# SubSS: A Protein-Protein Interaction Detection Tool

Nazar Zaki[1], Safaai Deris[2] and Saleh Alwahaishi[3]
[1, 3] *College of Information Technology, UAE University, Al-Ain, P.O. Box: 17555, UAE*
[2] *Faculty of Computer Science and Information Systems, Universiti Teknologi Malaysia, Johor*
[1] *nzaki@uaeu.ac.ae*

## Abstract

*Many essential cellular processes are mediated by protein-protein interaction the reason why predicting protein-protein interaction has recently received considerable attention from biologist around the globe. In this paper, we present a computational tool for detecting protein-protein interaction based on substring similarity measure. Two proteins may interact by the mean of the similarities of the substrings they contain. This friendly and easy to use tool helps biologist to distinguish between high-confidence interactions from low-confidence or unknown interactions. The tool performance is tested on the currently available protein-protein interaction data for the yeast Saccharomyces cerevisiae, and it delivered considerable improvement over the existing techniques.*

## 1. Introduction

Protein-Protein interaction is a central problem in computational biology. Information about these interactions improves our understanding of diseases and it can provide the basis for new therapeutic approaches. To solve this problem, vast of approaches have already been developed. Some of the earliest techniques predict interacting proteins through the similarity of expression profiles [1], phylogenetic profiles [2] or trees [3], and studying the patterns of domain fusion [4]. However, it has been noted that these methods predict protein interactions in a general sense, meaning joint involvement in a certain biological process, and not necessarily actual physical interaction [5].

Most of the recent works focus on employing the protein domain knowledge to predict the protein-protein interaction. [6]-[9]. However, most of these methods focus on domain structure and none of them consider all the sequence information. We understand that protein domains are highly informative for predicting protein interaction as it reflects the potential structural relation-ships 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 computational tool known as SubSS for detecting protein-protein interaction. SubSS is motivated by the success of the recently published method on predicting protein-protein interaction based on substring sensitivity measure [10]. The idea behind the method is to predict protein interaction through sequence similarity. Two protein sequences may interact by the mean of the similarities of the substrings of amino acids they contain. It's based on the observation that, Smith-Waterman (SW) algorithm [11], which measures the similarity score between two sequences by a local gapped alignment. SW provides a relevant measure of similarity between proteins sequences which incorporates biological knowledge about protein evolutionary structural relationships [12]. In SubSS, we aim to provide an easy to use and versatile tool to detect protein-protein interaction based on amino acid substring sensitivity measure.

## 2. Implementation

The background algorithm of SubSS 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 against the protein sequences of interest. These features are then used in conjunction with support vector machines (SVMs) [13]-[14] to predict the possible proteins interactions. The overview of the algorithm is presented in Fig 1.

