Automatic Annotation of Protein Functional Class from Sparse and Imbalanced Data Sets

Jaehee Jung[1] and Michael R. Thon[1,2]

[1] Department of Computer Science,
[2] Department of Plant Pathology & Microbiology,
Texas A&M University,College Station, TX, 77843 USA
{jaeheejung, mthon}@tamu.edu

Abstract. In recent years, high-throughput technologies such as DNA sequencing and microarrays have created the need for automated annotation and analysis of large sets of genes. The Gene Ontology (GO) provides a common controlled vocabulary for describing gene function however the process for annotating proteins with GO terms is usually through a tedious manual curation process by trained profession annotators. With the wealth of genomic data that are now available, there is a need for accurate automated annotation methods. In this paper, we propose a method for automatically predicting GO terms for proteins by applying statistical pattern recognition techniques. We employ protein functional domains as features and learn independent Support Vector Machine classifiers for each GO term. This approach creates sparse data sets with highly imbalanced class distribution. We show that these problems can be overcome with standard feature and instance selection methods. We also present a meta-learning scheme that utilizes multiple SVMs trained for each GO term, resulting in improved overall performance than either SVM can achieve alone.

Key Word: Gene Annotation, Feature Selection, Gene Ontology, InterPro, Imbalanced Data

1 Introduction

In recent years, high-throughput genome sequencing and gene annotation methods have resulted in the availability or large sets of genes and predicted gene products (proteins) and to a large extent, the functions of many of these genes are still unknown, i.e. they are unannotated. Biologists deduce protein function through experimentation and as such, knowledge of gene function derived in this fashion is laborious and inexpensive. Given the wealth of genome data that are available now, one of the central problems facing researchers is the accurate prediction of protein function based on computationally obtained features of the proteins and the genes from which they are derived. Such computationally predicted functions are useful to guide laboratory experimentation and as an interim annotation, until protein function can be validated experimentally. Traditionally, protein function is expressed as free text descriptions but recently controlled vocabularies of various types have been employed. The Gene Ontology (GO) [22] provides a controlled vocabulary or terms for annotating proteins. In addition, the GO consortium describes the relationships among the terms with a directed acyclic graph (DAG), providing a rich framework for describing the function of proteins. GO terms are often assigned to proteins by teams of curators, who examine references in the scientific literature as well as features of the proteins. One of the central problems facing computational biologists is how to emulate this process.

As the need for GO annotation increases, various kinds of annotation systems are being developed for automated prediction of GO terms. Most methods rely on the identification of similar proteins in large databases of annotated proteins. GOtcha [12] utilizes properties of the protein sequence similarity search results (BLAST) such as the p-score, for predicting an association between the protein and a set of nodes in the GO graph. Several other recently described methods, including GOFigure [8], GOblet [5], and OntoBlast [19] depend on sequence similarity searches of large databases to obtain features that are used for predicting GO terms. These tools employ only blast results as attributes for prediction of GO terms, however, several systems utilize features besides blast search results. Vinayagam et el. [14, 15] suggest a method to predict GO terms using SVM and feature sets including sequence similarity, frequency score the GO terms, GO term relationship between similar proteins. Al-shahib et el. [1] use amino acid composition, amino acid pair ratios protein length, molecular weight, isoelectric point, hydropathy and aliphatic index as features for SVM classifiers to predict protein function. King et el. [9] employ not only blast sequence similarity, but also bio-chemical attributes such as molecular weight, and percentage amino acid content. Pavlidis et el. [13] predict gene function from heterogeneous data sets derived from DNA microarray hybridization experiments and phylogenetic profiles.

A number of different methods have been developed to identify and catalog protein families and functional domains which serve as useful resources for understanding protein function. The European Bioinformatics Institute (EBI) has created a federated database called InterPro (IPR) [23] which serves as a central reference for several protein family and functional domain databases, including Prosite, Prints, Pfam, Prodom, SMART, TIGRFams and PIR SuperFamily. InterPro families and functional domains are usually assigned to proteins using a variety of automated search tools. In addition, the InterPro consortium also maintains an InterPro to GO translation table that allows GO terms to be assigned to proteins automatically, on the basis of the protein domain content of the protein.

