Analysis of Protein Interaction Sites through Separated Data Spaces

Byung-Hoon Park[†], Rajesh Munavalli, Al Geist and Nagiza F. Samatova[†]
Computational Biology Institute, Computer Science and Mathematics Division,
Oak Ridge National Laboratory[‡],
P.O Box 2008, Oak Ridge, TN 37831-6367

Abstract

Protein-protein interactions are reinforced by a number of contacting residues between interacting proteins. Although identification of such interacting residues plays a crucial role in deciphering many biological phenomena, most attempts to uncover conditions for a residue to be in a contact site have achieved limited successes. We conjecture that currently available data for protein interactions includes a number of classification structures that do not coincide with one another, which makes classification task difficult or even impossible. This paper illustrates that data sets indeed reveal clearer structure if properly separated into appropriate subsets. It also shows that proper aggregation of classifiers built from each subset will suffice to predict some protein interaction sites with high confidence, and suggests a potential classification scheme in this direction.

1 Introduction

Protein-protein interactions are fundamental to cellular processes. They are responsible for phenomena like DNA replication/transcription, regulation of metabolic pathways, immunologic recognition, signal transduction, etc. The identification of interacting proteins is therefore an important prerequisite step in understanding their physiological functions.

Studies of protein interactions are understood at two different levels: protein and residue level. The former identifies pairs of proteins that interact with each other. From a bioinformatics' standpoint, the problem is to build a model that can predict a possible interaction between two proteins given their structural or sequence information. Various methods using genomics context or data mining techniques have been designed to tackle this problem. Genomics-based approaches include gene fusion events [10, 17], conservation of gene-order or co-occurrence of genes in potential operons [6, 21], and the presence/absence of genes in different species [23]. In parallel to genomic context based developments, a number of data mining methods that attempt to "learn" from experimental data of interacting proteins have been reported in the literature [4, 26, 22]. These approaches, using experimentally verified data like the Database of Interacting Protein (DIP) [27, 28], either try to construct Support Vector Machines (SVM), or extract statistically significant profile pairs in terms of Pfam [1] InterPro [18] and Blocks [15].

In contrast, studies of protein interactions at residue level focus on identifying specific residues that interact with other residue(s) in different proteins. Identification of such interacting residues will help understand biological processes at the molecular level, which in turn will benefit recognition of interactions at the protein level as well. In particular, identification of protein interactions at the residue level also benefits the protein docking problem, which tries to locate interaction sites between two proteins, and then construct a model for a protein complex.

Earlier attempts to decipher protein interactions at the residue level studied various biological properties of interaction sites, such as hydrophobicity, propensities, accessible surface area, planarity, etc. Although researchers found that each of these properties revealed some information regarding the interaction, none of them was indicative enough to be used as a general purpose predictor. For example, Gallet et al. [14] claimed that the hydrophobicity moment [9] is sufficient to predict interaction sites. However, the overall sensitivity and specificity is too low to accept. Thus, the lesson learned from these studies is that interfaces in different types of protein complexes (e.g., heterocomplexes and homodimers) often have different biochemical properties which are difficult to generalize.

A recent trend in this direction embraces data mining techniques using amino acid sequence or a combination of sequence and spatial structural information. Zhou and Shan, and Fariselli et al. independently [30, 11] applied neural networks to the data set derived

[†]Corresponding authors parkbh@ornl.gov, samatovan@ornl.gov

[‡]Oak Ridge National Laboratory is managed by UT-Battelle for the LLC U.S. D.O.E. under Contract No. DE-AC05-00OR22725.

from spatial neighbors of interacting residues. Similar techniques are deployed to predict interaction sites from primary sequences only. Ofran and Rost [20] used neural network, and Yan et al. [29] applied SVM to predict protein interactions sites from sequences.

Despite many efforts to predict interacting residues, very limited successes have been reported. To the best of our knowledge, most approaches perform slightly better than a random guess in identifying interacting residues. Is it even then possible to draw any descriptive model for predicting interacting residues? In other words, is there any systematic knowledge structure in protein-protein interactions that is reflected in experimental data?

This paper attempts to answer this question by postulating that currently available data sets include classification structures that conflict with one another, thus proper separation of data will unveil much clearer locally meaningful structures. It particularly suggests a separation of data based on "easiness" which indicates the level of consensus within a committee of classifiers. It then shows superior performance of classifiers constructed from each separated subset over the data set with the same "easiness", when it is compared to that of a monolithic classifier (built from the entire data set). It also suggests that a proper aggregation of individual classifiers potentially suffices to accurately predict protein interaction sites in general.

