 10.1093/bioinformatics/btg1061

Inferring strengths of protein-protein interactions from experimental data using linear programming

*Morihiro Hayashida, Nobuhisa Ueda and Tatsuya Akutsu**

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

Received on March 17, 2003; accepted on June 9, 2003

ABSTRACT

Motivation: Several computational methods have been proposed for inference of protein-protein interactions. Most of the existing methods assume that protein-protein interaction data are given as binary data (i.e. whether or not each protein pair interacts). However, multiple biological experiments are performed for the same protein pairs and thus the ratio (strength) of the number of observed interactions to the number of experiments is available for each protein pair.

Results: We propose a new method for inference of protein-protein interactions from such experimental data. This method tries to minimize the errors between the ratios of observed interactions and the predicted probabilities in training data, where this problem is formalized as a linear program based on a probabilistic model. We compared the proposed method with the association method, the EM method and the SVM-based method using real interaction data. It is shown that a variant of the method is comparable to existing methods for binary data. It is also shown that the method outperforms existing methods for numerical data.

Availability: Programs transforming input data into LP format files are available upon request.

Supplementary information: http://sunflower.kuicr.kyoto-u.ac.jp/~morihiro/protint/supplement.html

Contact: takutsu@kuicr.kyoto-u.ac.jp

INTRODUCTION

Due to rapid progress of the genome sequencing projects, whole genomic sequences of more than several tens of organisms were already determined. As a next step of the genome projects, many researchers focus on understanding of functions of genes and/or proteins. Information about protein-protein interaction is important for understanding of protein functions because protein-protein interaction plays a key role in many cellular processes. Recently, large-scale two-hybrid systems were developed for comprehensive analysis of protein-protein interactions in *Saccharomyces cerevisiae* (budding yeast) (Ito *et al*., 2000, 2001; Uetz *et al*., 2000). Though these experiments revealed many unknown interactions, there were a large gap between the results by Ito *et al*. (2000, 2001) and Uetz *et al*. (2000). These suggest that current experimental techniques are not complete. Therefore, computational methods should be developed for inferring protein-protein interactions.

Several computational methods have been proposed for inference of protein-protein interactions. Enright *et al*. (1999) and Marcotte *et al*. (1999) proposed the gene fusion/Rosetta stone method. Marcotte *et al*. (1999) also proposed a method combining multiple sources of data. Wojcik *et al*. (2001) proposed the interaction domain pair profile method. Gomez *et al*. (2001) proposed probabilistic models for protein-protein interactions. Bock *et al*. (2001) applied the SVM (support vector machine) (Cortes *et al*., 1995) to inference of protein-protein interactions.

Recently, some methods were proposed for inferring domain-domain interactions (and/or signature-signature interactions) from protein-protein interaction data. Domain-domain interaction data are useful not only for more detailed understanding of protein-protein interactions but also for inferring protein-protein interactions: two proteins are expected to interact if these proteins contain an interacting domain pair(s). Sprinzak and Margalit proposed the association method for computing the score for each domain pair (Sprinzak *et al*., 2001). Kim *et al*. (2002) proposed similar scores and applied the scores to inference of protein-protein interactions. Deng *et al*. (2002) proposed an EM (expectation-maximization) algorithm for estimating the probability of interaction for each domain pair. They compared the EM method with the association method using protein-protein interaction data by Uetz *et al*. (2000) and Ito *et al*. (2000, 2001), and showed that the EM method was better than the association method.

Although most of the existing methods assume that protein-protein interaction data are given as binary data (i.e. whether or not each protein pair interact is given), multiple experiments are performed for the same protein

*To whom correspondence should be addressed.

pairs in practice and thus the ratio of the number of observed interactions to the number of experiments is available for each protein pair. For example, Ito *et al*. (2000, 2001) performed multiple experiments for each of protein-protein pairs. But, the results are not always the same for the same pair. Therefore, it is reasonable to use the ratio of the number of observed interactions to the number of experiments as input data, where the ratio is also referred to as the *strength* in this paper.

In this paper, we propose a new method for inferring domain-domain interactions from strength data of protein-protein interactions. This method tries to minimize the errors between the ratios of observed interactions and the predicted probabilities in training data. We formulate this minimization problem as a linear program based on a probabilistic model of protein-protein interaction, where the model was proposed by Deng *et al*. (2002). In order to minimize the errors, we use a technique similar to robust linear programming (Bennet *et al*., 1992) and soft margin (Cortes *et al*., 1995). Though we used the probabilistic model proposed by Deng *et al*. (2002), the proposed method is completely different from their method: their method uses an EM algorithm whereas our method uses linear programming, and their method assumes binary interaction data as input whereas our method assumes numerical interaction data as input. The proposed method has another advantage: several kinds of constraints can be easily put on and thus it is easy to combine the method with other methods.