The availability of protein data sets annotated with GO terms and InterPro domains provides an opportunity to study the extent to which InterPro can be used to predict GO terms. The InterPro database contains over 12,000 entries and the GO contains over 19,000 but proteins are usually annotated with a few terms from each database, resulting in a sparse data set. In addition, a large set of proteins will contain only a few positive examples of each GO term, leading to extremely biased class distribution in which less than 1% of the training instances represent positive examples of a GO term.

Many studies have shown that standard classification algorithms perform poorly with imbalanced class distribution [7, 10, 16]. The most common method to overcome this problem is through re-sampling of the data to form a balanced data set. Re-sampling methods may under-sample the majority class, over-sample the minority class, or use a combination of both approaches. A potential drawback of under-sampling is that effective instances can be ignored. Over-sampling, however, is not without its problems. The most common approach is to duplicate instances from the minority class but Ling et el. [11] show that often times this approach does not offer significant improvements in performance of the classifier, as compared to the imbalanced data set. The other approach is the Synthetic Minority Over-sampling Technique (SMOTE) [2], which is an over-sampling technique with replacement in which new synthetic instances are created, rather than simply duplicating existing instances. Under-sampling can potentially be used to avoid the problems of over-sampling [10, 20]. Under-sampling removes instances from the majority class to create a smaller, balanced data set. While other approaches such as feature weighting can be employed, under-sampling has the added benefit of reducing the number of training instances that are required for training, thus reducing the difficulties of training pattern recognition algorithms on very large data sets.

In this paper we consider the application of statistical pattern recognition techniques to classify proteins with GO terms, using InterPro terms as the feature set. We show that many of the problems associated with sparse and imbalance data sets can be overcome with standard feature and instance selection methods. Feature selection in an extremely sparse feature space can produce instances that lack any positive features, leading to a subset of identical instances in the majority class. By selectively removing these duplicated instances, or keeping them, we trained two SVMs that have different performance characteristics. We describe a meta-learning scheme that combines both models, resulting in improved performance than can be obtained by using either SVM alone.

2 Methods

2.1 Dataset

The data set used for this study was comprised of 4590 annotated proteins from the *Saccharomyces cerevisiae* (Yeast) genome obtained from the UniProt database [26]. This protein contains manually curated GO annotations as well as InterPro terms automatically assigned with InterProScan.

The data set contains 2602 InterPro terms and 2714 GO terms with an average of 2.06 InterPro terms and 3.99 GO terms assigned to each protein. Table 1 illustrates the imbalanced nature of the data set. In this study, each GO term was considered as an independent binary classification problem and therefore, all proteins annotated with a GO term are treated as positive instances

(GO+) and the remaining proteins treated as negative instances(GO-), resulting in highly biased class and feature distribution. For the purpose of this study, we only considered data sets that contained at least 10 GO+ proteins.

Table 1. Examples of randomly selected classes (GO terms) and features (InterPro terms) illustrating the imbalanced and sparse nature of the data set.

GO term	Number of Positive Examples	Number of Negative Examples	InterPro term	Number of Positive Examples	Number of Negative Examples
GO:0000001	22	4568	IPR000002	5	2597
GO:0000022	15	4575	IPR000009	2	2600
GO:0000776	12	4578	IPR000073	13	2589
GO:0005635	35	4555	IPR000120	2	2600

2.2 Under-Sampling

Several methods are available for creating balanced data sets. If the features are continuous, we can perform over-sampling using methods such as SMOTE [2] which created new interpolated value for each new instance. In our case, however, the data set is binary format so this method cannot be used. In most cases under-sampling is considered to be better than over-sampling in terms of changing in misclassification costs and class distribution [2]. Another issue about the under-sampling is how ratio positive verse negative to make balanced set is optimized for training. In the point of the dealing with the imbalanced data problem, Al-shahib et el.[1] applied various under-sampling rates from 0% to 100% and conclude that the fully balanced set which have same number of positives and negatives, give the best performance. In light of this prior work, we performed under-sampling to create fully balances data sets for each GO term.