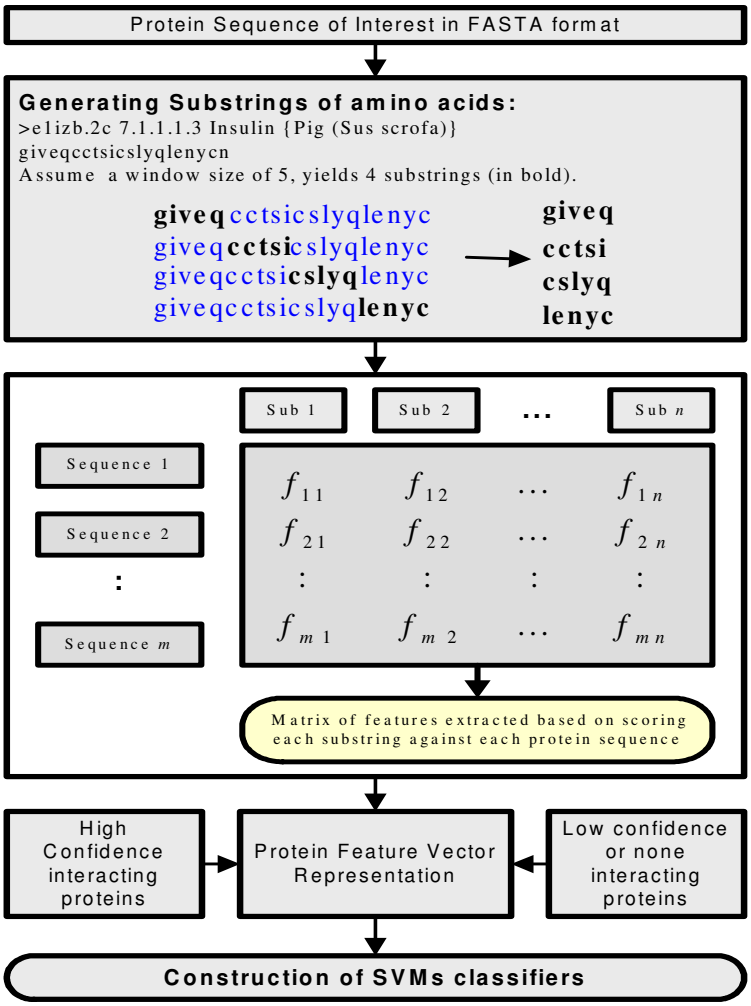


**Fig. 1:** Algorithm overview.

SubSS was built using visual basic.net under Microsoft Development Environment 2003. It performs its tasks in two manners, internal and external operations. SubSS interface visualizes the different options that could be used as arguments during the different phases and processes levels.

The tool needs three input files; the file contains the protein sequences of interest in a FASTA format, the high-confidence inter-acting proteins file and the low-confidence or unknown interacting proteins file, respectively. The tool has mainly three major phases. The first phase starts processing the data by generating the sub-strings dataset. This goal can easily be achieved by simply shifting a window of a size $k > 1$, over the protein examples. The number of substrings generated depends on the substring size. Following the preparation of the amino acids substrings, the sensitivity of each feature is measured using a simple pairwise sequence similarity algorithm as implemented in FASTA [15]. The tool's interface offers flexibility for user to change default parameters such as the gap opening/extension penalties, and the scoring matrix. Following the feature extraction step, the second Phase starts by concatenating the feature vectors of proteins based on whether the pair is interacting or not. If the concatenating proteins are confidently interacting, then, they will be included in the positive set; otherwise, they will be included in a negative set.

When the positive and negative sets are prepared, the program employs SVM to discriminate between the interacting and non-interacting proteins. In our implementation, we used Libsvm soft-ware implemented by Chang et al. [16]. SubSS interface offers flexiblity to change the SVM parameters, such as, the soft-margin parameter, the kernel function, feature selection options, and the cross-validation folds. In Fig 2, we show the SubSS user interface.

