World Academy of Science, Engineering and Technology
International Journal of Bioengineering and Life Sciences
Vol:1, No:1, 2007

Protein-Protein Interaction Detection Based on Substring Sensitivity Measure

Nazar Zaki, Safaai Deris and Hany Alashwal

***Abstract*—**Detecting protein-protein interactions is a central problem in computational biology and aberrant such interactions may have implicated in a number of neurological disorders. As a result, the prediction of protein-protein interactions has recently received considerable attention from biologist around the globe. Computational tools that are capable of effectively identifying protein-protein interactions are much needed. In this paper, we propose a method to detect protein-protein interaction based on substring similarity measure. Two protein sequences may interact by the mean of the similarities of the substrings they contain. When applied on the currently available protein-protein interaction data for the yeast Saccharomyces cerevisiae, the proposed method delivered reasonable improvement over the existing ones.

***Keywords*—**Protein-Protein Interaction, support vector machine, feature extraction, pairwise alignment, Smith-Waterman score

I. Introduction

THE more we know about the molecular biology of the cell, the more we see genes and proteins as part of networks or pathways instead of as isolated entities, and their function as a variable dependent of the cellular context and not only of the individual properties [1]. This is the reason why biologists are making the transition from studying structure-function relationships in individual protein families to high-throughput investigation of entire cellular networks [2]. The goal remains to elucidate the structure, interactions and functions of all proteins within cells and organisms. The expectation is that this will provide a fuller appreciation of cellular processes and networks at the protein level, ultimately leading to a better understanding of disease mechanisms and suggesting new means for intervention [3]. To solve this problem, vast of approaches have already been developed for predicting physical interactions which may lead to the identification of the functional relationships between proteins. Some of the earliest techniques predict interacting proteins through the similarity of expression profiles [4], coordination of occurrence of gene products in genomes, description of similarity of phylogenetic profiles [5] or trees [6], and studying the patterns of domain fusion [7]. However, it has been noted that these methods predict protein–protein interactions in a general sense, meaning joint involvement in a certain biological process, and not necessarily actual physical interaction [8].

Most of the recent works focus on employing the protein domain knowledge to predict the protein-protein interaction. The motivation for this choice is that molecular interactions are typically mediated by a great variety of interacting domains [9]. It is thus logical to assume that the patterns of domain occurrence in interacting proteins provide useful information for training protein-protein interaction prediction methods.

One of the previous works introduced was based on the assumption that protein–protein interactions are evolutionary conserved. It involves the use of high-quality protein interaction map with interacting domain information as input to predict an interaction map in another organism [10]. Kim et al. [11] developed a statistical scoring system to measure the intractability between protein domains which could be used to predict protein-protein interaction. In other study, the notion of potentially interacting domain pair (PID) was introduced to describe domain pairs that occur in interacting proteins more frequently than would be expected by chance. In a similar approach, Ng et al. [12] described an integrative approach to computationally derive putative domain interactions from multiple data sources, including rosetta stone sequences, protein interactions, and protein complexes. Gomez et al. [13] constructed an attraction-repulsion model associated with Pfam domains along the length of each protein.

Most the above methods focus on domain structure and none of them consider all the sequence information to predict the protein-protein interaction. We understand that protein domains are highly informative for predicting protein-protein interaction as it reflects the potential structural relationships between proteins, however, other sequence parts (not currying any domain knowledge) may contribute to the information by showing how different two proteins are.

In this paper, we present a simple yet effective method to predict protein-protein interaction. The idea is to predict protein-protein interaction through sequence similarity. Two protein sequences may interact by the mean of the similarities of the substrings they contain. This work is motivated by the

This work was supported by the Research Affairs, UAE University under Grant 04-01-9-11/06.

