Prediction for Essential Proteins with the Support Vector Machine*

Zih-Jie Yang, Chang-Biau Yang† and Chiou-Ting Tseng
Department of Computer Science and Engineering
National Sun Yat-sen University
Kaohsiung 80424, Taiwan

***Abstract*—Essential proteins affect the cellular life deeply, but it is hard to identify them. Protein-protein interaction is one of the ways to disclose whether a protein is essential or not. We notice that many researchers use the feature set composed of topology properties from protein-protein interaction to predict the essential proteins. However, the functionality of a protein is also a clue to determine its essentiality. The goal of this paper is to build SVM models for predicting the essential proteins. In our experiments, we download Scere20070107, which contains 4873 proteins and 17166 interactions, from DIP database. The ratio of essential proteins to nonessential proteins is nearly 1:4, so it is imbalanced. In the imbalanced dataset, the best values of F-measure, MCC, AIC and BIC of our models are 0.5197, 0.4671, 0.2428 and 0.2543, respectively. We build another balanced dataset with ratio 1:1. For balanced dataset, the best values of F-measure, MCC, AIC and BIC of our models are 0.7742, 0.5484, 0.3603 and 0.3828, respectively. Our results are superior to all previous results with various measurements.**

***Index Terms*—bioinformatics; essential protein; protein-protein interaction; support vector machine; feature set**

I. Introduction

Protein plays an important role in our life. Proteins join all cell activities, including cell budding, conjugation, cytokinesis, etc. The *protein-protein interaction* (PPI) is one of the significant characteristics of a protein. In the past, finding PPI is a time-consuming work. Recently, with the *yeast two-hybrid* [15] high-throughput technique, which can find a lot of PPIs in one experiment, it becomes easier to get the PPI information.

A PPI network is similar to a social network, thus some researchers apply some social network techniques to the PPI network [21]. Some researchers have focused on PPI networks, such as prediction and classification of protein functionality [19], analysis of protein phenotype [7], etc. Prediction of *protein essentiality* is one of the studies on protein phenotype. When an essential protein is removed, it will cause the cell to lose its life or functionality because the function of the essential protein cannot be replaced by other proteins. Essential proteins can be identified by the experiment with the technique of gene deletion [5], RNA interference [11] and combination of gene replacement and conditional gene expression, but all of them spend a lot of time and resources. Previous study [4] shows that essential proteins and nonessential proteins have discrimination in the topological properties of the PPI network.

Several researchers have developed some methods to solve the essential protein classification problem. Chin *et al.* [8] proposed a double screening scheme and built a framework called *hub analyzer* (http://hub.iis.sinica.edu.tw/Hubba/index.php) to rank the proteins. Hwang *et al.* [14] applied the *support vector machine* (SVM) [10] to classify the proteins. Marcio *et al.* [1] used *Waikato Environment for Knowledge Analysis* (WEKA) [22] to predict the essential proteins.

The goal of this paper is to find the essential proteins in a PPI network. We get the PPI dataset Scere20070107, which contains 4873 proteins and 17166 interactions, from the DIP database. The LIBSVM tool [13] is used to predict the essential proteins. Our feature set consists of the features extracted from the methods proposed by Chin *et al.* [8], Hwang *et al.* [14], Lin *et al.* [17] and Marcio *et al.* [1]. We build 45 SVM models with various subsets of features. Each model is performed on

*This research work was partially supported by the National Science Council of Taiwan under contract NSC100-2221-E-110 -050.
†Corresponding author: cbyang@cse.nsysu.edu.tw

both imbalanced and balanced datasets. The best values of F-measure, MCC, AIC and BIC of our experiments on the imbalanced dataset are 0.5197, 0.4671, 0.2428 and 0.2543, respectively. For the balanced dataset, we get 0.7742, 0.5484, 0.3603 and 0.3828, respectively. The best of our models outperforms other previous methods with various performance measurements.