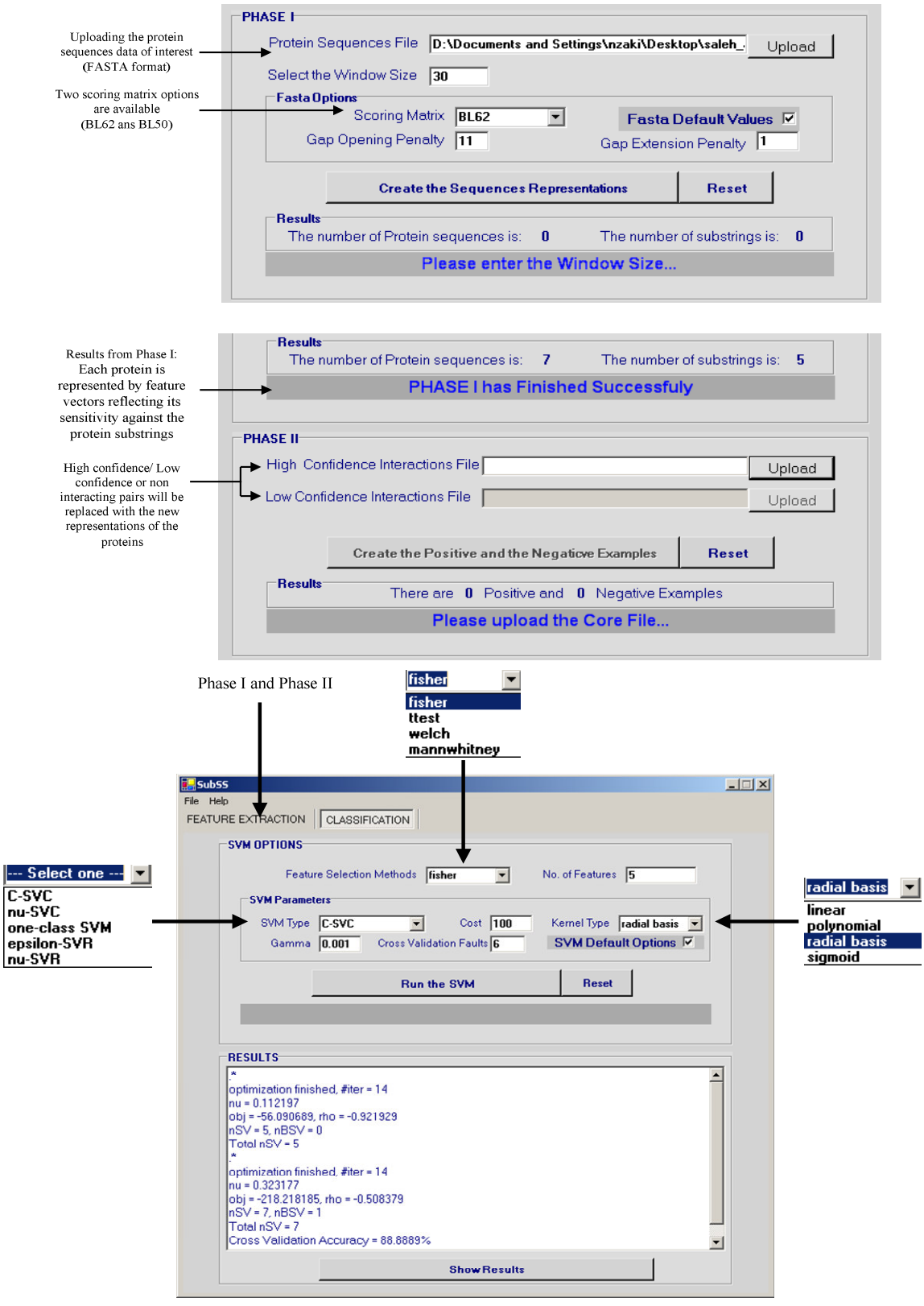


**Fig. 2:** SubSS user Interface.

## 3. Results and discussion

The performance of the tool could be tested by how well it can recognize the high-confidence interacting protein pairs, so the output is the classification accuracy. Therefore, the two evaluation measures used are:

- 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) [18]. 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, $fpp = \frac{tp}{(tp + fp)}$.

Where *tp* is the number of interacting sequences classified interacting, *tn* is the number of non-interacting sequences classified non- interacting, *fn* is the number of non-interacting sequences classified interacting, *fp* is the number of interacting sequences classified non- interacting and *n* here, is equal to the total of the $tp + fn + fp + tn$.

The performance is tested on the currently available protein-protein interaction data for the yeast Saccharomyces cerevisiae. 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) available at http://dip.doe-mbi.ucla.edu/. 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 [17].

In our testing; only default parameters are used such as BLOSUM 62 as scoring matrix with gap parameters set to 11 and 1, Gaussian Radial Basis Function kernel with scaling parameter set to 0.001, fisher algorithm for features selection, and 10-fold cross-validation.

The tool 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 size of 30 amino acids. Figs (3) and (4) show the comparison of different substring window sizes and their performance based on 10-fold cross validation and ROC.