Nazar Zaki is an Assistant Professor with the College of Information Technology, UAE University. Al-Ain 17555 UAE, (phone: +971-50-7332135; fax: +971-3-7626309; e-mail: nzaki@uaeu.ac.ae).
Safaai Deris is a Professor with the Faculty of Computer Science and Information Systems, Universiti Teknologi Malaysia, 81310 Skudai, Johor, Malaysia, (e-mail: safaai@fsksm.utm.my).
Hany Alashwal is a Ph.D. candidate at the Faculty of Computer Science and Information Systems, Univeristi Teknologi Malaysia, 81310 Skudai, Johor, Malaysia, (e-mail: hany@siswa.utm.my).

observation that the Smith-Waterman (SW), algorithm [14], which measures the similarity score between two sequences by a local gapped alignment, provides a relevant measure of similarity between protein sequences. This similarity incorporates biological knowledge about protein evolutionary structural relationships [15].

II. ALGORITHM

The proposed algorithm uses a transformation that converts protein sequence into fixed-dimensional representative feature vectors, where each feature records the sensitivity of a set of substrings of amino acids to the protein sequences of interest. These features are then used in conjunction with support vector machines (SVM) to predict the possible interactions between proteins. The overview of the algorithm which we call it Substring Scoring (SubSS) Method is presented in Fig 1. In the proceeding sections we will discuss the SubSS algorithm in details.

