GMR Genetics and Molecular Research
Online Journal
ISSN:1676-5680

A discriminative method for protein remote homology detection based on N-Gram

S. Xie[1], P. Li[1], Y. Jiang[1†] and Y. Zhao[2]

[1]School of Information Science and Technology,
Xiamen University, Xiamen, Fujian, China
[2]Information and Computer Engineering College,
Northeast Forestry University, Harbin, Heilongjiang, China

[†]*In memoriam*
Corresponding author: Y. Zhao
E-mail: zymyoyo@hotmail.com

Genet. Mol. Res. 14 (1): 69-78 (2015)
Received October 22, 2013
Accepted September 18, 2014
Published January 15, 2015
DOI http://dx.doi.org/10.4238/2015.January.15.9

ABSTRACT. Protein remote homology detection refers to detecting structural homology in proteins with an extremely low rate of sequence similarity. Such detection is primarily conducted using 3 methods: pairwise sequence comparisons, generative models for protein families, and discriminative classifiers. In this study, a discriminative classification method involving N-Grams was adopted to extract features using a random forest algorithm to classify data sets. Experiments in the SCOP 1.53 data set showed that our approach improved the receiver operating characteristic by 6% compared with well-known methods. To determine a score threshold that could be used to divide the data set, we also used a heuristic method through which the precision of positive examples and recall rate reached 0.5647 and 0.8647, respectively. Few other studies have investigated the recall and precision of such examples.

Key words: Protein remote homology; N-Gram; Machine learning; Random forest

INTRODUCTION

Rapid development of large-scale sequencing techniques has increased the number of known protein sequences (Wu et al., 2006; Cochrane et al., 2009; Shumway et al., 2010; Cheng et al., 2011, 2012; Zou et al., 2013a). These sequences must be classified into structural and functional classes based on homology. A newly sequenced protein can be annotated by transfer annotations from well-characterized homologous proteins. Therefore, development of algorithms for detecting protein homology is very important. Specifically, an algorithm for protein remote homology detection, which is the detection of evolutionary homology in proteins with low similarity, is an important challenge in bioinformatics.

Protein remote homology detection has been studied for several decades. Numerous algorithms have been proposed to address this problem, including pairwise comparison, generative models for protein families, and discriminative algorithms. Pairwise comparison methods, such as the pairwise method (Liao and Noble, 2003) and the Smith-Waterman dynamic programming algorithm (Smith and Waterman, 1981), measure pairwise similarities between protein sequences. These methods are effective for early detection but fail when applied to remote homology protein sequences with low similarity. Generative models determine a probability distribution over a protein family and then generate unknown proteins as new members of the family based on a stochastic model, such as a profile hidden Markov model (Karplus et al., 1998). Recent methods have applied discriminative algorithms for accurate remote homology detection (Vapnik, 1998). In contrast to generative methods, discriminative methods extract features from initial protein sequences and discriminate protein families based on features. Among these 3 methods, discriminative algorithms show state-of-the-art performance for detecting protein homology.

In this study, a discriminative method combining N-Gram with random forest was examined for its ability to detect protein remote homology, unlike top-N-Grams (Liu et al., 2008), which extracts profile-based patterns by considering the most frequent elements in profiles. We consider the text feature of protein sequences by extracting features directly from protein sequences and then use random forest to classify the data set. Experiments on a benchmark data set revealed that our method showed desired performance regarding the mean receiver operating characteristic (ROC). To improve recall and precision, we used a novel method to determine a score threshold, and used the threshold to reclassify the data set. Compared with the initial classification result, reclassification showed significantly improved recall and precision.

MATERIAL AND METHODS

Data set

A common benchmark (Liao and Noble, 2003) was used to evaluate the performance of the method proposed using a data set published at http://noble.gs.washington.edu/proj/svm-pairwise/. This benchmark has been used to evaluate the performance of various homology detection methods (Saigo et al., 2004; Lingner and Meinicke, 2006; Dong et al., 2006); thus, our results could be compared with those of previous studies. The data set contains 4352 proteins derived from the SCOP database version 1.53. These proteins were extracted from the Astral database (Brenner et al., 2000), with sequence similarity of any pair less than an E-value of 10^{-25}. The 4352 distinct protein sequences were classified into 54 families. In each family,

positive test samples were derived from proteins within the family. Proteins outside the family but within the same superfamily were considered to be positive training samples. Protein sequences outside the superfamily were selected as negative samples and were separated into training and test sets.