2 Systems and Methods

2.1 Data Set Data sets are chosen from the Protein Data Bank (PDB) [3]. Among several different types of protein complexes in the PDB database, we particularly decide to use protein complexes that account for transient interactions between non-identical chains. The data set is originally compiled and provided in [20, 19]. We further refine the data by filtering out protein chains in each complex that have more than 33% sequence identity with any other chain in different complexes. The final data set includes 52,106 residues from 303 protein complexes.

A residue is defined to be a contacting residue if its accessible surface area is reduced by more than 30% on complexation. From a primary sequence, segments of 11 consecutive residues are replaced with their corresponding HSSP (**H**omology-derived **S**tructure**S** of **P**roteins) scores [8], which constitute feature vectors. A feature vector is labeled as "interacting" if the centered residue (the 6th one from either end) is either a contacting residue or within 5 angstroms of it. Note that spatial information about residues is only used to assign labels that are needed to train a classifier; for classification, sequence information is sufficient.

2.2 Separation of Data using AdaBoost From a standpoint of data mining, a data set is difficult to mine when 1) it contains no systematic knowledge structure (i.e., randomly structured), or 2) a number of structures do not coincide with one another. We conjecture that the data set under consideration belongs to the second case. More specifically, we postulate that a knowledge structure is masked off by another one, thus it would reveal clearer structures if properly separated into subsets that each contains appropriate data only.

Given: $S_0 = \langle (x_1, y_1), (x_2, y_2), \cdots, (x_n, y_n) \rangle$, where (x_i, y_i) is the ith feature vector and its class label.
Initialize: $D_1(x_i) = 1/n, i = 1, \cdots, n$.
Begin
for $t = 1, \cdots, T$
1. Sample $m < n$ data points from S_{t-1} according to probability distribution D_t.
2. Generate a hypothesis h_t from the sample.
3. $\alpha_t = \frac{1}{2} ln(\frac{1-\varepsilon_t}{\varepsilon_t})$, where ε_t is the error rate of h_t.
4. Update D:
   $D_{t+1}(x_i) = \frac{D_t e^{-\alpha_t y_i h_t(x_i)}}{Z_t}$, where Z_t is a normalization constant chosen so that D_{t+1} is a probability distribution.

End
Output: the final hypothesis
$H(x) = sign(\sum_{t=1}^{T} \alpha_t h_t(x))$

Figure 1: The algorithm AdaBoost that implements sub-sampling. It assumes that $y_i \in \{1, -1\}$

We choose to divide the data set into three disjoint subsets based on "easiness". We particularly define a data point (or feature vector) to be easy when a clear consensus about its class label is observed among multiple classifiers and at the same time the prediction is correct. On the other hand, a data point is difficult when a consensus is drawn from the ensemble with an incorrect prediction. We argue that a monolithic classifier is bound to fail if it tries to accommodate both easy and difficult data points at the same time.

Boosting [24, 12, 13] is an ensemble learning technique that works by "resampling". It modifies the distribution according to which training data is resampled, while leaving the underlying learning algorithm unaltered. The first provably effective boosting algorithm, AdaBoost, was suggested by Freund and Schapire [12]. It has been extensively studied in a theoretical framework [16, 25, 5] as well as an empirical one [16, 2, 7]. AdaBoost has been proved to decrease the training error

of the data exponentially with the number of classifiers in the ensemble. Figure 1 shows an implementation of AdaBoost that is based on sub-sampling.

The data separation procedure utilizes the Boosting distribution updated after the final iteration step. The Boosting distribution (See Figure 1) gives higher weights to those instances that are hard to classify and lower weights to those that are easy to classify. With some arithmetic, it is not difficult to see that the Boosting distribution after the Tth iteration can be written as:

$$D_{T+1} = e^{-y_i \sum_{t=1}^{T} \alpha_t h_t(x_i)}/n \prod_{t=1}^{T} Z_t \tag{2.1}$$

where Z_t is the normalization constant for $D_t, 1 \leq t \leq T$.

Let us define the confidence of the ensemble of classifiers as the degree of consensus, i.e,

$$p = |\sum_{t=1}^{T} \alpha_t h_t(x)|$$

, where T denotes the number of classifiers in the ensemble.