The method is compared with the association method, the EM method and the SVM-based method using real protein-protein interaction data. It is shown that the method is comparable to existing methods when it is applied to binary data and outperforms existing methods when it is applied to numerical data (i.e. strength data).

ALGORITHMS

In this section, we describe the association method (Sprinzak *et al*., 2001), the EM method (Deng *et al*., 2002), and the proposing LP-based method along with its variants. We also describe a simple SVM-based method.

Association method (Sprinzak *et al*., 2001)

Let $P_1, \ldots, P_N$ denote the proteins in the training data set. We also use P_i to denote the set of domains contained in P_i. Let $D_1, \ldots, D_M$ denote the domains. Let P_{ij} and D_{mn} be the protein pair (P_i, P_j) and the domain pair (D_m, D_n), respectively. We also use P_{ij} to denote the set of domain pairs between P_i and P_j (i.e. $P_{ij} = \{D_{mn} | D_m \in P_i, D_n \in P_j\}$).

The association method assigns a simple 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 simply defined by

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

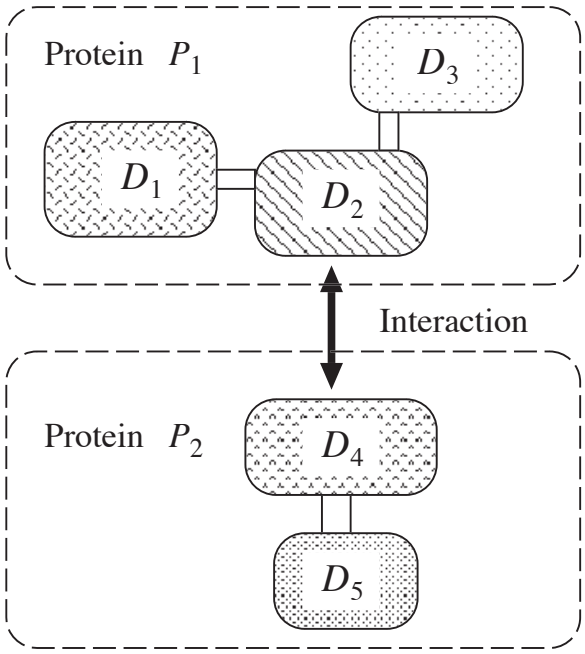

Fig. 1. Inference of protein-protein interactions through domain-domain interactions. In this case, we infer that proteins P_1 and P_2 interact with each other since domains D_2 and D_4 interact with each other.

EM method (Deng *et al*., 2002)

We use the probabilistic model proposed in Deng *et al*. (2002). We treat protein-protein interactions and domain-domain interactions as random variable: $P_{ij} = 1$ if P_i and P_j interact with each other, and $D_{mn} = 1$ if D_m and D_n interact with each other. We assume 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 Fig. 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. $\lambda_{mn} = \Pr(D_{mn} = 1)$).

Deng *et al*. (2002) considered two types of experimental errors: false positives, in which two proteins do not

interact in reality but were observed to be interacting in the experiments, and false negatives, in which two proteins interact in reality but were not observed to be interacting in the experiments. Let fp and fn denote the false positive rate and the false negative rate, respectively. Letting O_{ij} be the variable for the observed interaction result for P_i and P_j ($O_{ij} = 1$ if the interaction is observed), we have:

$$fp = \Pr(O_{ij} = 1 | P_{ij} = 0),$$
$$fn = \Pr(O_{ij} = 0 | P_{ij} = 1).$$

Then, $\Pr(O_{ij})$ is given by

$$\begin{aligned}&\Pr(O_{ij} = 1)\\ &= \Pr(O_{ij} = 1, P_{ij} = 1) + \Pr(O_{ij} = 1, P_{ij} = 0)\\ &= \Pr(P_{ij} = 1)(1 - fn) + (1 - \Pr(P_{ij} = 1))fp.\end{aligned}$$

Deng *et al*. (2002) defined the likelihood function (the probability of the observed whole proteome interaction data) by

$$L = \prod (\Pr(O_{ij} = 1))^{O_{ij}} (1 - \Pr(O_{ij} = 1))^{1-O_{ij}},$$

where $O_{ij} = 1$ if the interaction between P_i and P_j is observed. The likelihood L is a function of (λ_{mn}, fp, fn). Since it is difficult to directly compute (λ_{mn}, fp, fn) which maximize L, they developed an EM algorithm, where fp and fn were fixed to certain values.

LPBN: LP-based method for binary interaction data

In this subsection, we describe a basic version (called LPBN) of the proposing LP-based method.

