DUJASE Vol. 3(1) 39-45, 2015 (January)

A Novel Dynamic Edit Distance based Algorithm for Protein-Protein Interaction Prediction

Md. Abdul Wadud Akanda and Saifuddin Md.Tareeq

Department of Computer Science and Engineering, University of Dhaka

Email: smtareeq@cse.univdhaka.edu

Received on 30. 10. 2013. Accepted for publication on 14.07. 2014.

Abstract

In this paper a novel dynamic algorithm for predicting protein-protein interaction based on protein sequence information is proposed. The algorithm consists of two major steps namely feature extraction and classification. Feature extraction is performed by a new dynamic edit distance based approach and classification is done by using support vector machine. The proposed algorithm is evaluated in terms of accuracy and efficiency. With a cross validation accuracy of 87.3%, the proposed algorithm gives better result in terms of accuracy and sensitivity than most of the existing methods. With the proposed algorithm a competitive running time of $O(n^2m^2)$ is achieved where n is the number of sequences and m is the longest sequence.

Keywords: Sequence Similarity, Protein-Protein Interaction, Edit Distance, Support Vector Machine.

1. Introduction

Protein-protein interactions are important in almost all aspects of cellular function, such as signaling pathways, protein structure modeling, immunological recognition, DNA replication and repair, gene translation, enzyme reaction and molecular recognition, as well as protein synthesis. Detecting interaction between two proteins provides functional and structural information and helps in identifying pharmacological targets and guides drug designing. Hence predicting the interaction of proteins has a great importance in molecular recognition. Figure 1 shows a general form of protein-protein interaction.

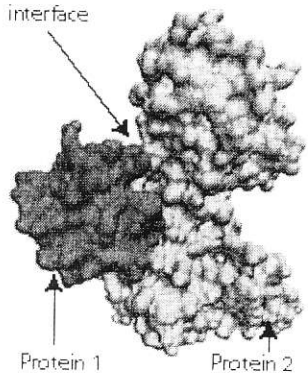

Fig. 1: Protein-Protein interaction

Computational methods based on sequence information employ domain knowledge to predict the protein-protein interaction. Molecular interactions are typically mediated by a great variety of interacting domains [1]. Sprinzak *et. al.* [2] developed the Association Method (AM) which defines a simple measure of interaction probability between two domains as the fraction of interacting protein pairs among all protein pairs containing the domain pairs. The limitation of this method lies in the possibility to assign high association scores to domain pairs with low frequency. Deng *et. al.* [3] developed the Maximum Likelihood Estimation (MLE) method which is based on the assumption that two proteins interact if at least one pair of domains of the two proteins interact. Huang *et. al.* [4] introduced the Maximum Specificity Set Cover (MSSC). Huang started by selecting high quality protein interactions based on a clustering measure and then used MSSC to assign probabilities to domain pairs. As most of the existing domain based methods consider only single-domain pairs and assume independence between domain-domain interactions, Xue-Wen *et. al.* [5] introduced a domain-based random forest of decision trees to infer protein interactions. This method is capable of exploring all possible domain interactions and making predictions based on all the protein domains. The tool termed PIPE (Protein-Protein Interaction Prediction Engine) was developed by Sylvain *et. al.* [6]. Based on the assumption that some of the interactions between proteins are mediated by a finite number of short polypeptide sequences, PIPE is developed. These short polypeptide sequences are typically shorter than the classical domains, and are used repeatedly in different proteins and contexts within the cell.

The methods discussed are based on previously identified domains and the identification of domain is a long and computationally expensive process. They are not universal because their accuracy and reliability is dependent on the domain information. They often have limited abilities to detect novel interactions and to differentiate them from false positives.

In this paper we proposed a predictive method based on analysis of protein sequence information without knowing protein domains. The idea is to predict protein-protein interaction through sequence similarity considering two protein sequences may interact by the mean of similarity of substrings they contain. A new dynamic edit distance based algorithm is proposed for feature generation and then used support vector machine for classification. Experimental result suggests that our algorithm outperform existing algorithms in terms of accuracy and sensitivity and stand second in terms of specificity.