N-Gram method

The lengths of various protein sequences vary widely. To classify proteins using the discriminative method, sequences must be transformed into fixed-length feature vectors. In this study, an N-Gram model (Manning and Schuetze, 1999) was applied to protein sequences to extract feature vectors. We counted the frequency of N consecutive amino acids in the protein sequence (N = 1, 2, 3). Because there are 20 different amino acids commonly observed in protein sequences, the vector length of the N-Gram was 20^N. For example, a vector length of 2-Gram is 400. The steps for transforming the protein sequences of a protein of length L into feature vectors are described as follows. A flow chart of the feature vector based on 2-Gram is shown in Figure 1.

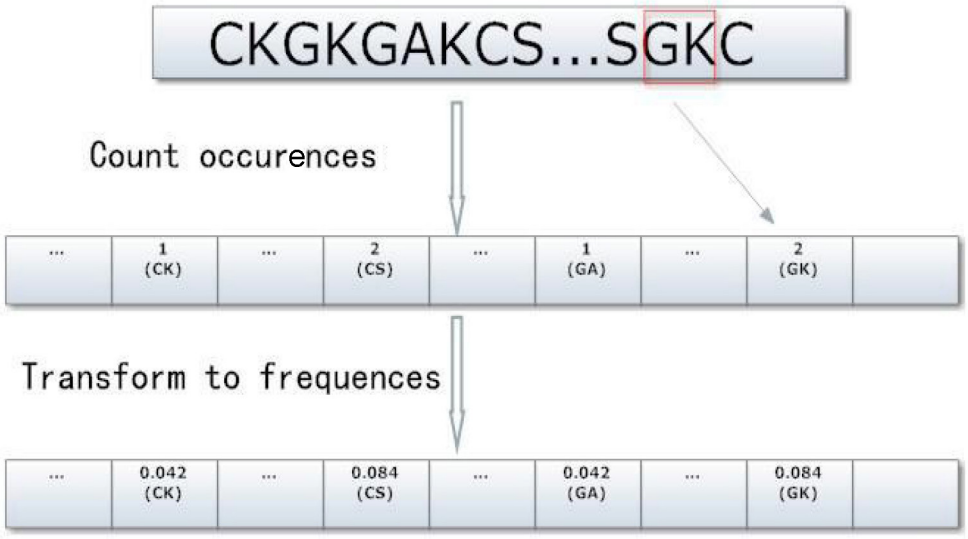


Figure 1. Flow chart of feature vector based on 2-Gram.

Step 1. Initialization of a feature vector for the protein sequence. The vector element represents one possible combination of an N-Gram, and its value was initialized to 0.

Step 2. Transversion of the protein sequence and counting all combinations of amino acids. In this step, an N-size window was applied to slide from the *i*th position to the $i + n$-1th position ($i = 1, 2, ...L - n + 1$). When the window slid on each position, the occurrence of each encountered N-Gram was stored.

Step 3. Normalization of the count vector into a frequency vector. We summed the values of the vector generated by the second step and divided each value by the accumulated value.

In this study, we used only N = 1-3 and the combination of the features of these 3 values. Increasing N significantly increased the calculation and reduced prediction accuracy. Thus, we extracted 20-D features for 1-Gram, 400-D features for 2-Gram, and 8000-D features for 3-Gram. The performance of various values of N is discussed below in the 'Optimizing random forest' section.

Improving recall and precision

When the random forest method was applied to the feature data set, the classifier returned a prediction score of every instance. In previous experiments, we set a score threshold and used the threshold to divide the data set into positive and negative classes. Mean recall and precision of various thresholds are listed in Table 1.

Table 1. Mean recall and mean precision of various thresholds.

Threshold	Negative class		Positive class	
	Recall	Precision	Recall	Precision
0.6	0.991465	0.994948	0.49997	0.55733
0.7	0.982034	0.99713	0.731501	0.50415
0.8	0.9666807	0.998174	0.864449	0.42416

Different thresholds showed different recall and precision rates. A trade-off was observed between recall and precision: an increased threshold raised the recall rate of the positive class and precision of the negative class but reduced the recall of the negative class and precision of the positive class.

To improve mean recall and precision, we used a heuristic method to discretely determine the thresholds for each family. For each family, we sampled training sets to form a new training set and test set, which we used to determine the best threshold according to the created metric. Finally, we used the threshold to classify the original test set. The detailed steps are described below.