Using the probabilistic model for the EM method and some threshold Θ, we can predict protein-protein interactions by the following rule:

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

The condition can be transformed as follows:

$$\begin{aligned}1 - \prod_{D_{mn} \in P_{ij}} (1 - \lambda_{mn}) &\geq \Theta,\\ \prod_{D_{mn} \in P_{ij}} (1 - \lambda_{mn}) &\leq 1 - \Theta,\\ \ln \left(\prod_{D_{mn} \in P_{ij}} (1 - \lambda_{mn}) \right) &\leq \ln(1 - \Theta),\\ \sum_{D_{mn} \in P_{ij}} \ln(1 - \lambda_{mn}) &\leq \ln(1 - \Theta),\end{aligned}$$

where 'ln' denotes the natural logarithm. Let $\gamma_{mn} = \ln(1 - \lambda_{mn})$ and $\beta = \ln(1 - \Theta)$. Then, the above condition can be written as

$$\sum_{D_{mn} \in P_{ij}} \gamma_{mn} \leq \beta.$$

This is a linear inequality. Therefore, if we can find γ_{mn} ($\gamma_{mn} \leq 0$) satisfying

$$O_{ij} = 1 \iff \sum_{D_{mn} \in P_{ij}} \gamma_{mn} \leq \beta$$

for all observed data (i.e. all training data) O_{ij}, we can obtain the parameters consistent with all training data.

However, it is usually impossible to satisfy all constraints. In such a case, it is reasonable to try to minimize the classification error. Though it is quite difficult to minimize the number of unsatisfied constraints (Amaldi *et al*., 1998), it is possible to minimize the sum of distances (Bennet *et al*., 1992; Cortes *et al*., 1995). Therefore, we use the following linear program:

$$\begin{aligned}&\text{minimize} \quad \sum_{P_{ij}} \xi_{ij},\\ &\text{subject to}\\ &\sum_{D_{mn} \in P_{ij}} \gamma_{mn} \leq \beta - const + \xi_{ij} \quad \text{for } P_{ij} \text{ such that } O_{ij} = 1,\\ &\sum_{D_{mn} \in P_{ij}} \gamma_{mn} > \beta + const - \xi_{ij} \quad \text{for } P_{ij} \text{ such that } O_{ij} = 0,\\ &\gamma_{mn} \leq 0 \quad \text{for all } \gamma_{mn},\\ &\xi_{ij} \geq 0 \quad \text{for all } \xi_{ij},\\ &\beta < 0,\end{aligned}$$

where *const* is an appropriate small constant (we currently use $const = 0.01$). Once γ_{mn} and β are determined, we can obtain λ_{mn} and Θ by $\lambda_{mn} = 1 - \exp(\gamma_{mn})$ and $\Theta = 1 - \exp(\beta)$, respectively.

LPNM: LP-based method for numerical interaction data

Here we describe an LP-based method for numerical interaction data (called LPNM), which is the most important variant of the LP-based method.

In LPBN, we used some threshold Θ to predict protein-protein interactions. On the other hand, in LPNM, we set Θ_{ij} to be the ratio of interactions between proteins P_i and P_j in a series of experiments, that is,

$$\Theta_{ij} = \frac{N(O_{ij})}{Z},$$

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

Since Θ_{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 Θ_{ij}, in other words, between the probability of observing an interaction in the above probabilistic model and the ratio of the interactions observed in the experiments.

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

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

From the above equation, we have a linear equation

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

for any $P_{i,j}$ by setting $\gamma_{mn} = \ln(1-\lambda_{mn})$ and $\beta_{ij} = \ln(1-\Theta_{ij})$. 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 sum of the difference $\sum_{P_{ij}} |\sum_{D_{mn} \in P_{ij}} \gamma_{mn} - \beta_{ij}|$. We therefore use the following linear program to minimize the difference:

$$\begin{aligned}
&\text{minimize} \quad \sum_{P_{ij}} \alpha_{ij}\,, \\
&\text{subject to} \\
&\qquad \sum_{D_{mn} \in P_{ij}} \gamma_{mn} - \beta_{ij} \le \alpha_{ij}, \\
&\qquad \beta_{ij} - \sum_{D_{mn} \in P_{ij}} \gamma_{mn} \le \alpha_{ij}, \\
&\qquad \gamma_{mn} \le 0 \quad \text{for all } \gamma_{mn}, \\
&\qquad \alpha_{ij} \ge 0 \quad \text{for all } \alpha_{ij}, \\
&\qquad \beta_{ij} < 0.
\end{aligned}$$

Combination of LPBN and EM

Due to the relation of $\lambda_{mn} = 1 - \exp(\gamma_{mn})$ (equivalently, $\gamma_{mn} = \ln(1 - \lambda_{mn})$), we can combine the LPBN method with the EM method. We examine two kinds of combinations: LPEM and EMLP.