2. Proposed Algorithm

The details of the proposed algorithm are described in the following subsections.

2.1 Collecting interacting protein sequence

The Database of Interacting Proteins (DIP) lists protein pairs that are known to interact. To test the proposed method the protein-protein interaction data from the DIP is obtained. The sequences of the proteins participating in DIP interactions are provided in FASTA format.

The DIP version used for our test was used by Nazar Zaki at el [7] which contains 4749 proteins involved in 15675 interactions. In this case, only high quality core set of 2609 yeast proteins was considered. This core set was involved in 6355 interactions, which have been determined by at least one small-scale experiment or two independent experiments [8]. Furthermore, it is followed that same dataset where proteins interact with only one protein and not involved in any other interactions. This process resulted in a dataset of 150 proteins with 75 positive interactions shown in the Table 1. Our main target here is to design an approach capable of predicting protein interaction partner where edit distance based algorithm will be used for generating feature values, which facilitates a way to construct protein-protein interaction using only sequence information.

Table 1: Dataset of interacting proteins used in the experiment

YBL045C	YPR191W	YDR098C	YGL220W	YLR317W	YNL140C
YBR127C	YDL185W	YDR139C	YLR306W	YLR366W	YMR242C
YDR045C	YQR207C	YDR140W	YNR046W	YLR417W	YPL002C
YDR190C	YPL235W	YDR469W	YLR015W	YML119W	YLL032C
YDR441C	YML022W	YER159C	YDR397C	YMR052W	YFR008W
YEL041W	YJR049C	YGL057C	YJL135W	YMR228W	YFL036W
YER017C	YMR089C	YGL090W	YOR005C	YNL311C	YKL001C
YGR180C	YJL026W	YGL174W	YIR005W	YOL108C	YDR123C
YGR240C	YMR205C	YGL195W	YFR009W	YOL111C	YOR007C
YGR261C	YBR288C	YGL125W	YGL154C	YOR269W	YLR254C
YHL027W	YJL056C	YGR057C	YKL015W	YPL003W	YPR066W
YHR024C	YLR163C	YGR074W	YKL183W	YPL209C	YBR156C
YHR056C	YDR303C	YGR208W	YKL177W	YPR046W	YJR135C
YIL103W	YKL191W	YGR229C	YGR185C	YPR051W	YEL053C
YLR238W	YDR200C	YHL044W	YKR035C	YBR107C	YDR254W
YLR456W	YPR172W	YHR193C	YDR252W	YDR080W	YDL077C
YNL007C	YIR040C	TJL006C	YML112W	YER069W	YJL071W
YML329C	YKL197C	YJL035C	YLR316C	YER090W	YKL211C
YOR136W	YNL037C	YJL090C	YKL108W	YGL008C	YCR024C-A
YPL195W	YJL024C	YKL160W	YKL036C	YGL236C	YMR023C
YPR029C	YLR170C	YLL059C	YML011C	YGR075C	YBR152W
YBR228W	YLR135W	YLR036C	YKR065C	YHR004C	YAL009W
YDR001C	YLR270W	YLR065C	YDL149W	YKL182W	YPL231W
YDR013W	YDR489W	YLR226W	YPR161C	YLR075W	YIR012W
YDR086C	YLR378C	YLR240W	YBR097W	YNL259C	YDR270W

2.2 Generating non-interacting protein sequence

Though the number of interacting proteins is much smaller than non-interacting proteins, obtaining identified and standard non-interacting protein pairs remains to be a concern of all researchers working in predicting protein-protein interaction. Therefore, in this case, the following two steps are used for generating non-interacting protein sequences. In the first step, it adopted a random method using the amino acids (A, V, Y, P, M, I, L, D, E, K, R, S, T, Y, H, C, N, Q, W, Z) to generate non-interacting protein pairs. In the second step it deleted all pairs that appear in the DIP by chance for getting non-interacting data set. This technique of generating non-interacting dataset is acceptable for the purpose of comparing the feature representation since the resulting inaccuracy will be approximately uniform with respect to each feature representation [9]. The dataset that is considered for testing contains 150 protein sequences which are involved in 75 interactions. For this reason, the equal number of non-interacting protein sequences using the above technique is generated.