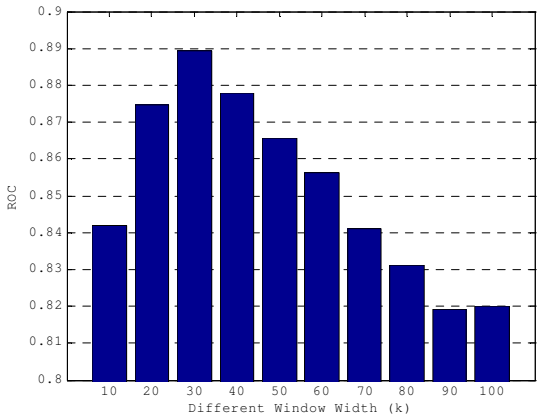


**Fig. 3:** comparing different window size values (k) based on the ROC scores.

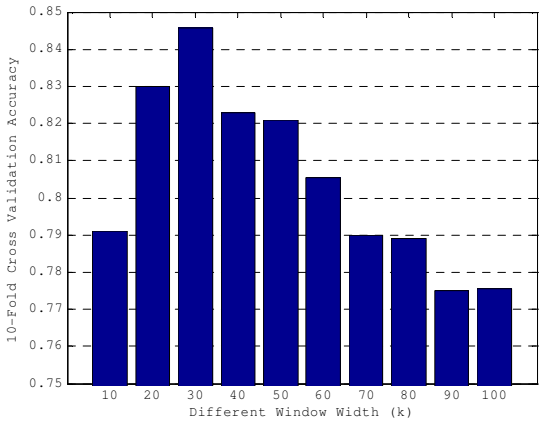


**Fig. 4:** comparing different window size values (k) based on 10-Fold cross validation accuracy.

Both two figures show that, 30 amino acids substring leads to a better results. We can also notice that as the window grow wider or smaller the performance decrease accordingly.

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.

In [7], the prediction system gives about 50% sensitivity and more than 98% specificity. [8] 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%. While in [9], the best result achieved was a ROC score of 0.818. It's clear that, SubSS is outperformed most of the existing methods with cross-validation accuracy of 84.57% and ROC score reaches 0.8892. However, the tool has two major limitations; First: despite the fact that, SW algorithm provides relevant measures of similarities between protein sequences, it has a down-side that, it is too slow since it depends on dynamic programming algorithm. Second; depending on only similarities between proteins is not enough evidence to predict the possible interaction.

## 4. Conclusion

The second and following pages should begin 1.0 inch (2.54 cm) from the top edge. On all pages, the bottom margin should be 1-1/8 inches (2.86 cm) from the bottom edge of the page for 8.5 x 11-inch paper; for A4 paper, approximately 1-5/8 inches (4.13 cm) from the bottom edge of the page.

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 behind the tool presented in this work is to predict protein-protein interaction through sequence similarity. Two protein sequences may interact by the mean of the similarities of the amino acids substrings they contain. The proposed tool termed SubSS, can effectively predict protein-protein interaction. This friendly and easy to use tool helps biologist to distinguish between high-confidence interactions from low-confidence or unknown interactions. The tool performance is tested on the currently available protein-protein interaction data for the yeast Saccharomyces cerevisiae, and it delivered considerable improvement. SubSS achieved cross-validation accuracy of 84.57% and ROC score reaches 0.8892. The accuracy of our tool comes from the combination of SVM algorithm and the SW 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 [12]. 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 our tool, SubSS has an important cost in terms of computation time. It includes an SVM optimization, which is roughly $O(n^2)$, where $n$ is the number of training set examples. The feature sensitivity measure phase of SubSS involves computing $n^2$ pairwise scores. Using SW, 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 SubSS on predicting protein-protein interaction encouraged us to plan future directions such as optimizing the substring width and finding suitable threshold score.

## 5. 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 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).

## 6. References

[1] 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.

[2] 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.

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

[4] 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.

[5] 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.

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

[7] 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.

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

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

[10] N. Zaki, S. Deris, & H. Alashwal, "Protein-protein interaction detection based on substring sensitivity measure", *International journal of biomedical sciences*, 1, pp: 1306-1216, 2006.

[11] T. Smith & M. Waterman, "Identification of common molecular subsequences", *J. of Molecular Biology*, 147, pp: 195-197, 1981.

[12] H. Saigo, J. Vert, N. Ueda, & T. Akutsu, "Protein homology detection using string alignment kernels", *Bioinformatics*, 11, pp: 1682-1689, 2004.

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

[14] V. N. Vapnik "Statistical Learning Theory," Wiley, 1998.

[15] W. R. Pearson, "Rapid and sensitive sequence comparisons with FASTAP and FASTA Method", *Enzymol*, 183, pp: 63, 1985.

[16] C. C. Chang & C. L. Lin, LIBSVM: a library for support vector machines, 2001. Software available at http://www.csie.ntu.edu.tw/~cjli-n/libsvm.

[17] 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.

[18] Swets, "Measuring the accuracy of diagnostic systems," Science, 270, pp: 1285-1293, 1988.