The rest of this paper is organized as follows. In Section II, we will introduce some preliminary knowledge. In Section III, we will present our method. Section IV shows the experimental results and compares them with some previous results. Finally, Section V gives a conclusion and some possible future works.

II. Preliminaries

In this section, we will introduce some related works. We will introduce the support vector machine (SVM), the position specific scoring matrix (PSSM), and the performance measurements we use.

A. Support Vector Machine

The *support vector machine* (SVM) [9] is a method that can classify a set of samples into two classes. Let $\{(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)\}$ denote the set of samples, where x_i is the feature vector of sample i, y_i is its label, n denotes the number of samples and d denotes the dimension of x_i. SVM is to find a hyperplane which has the maximal margin to separate the samples with labels +1 and -1. The margin is defined as the shortest distance from each feature vector to the support hyperplane. The SVM provides some various *kernel functions* to transform the original feature space into higher dimensional space. Then, the cases will become nearly linearly separable. Four kernel functions are usually used, including *linear*, *polynomial*, *radial basis function* and *sigmoid*.

B. Position Specific Scoring Matrix

Altschul *et al.* [3] proposed the *Position Specific Scoring Matrix* (PSSM). The *basic local alignment search tool* (BLAST) [2] and *position specific iterated-BLAST* (PSI-BLAST) are two popular tools related to PSSM, both implemented by National Center for Biotechnology Information (NCBI) (http://www.ncbi.nlm.nih.gov/). Querying primary biological sequences is the main task of the BLAST. PSI-BLAST is based on BLAST. It first obtains the list of similar sequences produced by BLAST. Then, PSI-BLAST produces a profile of the list, and uses it as an input to perform the next iteration in PSI-BLAST. The PSSM is calculated by PSI-BLAST in each iteration. The scores of PSSM represent the occurrence probabilities over the background probability.

C. Performance Evaluation

We use *receiver operating characteristic* (ROC) curve and *area under curve* (AUC) value to evaluate the classifiers. And we measure the prediction result generated by a classifier by various kinds of accuracy measurements, including *precision*, *recall*, *F-measure*, *Matthews correlation coefficient* (MCC) and *percentage* of essential proteins. The formulas are given as follows.

1. Precision: $\frac{TP}{TP+FP}$
2. Recall: $\frac{TP}{TP+FN}$
3. F-measure: $\frac{2\times precision\times recall}{precision+recall}$
4. MCC: $\frac{TP\times TN-FP\times FN}{\sqrt{(TP+FN)(TP+FP)(TN+FP)(TN+FN)}}$
5. Percentage of essential protein: $\frac{Number\ of\ essential\ proteins\ in\ top\ n}{n}$.

In the above, TP (*true positive*) denotes the number of true objects (essential proteins) correctly predicted, FP (*false positive*) denotes the number of false objects (nonessential proteins) wrongly predicted, FN (*false negative*) denotes the number of true objects (essential proteins) wrongly predicted, TN (true negative) denotes the number of false objects (nonessential proteins) correctly predicted and n is the number of proteins.

With *Akaike's information criterion* (AIC) and *Bayesian information criterion* (BIC) [12], if a classifier involves more features, it will get more penalty. The formulas of AIC and BIC are given as follows.

1. AIC: $\frac{\sum_{i=1}^{N} err(i)}{N} + \frac{2\times(k+1)}{N}$
2. BIC: $\frac{\sum_{i=1}^{N} err(i)}{N} + \frac{(\log N)\times(k+1)}{N}$

In the above, $err(i)$ denotes the error of data element i, N denotes the number of samples in the dataset and k denotes the number of features. By the definitions, BIC gets more penalty on the number of features than AIC.

TABLE I
THE NAMES AND AUC VALUES OF PROTEIN PROPERTIES.