For each data set, we performed under-sampling of the majority class (GO-negative proteins) to create a balanced data set for SVM induction. We compared the performance of four under-sampling methods: Farthest, Nearest, Cluster and Random. In the first two cases, we used Euclidean distance, computed on the basis of the InterPro term content of each protein as a measure of distance. The Farthest and Nearest methods select proteins from the negative class that have the greatest and least distance from the positive class respectively. The Cluster method first performs hierarchical clustering of the negative class where the number of clusters formed equals the number of instances in the positive class. A single protein from each cluster is selected randomly. The Random method randomly selects proteins from the negative class.

Let D_{All} be the set of all of IPR and GO data. We define the example of datset as D_{All} =$\{(X_i, Y_j)|$ i=1,$\cdots$,l,j=1,$\cdots$,m $\}$, where x=$(x_1, x_2, \cdots, x_k)$ $\in$ IPR$\{0,1\}$ is a feature vectors, Y =$(y_1, y_2, \cdots, y_k)$ $\in$ GO=$\{0,1\}$ is the class designation.

2.3 Feature Selection

We employed four different objective functions for feature selection: chi-squared ($\chi2$), information gain, symmetrical uncertainty, and correlation coefficient.

Classical linear correlation [18] measures the degree of correlation between features and classes, and ranges from -1 to 1. If the features and the class are totally independent, then the correlation coefficient is 0. The traditional linear correlation method is very simple to calculate, but it assumes that there is a linear relationship between the class and the feature, which is not always true [18]. To overcome this shortcoming, the other correlation measures based on the theoretical concept of entropy were also assessed for feature selection. Information gain is a measurement based on entropy, and measures the number of bits of information obtained for class prediction [17]. However, information gain have non-normalized value and it is biased toward of feature with more value. To compensate for this disadvantage, symmetrical uncertainty value is normalized from 0 to 1 and un-biased in terms of feature content. The idea of symmetrical uncertainty is based on the information gain, but applied value is normalized and un-biased toward feature with more value [18]. When calculating the contingency between features and a class of interest, the $\chi2$ statistic measures the lack of independence. As the $\chi2$ statistic values increases, the dependency between features and classes also increases [17, 18].

The features were ranked using each of the objective values and a sequential forward selection search algorithm was used for feature selection. Forward selection was used since it is considered to be computationally more efficient than backward elimination [4]. The feature inclusion threshold for 12 randomly selected data sets was determined by computing the error rate during each stepwise feature addition and finding the minimal error rate. The average threshold value for the 12 data sets was used for the remaining data sets.

2.4 Implementation

Feature selection experiments were performed with WEKA [27]. Under-sampling and SVM induction were performed with MATLAB [24] using the pattern recognition toolbox [25].

3 Experiments

Individual data sets are constructed for each GO term which are then subjected to feature selection and instance selection prior to SVM model induction. Because of the extremely sparse nature of the data set, the feature selection step can remove all InterPro terms from some proteins, resulting in proteins that completely lack features. In most cases, feature selection resulted in a large number of GO- proteins in each data set. We theorized that such a large number of redundant proteins in the data sets could lead to skewed performance of the SVM so for each GO term, we constructed two data sets. ***Model 1*** refers to the SVM learned from the data set containing the redundant GO- proteins and ***Model 2*** refers to a smaller set in which redundant proteins were removed prior to model induction (Fig. 1). We expected that ***Model 2*** would result in SVM with higher accuracy than only ***Model 1***.