Now we can rewrite equation 2.1 as:

$$D_{T+1}(x_i) = \begin{cases} e^{-p}/K & \text{if classification is correct} \\ e^{p}/K & \text{otherwise} \end{cases}$$

where $K = n \prod_{t=1}^{T} Z_t$. Note K does not depend on the training data, and can be considered as a constant for the current discussion. This suggests that the Boosting distribution is an either exponentially increasing or decreasing function of the confidence p depending on the correctness of the prediction. Since p essentially illustrates the degree of consensus among classifiers in the ensemble, the final Boosting distribution supposedly illustrates a degree of easiness of each data point; low values correspond to data points that are correctly classified with high consensus, whereas high values correspond to data points that are incorrectly classified again with high consensus.

Based on the final Boosting distribution, the data set is split into three sets, S_E (or E-set), S_M (or M-set) and S_H (or H-set) based on given cut-off thresholds. We particularly choose a logodd value to decide the threshold:

$$Logodd(x_i) = log_2(D_{T+1}(x_i)/(\frac{1}{n}))$$

Logodd value indicates the relative easiness (or hardness) compared to what is expected at random.

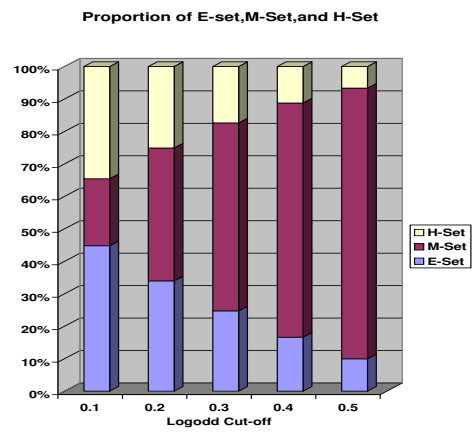

Figure 2: Distribution of E-set,M-set, and H-set when they are split at different cut-off values.

Given a pair of threshold values (δ_E, δ_H) a data point x_i belongs to:

$$\begin{array}{ll} S_E & \text{if } D_{T+1}(x_i) \leq \delta_E \\ S_H & \text{if } D_{T+1}(x_i) \geq \delta_H \\ S_M & \text{otherwise.} \end{array}$$

For the current discussion, we will use $\delta_E = -\delta_H$. Figure 2 shows how S_E, S_M and S_H are split with respect to different threshold values after 500 Boosting iterations.

3 Results

This section discusses properties of separated data subsets that are observed from some empirical studies. First, we compare complexity of the classifiers constructed from E-set, M-set and H-set, respectively. The aim is to confirm the assumption that E-set and H-set contain a clear knowledge structure that can be modeled by simple classifiers, whereas M-set includes more sophisticated one. Second, cross validation is performed between E-sets (and likewise between H-sets) from separately maintained data sets. This is particularly helpful to learn whether an E-set (or likewise an H-set) shares unique knowledge structure with other E-sets. For every case, we use AdaBoost with linear kernel based Support Vector Machine (SVM) as the base classifier.

3.1 Complexities of E-set, M-set and H-set We choose to measure ratios of support vectors in classifiers

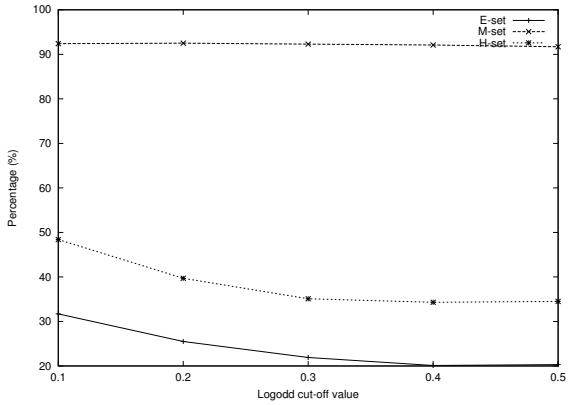

Figure 3: The ratios of support vectors in each SVM constructed from E-set, M-set, and H-set, respectively. E-sets/H-sets are generated at cut-off values of ± 0.1, ± 0.2, ± 0.3, ± 0.4 and ± 0.5.

(SVM) constructed from E-set, M-set, and H-set to compare their complexities. By definition, support vectors are data points that lie on the decision boundary of a SVM. In essence they are most difficult instances to classify. Thus the proportion of support vectors is a good measure of a machine complexity. Figure 3 illustrates ratios of support vectors to the size of the training set that constitute SVMs for E-set, M-set, and H-set, and shows how those ratios change over data sets that are generated around different cut-off threshold values. The figure demonstrates that complexities of E-set and H-set are clearly lower than that of M-set. Furthermore, regardless of threshold values (or, data sizes), the ratio of M-set is pretty much invariant, whereas the higher the ratio is (more precisely, the higher the absolute value of threshold), the lower the complexity is observed for both E-set and H-set.