2.3 Preparation of experimental dataset

Interacting and non-interacting protein sequences are grouped separately. All the protein sequences in each group are then merged. The merged sequence is divided into some substrings based on the window size. Then the similarity score for each protein sequence against each substring of the merged sequence of each group is measured using proposed edit distance based algorithm. Finally, the similarity scores are concatenated based on the prior knowledge of interaction. After getting similarity scores from both the interacting and non-interacting group, all the scores are accumulated to prepare the final dataset.

Let us explain the procedure with an example using interacting protein sequences. The interacting protein sequences are arranged into a group named Interacting Group (G_I). So, $G_I = \{S_1, S_2, S_3 \ldots\ldots\ldots\ldots\ldots, S_N\}$ where S indicates protein sequence. Let, S_1, S_2, S_3, S_4, S_5 and S_6 are six protein sequences where,S_1= {MSSSTPFDPYAL}, S_2={QNVQSKSR}, S_3= {EDKAD}, S_4= {VRKI}, S_5= {IILLVVI} and S_6={LAVIIVPIAPSR}.

It is also assumed that prior knowledge about the interaction information between these proteins is known. Let, these 6 proteins interact in the following manner: S_1 interact with S_3, S_2 interact with S_6 and S_4 interact with S_5.

These six proteins sequences is then merged to make a sequence named 'MergedSequence' as MergedSequence ($S_1, S_2, S_3, S_4, S_5, S_6$) ={

MSSSTPFDPYALQNVQSKSREDKADVRKIIILLVVI LAVIIVPIA }. This merged sequence was divided into substrings according to a window size. Let, for the above example, window size is 7.Then the following substrings will be available in this case:$SubString_1$= {MSSSTPF}, $SubString_2$= {DPYALED}, $SubString_3$= {KADVRKI}, $SubString_4$= { IILLVVI }, $SubString_5$= {LAVIIVP} and $SubString_6$= {IAPSR}.

It is notable here that the last substring is not necessarily being equal to 7; however, it should not be a problem since the sensitivity against all the protein sequences of interest is tested. Then similarity score between each of the substrings of the merged sequence and each protein sequence is generated. This process was continued for each of the protein sequences of our interest. This procedure of similarity score measurement is illustrated in subsection 2.4.

In the next step, the scores of the interacting protein sequences were concatenated. For the above example, as protein sequence S_1 interact with sequence S_3, the respective scores of these sequences will be concatenated for preparing the training dataset. In this way, the scores of S_2 and S_6, S_4 and S_5 will be concatenated. Using all of these scores, the training dataset is prepared. In case of non-interacting protein sequences, similar procedure is followed.

2.4 Similarity score measurement

The proposed algorithm uses a transformation that converts protein sequence into fixed-dimensional representative feature vectors, where each feature records the similarity 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 similarity of each feature was measured using a pair wise sequence similarity algorithm. An edit distance based algorithm was used to measure the similarity score between a substring of the merged sequence and each protein sequence. The score generated here is eventually used as training data for the future step. The feature vector for each protein is thus formulated as follows: The merged sequence was arranged into a number of substrings ($SubString_1$, $SubString_2$, $SubString_3$, ..., $SubString_N$) based on the window size. Then the feature values in that case might look like as follows.