Step 1: We randomly divided the training set (denoted as TR) into a new test set (NTE) and new training set (NTR) based on a ratio. For instance, 70% training data were extracted randomly to construct the new training set and the remaining data were used to create the new test set.

Step 2: NTR was used as the training set and the random forest method was used as a classifier for NTE to restore the prediction scores and true class label of NTE.

Step 3. We found the best score threshold for NTE. As the data set was unbalanced (more negative samples than positive samples), the recall and precision of positive samples were mainly considered in threshold determination. As described, a trade-off exists between recall and precision; thus, when the score threshold was determined, an F-measure was used as a metric:

$$F_\beta = \frac{(\beta^2 + 1)PR}{\beta^2 P + R} \qquad \text{(Equation 1)}$$

where *P* (precision) was calculated according to the following formula:

$$P = \frac{TruePositive}{TruePositive + FalsePositive} \quad \text{(Equation 2)}$$

and *R* (recall) was calculated by

$$R = \frac{TruePositive}{TruePositive + FalseNegative} \quad \text{(Equation 3)}$$

β was a parameter used to adjust the weight between *P* and *R*.

Consequently, only the scores of the positive samples were considered to be potential thresholds. When the prediction score of the positive sample was set as the threshold and if the score of an instance was higher than the threshold, it was labeled as negative; otherwise, it was classified as positive. We then calculated the F-value of this positive sample. Finally, we selected the prediction score with the highest F-value as the final threshold.

Step 4: We found the score threshold for the test set. In step 3, each family showed the best threshold for the new test set. First, this threshold was used on the test set directly, and performance was found to be less than satisfactory. We then assumed that the new and original test sets were identically distributed. Thus, the position of the best threshold, rather than its value, was used. Experiments indicated that using position information enhances performance.

The algorithm to reclassify the data set is described in Legend 1, and the function *Threshold* in Legend 1 was used to determine the position of the NTE threshold. The algorithm is described in Legend 2. We denoted E<n,p,k> as a structure to calculate recall and precision, where *n* is the number of negative instances of the test set, *p* is the number of positive instances, and *k* expresses the threshold.

RESULTS

ROC and ROC50

The simply measured error rates did not precisely indicate performance in terms of the unbalanced data set. In this case, a ROC score is typically used as a metric to evaluate the method (Gribskov and Robinson, 1996). A ROC score is the normalized area under a curve that plots true-positives against false-positives for different classification thresholds. If the method perfectly separates positive samples from negative samples, the ROC score equals 1, whereas 0 indicates that no sequence selected by the algorithm is positive. Another performance metric is the ROC50 score (Liu et al., 2013), which represents the area under the ROC curve up to the first 50 false-positives. The algorithm used to calculate the ROC value (Shah et al., 2008) is described in Legend 3.

Comparison of methods

In this study, all 4 types of N-Grams, including 1-Gram, 2-Gram, 3-Gram, and C3-

Gram (a combination of 1-Gram to 3-Gram) were used to extract features. The ROC and ROC50 for these 4 N-Grams are shown in Table 2. An additional 6 previous methods are shown in the table for comparison with our methods. The PseAAC Index (Liu et al., 2013), N-Gram (Leslie et al., 2002), pattern (Dong et al., 2005), motif (Ben-Hur and Brutlag, 2003), and binary profiles (Dong et al., 2007) are based on 5 building blocks of proteins. HHsearch (Söding, 2005) employs a novel profile based on hidden Markov models.

Table 2. ROC and ROC50 of different methods.

Methods	ROC	ROC50	Source
1-Gram	0.950	0.937	This study
2-Gram	0.971	0.956	This study
3-Gram	0.962	0.955	This study
C3-Gram	**0.975**	0.911	This study
PseAAC Index	0.880	0.620	(Shah et al., 2008)
PseAAC Index-Profile	0.922	0.712	(Shah et al., 2008)
SVM-N-Gram	0.791	0.584	(Söding, 2005)
SVM-Pattern	0.835	0.589	(Söding, 2005)
SVM-Motif	0.814	0.616	(Söding, 2005)
HHsearch	0.915	**0.990**	(Söding, 2005)

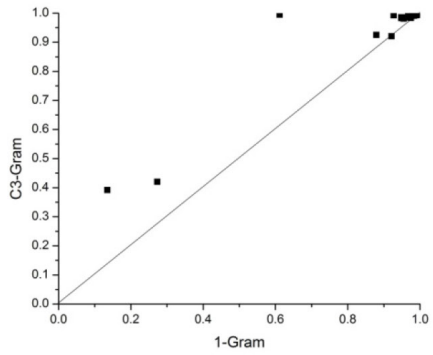
Bold numbers indicate the best results.