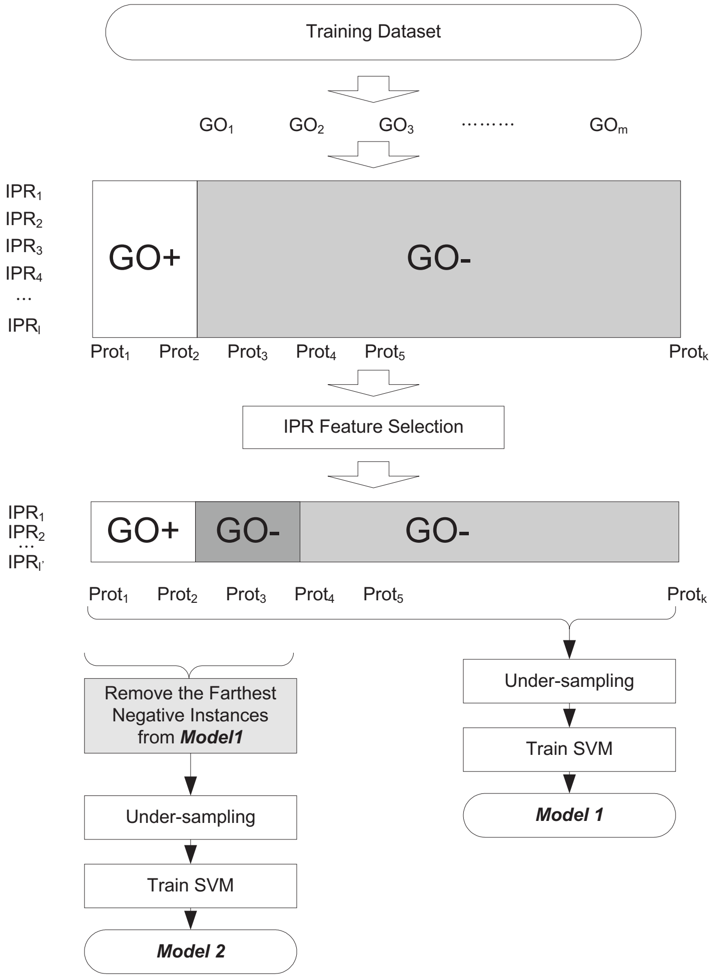

Fig. 1. Flow chart for the training process.

3.1 Feature and Instance Selection

We randomly selected 50 ***Model 1*** data sets and compared the performance of the feature selection and instance selection methods. The relative performance of the various methods were compared using error rate and AUC by 10-fold cross validation. The chi-squared method outperformed the other feature selection methods (Table 2) and was used to prepare data sets for instance selection. The Farthest method provided the best instance selection performance (Table 3) and was selected to create balanced data sets for SVM induction.

Table 2. Performance comparison of 4 different feature selection methods. Features were ranked using one of four objective functions (SU: symmetrical uncertainty, INFO: information gain, CHI: chi-squared, ABS: absolute correlation coefficient) and sequential forward selection was performed to optimize. Values represent average over 50 data sets.

Method	Sensitivity	Specificity	AUC	Error Rate
SU	0.976	0.832	0.726	0.01
CHI	0.988	0.931	0.870	0.01
INFO	0.783	0.980	0.846	0.12
ABS	0.767	0.236	0.792	0.43

Table 3. Performance comparison of 4 different under-sampling methods. Nearest is the result of applying for choosing negative instances as nearest method, and Farthest is farthest negative instances. Cluster is clustering and choosing randomly negatives. Random is randomly selected.

Method	Sensitivity	Specificity	AUC	Error Rate
Farthest	0.942	0.942	0.782	0.033
Nearest	0.731	0.789	0.737	0.517
Cluster	0.897	0.896	0.767	0.088
Random	0.872	0.928	0.765	0.083

We used 10-fold cross validation to compare the performance of SVMs trained using ***Model 1*** and ***Model 2***. In general, ***Model 1*** SVMs had very low false negative rates but had high false positive rates whereas ***Model 2*** SVMs tended to have lower false positive rates (Fig. 2). On average, ***Model 1*** has 0.32 false negative instances per SVM but 297.54 false positive instances and 4024 true negative instances per SVM among 4347 proteins. Of the 374 SVMs trained, 84% have less than 1 false negative instance using ***Model 1*** (Fig. 2(a)). Therefore, we conclude that this model is effective at classifying positive instances, although it should be noted that ***Model 1*** trained SVMs have high false positive rates. Since properties of both models were desirable for our classifier, we developed a meta-learning scheme that incorporated both models and includes a final filtering step, in order to reduce the false positive rate.