$Score_{1S1}$= Edit_Distance_Based_Algorithm{ S_1, $SubString_1$}

$Score_{2S1}$= Edit_Distance_Based_Algorithm { S_1, $SubString_2$}

$Score_{NS1}$= Edit_Distance_Based_Algorithm { S_1, $SubString_N$}

$Score_{1S2}$= Edit_Distance_Based_Algorithm { S_2, $SubString_1$}

$Score_{2S2}$= Edit_Distance_Based_Algorithm { S_2, $SubString_2$}

$Score_{NS2}$= Edit_Distance_Based_Algorithm { S_2, $SubString_N$}

$Score_{1SN}$= Edit_Distance_Based_Algorithm { S_N, $SubString_1$}

$Score_{2SN}$= Edit_Distance_Based_Algorithm { S_N, $SubString_2$}

$Score_{NSN}$= Edit_Distance_Based_Algorithm { S_N, $SubString_N$}

The proposed Edit Distance Based Algorithm is explained in subsection 2.5. Using a shifting window over the merged sequence of the training set may lead to generating a subsequence comprises of the end of one sequence and the beginning of the next sequence. This, however, is not a problem since all protein sequences of interest score against the same subsequence. Similarly, feature values for the non-interacting dataset were also calculated. The block diagram of overall procedure is shown in Figure 2 and the pseudo-code for the algorithm is given in Figure 3.

2.5 The Algorithm

Edit distance is used to measure distance as the number of operations required to transform a string into another. Given two character strings, S1 and S2, the edit distance between them is the minimum number of edit operations required to transform S1 into S2. Most commonly, the edit operations allowed for this purpose are: inserting a character into a string, deleting a character from a string and replacing a character of a string by another character; for example, the edit distance between cat and dog is 3. In fact, the notion of edit distance can be generalized to allowing different weights for different kinds of edit operations, for instance a higher weight may be placed on replacing the character s by the character p, than on replacing it by the character a. Setting weights in this way depending on the likelihood of letters substituting for each other is very effective in practice.