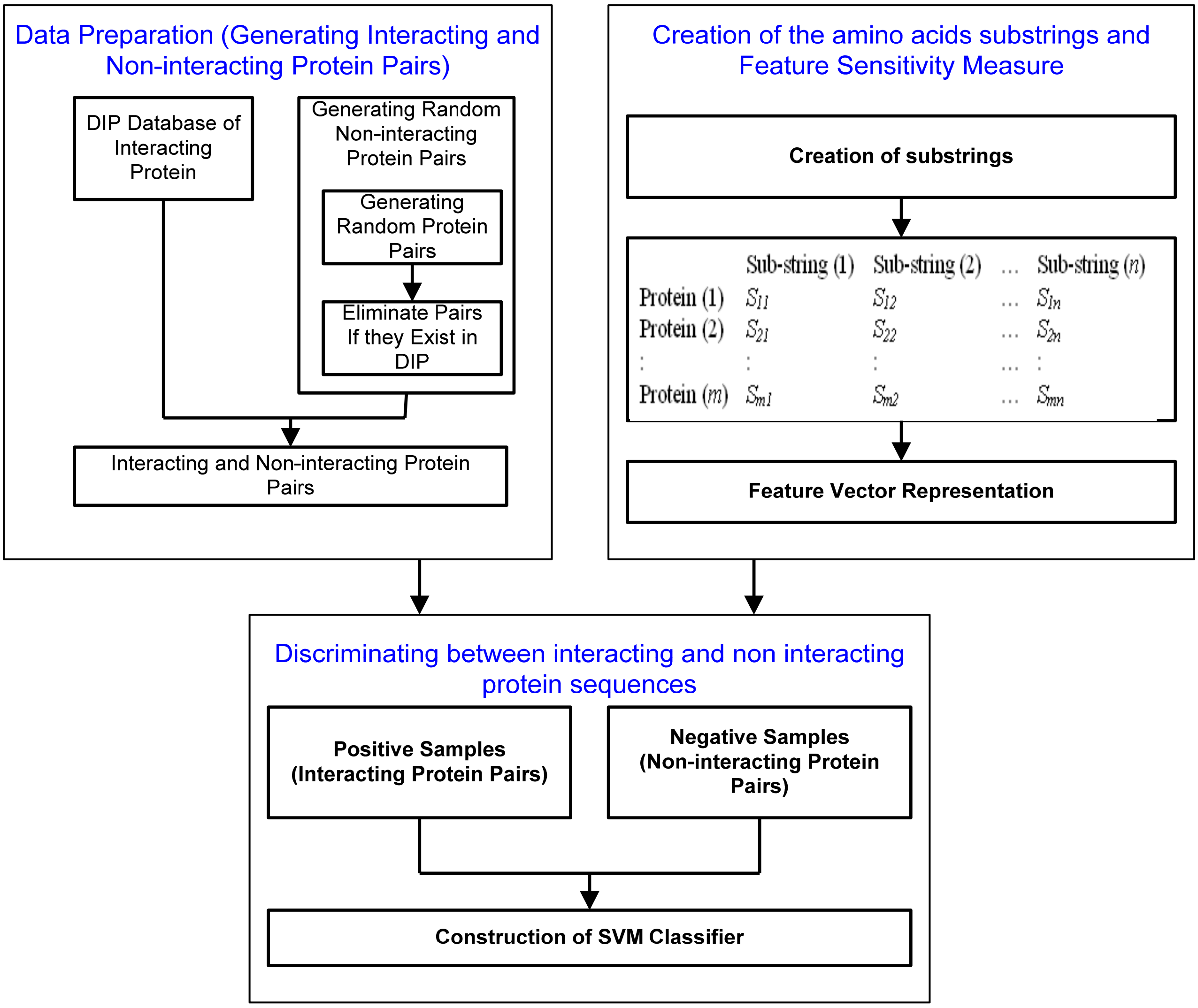

Data Preparation (Generating Interacting and Non-interacting Protein Pairs)
DIP Database of Interacting Protein
Generating Random Non-interacting Protein Pairs
Generating Random Protein Pairs
Eliminate Pairs If they Exist in DIP
Interacting and Non-interacting Protein Pairs
Creation of the amino acids substrings and Feature Sensitivity Measure
Creation of substrings
Sub-string (1) Sub-string (2) ... Sub-string (n)
Protein (1) S_{11} S_{12} ... S_{1n}
Protein (2) S_{21} S_{22} ... S_{2n}
Protein (m) S_{m1} S_{m2} ... S_{mn}
Feature Vector Representation
Discriminating between interacting and non interacting protein sequences
Positive Samples (Interacting Protein Pairs)
Negative Samples (Non-interacting Protein Pairs)
Construction of SVM Classifier

Fig. 1 algorithm overview

A. Data Preparation (generation of interacting and non interacting protein sequences)

This step starts by generating a dataset of interacting and non interacting protein pairs. For the interacting pair, it is simply obtained from the Database of Interacting Protein (DIP). However, obtaining identified and standard non-interacting proteins pairs remains to be the concern of all researchers working in predicting protein-protein interaction. Therefore, in our case we use a random method to generate proteins pairs, and then delete all pairs that appear in DIP. This is acceptable for the purposes of comparing the feature representation since the resulting inaccuracy will be approximately uniform with respect to each feature representation [16].

Two protein sequences $p_1 = (a_{11}, a_{12}, ..., a_{1n})$ and $p_2 = (a_{21}, a_{22}, ..., a_{2m})$, where a_{1n} refers to the n^{th} amino acid in protein sequence p_1, are represented as $seq_{pos1} = (a_{11}, a_{12}, ..., a_{1n}, a_{21}, a_{22}, ..., a_{2n})$, if the pair p_1 and p_2 are confidently interact with each other. Where seq_{pos1} shows that the new first protein sequence created by concatenating the two interacting pair p_1 and p_2 is placed in the positive class. However, if the two protein pair p_1 and p_2 are not interacting with each other then it's represented as $seq_{neg1} = (a_{11}, a_{12}, ..., a_{1n}, a_{21}, a_{22}, ..., a_{2n})$, where seq_{neg1} shows that the new first protein sequence created by concatenating the non-interacting pair p_1 and p_2 is placed in the negative class.

B. Creation of the amino acids substrings

In this step, we consider each protein sequence as a string of amino acids and then, we try to find out all possible substrings that the protein sequence contains. Unlike the string kernel method used for protein homology detection [17], we consider only the contiguous substrings. This goal can easily be achieved by simply shifting a window of a length $k > 1$, over the protein training examples. This process can be illustrated as follows:

If we have a protein sequence
>YAL030W SNC1 SGDID:S0000028
MSSSTPFDPYALSEHDEERPQNVQSKSRTAELQAEIDDTVGIM
RDNINKVAERGERLTSIEDKADNLAVSAQGFKRGANRVRKA
MWYKDLKMKMCLALVIIILLVVIIVPIAVHFSR*

Assuming $k = 20$, yields 6 substrings (note that the last substring is not necessary equal to k, however, it should not be a problem since we test the sensitivity against all the protein sequences of the interest).