Rank	Property name	Type	Size	AUC value
1	Phyletic retention [14]	O	1	0.7046
2	Bit string of double screening scheme	T	1	0.659
3	Amino acid occurrence [17]	S	20	0.6559
4	Nucleus [1]	P	1	0.6534
5	Betweenness centrality related to physical interactions [1]	T	1	0.62
6	Neighbors' intra-degree [14]	T	1	0.6181
7	Essential index [14]	O	1	0.6171
8	Clique level [14]	T	1	0.6141
9	Degree related to all interactions [16]	T	1	0.6067
10	Common function degree [14]	O	1	0.6058
11	Clustering coefficient [14]	T	1	0.5981
12	Betweenness centrality related to all interactions [24]	T	1	0.5981
13	Other process [1]	P	1	0.598
14	Density of maximum neighborhood component [8]	T	1	0.5859
15	Maximum neighborhood component [8]	T	1	0.5841
16	Closeness centrality [23]	T	1	0.5689
17	Degree related to physical interactions [1]	T	1	0.5686
18	Edge percolated component [7]	T	1	0.5584
19	Other localization [1]	P	1	0.5582
20	Open reading frames length [14]	O	1	0.5449
21	Cytoplasm [1]	P	1	0.534
22	Average amino acid PSSM [17]	S	20	0.5325
23	Cell cycle [1]	P	1	0.5313
24	Transcription [1]	P	1	0.5298
25	Mitochondrion [1]	P	1	0.517
26	Metabolic process [1]	P	1	0.5119
27	Bottleneck [19] [24]	T	1	0.5107
28	Cysteine count [17]	S	1	0.5094
29	Endoplasmic reticulum [1]	P	1	0.5093
30	Cysteine odd-even index [17]	S	1	0.5088
31	Average hydrophobic [17]	S	1	0.506
32	Signal transduction [1]	P	1	0.505
33	Average cysteine position [17]	S	1	0.5032
34	Outdegree related to metabolic interaction [1]	T	1	0.5032
35	Outdegree related to transcriptional regulation interaction [1]	T	1	0.4993
36	Indegree related to metabolic interaction [1]	T	1	0.499
37	Average distance of every two cysteines [17]	S	1	0.4973
38	Transport [1]	P	1	0.4971
39	Betweenness centrality related to metabolic interactions [1]	T	1	0.4964
40	Protein length [17]	S	1	0.4958
41	Betweenness centrality transcriptional regulation interactions [1]	T	1	0.4956
42	Identicalness [1]	O	1	0.4936
43	Indegree related to transcriptional regulation [1]	T	1	0.4913
44	Cysteine location [17]	S	5	0.4867
45	Average hydrophobicity around cysteine [17]	S	4	0.4832

III. Our Method

In this section, we will introduce our features and our prediction method with SVM.

A. Feature Extraction

The feature set includes *topological properties* (T), such as bit string of double screening scheme and betweenness centrality related to physical interactions; *protein properties* (P), such as cell cycle and metabolic process; *sequence properties* (S), such as amino acid occurrence and average amino acid PSSM; and *other properties* (O), such as phyletic retention and essential index. There are totally 45 properties, including 90 features, as shown in Table I.

Lin *et al.* [18] and Chin *et al.* [8] proposed the double screening scheme. They use a ranking scheme to find the essential proteins. But each protein does not have a unique score. Thus, we propose a bit string implementation to transform the ranking to a score of protein. We select two topological properties for ranking, A and B. For the n target proteins, we rank them by A and B individually. Then, $\frac{n}{2}$ iterations are performed to construct the bit vector for each protein. A 2D-bit array M is built, where each bit $M[i,j]$ corresponds to a protein j in iteration i. In the ith iteration, we find the top i ranked proteins, described as follows. We select the top $2i$ proteins by ranking method A, then we select the top i proteins by ranking method B among the $2i$ proteins. If a protein j is selected, we set its $M[i,j]$ to 1; otherwise, $M[i,j]$ is set to 0. Hence, each protein will have $\frac{n}{2}$ bits and its score is the sum of its bits.