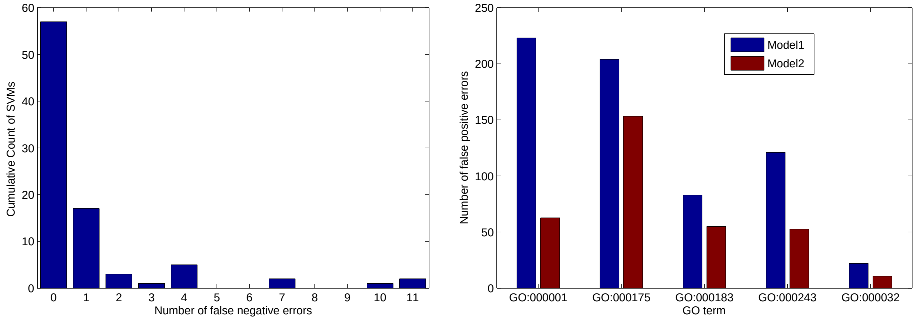

Fig. 2. Performance of SVMs estimated by 10-fold cross validation. (a) Cumulative count of number of false negative errors produced by 374 independent SVMs (b) False positive errors produced by ***Model 1*** and ***Model 2*** for five randomly selected GO term datasets.

3.2 Test Procedure

Data flow for the prediction step is shown Fig. 3. We focus on keeping the true positive rate as high as possible so ***Model 1*** is utilized as first step. The ***Model 1*** classifier plays a role of excluding most negative instances, but has the risk of making false positive classifications. Proteins classified as positive by ***Model 1*** are classified again using ***Model 2***, thereby reducing the number false positive proteins from 297.54 to 110.63 on average.

The third step is comprised of a decision rule that we devised based on observations we made of the dataset. Under the assumption that a positive relationship exists between GO terms and InterPro terms, we define the following decision rule: For each GO term assigned to a protein, we identify whether a training proteins exists with that GO term and an InterPro term assigned to the predicted protein. If at least one association exists, the the predicted GO term is retained, otherwise it is removed from the set of predicted GO terms.

We compared the precision of the suggested classification procedure (Fig. 3 Process B) with the precision of ***Model 1*** alone (Process A), where precision is measured as the number of true positive GO terms divided by the number of predicted GO terms. Among the data set of 4347 proteins, we held out 40 proteins to use for comparative analysis of the two classification procedures as well as for comparison to other GO classification methods. Performance comparisons were made using the 40 proteins held out from the training set. Precision was slightly higher with the suggested classification method (Fig. 4).

3.3 Comparison to Other Methods

Using the training set, we prepared SVMs for each GO term. Precision is employed again as a metric to compare the performance of our method to that

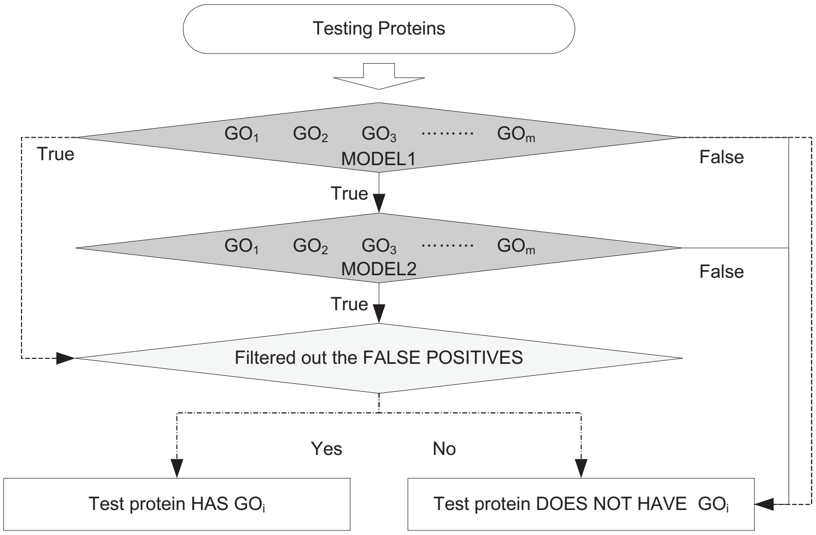

Fig. 3. Flow chart for the testing process. The dotted line represents use of ***Model 1*** only (Process A). The solid line represents use of both ***Model 1*** and ***Model 2*** (Process B). The filtering step is used in both cases.