>YAL030W sub 1
MSSSTPFDPYALSEHDEERP
>YAL030W sub 2
QNVQSKSRTAELQAEIDDTV
>YAL030W sub 3
GIMRDNINKVAERGERLTSI
>YAL030W sub 4
EDKADNLAVSAQGFKRGANR
>YAL030W sub 5
VRKAMWYKDLKMKMCLALVI
>YAL030W sub 6
IILLVVIIVPIAVHFSR*

C. Feature Sensitivity Measure

The sensitivity of each feature is measured using a simple pairwise sequence similarity algorithm. Smith-Waterman algorithm [14] is used to measure the sensitivity score between each substring generated in the previous step and the protein sequence. The score generated here is eventually used as a representation of the protein sequence. Our expectation here is to show that two proteins are likely to interact if they contain similar substrings of amino acids. The feature vector for each protein is thus formulated as follows:

$$\begin{aligned} p_1 &= (s_{11}, s_{12}, ..., s_{1n}) \\ p_2 &= (s_{21}, s_{22}, ..., s_{2n}) \\ &\vdots \\ p_m &= (s_{m1}, s_{m2}, ..., s_{mn}) \end{aligned} \qquad (1)$$

Where $s_{m1}, s_{m2}, ..., s_{mn}$ represent the scores of n substrings against a total number of m proteins. Proteins p_1 and p_2 are likely to interact if they contain similar substrings of amino acids. It's believed that, the possibility of two proteins to interact with each other is associated with their structural and functional similarities. Please note that, the idea of using protein substring sequence or pairwise is not novel, as many research have already been done to detect protein homology as a way to identify functional relationships [17], [18] and [19].

D. Discriminating between interacting and non interacting protein sequences

To discriminate between interacting and non interacting protein pairs, we employed support vector machine (SVM). SVM [20], [21] is a powerful classification algorithm and well suited the given task. It addresses the general problem of learning to discriminate between positive and negative members of a given class of n-dimensional vectors. The algorithm operates by mapping the given training set into a possibly high-dimensional feature space and attempting to learn a separating hyperplane between the positive and the negative examples for possible maximization of the margin between them [22]. The margin corresponds to the distance between the points residing on the two edges of the hyperplane. Having found such a plane, the SVM can then predict the classification of an unlabeled example. In fact, much of the SVM's power comes from its criterion for selecting a separating plane when many candidate planes exist: the SVM chooses the plane that maintains a maximum margin from any point in the training set [19]. SVM classifiers do not require any complex parameters to be tuned and optimized, and they exhibit a great ability to generalize even when given a small number of training examples. The only significant parameters to be tuned are the choice of the kernel function and the soft-margin parameter (capacity or regularization parameter). The kernel projects the data to higher dimensional space to increase the computational ability. The formulation of the SVM is described as follows:

Suppose our training set S consists of labeled input vectors (x_i, y_i), $i = 1...m$ where $x_i \in \Re^n$ and $y_i \in \{\pm 1\}$. We can specify a linear classification rule f by a pair (w, b), where the normal vector $w \in \Re^n$ and the bias $b \in \Re$, via

$$f(x) = (w, b) + b \qquad (2)$$

where a point x is classified as positive if $f(x) > 0$. Geometrically, the decision boundary is the hyperplane

$$\{x \in \Re^n : (w, x) + b = 0\} \qquad (3)$$

The idea makes it possible to efficiently deal with vary high dimensional futures spaces is the use of kernels:

$$K(x, z) = \langle \phi(x) \cdot \phi(z) \rangle \quad \text{for all } x, z \in X \qquad (4)$$

where ϕ is the mapping from X to an inner product feature space. We thus get the following optimization problem:

$$\max_{\lambda} \sum_{i=1}^{m} \lambda_i - \frac{1}{2} \sum_{i,j=1}^{m} \lambda_i \lambda_j y_i y_j K(x_i, x_j) \qquad (5)$$

subject to the constraints

$$\lambda_i \geq 0 \qquad \sum_{i=1}^{m} \lambda_i y_i = 0 \qquad (6)$$

III. Material and implementation

In this section, we describe the implementation and the materials used to test the algorithm on its ability to predict the protein-protein interaction.