An example of our bit string implementation is shown in Tables II and III. In the first iteration, our goal is to find the top 1 essential protein by double screening scheme. We first select top 2 essential proteins by ranking method A, which are W and X. Next, W and X are 2nd and 1st ranked by B, respectively. Hence, in the first iteration, X is finally selected. Then, we set the bit $M[1,X]$ for the first iteration to 1, and the others are set to 0. In the second iteration, the top 2 essential proteins are desired to be found. First, four proteins W, X, Y and Z are selected, because they are the top 4 by ranking method A. By ranking B, we select the top 2 proteins from them, which are X and Y. Hence, the bits $M[2,X]$ and $M[2,Y]$ are 1, and the others are 0 in the second iteration. Finally, we sum up the bits of each protein, as shown in the fourth column of Table III.

TABLE II
RANKING BY TWO DIFFERENT METHODS.

	Ranking method	
Protein name	A (DMNC)	B (MNC)
W	1st	4th
X	2nd	2nd
Y	3rd	1st
Z	4th	3rd

TABLE III
BIT STRING BY THE DOUBLE SCREENING SCHEME.

	ith iteration				
Protein name	1st	2nd	Sum of bit string	$n-r$	Sum
W	0	0	0	0	0
X	1	1	2	2	4
Y	0	1	1	3	4
Z	0	0	0	1	1

There is still a problem in our bit string implementation, the number of proteins scarcely being selected is around $\frac{n}{2}$. Thus, about $\frac{n}{2}$ sums are close to 0. It would cause SVM being difficult to classify these $\frac{n}{2}$ proteins. In order to overcome this problem, for each protein, we add another score $n-r$ to the sum of the bit vector, where r is the rank of the protein by ranking method B. In this paper, we use DMNC as ranking A and MNC as ranking B. In this example, $n=4$, so the values $n-r$ of W, X, Y and Z are 0, 2, 3 and 1, respectively. We sum it up with the bit string, hence the final scores are 0, 4, 4 and 1.

B. Our Method with SVM

This paper uses the SVM program developed by Chang and Lin [6], called *LIBSVM* [13]. We apply *10-fold cross-validation* for 10 times on the dataset. The procedure of our method is described as follows.

Step 1. Retrieve the protein-protein interaction dataset from the DIP database.
Step 2. Extract features described in Section III-A.
Step 3. Measure the performance of each property.
Step 4. Build 45 feature sets (models) by adding

the properties one by one with the decreasing order of their performance.

Step 5. Perform 10-fold cross-validation for 10 times on the imbalanced dataset and balanced dataset, respectively.

IV. Experimental Results

In this section, we will show our experimental results and compare them with the methods proposed by Hwang *et al.* [14], Marcio *et al.* [1] and Chin *et al.* [8].

A. PPI Dataset

We download the PPI data from the DIP (http://dip.doe-mbi.ucla.edu/) database [20] and use Scere20070107, which contains 4873 proteins and 17166 interactions. We consider the largest connected component in the PPI, which has 4815 proteins, including 975 essential proteins and 3840 nonessential proteins. The information of essential proteins is extracted from the SGD. In August 2011, there are 1222 essential proteins in SGD (http://www.yeastgenome.org/).

B. Data Balancing

There are 975 essential proteins and 3840 nonessential proteins in our dataset. The ratio of essential proteins to nonessential proteins is low, nearly 1:4, which will lead to biased fitting to nonessential proteins. Hence, we build another dataset with a balanced situation. We randomly select 975 nonessential proteins and mix with the essential proteins to form the balanced dataset.

C. Experimental Results and Comparison

We first calculate the AUC value for each property, and then rank the AUC values, as shown in Table I. We build 45 feature subsets (models) for imbalanced and balanced datasets. Each model is formed by gradually including the properties one by one with the decreasing order of AUC values. In other words, the ith model is composed of the properties with the top i AUC values. And then we run 10-fold cross-validation for 10 times with SVM on imbalanced datasets and balanced datasets. Table IV show our results of the dataset Scere20070107 in imbalanced dataset, and compare them with the results of Hwang *et al.* [14] and Marcio *et al.* [1]. Table V show the results of balanced dataset. The value with an underline means that it has the highest performance under that measurement. Due to page limit, we only keep those lines with peak values in at least one of the measurements.