The LPEM method first computes γ_{mn} using LPBN. Then, it converts γ_{mn} into λ_{mn} and applies the EM method using these λ_{mn} as the initial values.

The EMLP method first computes λ_{mn} using the EM method. Next, the following constraints are added to the linear program:

$$\ln((1 + \delta)(1 - \lambda_{mn})) \le \gamma_{mn} \le \ln((1 - \delta)(1 - \lambda_{mn})),$$

where δ is an appropriate fixed constant (we currently use $\delta = 0.05$ and $\delta = 0.2$). Then, γ_{mn} are obtained by solving the linear program.

SVM-based method

It is reasonable to apply SVM to inference of protein-protein interactions because LPBN is similar to SVM (Cortes *et al.*, 1995). Although SVM was already applied to inference of protein-protein interactions by Bock *et al.* (2001), they did not compute scores or probabilities of domain-domain interactions. In oder to apply SVM to inference of domain-domain interactions, we treat observed interacting pairs as positive examples and non-observed pairs as negative examples. For each protein pair (P_i, P_j), we define the feature vector $\boldsymbol{f}_{ij}$ by

$$\begin{aligned}
(\boldsymbol{f}_{ij})^{mn} &= 1 \text{ if } D_{mn} \in P_{ij}, \\
(\boldsymbol{f}_{ij})^{mn} &= 0 \text{ otherwise},
\end{aligned}$$

where $(\boldsymbol{f}_{ij})^{mn}$ denotes the mnth element of the vector $\boldsymbol{f}_{ij}$. If we use the linear kernel and the soft margin in SVM, it will be quite similar to LPBN. But, there is a big difference. In the SVM formulation, we can not guarantee $\gamma_{mn} \le 0$ (recall that $\gamma_{mn} = \ln(1 - \lambda_{mn})$). This condition is very important to give the probabilistic interpretation for the obtained parameters.

RESULTS

Data and implementation

We compared the LP-based methods (LPBN, LPNM, LPEM, EMLP) with the association method (ASSOC), the EM method (EM) and the SVM-based method (SVM). For the training and test data of protein-protein interactions, we used two data sets, the core data set of *Saccharomyces cerevisiae* (core20020404.lst) of the DIP database (Xenarios *et al.*, 2002) and the full data of Ito's Yeast Interacting Proteins (YIP) database (Ito *et al.*, 2000, 2001). We used the DIP database and the YIP database as for binary interaction data and for numerical interaction data, respectively. The main reason is that the DIP database seems to consist of the most reliable interaction data, and the YIP database provides numerical interaction data for pairs of proteins as the number of IST (Interaction Sequence Tags) hits. For each protein in these databases, we obtained its sequence data from the Swissprot/TrEMBL database (Bairoch *et al.*, 2000). In order to derive domains from the sequences, we used InterProScan (version 3.1) (Zdobnov *et al.*, 2001) as in Kim *et al.* (2002); Sprinzak *et al.* (2001). Though InterProScan identified not only protein domains but also protein signatures such as functional sites and sequence motives, we used all the hits because signatures may also play an important role in protein-protein interaction. As in Kim *et al.* (2002); Sprinzak *et al.* (2001), InterPro signatures in the same parent-child relationship were also merged into one signature. The sequence and signature pairs we used can be found at http://sunflower.kuicr.kyoto-u.

ac.jp/∼morihiro/protint/supplement.html. We also used PFAM (Bateman *et al.*, 2002) to find protein domains, and obtained results similar to those with InterPro. These supplementary data are also provided from the above web page.

We used SVM^{light} (Joachims *et al.*, 1999) for SVM learning, and used LOQO (version 1.08) on SUN UNIX (Vanderbei *et al.*, 1996) and lp_solve (version 4.0) on LINUX (ftp://ftp.es.ele.tue.nl/lp_solve) for solving linear programs. The experiments were mostly performed on a PC cluster with 8 Pentium Xeon 2.8 GHz processors, where only one CPU was used in all experiments. In each case, both training and test could be done in a few minutes.

The scores obtained by ASSOC were used as the initial values of λ_{mn} for EM since it was much better to use these scores than to use random initial values. EM steps were repeated until the difference of log-likelihood between two consecutive steps became less than 0.01 or until the number of repeats exceeded 200. Following to (Deng *et al.*, 2002), $fp = 2.5\text{E-}4$ and $fn = 0.80$ were used for EM. Though we examined several other parameter sets for EM, the results did not change significantly. We used the linear kernel for SVM with the default value of the trade-off parameter. Though we examined other kernels and parameters, the results did not change significantly.