A. Data Used

This step starts by generating a dataset of interacting and non interacting protein pairs. For the interacting pair, it is simply obtained from the Database of Interacting Protein (DIP).

We obtained the protein interaction data from the Database of Interacting Proteins (DIP). The DIP database provides sets of manually created protein-protein interactions in *Saccharomyces cerevisiae*. The current version contains 4749 proteins involved in 15675 interactions for which there is domain information. DIP also provides a high quality core set of 2609 yeast proteins that are involved in 6355 interactions which have been determined by at least one small-scale experiment or at least two independent experiments and predicted as positive by a scoring system [23]. Table I shows detailed description of the datasets that are comprised by DIP.

TABLE I
The protein interactions of yeast S. cerevisiae identified by wet-lab experiments

Number of Proteins	Number of Interactions	Number of Experiments	Number of Interactions
4749	15675	1	13653
		2	1278
		3	407
		4	167
		5	84
		6+	13653

B. Data Processing

We started processing the data by generating the substrings dataset. The number of substrings generated depends on the width of the window k. In our case we used different values of k such as 10, 20, 30, 40, 50, 60, 70, 80, 90, and 100. Following the preparations of the amino acids substrings datasets, we start extracting the feature vectors by scoring each substring against the core set of the yeast proteins. This process transforms all the 2609 protein sequence into fix dimension of features using pairwise algorithm. This feature extraction step uses Smith-Waterman [14] as implemented in FASTA [24]. In order to be consistent with the SVM-pairwise method [19], the substitution matrix is always the BLOSUM 62 matrix and the gap parameters are always set to 11 and 1. Following the feature extraction step, we concatenate the feature vectors of proteins based on whether the pair is interacting or not. If the concatenating proteins are interacting we place them in a positive set, otherwise, they are placed in a negative set. When the positive and negative sets are prepared, we employ SVM to discriminate between the interacting and non-interacting proteins. In our implementation, we used Libsvm software implemented by Chang et al. [25]. In all the experiments, the soft-margin parameter was set to 10 and employed the Gaussian Radial Basis Function kernel (RBF kernel). The Gaussian Radial Basis function is used as it allows pockets of data to be classified which is more powerful way than just using a linear dot product. The function has the form $K(x, z) = e^{-\gamma \|x-z\|^2}$, where $x, z \in X$ and $\gamma > 0$. In this case, the scaling parameter γ was set to 0.001. Ten-fold cross-validation was used to measure the training accuracy. The entire set of training pairs was split into 10 folds so that each fold contained approximately equal number of positive and negative pairs.

The algorithm is developed using Perl. To make a positive interaction set, we represent an interaction pair by concatenating feature vectors of each proteins pair that are listed in the DIP-CORE as interacting proteins. All proteins in DIP-CORE were included which yielded 3002 protein pairs. Constructing a negative interaction set is not an easy task. This is due to the fact that there are no experimental data in which protein pairs have confirmed to be non-interacting pairs [16]. As a result, we used a random approach to construct the negative data set since no other valid option is available in the literature. The negative interaction set was constructed by generating random protein pairs. Then, all protein pairs that exist in DIP were eliminated. This random approach can generate as many as 20202318 potentially negative candidates. Hence, the number of positive protein pairs is quite small compared to that of potentially negative pairs. The excessive potentially negative examples in the training set may lead to yield many false negatives because many of the positive examples are ambiguously discriminative from the negative examples in the feature space. For this reason, a negative interaction set was constructed containing the same number of protein pairs as for the positive interaction set.

IV. RESULTS

The performance of system is measured by how well a system can recognize interacting protein pairs. In order to analyze the evaluation measures in protein-protein interaction prediction, we first explain the contingency table (Table II). The entries of the four cells of the contingency table and a number n are described as follows:

$tp =$ number of interacting sequences classified interacting
$fn =$ number of non-interacting sequences classified interacting
$fp =$ number of interacting sequences classified non-interacting
$tn =$ number of non- interacting sequences classified non-interacting
$n = tp + fn + fp + tn$ (Total number of sequences).