Table 2 shows that the performance of the proposed method was highly comparable to that of HHsearch, a state-of-the-art method. The ROC50 of C3-Gram was 7.9% lower than that of HHsearch. However, its ROC was higher than that of HHsearch by 6.0%. The proposed method outperformed the other methods in terms of ROC and ROC50. The proposed N-Gram is an efficient method of remote homology detection.

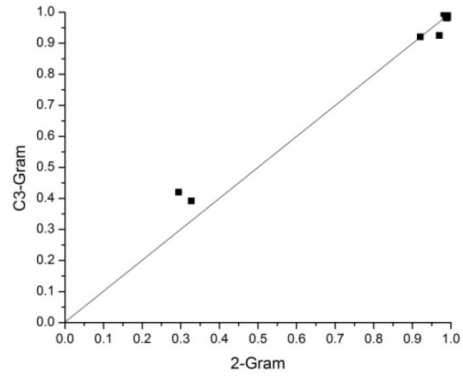
Based on Dong et al. (2006), the difference between N-Grams can be revealed through a family-by-family comparison of ROC scores between C3-Gram and the 3 other N-Grams plotted in Figure 2. Every point in the graph represents one of the 54 SCOP families. The point above the diagonal indicates that C3-Gram outperformed the method labeled by the x-axis in the family it represents. From (A), (B), and (C), we can see that C3-Gram outperformed the other N-Grams, and 2-Gram was highly comparable with C3-Gram, whereas 3-Gram showed the worst performance. Thus, each N-Gram positively contributed to C3-Gram, and the contribution of 2-Gram was the most significant. Based on computational complexity, 2-Gram is preferable to C3-Gram.

Optimizing random forest

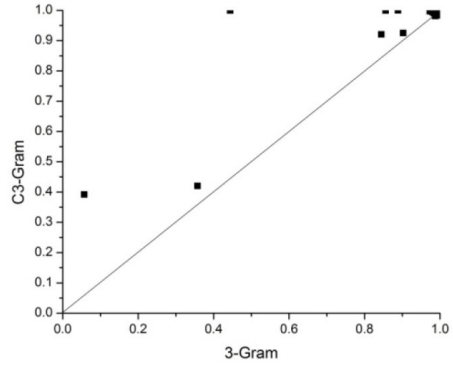
In studies on protein detection, the ensemble classifier is typically considered to be a priority classifier (Cai et al., 2010; Chen et al., 2012; Lin et al., 2013). In this study, random forest was selected as a classifier, which is a powerful ensemble classification algorithm for protein remote homology detection. Compared with other classifiers, the random forest algorithm showed unique advantages for dealing with high-dimensional feature sets and was highly suitable for balancing errors between imbalanced data sets. To optimize the parameters of random forest, an experiment was designed to evaluate its performance with trees of different numbers. We calculated the ROC values of trees of different numbers (Figure 3).



(a)



(b)



(c)

Figure 2. Family-by-family comparison of C3-Gram and other N-Grams.