To compare our results with other previous results, we also get the feature sets used by Hwang *et al.* and Marcio *et al.*, and rebuild their models by LIBSVM. In redoing their experiments, the parameters c and γ of LIBSVM are tuned to get the best result and we execute 10-fold cross-validation for 10 times to get the average performance. The job for parameter tuning and repeatedly testing is the same as our experiments.

As shown in Table IV, the performance of our models in the imbalanced dataset is better than Hwang *et al.* and Marcio *et al.* starting from the 7th model. The best values of AUC, precision, recall, F-measure and MCC of our models are 0.8413, 0.7504, 0.4031, 0.5197 and 0.4671, respectively. The best F-measure and MCC occur at the 36th model, that is, we can select this model to get the best accuracy.

For the balanced dataset, starting from the 7th model, the results are also better than the results of Hwang *et al.* and Marcio *et al.*, as shown in Table V. The best values of AUC, precision, recall, F-measure and MCC we get are 0.8502, 0.7872, 0.8069, 0.7742 and 0.5484, respectively. The 32nd model has the best F-measure and MCC. In other words, if we desire to get the highest prediction accuracy, we can use the 32nd model.

As one can see, the recall of the imbalanced dataset is lower than that of the balanced dataset. The ratio of essential proteins to nonessential proteins in the imbalanced is 1:4. The SVM model for the imbalanced dataset could favor nonessential proteins, so the prediction would have more nonessential proteins, even some of them are truly essential. Hence, the imbalanced dataset has lower recall than the balanced dataset has.

In addition to accuracy comparison, we also use AIC and BIC to evaluate all models. Tables IV and V also show AIC and BIC comparisons of our models with Hwang *et al.* and Marcio *et al.* Some of our models have AIC and BIC values lower than those of Hwang *et al.* and Marcio *et al.* The lowest values of AIC and BIC for the imbalanced case are

TABLE IV
THE PERFORMANCE COMPARISON OF THE IMBALANCED DATASET.

*i*th model (number of features)	AUC	Precision	Recall	F-measure	MCC	AIC	BIC
Hwang *et al.* (10)	0.7781	0.7382	0.3299	0.4525	0.4179	0.2608	0.2646
Marcio *et al.* (23)	0.7245	0.6574	0.1714	0.2716	0.267	0.3016	0.31
1st (1)	0.7046	0.6049	0.2513	0.3551	0.3035	0.276	0.2767
2nd (2)	0.7338	0.6494	0.3143	0.4236	0.3666	0.2626	0.2636
3rd (22)	0.7616	0.6495	0.3202	0.4289	0.3704	0.2685	0.2765
4th (23)	0.7531	0.6709	0.2987	0.4131	0.3668	0.269	0.2774
5th (24)	0.7829	0.6706	0.2968	0.4111	0.3653	0.2653	0.274
6th (25)	0.7539	0.6739	0.3024	0.4172	0.3707	0.2698	0.2789
7th (26)	0.7851	0.748	0.3416	0.4686	0.4331	0.2509	0.2603
8th (27)	0.7849	0.7504	0.3481	0.4754	0.4389	0.2519	0.2617
9th (28)	0.7765	0.7401	0.3662	0.4898	0.4463	0.2517	0.2618
10th (29)	0.8123	0.7333	0.3783	0.4977	0.4502	0.246	0.2565
11th (30)	0.7814	0.7368	0.3685	0.4911	0.4462	0.2533	0.2641
12th (31)	0.8122	0.7373	0.3703	0.4917	0.4469	0.247	0.2582
13th (32)	0.8219	0.7436	0.3711	0.4948	0.4513	0.2428	0.2543
34th (72)	0.8371	0.7237	0.4031	0.5162	0.4614	0.2556	0.2812
36th (74)	0.8376	0.7333	0.4026	0.5197	0.4671	0.2547	0.2809
40th (78)	0.8413	0.727	0.3938	0.51	0.4576	0.2591	0.2867

TABLE V
THE PERFORMANCE COMPARISON OF THE BALANCED DATASET.