TABLE II
THE CONTINGENCY TABLE

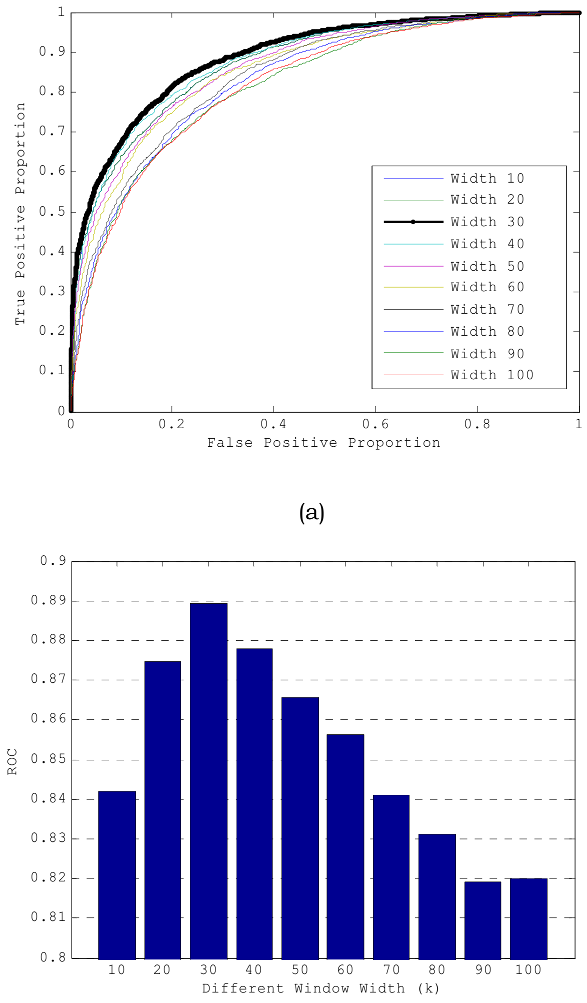
	Related Sequence	Unrelated Sequence
Classified Related	True positives (*tp*)	False negatives (*fn*)
Classified Unrelated	False positives (*fp*)	True negatives (*tn*)

The information encoded in the contingency table is used to calculate the protein-protein interaction evaluation measures. The performance of the algorithm is measured using two evaluation measures:

- Cross-validation accuracy = $\frac{tp + tn}{n}$, In this paradigm, the data are split into ten equal sized parts and calculates cross-validation accuracy.
- We further more calculated the receiver operating characteristic (ROC) [26]. The ROC statistic is the integral of the ROC curve, which plots the True Positive Proportion, $tpp = \frac{tp}{(tp + fn)}$, versus the False Positive Proportion, $fpp = \frac{tp}{(tp + fp)}$.

-

Based on the above mentioned performance measures, our algorithm was able to achieve cross-validation accuracy of 0.8457 and ROC score reaches 0.8892. This was the best performance based on a substring length of 30 amino acids. Different lengths are investigated to optimize the algorithm performance. Figs 2 (a), 2(b) and 3 show the comparison of different substring lengths and their performance based on 10-fold cross validation and ROC. All the three figures show that, the length 30 is the perfect window size. We can also notice that as the window grow wider or smaller the performance decrease accordingly.



(a)

(b)

Fig. 2 comparing different window size values (k) based on the ROC scores

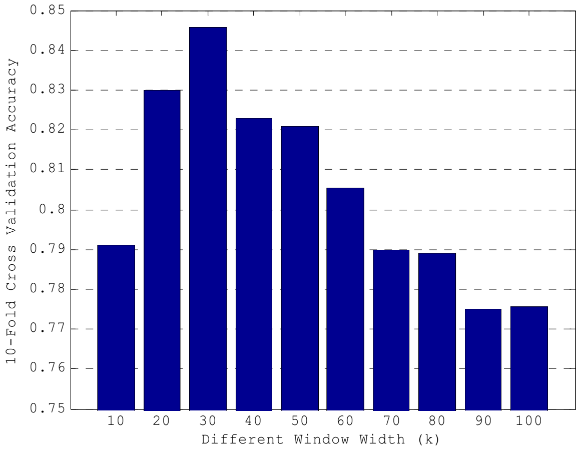


Fig. 3 comparing different window size values (k) based on 10-Fold cross validation accuracy.