For binary interaction data set, we evaluated the methods using the relationship between sensitivity and specificity. We call a protein pair a *true positive* if it is both predicted and observed, a *false positive* if it is predicted but is not observed, a *true negative* if it is neither predicted nor observed, and a *false negative* if it is not predicted but is observed. The *sensitivity* is defined to be the ratio of the number of true positives to the number of true positives and false negatives. The *specificity* is defined to be the ratio of the number of true negatives to the number of true negatives and false positives.

For numerical interaction data, we evaluated the methods by root mean squared error (RMSE) between the predicted probability $\Pr(P_{ij} = 1)$ and the observed ratio Θ_{ij} from the YIP database. In precise, for a set of proteins $\mathcal{P}$,

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

Results on binary training data

In order to evaluate the classification abilities of the methods for binary data, we first used the same data set for both training and test. Among 3003 pairs in the DIP core data set, we used 1767 pairs as positive data (POS), for each of which at least one hit was found by InterProScan. The other protein pairs were used as negative data (NEG), where we only considered the proteins that appeared in POS. Because of the limit of memory space, only (randomly selected) 40% of NEG were given for LPBN and SVM.

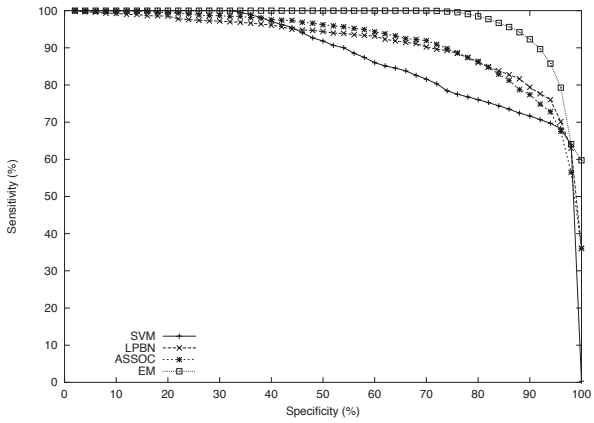

Fig. 2. Comparison of specificity and sensitivity for several methods on training data. It is seen that EM is the best, LPBN and ASSOC are comparable, and SVM is poor.

The result is shown in Figure 2. Since performances of LPEM and EMLP were almost the same as EM, the curve for LPEM or EMLP is not drawn in Figure 2. It is seen that EM is the best, LPBN and ASSOC are comparable, and SVM is poor. It is suggested from the figure that the probabilistic model proposed by Deng *et al.* (2002) is appropriate because SVM is not based on the model whereas the other methods are based on the model.

Results on binary test data

Next, we compared the methods for binary data using a standard evaluation procedure: parameters were learned using the training data set and then the relationship between sensitivity and specificity was measured using the test data set. We randomly select 2/3 of POS as positive training data and the remaining 1/3 of POS as positive test data. We randomly selected about 100 000 pairs not contained in POS as negative training data. It should be noted that about 28 000 pairs among 100 000 pairs were really used for training since we only used pairs each of which contained at least one domain pair appearing in positive training data. We used the remaining set of the pairs as negative test data. We repeated the above procedure 10 times and took the average over 10 trials.

The relationship between sensitivity and specificity for the test data set is shown in Figure 3. It should be noted that we removed protein pairs in the test data set which did not have domain pairs appearing in the positive training data set because the scores of such pairs are always 0. If such pairs are included, the sensitivity will decrease significantly. For example, the sensitivity decreases to $50 \sim 60\%$ at the specificity $= 80\%$ in each method.

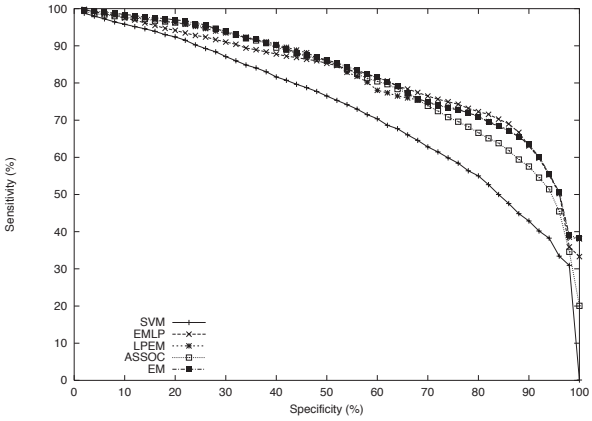

Fig. 3. Comparison of specificity and sensitivity for SVM, EMLP, LPEM, ASSOC and EM on test data.