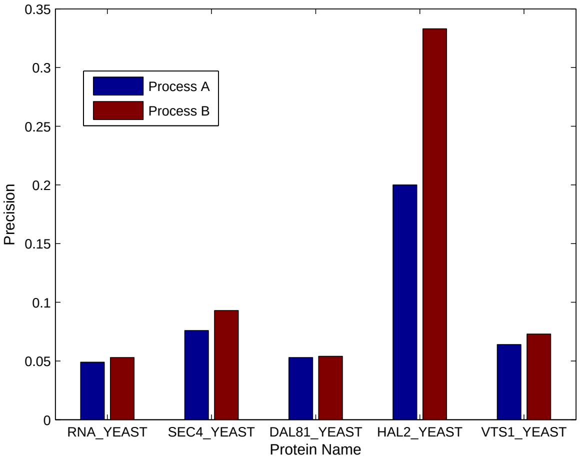

Fig. 4. Precision of Process A and Process B

of other described methods. Because most of the automated GO annotation methods are only available as web-based forms designed to process one protein at a time, we compared the performant of our method to that of five different GO annotation tools using nine proteins randomly selected from the hold out set (Fig. 5). We used the author recommended confidence thresholds of 20% and 50% for the GOtcha and GOPET methods, respectively, and employed an e-value cutoff of 1e-5 for GOFigure. IPR2GO is a manually curated mapping of InterPro terms to GO terms maintained by the InterPro consortium. Out method, which we term Automatic Annotation of Protein Functional Class (AAPFC) includes trained SVMs for every GO term in which ten or more positive protein instances could be found in the training data set. On average, precision is 0.53 for AAPFC, 0.17 for GOPET, 0.05 for GOtcha, 0.29 in GOFigure, and 0.20 in IPR2GO. Surprisingly, AAFPC outperformed IPR2GO, suggesting that there are many protein functions that can be prediction from InterPro terms, that cannot be described an a simple one-to-one translation table.

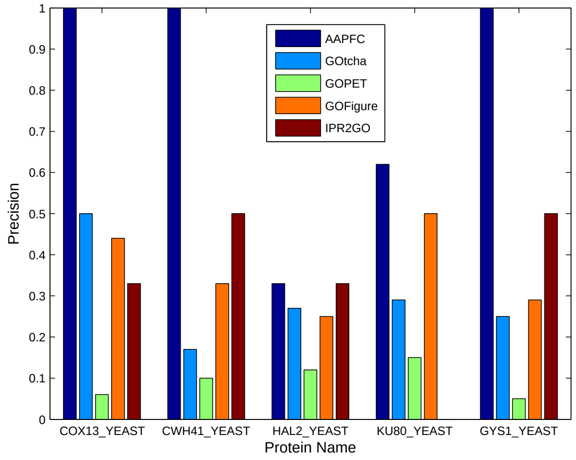

Fig. 5. Comparison of the proposed classification method (AAPFC) to four other methods.

4 Conclusions

In this paper, we propose a method for assigning GO terms to proteins using InterPro terms as features and learning independent SVMs for each GO term. By creating two data sets, each having different properties, and learning two SVMs for each GO term, we developed a meta-learning scheme that benefits from the strengths of each model. Our long-term plans are to develop a system for assigning GO terms to proteins using multiple feature types, including biochemical properties (amino acid content, etc.), phylogenetic profile, sequence similarity, and others. Our current strategy treats each GO term as an independent learning problem. This has some practical benefits in that individual classifiers or sets of classifiers could be learned or re-learned over time without the need to re-learn the whole system. On the other hand, this approach assumes that all GO terms are independent. During our initial steps of data exploration and data cleaning, we observed a high correlation coefficient among some pairwise GO terms, indicating that there is dependency among some GO terms. Therefore, future work we propose to utilize stacked generalization as an approach to capture dependence among GO terms into the learning method. The outputs of the classifiers described here can be used as inputs to another classifier, thus enabling the dependence among GO terms to be utilized for classification.