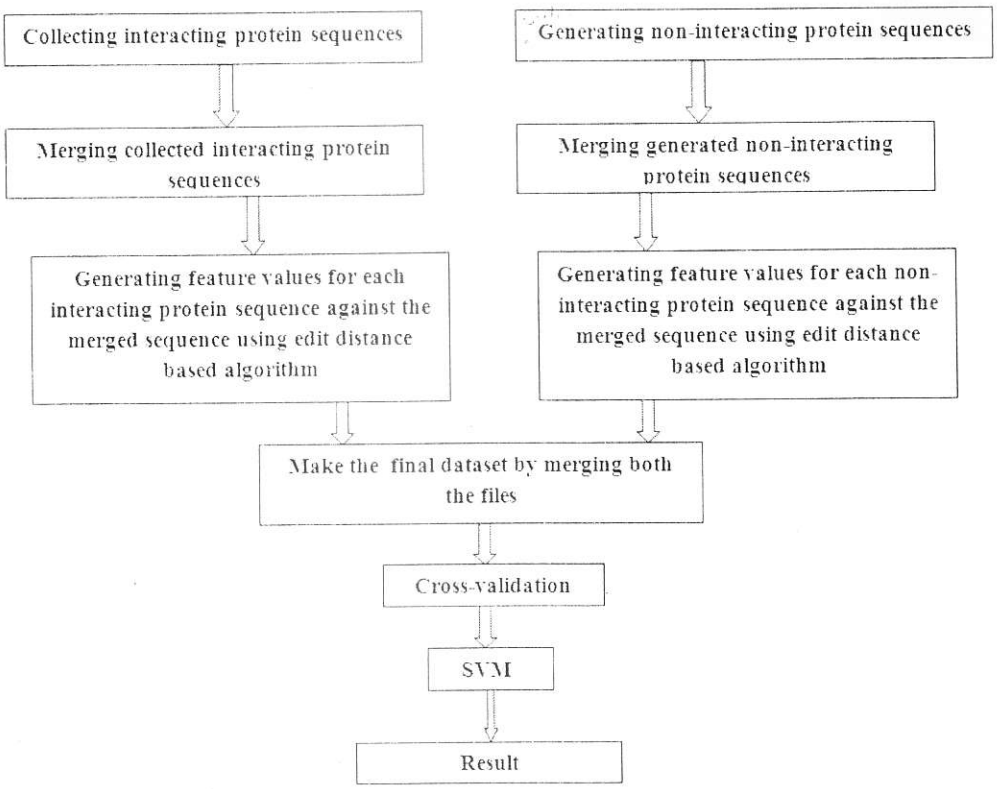

Fig. 2: Block diagram for the proposed algorithm

Algorithm 1 Proposed Edit Distance based Algorithm

Require: Two Strings(String1, String2)
Ensure: Similarity Score between Two Strings

1: for $i = 0 \rightarrow m$ do ▷ m → Maximum length between the two strings
2: $Table[0][i] = 0$ ▷ Table →Holds scores of the respective positions
3: $Table[i][0] = 0$
4: end for
5: for $i = 0 \rightarrow m$ do
6: for $j = 0 \rightarrow m$ do
7: $a = 0$
8: if $String1[i] == String2[j]$ then
9: $a = 2$
10: end if
11: $a = Table[i][j] + a$
12: $b = Table[i + 1][j] - 1$
13: $c = Table[i][j + 1] - 1$
14: $Table[i + 1][j + 1] = max(a, b, c)$
15: end for
16: end for
17: return $Table[m - 1][m - 1]$

Fig. 3: Proposed Edit Distance based Algorithm

In our algorithm edit distance has been used in the following way:

1) The main target of our proposed Edit Distance based Algorithm is to find out the similarity ratio. That means, it indicates the maximum number of matches between the two strings. On the contrary, conventional edit distance vector algorithm finds out the minimum cost needed to make the two strings similar.

2) Edit Distance Algorithm minimizes the result whereas our proposed Edit Distance based Algorithm maximizes the output.

With proposed algorithm if two strings are considered, for example, String1 ={ababababbab} and String2 ={ababbabb} then the scenario that will be obtained is shown in Figure 4.

		a	*b*	*a*	*b*	*a*	*b*	*b*	*a*	*b*
	0	0	0	0	0	0	0	0	0	0
a	0	2	1	2	1	2	1	0	2	1
b	0	1	4	3	4	3	4	3	2	4
a	0	2	3	6	5	6	5	4	5	4
b	0	1	4	5	8	7	8	7	6	7
b	0	0	3	4	7	7	9	10	9	8
a	0	2	2	5	6	9	8	9	12	11
b	0	1	4	4	7	8	11	10	11	14
b	0	0	3	3	6	7	10	13	12	13

Fig. 4: Similarity score obtained by using proposed algorithm

Proposed edit distance based algorithm compares two sequences and when both the positions of the two strings match, adds 2 whereas if the positions mismatch, negates 1. Using this rule, the final feature values of the strings which determine their respective similarity scores is obtained. Here, the problem is solved by identifying a collection of sub-problems and tackling them one by one, smallest first, using the answers to small problems to help figure out larger ones, until the whole lot of them is solved. As proposed algorithm follows above mentioned criteria, the algorithm is dynamic in nature. It needs to mention here again that the edit distance based dynamic algorithm has been used in protein-protein interaction for the first time.

2.6 Cross Validation

In k-fold cross-validation, the original sample is randomly partitioned into k sub samples. Of the k sub samples, a single sub sample is retained as the validation data for testing the model, and the remaining k-1 sub samples are used as training data. The cross-validation process is then repeated k times (the folds), with each of the k sub samples used exactly once as the validation data. The k results from the folds then can be averaged (or otherwise combined) to produce a single estimation. The advantage of this method over repeated random sub-sampling is that all observations are used for both training and validation, and each observation is used for validation exactly once. A 10-fold cross-validation is carried in our experiment. The training and testing sets were used in Support vector machine (SVM).