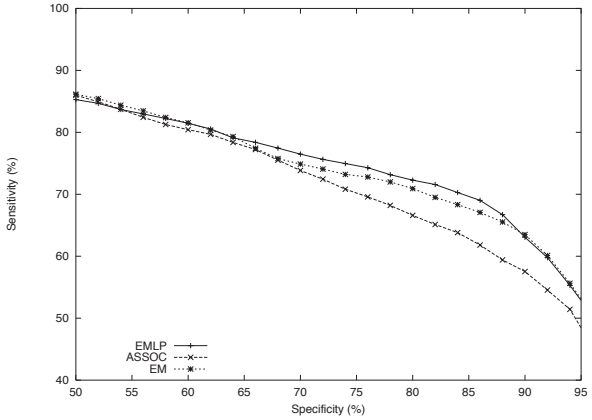

Fig. 4. Detailed comparison of specificity and sensitivity for EMLP, ASSOC and EM on test data.

It is seen from Figure 3 that performance of SVM was poor. As in the case of training data, performance of LPEM is similar to that of EM. We did not examine LPBN because its performance for training data was similar to that of ASSOC. Since the differences among EMLP, ASSOC and EM are unclear from Figure 3, the details of a part of Figure 3 are shown in Figure 4 for these three methods. It is seen that EMLP is slightly better than EM, and EM is slightly better than ASSOC. Though EM is better than EMLP in the region of specificity $<50\%$, the region of specificity $\geq 50\%$ is much more important.

Results on numerical interaction data

Lastly, we show results on numerical interaction data. We evaluated LPNM, EM and ASSOC. We did not evaluate LPEM, EMLP or SVM, because parameters obtained by LPEM and EMLP were similar to those by EM for binary data and the performance of SVM was poor even for binary data. We evaluated the methods by 5-fold cross validation. We used 1586 interaction pairs of proteins and the numbers of their IST hits as a whole data set.

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

		LPNM	EM	ASSOC
Error	1st	0.0244429	0.300659	0.284627
	2nd	0.0325133	0.31521	0.287918
	3rd	0.030796	0.299403	0.287875
	4th	0.0346763	0.292925	0.268931
	5th	0.0317004	0.276053	0.271517
	Average	0.03082580	0.2968499	0.2801738
Time	(sec)	1.295985	1.379543	0.0064746

In numerical interaction data, the ratio of the number of IST hits to the number of experiments is given for each pair of proteins. On the other hand, EM and ASSOC require labels (positive (interact) or negative (not interact)) to find appropriate parameters. We then have to set some threshold to divide the set of protein pairs into positive and negative data. We set here the threshold for IST hits to be 3, that is, interaction pairs whose IST hits are less than 3 are regarded as negative data, and the others as positive data. This threshold might seem to be too small compared with the total number of experiments ($192 = 96 \times 2$). But, the numbers of IST hits for most protein pairs are very low and thus we use this threshold. We examined several other threshold values, but the results did not change significantly.

Table 1 shows root mean squared errors and average elapsed time for test data sets using LPNM, EM and ASSOC. It should be noted that we employed 5-fold cross validation and the kth row means that the kth block among five blocks of the data is used as a test data set.

It is seen from the table that the errors for LPNM are much smaller than those for ASSOC and EM. Since the strength (i.e. the ratio of the number of IST hits to the number of experiments) takes a value between 0.0 and 1.0, the errors for LPNM are considerably small whereas the errors for EM and ASSOC are large. These results suggest that, in the sense of minimizing RMSE, LPNM was able to find much better parameters (i.e. probabilities of domain-domain interactions) than existing methods. It is reasonable because LPNM tries to minimize the error, whereas EM or ASSOC does not try to minimize the error. It is also seen that RMSE for EM is always greater than that for ASSOC. This is reasonable because EM tries to make the probabilities for interacting pairs in the training data close to 1.0 whereas strengths of most interacting

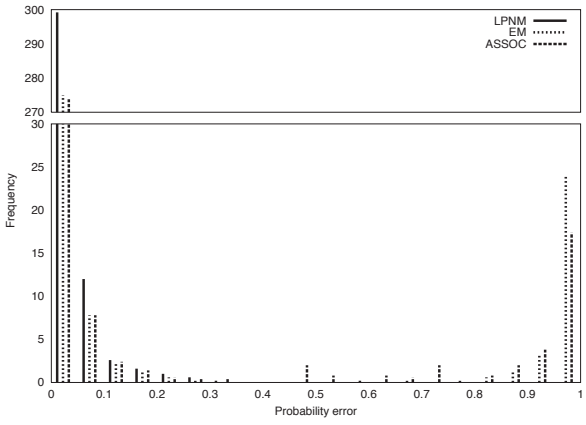

Fig. 5. Distributions of probability errors for LPNM, EM and ASSOC. Y-axis shows the number of interacting protein pairs for which the errors (between the predicted probabilities and the observed probabilities) are within the specified range. The average numbers over 5 test data sets are shown.