A. *Comparing SubSS method with other existing works*

Comparing protein-protein interaction prediction systems with the other existing systems is always a difficult task. The reason is that, most of the authors used different type of data, experimental setup, and evaluation measures. In this section we will try to describe some of the good results achieved so far and compare them to our results. We will presents some of results achieved with an experimental work similar to ours in terms of the data used and experimental setup.

Kim et al developed a statistical scoring system to measure the intractability between protein domains which could be used to predict protein-protein interaction. The prediction system gives about 50% sensitivity and more than 98% specificity.

Ng et al. developed an integrative approach to computationally derive putative domain interactions from multiple data sources. He reported true positive value of 58.97% and false positive value of 12.51%., which approximately yields sensitivity of 58.97%, specificity of 82.5% and accuracy of 73.23%.

Gomez et al. constructed an attraction-repulsion model associated with Pfam domains. The best result achieved in this study was a ROC score of 0.818.

It's clear that our algorithm is outperformed most of the existing methods with cross-validation accuracy of 84.57% and ROC score reaches 0.8892.

V. Conclusion and discussion

Protein-protein interactions are operative at almost every level of cell function, in the structure of sub-cellular organelles, the transport machinery across the various biological membranes, packaging of chromatin, the network of sub-membrane filaments, muscle contraction, and signal transduction, regulation of gene expression, to name a few. The idea of this work is to predict protein-protein interaction through sequence similarity. Two protein sequences may interact by the mean of the similarities of the substrings they contain. The proposed algorithm termed SubSS, can effectively predict protein-protein interaction. The algorithm is able to outperform the currently available generic biochemical assays used for large-scale detection of protein-protein interactions. SubSS algorithm achieved cross-validation accuracy of 84.57% and ROC score reaches 0.8892. The accuracy of our algorithm comes from the combination of SVM algorithm and the Smith-Waterman score which have been developed to quantify the similarity of biological sequences. The SVM algorithm is based on a sound mathematical framework and has been shown to perform very well on many real-world applications [15]. The experimental work shows that, pairwise sequence comparison can be extremely powerful when used in conjunction with SVM.

One significant characteristic of any protein-protein interaction prediction algorithm is whether the method is computationally efficient or not. In order to gauge the computational cost of the proposed approach, SubSS method has an important cost in terms of computation time. SubSS method includes an SVM optimization, which is roughly O(n^2), where n is the number of training set examples. The feature sensitivity measure step of SubSS method involves computing n^2 pairwise scores. Using Smith-Waterman, itself is computed by dynamic programming and each computation is O(m^2), where m is the length of the longest training set sequence, yielding a total running time of O(n^2m^2). However, it can be worth the cost when one is interested in precision more than in speed.

Finally, the success of applying the SubSS method on predicting protein-protein interaction encouraged us to plan future directions such as optimizing the substring width and finding suitable threshold score.

Acknowledgment

This work was financially supported by the Research Affairs at the UAE University under a contract no. 04-01-9-11/06. The authors would also like to acknowledge the help provided by the School of the Graduate Studies and the AI & Bioinformatics Lab (AIBIL) at the Universiti Teknologi Malaysia (UTM).

References

[1] P. E. Bourne and H. Weissig, "Structural bioinformatics," John Wiley and sons, 2003.
[2] Y. Huang, D. Frishman and I. Muchnik, "Predicting protein-protein interactions by a supervised learning classifier," Computational Biology and chemistry, no. 28, pp: 291-301, 2004.
[3] J. R. Bock and D. A. Gough, "Predicting protein-protein interactions from primary structure," Bioinformatics, Vol. 17 no. 5, pp:455-460, 2001.
[4] E. M. Marcotte, M. Pellegrini, M. J. Thompson, T. O. Yeates, and D. Eisenberg, "A combined algorithm for genome-wide prediction of protein function," Nature, vol. 402, pp: 83–86, 1999.
[5] M. Pellegrini, E. M. Marcotte, M. J. Thompson, D. Eisenberg, and T. O. Yeates, "Assigning protein functions by comparative genome analysis: protein phylogenetic profiles," In the proceedings of National Academy of Sciences, USA, vol. 96, pp: 4285–4288, 1999.