2.7 Classification using SVM

Support vector machines (SVMs) are a set of related supervised learning methods used for classification and regression. In simple words, given a set of training examples, each marked as belonging to one of two classes, an SVM training algorithm builds a model that predicts whether a new example falls into one class or the other. Support Vector Machine (SVM) with our prepared dataset is used. SVM was used for both training and testing data. By the result of SVM, it was possible to predict whether two protein sequences interact or not.

3. Experimental results

3.1 Window size determination

The proposed method is based on the assumption that two proteins may interact if their pair wise scores against large subsequences of amino acids created by shifting a window over concatenated protein training sequences are similar. As window size is a considering factor in this case, the first step in our investigation was to determine the optimal sliding window length. It is tested with 5 different window sizes starting from 500. The others were 1000, 1500, 1800 and 2000 respectively. Sensitivity, specificity and accuracy recorded from testing our method on 150 interacting protein sequences and 150 non-interacting protein sequences for various window sizes are shown in the Table 2.

Table 2: Sensitivity, specificity and accuracy recorded from our experiment

Window Size	Sensitivity (%)	Specificity (%)	Accuracy (%)
500	68.38	75.47	78.44
1000	79.47	81.91	83.13
1500	77.60	85.47	87.33
1800	65.42	78.70	80.39
2000	79.18	84.20	79.87

In this way it is empirically determined that the method works well when the window size is 1500 and at that time sensitivity, specificity and accuracy are 77.60%, 85.47% and 87.33% respectively.

3.2 Performance Analysis

3.2.1 Performance measurement terms

Interaction prediction has to fulfill two competing demands. The predictor should cover as many of the real interacting residues as possible, but at the same time should predict as few false positive as possible. These two demands are measured by sensitivity and specificity respectively. Including these two criteria, the results reported in this paper concern the evaluation of protein-protein interaction prediction based on the following quantities:

- The number of true positives (TP) (residues correctly classified as interacting)

- The number of true negatives (TN) (residues correctly classified as non-interacting)
- The number of false positives (FP) (non-interacting residues incorrectly classified as interacting)
- The number of false negatives (FN) (interacting residues incorrectly classified as non-interacting).

Based on the above definition and according to Xue wen Chen *et. al.* [10]:

$$Sensitivity = \frac{TP}{FN + TP}$$

$$Specificity = \frac{TN}{FP + TN}$$

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

3.2.2 Performance Comparison

In this section a comparative analysis of result is given. It is to be noted that comparing protein-protein interaction prediction systems with the other existing systems is always a difficult task because most of the authors used different types of data, experimental setup, and evaluation measures. Table 3 summarizes the performance of proposed method in terms of different performance measurement terms and shows the comparative results with other methods.

Table 3: Performance comparison among different interaction prediction methods

Method	Sensitivity (%)	Specificity (%)	Accuracy (%)
Proposed Algorithm	77.60	85.47	87.33
Ensemble Method[14]	76.76	63.16	79.76
Integrative approach[12]	58.97	82.50	73.23
Statistical Scoring System[11]	50.00	98.00	71.42
PPI-GS[15]	51.65	38.78	68.42

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. The prediction system gives about 50% sensitivity and more than 98% specificity. Ng *et. al.* [12] developed an integrative approach to computationally derive putative domain interactions from multiple data sources. Authors 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%. PIPE [13] produced a sensitivity of 61% for detecting yeast protein interaction with specificity 89% and an overall accuracy of 75%. Figure 5 shows the comparison of our method with other different existing methods in terms of accuracy, specificity and sensitivity.