Table 2. Examples of inferred number of IST hits by LPNM, EM and ASSOC

Protein pair		YIP	LPNM	EM	ASSOC
Q06178	P53204	36	19	192	192
Q12518	Q99210	23	14	192	192
P53949	P50946	23	5	192	192
P32458	P32468	11	1	0	0
P27472	P47011	11	11	192	192
P07278	P05986	10	4	192	192
Q04739	P12904	9	3	192	192
P40054	P40054	9	3	191	187
P40917	P32366	7	15	192	192
P36017	P50079	7	2	0	0
P25383	Q99303	7	1	192	87
P23291	P39010	7	5	192	192
Q12084	Q12084	6	0	192	192
Q06169	Q12402	6	6	192	192
Q02821	P40892	6	1	0	0
P38697	Q02821	6	2	186	144

pairs are much lower than 1.0. As for the elapsed time, LPNM and EM are comparable, but ASSOC is much faster than them.

Figure 5 shows the average frequencies of probability errors of protein-protein interactions for the test data during the cross validation by LPNM, ASSOC and EM respectively. Note that distributions of errors for EM (and ASSOC) are large around 1.0 whereas these are small for LPNM. It is reasonable because EM tries to maximize the probabilities for interacting protein pairs, but the real probabilities are small.

For training data sets, the errors for LPNM are much smaller than those for EM and ASSOC (See our supplemental web pages). But, in EM and ASSOC, the errors for the training data are larger than those for the test data. It is also reasonable because EM or ASSOC was not designed for inference of strengths of interactions.

Table 2 shows examples of inferred strengths (the number of IST hits) of protein-protein interactions for LPNM, EM and ASSOC. In this table, data are shown for protein pairs (in one test data set) for which the numbers of IST hits in the YIP database are greater than 5 and at least one method output non-zero probabilities. It can be seen that inferred numbers of IST hits by LPNM are much closer to the numbers in the YIP database than those by EM and ASSOC. It is also seen that in most cases, the inferred numbers by EM and ASSOC are close to the maximum number of IST hits (i.e. $192 = 96 \times 2$).

DISCUSSION

We proposed an LP-based method (along with several variants) for inferring strengths of protein-protein interactions from experimental data. We compared the proposed method with existing methods such as the association method and the EM method. For numerical interaction data, the LPNM method outperformed existing methods.

The most important feature of the proposed method is that strengths of protein-protein interactions are taken into account for both training and test data. Although most of existing methods (e.g. ASSOC, EM, SVM) output scores ($\approx$ strengths) of protein-protein interactions, training data should be given as binary data. It seems difficult to modify existing methods so that numerical interaction data can be given as training data.

Another feature of the proposed method is that several kinds of constraints can be put on. In this paper, we used the following two types of constraints: constraints on the strengths of interactions (LPNM), and constraints on the ranges of the parameter values (EMLP). The former was quite useful as mentioned above. The latter was useful to combine the LP-based method with the EM method. It would be interesting to seek other types of constraints.

Though the LPNM method outperformed existing methods for numerical interaction data, its performance is not satisfactory as seen from Table 2. Therefore, improved methods for numerical data should be developed.

For the binary training data set, the EM method was better than LP-based methods and the association method. But, the differences for the test data set were small. In fact, the EM method was worse than the association method for several cases in which a lot of negative training data were given. It is probably due to overfitting. Thus, we might be able to improve the prediction accuracy for the test data set if some technique for avoiding overfit can be incorporated to the EM method and/or the LP-based method.

As mentioned before, all examined methods except the SVM-based method are based on the probabilistic model proposed by Deng *et al*. (2002) and are better than the SVM-based method. This suggests that the probabilistic model by Deng *et al*. (2002) is adequate and might capture some features of the relationship between domain-domain interactions and protein-protein interactions.

Though the LPBN method was better the SVM-based method, it is similar to the SVM-based method in the sense that both methods use a hyperplane to separate positive examples from negative examples, and try to minimize the sum of classification errors. If SVM can be modified for cooperating with constraints that the parameters must be negative, better results might be obtained. It would be interesting to study such modifications since SVMs have been successfully applied to many problems in Bioinformatics. It would also be interesting to modify SVM so that it can cooperate with numerical training data.