*i*th model (number of features)	AUC	Precision	Recall	F-measure	MCC	AIC	BIC
Hwang *et al.* (10)	0.7781	0.7737	0.7019	0.7356	0.4985	0.3839	0.3912
Marcio *et al.* (23)	0.7245	0.6951	0.7125	0.7036	0.4004	0.4319	0.4478
1st (1)	0.7046	0.7306	0.6903	0.7098	0.4365	0.3921	0.3934
2nd (2)	0.7338	0.7701	0.6674	0.715	0.4725	0.3879	0.3899
3rd (22)	0.7616	0.7409	0.7137	0.7268	0.4643	0.396	0.4113
4th (23)	0.7531	0.7426	0.7593	0.7506	0.4964	0.382	0.3979
5th (24)	0.7829	0.7494	0.723	0.7357	0.4813	0.3872	0.4037
6th (25)	0.7539	0.7384	0.7478	0.743	0.4833	0.3896	0.4068
7th (26)	0.7851	0.7625	0.7325	0.747	0.5049	0.375	0.3929
8th (27)	0.7849	0.7646	0.7358	0.7499	0.5099	0.3747	0.3932
9th (28)	0.7765	0.7495	0.7425	0.7459	0.4947	0.3833	0.4024
10th (29)	0.8123	0.7872	0.7119	0.7474	0.5219	0.3627	0.3826
13th (32)	0.8219	0.7854	0.7299	0.7565	0.5321	0.361	0.3828
16th (35)	0.8194	0.781	0.7538	0.767	0.5429	0.3603	0.3841
32nd (70)	0.8502	0.7741	0.7744	0.7742	0.5484	0.3883	0.4352

0.2428 and 0.2543, respectively, which occur at the 13th model. For the balanced dataset, the lowest AIC is 0.3603 in the 16th model and the lowest BIC is 0.3828 in the 13th model.

The prediction method proposed by Chin *et al.* [8] gives a score to each target protein. Thus, the protein ranks can be obtained by the scores. LIBSVM also gives us a probability in the two-class classification, which represents the confidence of the classification. We use these probabilities to rank the proteins which are predicted as essential ones. Table VI show the percentages of proteins which are predicted as essential in the top ranked proteins in the imbalanced experiment. The value with an underline is the highest. In the balanced dataset, since nearly half of the proteins are not put into the dataset, we cannot get the probabilities for all proteins. Hence, we do not this ranking experiment

for the balanced case.

As shown in Table VI, the accuracies of essential proteins in predicted TOP 100, TOP 30%, TOP 50%, TOP 80% and TOP 100% of our models is better than Chin *et al.*, Hwang *et al.* and Marcio *et al.* starting from the 7th model, whose accuracies are 0.922, 0.8259, 0.734, 0.6283 and 0.5733, respectively. Here, TOP 30 % means the top 293 proteins, since there are 975 essential proteins in the dataset and $975 \times 30\% = 293$.

As the experimental results show, the performance measurements of our models are almost better than others beginning from the 7th model. The 7th model is composed of phyletic retention, bit string of the double screening scheme, amino acid occurrence, nucleus, betweenness centrality related to physical interactions, neighbors' intra-degree and essential index. With these properties, we get a conclusion that essential proteins have the following characteristics. An essential protein has more ortholog. The amino acid composition, cellular component of nucleus and interaction type are important factors to identify essential proteins. By neighbors' intra-degree and essential index, the essential proteins are clustered. The bit string implementation of the double screening scheme is also a good property for predicting essential proteins.

V. Conclusion and Future Work

In this paper, we apply the SVM classification to the identification of essential proteins. We build 45 SVM models for imbalanced dataset and balanced dataset. To compare with some previous results, we invoke several kinds of performance measurements. The F-measure, MCC, AIC and BIC of our experiment on the imbalanced dataset are 0.5197, 0.4671, 0.2428 and 0.2543, respectively. For getting high accuracy, we suggest the 36th model (74 features). If one would focus on the feature set with small size, we suggest to take the 13th model (32 features). For the balanced dataset, we get 0.7742, 0.5484, 0.3603 and 0.3828, respectively. We suggest the 32nd model for high accuracy, and the 13th and16th models for fewer features.