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 approach with the proposed algorithm, edit distance based algorithm has an important benefit in terms of computation time. This method includes

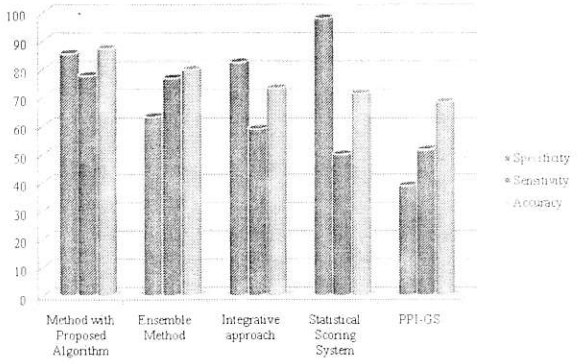

Fig. 5: Different interaction prediction methods and their Accuracy

an SVM optimization, which is roughly $O(n^2)$, where n is the number of training set examples. The feature sensitivity measure step of the method involves computing n^2 pair-wise scores. Using edit distance based algorithm, 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 worth the cost as life scientists is interested in precision more than in speed.

4. Conclusion

In this study a dynamic method for protein-protein interaction prediction using only sequence information is proposed. The method was developed based on a combination of similarity score measurement by using edit distance based algorithm and support vector machine. It is shown that similarity score provides relevant measure of similarity between protein sequences. This similarity incorporates biological knowledge about proteins and it is extremely powerful when combined with support vector machine to predict protein-protein interaction. The experimental result shows that accuracy and sensitivity of our algorithm is better than most of the existing algorithms and stand second in terms of specificity.

References

1. T. Pawson and P. Nash. Assembly of cell regulatory systems through protein interaction domains. Science 2003, 300:445-452, 2003.
2. E. Sprinzak and H. Margalit. Correlated sequence-signatures as markers ofprotein-protein interaction. J Mol Biol, 311:681-692, 2001.
3. M. Deng, S. Mehta, F. Sun, and T. Cheng. Inferring domain-domain interactions from protein protein interactions. Genome Res, 12:1540-1548, 2002.
4. T.W. Huang, A.C. Tien, W. S. Huang, Y.C. Lee, C.L. Peng, H.H. Tseng, C. Y. Kao, and C.Y. Huang. A database for the prediction of protein- protein interactions based on the orthologous interactome. Bioinformatics, 20:3273-3276, 2004.
5. C. Xue-Wen and L. Mei. Prediction of protein-protein interactions using random decision forest framework. Bioinformatics, 21:4394-4400, 2005.
6. P. Sylvain, D. Frank, C. Albert, C. Jim, D. Alex, E. Andrew, G. Marinella, G. Jack, J. Mathew, K. Nevan, L. Xuemei, and G. Ashkan. A protein-protein interaction prediction engine based on the re-occurring short polypeptide sequences between known interacting protein pairs. BMC Bioinformatics, 7(365), 2006.
7. N. Zaki. Prediction of protein-protein interactions using pairwise alignment and inter-domain linker region. Engineering Letter, 16(4), 2008.
8. 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, 1:349-56, 2002.
9. H. Alashwal, S. Deris, and R. M Othman. Comparison of domain and hydrophobicity features for the prediction of protein-protein interactions using support vector machines. International Journal of Information Technology, 3(1):18-24, 2007.
10. X. Chen and J.C. Jeong. Sequence-based prediction of protein interaction sites with an integrative method. Bioinformatics, 25(5):585-591, 2009.
11. W.K. Kim, J. Park, J.K. Suh, et al. Large scale statistical prediction of protein-protein interaction by potentially interacting domain (pid) pair. Genome Informatics Series, pages 42-50, 2002.
12. S.K. Ng, Z. Zhang, and S. H. Tan. Integrative approach for computationally inferring protein domain interactions. Bioinformatics, 19:923-929, 2002.
13. S.M. Gomez, W. S. Noble, and A. Rzhetsky. Learning to predict protein- protein interactios from protein sequences. Bioinformatics, 19:1875-1881, 2003.
14. L. Deng, J. Guan, Q. Dong, and S. Zhou. Prediction of protein-protein interaction sites using an ensemble method. BMC bioinformatics, 10(1):426, 2009.
15. X. Du and J. Cheng. Prediction of protein-protein interaction sites using granularity computing of quotient space theory. 1:324-328, 2008.