ACKNOWLEDGEMENTS

We would like to thank Dr. Hiroshi Mamitsuka for valuable discussions. This work was partially supported by Grant-in-Aid for Scientific Research on Priority Areas (C) 'Genome Information Science' and Grant-in-Aid #13680394 from the Ministry of Education, Science, Sports and Culture of Japan. This work was also partially supported by HITOCC (Hyper Information Technology Oriented Corporation Club), Japan.

REFERENCES

Amaldi,E. and Kann,V. (1998) On the approximability of minimizing nonzero variables or unsatisfied relations in linear systems. *Theor. Comput. Sci.*, **209**, 237–260.

Bairoch,A. and Apweiler,R. (2000) The SWISS-PROT protein sequence database and its supplement TrEMBL in 2000. *Nucleic Acids Res.*, **28**, 45–48.

Bateman,A., Birney,E., Cerruti,L., Durbin,R., Etwiller,L., Eddy,S.R., Griffiths-Jones,S., Howe,K.L., Marshall,M. and Sonnhammer,E.L.L. (2002) The Pfam protein families database. *Nucleic Acids Res.*, **30**, 276–280.

Bennet,K.P. and Mangasarian,O.L. (1992) Robust linear programming discrimination of two linearly inseparable sets. *Optimization Methods and Software*, **1**, 23–34.

Bock,J.R. and Gough,D.A. (2001) Predicting protein-protein interactions from primary structure. *Bioinformatics*, **17**, 455–460.

Cortes,C. and Vapnik,V. (1995) Support-vector networks. *Machine Learning*, **20**, 273–297.

Deng,M., Mehta,S., Sun,F. and Chen,T. (2002) Inferring domain-domain interactions from protein–protein interactions. *Genome Res.*, **12**, 1540–1548.

Enright,A.J., Iliopoulos,I., Kyrpides,N.C. and Ouzounis,C.A. (1999) Protein interaction maps for complete genomes based on gene fusion events. *Nature*, **402**, 86–90.

Gomez,G.M., Lo,S.H. and Rzhetsky,A. (2001) Probabilistic prediction of unknown metabolic and signal-transduction networks. *Genetics*, **159**, 1291–1298.

Ito,T., Tashiro,K., Muta,S., Ozawa,R., Chiba,T., Nishizawa,M., Yamamoto,K., Kuhara,S. and Sakaki,Y. (2000) 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. USA*, **97**, 1143–1147.

Ito,T., Chiba,T., Ozawa,R., Yoshida,M., Hattori,M. and Yoshiyuki,S. (2001) A comprehensive two-hybrid analysis to explore the yeast protein interactome. *Proc. Natl Acad. Sci. USA*, **98**, 4569–4574.

Joachims,T. (1999) Making large-scale SVM learning practical. In Schölkopf,B., Burges,C. and Smola,A. (eds), *Advances in Kernel Methods—Support Vector Learning*. MIT Press, pp. 169–185.

Kim,W.K., Park,J. and Suh,J.K. (2002) Large scale statistical prediction of protein-protein interaction by potentially interacting domain (PID) pair. *Genome Informatics*, **13**, 42–50.

Marcotte,E.M., Pellegrini,M., Ng,H., Rice,W.D., Yeates,T.O. and Eisenberg,D. (1999) Detecting protein function and protein-protein interactions from genome sequences. *Science*, **285**, 751–753.

Marcotte,E.M., Pellegrini,M., Thompson,M.J., Yeates,T.O. and Eisenberg,D. (1999) A combined algorithm for genome-wide prediction of protein function. *Nature*, **402**, 83–86.

Sprinzak,E. and Margalit,H. (2001) Correlated sequence-signatures as markets of protein–protein interaction. *J. Mol. Biol.*, **311**, 681–692.

Uetz,P., Giot,L., Cagney,G, Mansfield,A.T., Judson,R.S., Knight,J.R., Lockshon,D., Narayan,V., Srinivasan,M., Pochart,P. *et al*. (2000) A comprehensive analysis of protein-protein interactions in *Saccharomyces cerevisiae*. *Nature*, **403**, 623–627.

Vanderbei,R.J. (1996) *Linear Programming. Foundations and Extensions*. Kluwer Academic Publishers, Boston.

Wojcik,J. and Schächter,C. (2001) Protein-protein interaction map inference using interacting domain profile pairs. *Bioinformatics*, **17**, S296–S305.

Xenarios,I., Salwínski,L., Duan,X.J., Higney,P., Kim,S. and Eisenberg,D. (2002) DIP: The database of interacting proteins. A research tool for studying cellular networks of protein interactions. *Nucleic Acids Res.*, **30**, 303–305.

Zdobnov,E.M. and Apweiler,R. (2001) InterProScan—an integration platform for the signature-recognition methods in InterPro. *Bioinformatics*, **17**, 847–848.