References

1. Al-shahib, A., Breitling, R. and Gilbert, D. : Feature Selection and the Class Imbalance Problem in Predict Protein Function form sequence. Applied Bioinformatics Vol.4 (2005) 195-203
2. Chawla, N.V., Bowyer, K. and Hall, L.O. : Kegelmeyer,W.P. SMOTE: Synthetic minority over sampling technique. Journal of artificial Intelligence Research , Vol.16 (2002) 321-357
3. Drummond, C. and Holte, R.C. : C4.5,Class Imbalance, and Cost sensitivity: Why Under-sampling beats Oversampling. ICML'2003 Workshop on Learning from Imbalanced Datasets II. (2003)
4. Guyon, I. and Elisseeff, A. : An introduction to variable and feature selection. Journal of Machine Learning Research Vol.3 (2003) 1157-1182
5. Hennig, S., Groth, D. and Lehrach, H. : Automated Gene Ontology annotation for anonymous sequence data. Nucleic acids Research (2003) 3712-3715
6. Huang, J., Lu, J. and Ling, C.X. : Comparing Naive Bayes ,Decision Trees, and SVM using Accuracy and AUC. Proc. of The Third IEEE Inter. Conf. on Data Mining (ICDM) (2003) 553-556
7. Japkowics, N. and Stepen, S. : The class imbalanced problem : A systematic study. Intelligent Data Analysis Vol.6 (2002)
8. Khan, S., Situ, G., Decker,K. and Schmidt,C.J. : GoFigure:Automated Gene Ontology annotation. Bioinformatics Vol.19 (2003)
9. King, R.D., Karwath, A.,Clare,A. and Dephaspe,L. : Genome scale prediction of protein functional class from sequence using data mining. Proc. of the sixth ACM SIGKDD Inter. Conf. on Knowledge discovery and data mining (2003)
10. Kubat, M. and Matwin, S. : Addressing the curse of Imbalanced Training sets : One-sided Selection. Proc. of the Fourteenth Inter. Conf. on Machine Learning Proc. (ICML) (1997) 179-186
11. Ling, C. and Li, C. : Data mining for direct marketing :problem and solution. Proc. of the Fourth Inter. Conf. on Knowledges Discovery and Data Mining(KDD) (1998) 73-79
12. Martin, D.M., Berriman, M. and Barton, G.J. : GOtcha : A new method for prediction of protein function assessed by the annotation of sever genomes. BMC bioinformatics Vol.5 (2004)

13. Pavalidis, P., Weston, J., Cai, J. and Grundy, W.B. : Gene Functional Classification From Heterogeneous Data. Proc. of the Fifth Inter. Conf. on Research in Computational Molecular Biology (RECOMB) (2001) 249-255
14. Vinayagam, A., Konig, R., Moormann, J., Schubert, F., Elis, R. , Glatting, K.H. and Suhai, S. : Applying support vector machine for gene ontology based gene function prediction. BMC Bioinformatics Vol.19 (2003)
15. Vinayagam, A., Val, C.D, Schubert, F., Elis, R., Glatting, K.H., Suhai, S. and Konig, R. : GOPET : A tool for automated predictions of Gene Ontology terms. BMC Bioinformatics Vol.7 (2006)
16. Weiss,G.M. : Mining with rarity : A unifying framework. ACM SIGKDD Explorations Newsletter Vol.6 (2004) 7-19
17. Yang, Y. and Pedersen, J.O. : A comparative study on feature selection in text categorization. Proc. of the Fourteenth Inter. Conf. on Machine Learning (ICML) (1997) 412-420
18. Yu, L. and Liu, H. : Feature Selection for high-Dimensional Data: A Fast Correlation-based filter solution. Proc. of the Twentieth Inter. Conf. on Machine Learning (ICML) (2003)
19. Zehetner, G. : OntoBlast function: from sequence similarities directly to potential functional annotations by ontology terms. Nucleic acids Research (2003) 3799-3803
20. Zhang, J. and Mani, I. : kNN Approach to Unbalanced Data Distributions: A case study involving Information Extraction. ICML'2003 Workshop on learning from imbalanced datasets II (2003)
21. Zheng, Z., Wu, X. and Shrihari, R. : Feature selection for text categorization on imbalanced data. ACM SIGKDD Exploration Newsletter Vol.6 (2004) 80-89
22. Gene Ontology(GO) Consortium, `http://www.geneontology.org/`
23. InterPro, `http://www.ebi.ac.uk/interpro/`
24. MATLAB, `http://www.mathworks.com/`
25. Pattern Recognition Toolbox for MATLAB, `http://cmp.felk.cvut.cz/~xfrancv/stprtool/`
26. UniProt, `www.uniprot.org/`
27. WEKA, `http://www.cs.waikato.ac.nz/~ml/`