3.2 Cross Validation between E-sets, and between H-sets To verify whether the data is split into E-set, M-set, and H-set by chance or by some biological properties, we split the entire data into three disjoint subsets, and compare classification performances among E-sets, and among H-sets, respectively. More specifically, the entire set of 303 heterocomplexes is split randomly into 3 disjoint subsets; i.e. each subset includes 101 complexes. From each subset, E-set, M-set and H-set are found, and AdaBoost is applied to construct two ensemble classifiers for the E-set and the H-set, respectively. The ensemble for E-set is cross-validated with the other E-sets. Likewise, the ensemble for H-set is also cross-validated. This process is repeated for the remaining two subsets (3-fold validation), and the overall

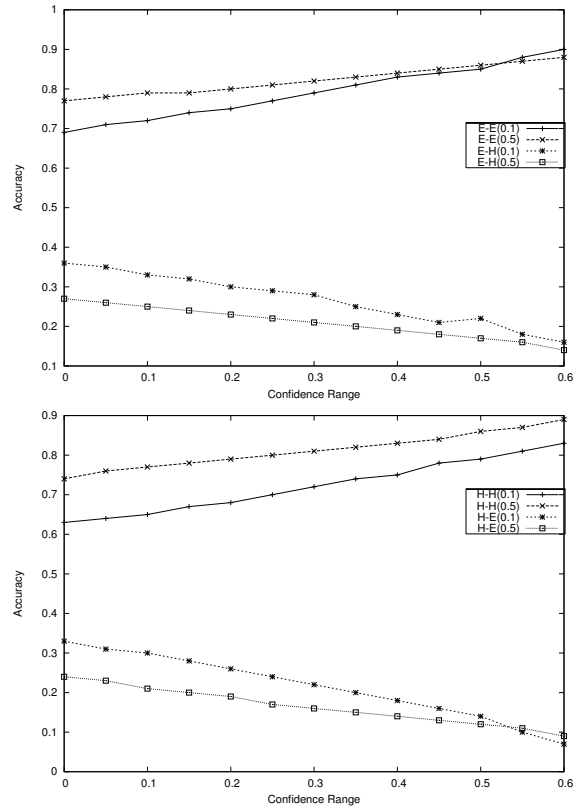

Figure 4: Cross validation results of classifiers constructed from E-set (Top) and H-set (Bottom), respectively. E-set(s) are split at -0.1 and -0.5, and the resulting classifiers are tested against other E-sets and H-sets. Likewise, H-sets are split at 0.1 and 0.5 and the resulting classifiers are tested over other E-sets and H-sets.

accuracies are measured with respect to different confidence levels. Note that confidence is the degree of agreement among different classifiers in an ensemble.

Figure 4 (Top) illustrates that classifiers constructed from an E-set perform significantly better over E-sets than H-sets (note these validation E-sets and H-sets are left out during the learning process). Likewise, Figure 4 (Bottom) a similar trend for classifiers constructed from H-sets. Furthermore, between E-sets and E-sets (and between H-sets and H-sets) classification accuracy is higher when predictions are made with higher confidences. In contrast, classification accuracy is lower with higher confidence if predictions are made between distinct sets: between E-set and H-set or vice versa. This clearly demonstrates that E-set and H-set contain classification structures that disagree with each other.

The entire cross validation procedure is repeated with the same data set but with random classification

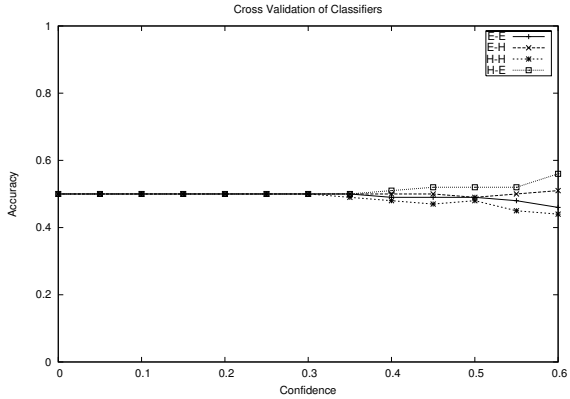

Figure 5: Cross validation results of classifiers constructed from E-set and H-set that are extracted from randomly labeled data set. Here E-set and H-set are split around logodd value of 0.1 and -0.1, respectively.

structures where labels Y are assigned randomly. It intends to verify that if the currently available data set has no systematic classification structures in it, the separation into E-set, M-set and H-set would not reveal any classification structures. Figure 5 essentially illustrates this, where the analogous cross validation results are shown with respect to different confidence levels and both E-set and H-set are split at logodd value 0.1. The figure confirms that there exist no common classification structures between E-sets and between H-sets.