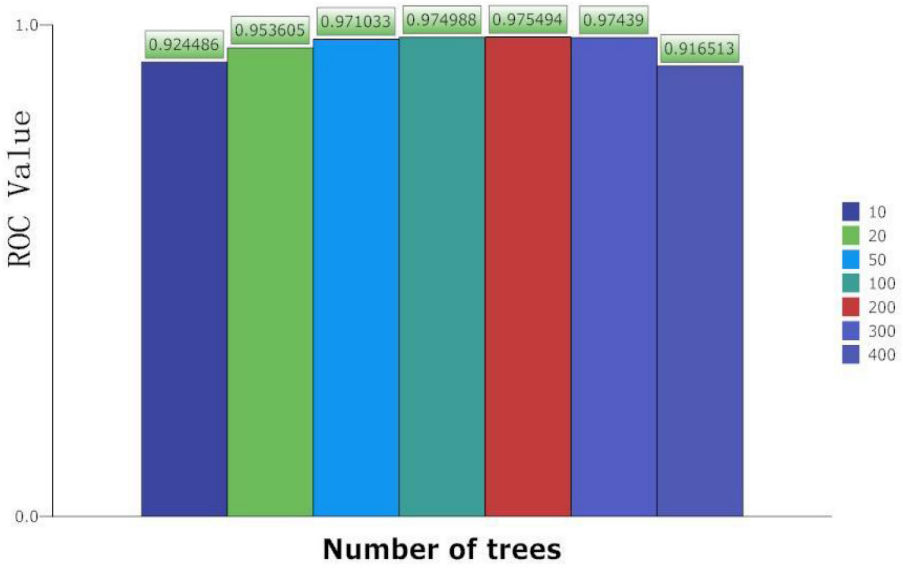


Figure 3. ROC scores of trees of different numbers.

Increasing the number of trees gradually increased the ROC value, which showed a peak when the trees were increased to 200. The ROC value then decreased, indicating that random forest performs best when trees were set to 200, until it reached 0.975494. Therefore, 200 was selected as the number of random forest trees in this study.

We also selected another 4 classifiers commonly used in protein classification to compare with random forest. The performance of each classifier is described in Table 3.

Table 3. Performance of different classifiers.

Classifier	ROC	ROC50
SMO	0.428	0.428
Naive Bayes	0.858	0.105
J48	0.132	0.132
Random Tree	0.393	0.384
Random Forest	**0.975**	**0.911**

SMO = sequential minimal optimization; J48 = decision tree. Bold numbers indicate the best results.

All classifiers used were implemented in Weka (Hall et al., 2009), a common data mining tool, and the 4 compared classifiers were used as default parameters in Weka. Random forest outperformed the other classifiers. The compared classifiers were unable to distinguish the data set and classified all data sets into the negative class.

Trade-off between recall and precision

Although the proposed method showed the desired performance in terms of ROC, the recall and precision of positive samples were unsatisfactory. The heuristic method proposed

in the 'Improving recall and precision' section was used to improve recall and precision. The average recall and precision of positive samples with different β values are shown in Table 4. Feature vectors were extracted by C3-N-Gram, and the number of trees used in random forest was 200.

Table 4. Mean recall and precision with different β values.

β	Mean recall	Mean precision	F1-Measure
0.5	0.59025	0.66593	0.6258
1.0	0.76207	0.61550	0.6810
1.5	0.83841	0.57204	0.6801
2.0	0.86752	0.56470	**0.6841**
2.5	0.87387	0.55120	0.6760
3.0	0.88551	0.54301	0.6732

Bold number indicates the best results.

The results shown in Table 4 indicate the trade-off between recall and precision. Increased recall reduced the corresponding precision, and β adjusted for this trade-off. Increased β also increased recall but reduced precision. Few studies have investigated recall and precision; therefore, previous data could not be compared with the results of this study. We used F1-measure, a simplified F-measure when β is set to 1, to judge the performance of different β values. F-measure was used both as a parameter and as a measurement of this method: it was first used to adjust for the trade-off between recall and precision with different β values, and then used as a measurement with $\beta = 1$. Using this measure, the method exhibited the best performance when $\beta = 2$ and yielded 0.86752 and 0.56470 for recall and precision, respectively.

DISCUSSION

In this study, the N-Gram model was successfully used to detect protein remote homology. Experimental evaluation through the benchmark data set showed that the proposed method effectively improved prediction performance. A novel method was also proposed to improve the recall and precision of positive samples. This method yielded values of 0.86752 and 0.56470 for mean recall and precision, respectively. Future studies will involve the development a web server such as that proposed by Zou et al. (2013b) for the method presented in this paper.

ACKNOWLEDGMENTS

Research supported by the Fundamental Research Funds for the Central Universities (#DL10BB02) and the Natural Science Foundation of China (#61001013 and #61370010).

REFERENCES

Ben-Hur A and Brutlag D (2003). Remote homology detection: a motif based approach. *Bioinformatics* 19 (Suppl 1): i26-i33.

Brenner SE, Koehl P and Levitt M (2000). The ASTRAL compendium for protein structure and sequence analysis. *Nucleic Acids Res.* 28: 254-256.

Cai YD, Lu L, Chen L and He JF (2010). Predicting subcellular location of proteins using integrated-algorithm method. *Mol. Divers.* 14: 551-558.

Chen W, Liu X, Huang Y, Jiang Y, et al. (2012). Improved method for predicting protein fold patterns with ensemble classifiers. *Genet. Mol. Res.* 11: 174-181.