[6] F. Pazos and A. Valencia, “Similarity of phylogenetic trees as indicator of protein-protein interaction,” Protein Engineering, vol. 14(9), pp: 609-614, 2001.

[7] J. Enright, I. N. Ilipoulos, C. Kyrpides, and C. A. Ouzounis, “Protein interaction maps for complete genomes based on gene fusion events,” Nature, vol. 402, pp: 86–90, 1999.

[8] D. Eisenberg, E. M. Marcotte, I. Xenarios, and T. O. Yeates, “Protein function in the post-genomic era,” Nature, vol. 405, pp: 823-826, 2000.

[9] T. Pawson and P. Nash, “Assembly of cell regulatory systems through protein interaction domains,” Science, vol. 300, pp: 445-452, 2003.

[10] J. Wojcik and V. Schachter, “Protein-Protein interaction map inference using interacting domain profile pairs,” Bioinformatics, vol. 17, pp: S296-S305, 2001.

[11] W. K. Kim, J. Park, and J. K. Suh, “Large scale statistical prediction of protein-protein interaction by potentially interacting domain (PID) pair,” Genome Informatics, vol. 13, pp: 42-50, 2002.

[12] S. K. Ng, Z. Zhang, and S. H. Tan, “integrative approach for computationally inferring protein domain interactions,” Bioinformatics, 19, pp: 923-929, 2002.

[13] S. M. Gomez, W. S. Noble, and A. Rzhetsky, “Learning to predict protein-protein interactios from protein sequences,” Bioinformatics, 19, pp: 1875-1881, 2003.

[14] Smith, T. and Waterman, M. Identification of common molecular subsequences. J. Mol. Bio., 147, pp: 195-197, 1981.

[15] H. Saigo, J. Vert, N. Ueda and T. Akutsu, “Protein homology detection using string alignment kernels,” Bioinformatics, Vol. 20 no. 11, pp: 1682-1689, 2004.

[16] H. Alashwal, S. Deris and R. Othman, “Comparison of Domain and Hydrophobicity Features for the Prediction of Protein-Protein Interactions using Support Vector Machines,” International Journal of Information Technology, Vol. 3, no. 1, 1305-2403, 2006.

[17] N. M. Zaki, S. Deris, and R. M. Illias, “Application of string kernels in protein sequence classification” App. Bioinformatics, 1, pp: 45, 2005.

[18] C. Leslie, E. Eskin, J. Weston and W. Noble, “Mismatch String Kernels for Discriminative Protein Classification,” Bioinformatics, 20, pp: 67, 2004.

[19] L. Liao, and W. S. Noble, “Combining Pairwise Sequence Similarity and Support Vector Machines for Detecting Remote Protein Evolutionary and Structural Relationships,” J. Comp. Biol., 10, pp: 857, 2003.

[20] Cristianini, N., and J. Shawe-Taylor, “An introduction to Support Vector Machines,” Cambridge, UK: Cambridge University Press. 2000.

[21] Vapnik, V. N. “Statistical Learning Theory,” Wiley, 1998.

[22] N. M. Zaki, S. Deris, and R. M. Illias, “Feature Extraction for Protein Homologies Detection Using Markov Models Combining Scores,” Int. J. on Comp. Intelligence and Appl., 1, pp: 1, 2004.

[23] C. M. Deane, L. Salwinski, I. Xenarios, and D. Eisenberg, “Protein interactions: two methods for assessment of the reliability of high throughput observations,” Molecular & Cellular Proteomics, vol. 1(5), pp: 349-56, 2002.

[24] W. R. Pearson, “Rapid and sensitive sequence comparisons with FASTAP and FASTA Method”, Enzymol, 183, pp: 63, 1985.

[25] C. C. Chang and C. J. Lin, “LIBSVM : a library for support vector machines,” 2001. Software available at http://www.csie.ntu.-edu.tw/~cjlin/libsvm. (24th March 2005).

[26] Swets, “Measuring the accuracy of diagnostic systems,” Science, 270, pp: 1285-1293, 1988.