4 Conclusion

Identification of residues in one protein sequence that are in contact with another protein sequence on complexation is important to elucidate a number of biological processes. However, the construction of such a predictor from protein sequences has been intractable so far; accuracies are slightly better than from a random guess. This paper suggests that clearer classification structures can be found by a proper separation of the data, which essentially shows that protein interaction sites are indeed governed by some biological patterns, not by randomness.

Our study will eventually help produce a more accurate classifier. Although we need a systematic way of segregating residues, which are previously unobserved, into proper sets (whether E-set or H-set), the empirical study suggests that proper aggregation of classifiers constructed from E-set, and H-set will suffice to predict some interaction sites with high accuracy.

Our study can also help identify additional features that may help predict protein interaction sites. It is particularly interesting to study biological properties of residues in either tail of the Boosting distribution. In fact, we are currently investigating those protein complexes that are abundant with residues in either tail to sift descriptive biological properties that can serve as discriminating features.

Acknowledgments

This work was funded in part or in full by the US Department of Energy's Genomes to Life program (www.doegenomestolife.org) under project, "Carbon Sequestration in Synechococcus Sp.: From Molecular Machines to Hierarchical Modeling," (www.genomes-to-life.org). The work of NFS was supported by the Laboratory Directed Research and Development Program of Oak Ridge National Laboratory (ORNL). ORNL is managed by UT-Battelle for the LLC U.S. D.O.E. under Contract No. DE-AC05-00OR22725. This research used resources of the Center for Computational Sciences at ORNL.

References

[1] A. Bateman, E. Birney, L. Cerruti, R. Durbin, L. Etwiller, S. R. Eddy, S. Griffiths-Jones, K. L. Howe, M. Marshall, and E. L. Sonnhammer. The pfam protein families database. *Nucleic Acids Res*, 30(1):276–80., 2002.

[2] Eric Bauer and Ron Kohavi. An empirical comparison of voting classification algorithms: Bagging, boosting, and variants. *Machine Learning*, 36(1–2):105–139, 1999.

[3] Helen M. Berman, John Westbrook, Zukang Feng, Gary Gililand, T. N. Bhat, Helge Weissig, Ilya N. Shindyalov, and Philip E. Bourne. The protein data bank. *Nucleic Acids Res*, 28:235–242, 2000.

[4] J. R. Bock and D. A. Gough. Predicting protein–protein interactions from primary structure. *Bioinformatics*, 17(5):455–60., 2001.

[5] Leo Breiman. Arcing classifiers. *The Annals of Statistics*, 26(3):801–849, 1998.

[6] T. Dandekar, B. Snel, M. Huynen, and P. Bork. Conservation of gene order: a fingerprint of proteins that physically interact. *Trends in Biochemical Sciences*, 23(9):324–328, 1998.

[7] Thomas G. Dietterich. An experimental comparison of three methods for constructing ensembles of decision trees: Bagging, boosting and randomization. *Machine Learning*, 40(2):139–158, 2000.

[8] Chris Dodge, Reihard Schneider, and Chris Snader. The hssp database of protein structure-sequence alignments and family profiles. *Nucleic Acids Res*, 26(1):313–315, 1998.

[9] D. Eisenberg, E. Schwarz, M. Komaromy, and R. Wall. Analysis of membrane and surface protein sequences

with the hydrophobicity moment plot. *J. Mol. Biol.*, 179:125–142, 1984.

[10] A. J. Enright, I. Iliopoulos, N. C. Kyrpides, and C. A. Ouzounis. Protein interaction maps for complete genomes based on gene fusion events. *Nature*, 402(6757):86–90, 1999.

[11] P. Fariselli, F. Pazos, A. Valencia, and R. Casadia. Prediction of protein-protein interaction sites in heterocomplexes with neural networks. *Eur. J. Biochem.*, 269:1356–1361, 2002.