Cheng L, Hou ZG, Lin Y, Tan M, et al. (2011). Recurrent neural network for non-smooth convex optimization problems with application to the identification of genetic regulatory networks. *IEEE Trans. Neural Netw.* 22: 714-726.

Cheng XY, Huang WJ, Hu SC, Zhang HL, et al. (2012). A global characterization and identification of multifunctional enzymes. *PLoS One* 7: e38979.

Cochrane G, Akhtar R, Bonfield J, Bower L, et al. (2009). Petabyte-scale innovations at the European Nucleotide Archive. *Nucleic Acids Res.* 37: D19-D25.

Dong QW, Lin L, Wang XL and Li MH (2005). A Pattern-Based SVM for Protein Remote Homology Detection. In: Machine Learning and Cybernetics, 2005. Proceedings of 2005 International Conference.

Dong QW, Wang XL and Lin L (2006). Application of latent semantic analysis to protein remote homology detection. *Bioinformatics* 22: 285-290.

Dong Q, Lin L and Wang X (2007). Protein Remote Homology Detection Based on Binary Profiles. In: Bioinformatics Research and Development (Hochreiter S and Wagner R, eds.). Springer, Berlin, 212-223.

Gribskov M and Robinson NL (1996). Use of receiver operating characteristic (ROC) analysis to evaluate sequence matching. *Comput. Chem.* 20: 25-33.

Hall M, Frank E, Holmes G and Pfahringer B (2009). The WEKA data mining software: an update. *ACM SIGKDD Explorations Newsletter* 11: 10-18.

Karplus K, Barrett C and Hughey R (1998). Hidden Markov models for detecting remote protein homologies. *Bioinformatics* 14: 846-856.

Leslie C, Eskin E and Noble WS (2002). The spectrum kernel: a string kernel for SVM protein classification. *Pac. Symp. Biocomput*. 564-575.

Liao L and Noble WS (2003). Combining pairwise sequence similarity and support vector machines for detecting remote protein evolutionary and structural relationships. *J. Comput. Biol.* 10: 857-868.

Lin C, Zou Y, Qin J, Liu X, et al. (2013). Hierarchical classification of protein folds using a novel ensemble classifier. *PLoS One* 8: e56499.

Lingner T and Meinicke P (2006). Remote homology detection based on oligomer distances. *Bioinformatics* 22: 2224-2231.

Liu B, Wang X, Lin L, Dong Q, et al. (2008). A discriminative method for protein remote homology detection and fold recognition combining Top-n-Grams and latent semantic analysis. *BMC Bioinformatics* 9: 510.

Liu B, Wang X, Zou Q and Dong Q (2013). Protein remote homology detection by combining Chou's pseudo amino acid composition and profile-based protein representation. *Mol. Informatics* 32: 775-782.

Manning C and Schuetze H (1999). Foundations of Statistical Natural Language Processing. MIT Press, Boston, 78.

Saigo H, Vert JP, Ueda N and Akutsu T (2004). Protein homology detection using string alignment kernels. *Bioinformatics* 20: 1682-1689.

Shah AR, Oehmen CS and Webb-Robertson BJ (2008). SVM-HUSTLE-an iterative semi-supervised machine learning approach for pairwise protein remote homology detection. *Bioinformatics* 24: 783-790.

Shumway M, Cochrane G and Sugawara H (2010). Archiving next generation sequencing data. *Nucleic Acids Res.* 38: D870-D871.

Smith TF and Waterman MS (1981). Identification of common molecular subsequences. *J. Mol. Biol.* 147: 195-197.

Söding J (2005). Protein homology detection by HMM-HMM comparison. *Bioinformatics* 21: 951-960.

Vapnik VN (1998). Statistical Learning Theory. Wiley-Interscience, New York.

Wu CH, Apweiler R, Bairoch A, Natale DA, et al. (2006). The Universal Protein Resource (UniProt): an expanding universe of protein information. *Nucleic Acids Res.* 34: D187-D191.

Zou Q, Chen WC, Huang Y and Liu XG (2013a). Identifying multi-functional enzyme by hierarchical multi-label classifier (2013a). *J. Comput. Theor. Nanos.* 10: 1083-1043.

Zou Q, Li X, Jiang Y and Zhao Y (2013b). BinMemPredict: a web server and software for predicting membrane protein types. *Current Proteomics* 10: 2-9.