In our experimental results, we discover that the performance of our models is better than others starting from the 7th model. We think that the essential proteins have more ortholog and they are clustered.

The features used by Lin *et al.* [17] have high accuracy for predicting disulfide bond state. Essential proteins may depend on the protein functions and the disulfide bond will affect the protein functions and structure. So, we also involve their features and expect that a good accuracy on essential protein prediction is obtained. However, most of these features do not have significant effect on predicting essential proteins. From our experimental results, we think that the prediction of essential proteins is not so much related to disulfide bond states.

There are some possible ways to improve the prediction accuracy. We may try to cluster the proteins together with similar feature values before the prediction job. Since the essential proteins may have more ortholog and they are clustered, we can do protein classification according to the number of orthologs and degrees of the proteins. The dataset may be clustered into several subsets. We can perform the prediction task for each subset independently to get higher accuracy. We may also try to find more features related to the proteins essentiality and try to invoke other tools or hybrid tools to improve the performance.

References

[1] M. L. Acencio and N. Lemke, "Towards the prediction of essential genes by integration of network topology, cellular localization and biological process information," *BMC Bioinformatics*, vol. 10, no. 1, pp. 290–307, 2009.

[2] S. F. Altschul, W. Gish, W. Miller, E. W. Myers, and D. J. Lipmanl, "Basic local alignment search tool," *Journal of Molecular Biology*, vol. 215, no. 3, pp. 403–410, 1990.

[3] S. F. Altschul, T. L. Madden, A. A. Schaffer, J. Zhang, Z. Zhang, W. Miller, and D. J. Lipman, "Gapped BLAST and PSI-BLAST: a new generation of protein database search programs," *Nucleic Acids Research*, vol. 25, no. 17, pp. 3389–3402, 1997.

[4] A. L. Barabasi and Z. N. Oltvai, "Network biology: understanding the cell's functional organization," *Nature Reviews Genetics*, vol. 5, no. 2, pp. 101–113, 2004.

[5] K. M. Cadigan, U. Grossniklaus, and W. J. Gehring, "Functional redundancy: The respective roles of the two sloppy paired genes in drosophila segmentation," *Proceedings of the National Academy of Sciences of the United States of America*, vol. 91, no. 14, pp. 6324–6328, 1994.

[6] 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.

[7] C.-S. Chin and M. P. Samanta, "Global snapshot of a protein interaction networka percolation based approach," *Bioinformatics*, vol. 19, pp. 2413–2419, 2003.

TABLE VI
THE PERCENTAGE OF ESSENTIAL PROTEINS IN THE IMBALANCED DATASET.

	Percentages of essential proteins in predicted n proteins				
ith model (number of features)	TOP 100 (10.25%)	TOP 30%	TOP 50%	TOP 80%	TOP 100%
Chin *et al.* (1)	0.64	0.5324	0.498	0.4897	0.4215
Hwang *et al.* (10)	0.82	0.7843	0.7223	0.6047	0.5586
Marcio *et al.* (23)	0.693	0.6478	0.5654	0.4858	0.46
1st (1)	0.694	0.6471	0.6008	0.5477	0.5349
2nd (2)	0.757	0.6945	0.6451	0.5722	0.5319
3rd (22)	0.799	0.6843	0.6459	0.5719	0.5318
4th (23)	0.798	0.7222	0.6609	0.5822	0.535
5th (24)	0.819	0.7092	0.6621	0.5832	0.5393
6th (25)	0.804	0.7222	0.6658	0.5844	0.5357
7th (26)	0.922	0.8259	0.734	0.6283	0.5733
8th (27)	0.915	0.8198	0.7342	0.6272	0.5723
9th (28)	0.9	0.8116	0.7373	0.6306	0.5677
10th (29)	0.905	0.8171	0.7387	0.6308	0.5795
29th (67)	0.895	0.8468	0.749	0.6403	0.5887
32nd (70)	0.898	0.8406	0.7492	0.6463	0.5931
37th (75)	0.891	0.8382	0.7512	0.6399	0.5907

[8] C.-H. Chin, C.-W. Ho, and M.-T. Ko, "Prediction of essential proteins and functional modules from protein-protein interaction networks," Ph.D. dissertation, National Central University, Chung-Li, Taiwan, 2010.