[12] Yoav Freund and Robert E. Schapire. Experiments with a new boosting algorithm. In *13th International Conference on Machine Learning*, pages 138–146. Morgan Kaufmann, 1996.

[13] Yoav Freund and Robert E. Schapire. A decision theoretic generalization of online learning. *Computer and System Sciences*, 55(1):97–119, 1997.

[14] X. Gallet, B. Charloteaux, A. Thomas, and T. Brasseur. A fast method to predict protein interaction sites from sequences. *J. Mol. Biol.*, 302:917–926, 2000.

[15] J. G. Henikoff, S. Pietrokovski, C. M. McCallum, and S. Henikoff. Blocks-based methods for detecting protein homology. *Electrophoresis*, 21(9):1700–6. [pii], 2000.

[16] R. Maclin and D. Opitz. An empirical evaluation of bagging and boosting. In *Fourteenth International conference on Machine Learning*, pages 546–551, 1997.

[17] E. M. Marcotte, M. Pellegrini, H. L. Ng, D. W. Rice, T. O. Yeates, and D. Eisenberg. Detecting protein function and protein-protein interactions from genome sequences. *Science*, 285(5428):751–753, 1999.

[18] N. J. Mulder, R. Apweiler, T. K. Attwood, A. Bairoch, D. Barrell, A. Bateman, D. Binns, M. Biswas, P. Bradley, P. Bork, P. Bucher, R. R. Copley, E. Courcelle, U. Das, R. Durbin, L. Falquet, W. Fleischmann, S. Griffiths-Jones, D. Haft, N. Harte, N. Hulo, D. Kahn, A. Kanapin, M. Krestyaninova, R. Lopez, I. Letunic, D. Lonsdale, V. Silventoinen, S. E. Orchard, M. Pagni, D. Peyruc, C. P. Ponting, J. D. Selengut, F. Servant, C. J. A. Sigrist, R. Vaughan, and E. M. Zdobnov. The interpro database, 2003 brings increased coverage and new features. *Nucleic Acids Research*, 31(1):315–318, 2003.

[19] Yanay Ofran and Burkhard Rost. Analysing six types of protein-protein interfaces. *J. Mol. Biol.*, 325:377–387, 2003.

[20] Yanay Ofran and Burkhard Rost. Predicted protein-protein interaction sites from local sequence information. *Febs Letters*, 544:236–239, 2003.

[21] R. Overbeek, M. Fonstein, M. D'Souza, G.D. Pusch, and N. Maltsev. Use of contiguity on the chromosome to predict functional coupling. *In Silico Biol.*, 1:93–108, 1999.

[22] Byung-Hoon Park, George Ostrouchov, Andrey Gorin, Al Geist, and Nagiza F. Samatova. Inference of protein-protein interactions by unlikely profile pair. In IEEE, editor, *Third IEEE International conference on data mining*, Melbourne, FL, 2003.

[23] M. Pellegrini, E. M. Marcotte, M. J. Thompson, D. Eisenberg, and T. O. Yeates. Assigning protein functions by comparative genome analysis: protein phylogenetic profiles. *Proc Natl Acad Sci U S A*, 96(8):4285–4288, 1999.

[24] Robert E. Schapire. The strength of weak learnability. *Machine Learning*, 5(2):197–227, 1990.

[25] Robert E. Schapire, Yoav Freund, Peter Bartlett, and Wee Sun Lee. Boosting the margin: A new explanation for the effectiveness of voting methods. In *Fourtheenth International conference on Machine Learning*, 1997.

[26] E. Sprinzak and H. Margalit. Correlated sequence-signatures as markers of protein-protein interaction. *J Mol Biol*, 311(4):681–92., 2001.

[27] I. Xenarios, D. W. Rice, L. Salwinski, M. K. Baron, E. M. Marcotte, and D. Eisenberg. Dip: the database of interacting proteins. *Nucleic Acids Research*, 28(1):289–291, 2000.

[28] I. Xenarios, L. Salwinski, X. Q. 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 Research*, 30(1):303–305, 2002.

[29] Changhui Yan, Vasant Honavar, and Drena Dobbs. Predicting protein-protein interactions sites from amino acid sequence. Technical Report ISU-CS-TR-02-11, Department of computer science, Iowa State University, 2002.

[30] Huan-Xiang Zhou and Yibing Shan. Prediction of protein interaction sites from sequence profile and neighbor list. *Proteins: Structure, Function, and Genetics*, 44:336–343, 2001.