[9] C. Cortes and V. Vapnik, "Support-vector networks," *Machine Learning*, vol. 20, no. 2, pp. 273–297, 1995.

[10] N. Cristianini and J. Shawe-Taylor, *An Introduction to Support Vector Machines and other kernel-based learning methods*. Cambridge University Press, 2000.

[11] L. M. Cullen and G. M. Arndt, "Genome-wide screening for gene function using rnai in mammalian cells," *Immunology and Cell Biology*, vol. 83, no. 3, pp. 217–223, 2003.

[12] T. Hastie, R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning: Data Mining, Inference, and Prediction.*, 2nd ed., 2009.

[13] C.-W. Hsu, C.-C. Chang, and C.-J. Lin, "A practical guide to support vector classification," http://www.csie.ntu.edu.tw/∼cjlin/papers/guide/guide.pdf.

[14] Y.-C. Hwang, C.-C. Lin, J.-Y. Chang, H. Mori, H.-F. Juan, and H.-C. Huang, "Predicting essential genes based on network and sequence analysis," *Molecular BioSystems*, vol. 5, no. 12, pp. 1672–1678, 2009.

[15] T. Ito, T. Chiba, R. Ozawa, M. Yoshida, M. Hattori, and Y. Sakaki, "A comprehensive two-hybrid analysis to explore the yeast protein interactome," *Proceedings of the National Academy of Sciences of the United States of America*, vol. 98, no. 8, pp. 4569–4574, 2001.

[16] H. Jeong, S. P. Mason, A.-L. Barabsi, and Z. N. Oltvai, "Lethality and centrality in protein networks," *Nature*, vol. 411,, pp. 41–42, 2001.

[17] C.-Y. Lin, C.-B. Yang, C.-Y. Hor, and K.-S. Huang, "Disulfide bonding state prediction with svm based on protein types," *Bio-Inspired Computing: Theories and Applications*, pp. 1436–1442, 2010.

[18] C.-Y. Lin, C.-H. Chin, H.-H. Wu, S.-H. Chen, C.-W. Ho, and M.-T. Ko, "Hubba: hub objects analyzer- framework of interactome hubs identification for network biology," *Nucleic Acids Research*, vol. 36, pp. W438–W443, 2008.

[19] N. Pržulj, D. Wigle, and I. Jurisica, "Functional topology in a network of protein interactions," *Bioinformatics*, vol. 20, pp. 340–348, 1998.

[20] L. Salwinski, C. S. Miller, A. J. Smith, F. K. Pettit, J. U. Bowie, and D. Eisenberg, "The database of interacting proteins: 2004 update." *Nucleic Acids Research*, vol. 32, pp. D449–D451, 2004.

[21] S. Wasserman and K. Faust, *Social Network Analysis: Methods and Applications*. Cambridge University Press, 1994.

[22] I. H. Witten and E. Frank, *Data Mining:Practical Machine Learning Tools and Techniques with Java Implementations*. Morgan Kaufmann, 2000.

[23] S. Wuchty and P. F. Stadle, "Centers of complex networks," *Journal of Theoretical Biology*, pp. 45–53, 2003.

[24] H. Yu, P. M. Kim, E. Sprecher, V. Trifonov, and M. Gerstein, "The importance of bottlenecks in protein networks: Correlation with gene essentiality and expression dynamics," *PLoS Computational Biology*, vol. 3, pp. 713–720, 2007.