

# A Novel Method for Protein Subcellular Localization Based on Boosting and Probabilistic Neural Network

**Jian Guo**

Student of Mathematical Sciences  
Tsinghua University  
Beijing, 10084, China

guojian99@tsinghua.org.cn

**Yuanlie Lin**

Faculty of Mathematical Sciences  
Tsinghua University  
Beijing, 100084, China

ylin@math.tsinghua.edu.cn

**Zhirong Sun**

Faculty of Institute of Bioinformatics  
Tsinghua University  
Beijing, 100084, China

sunzhr@mail.tsinghua.edu.cn

## Abstract

Subcellular localization is a key functional characteristic of proteins. An automatic, reliable and efficient prediction system for protein subcellular localization is needed for large-scale genome analysis. In this paper, we introduce a novel subcellular prediction method combining boosting algorithm with probabilistic neural network algorithm. This new approach provided superior prediction performance compared with existing methods. The total prediction accuracy on Reinhardt and Hubbard's dataset reached up to 92.8% for prokaryotic protein sequences and 81.4% for eukaryotic protein sequences under 5-fold cross validation. On our new dataset, the total accuracy achieved 83.2%. This novel method provides superior prediction performance compared with existing algorithms based on amino acid composition and can be a complementing method to other existing methods based on sorting signals.

*Keywords:* Subcellular localization; Boosting; Probabilistic neural network; Amino acid composition;

## 1 Introduction

High throughout genome sequencing projects are producing an enormous amount of raw nucleic acid sequences and protein sequences. The next step is to analysis these sequences begging for finding new gene functions and key regulatory pathways. As a hot topic in genome science, genome function annotation including the assignment of a function for a potential gene in the raw sequence is a vitally important work in genome research. Subcellular location is a key function characteristic of potential gene expressing the protein because the protein functions in the specific location in the intact cells to maintain the cell survival. As a result, the knowledge of protein subcellular location could provide useful information for the gene function prediction. However, subcellular localization analysis based on experiment is time consuming and could not be

performed for genome scale proteins. With the rapidly increasing number of sequences in database, it is highly necessary to develop an accurate, reliable and efficient system for protein subcellular localization automatically. Several efforts have been made in the prediction of protein subcellular localization. Up to now mainly two categories of prediction methods have been proposed. One was mainly based on the existence of sorting signals in N-terminal sequences (Nakai, 2000) including signal peptides, mitochondrial targeting peptides and chloroplast transit peptides (Nielsen et al, 1997, 1999). For the improvement of this method, Emanuelsson et al (Emanuelsson et al, 2000) proposed an integrated prediction system with artificial neural network based on individual sorting signal predictions. This system could be use to find cleavage sites in sorting signals and simulate the real sorting process to a certain extent. Nevertheless, the prediction accuracy of those methods based on sorting signals was highly correlated with the quality of protein N-terminal sequence assignment. Unfortunately, it is usually unreliable to annotate the N-terminal using known gene identification methods (Frishman et al, 1999). As a result, the prediction accuracy and reliability decreased when signals were missing or only partially included.

The other category of methods was mainly based on the amino acid composition of protein sequences in different subcellular localizations. This approach was first suggested by Nakashima and Nishikwa (1994). They found that the intracellular and the extra cellular proteins could be discriminated with high accuracy only by amino acid composition. From then on, different statistical methods and machine learning methods have been used based on amino acid composition of protein sequences to improve prediction accuracy. Cedano et al (1997) adopted a statistical method with Mahalanobis distance for prediction. Reinhardt and Hubbard (1998) predicted subcellular locations with neural networks and reached the accuracy 66% for eukaryotic sequences and 81% for prokaryotic sequences. Chou et al (1999) proposed the covariant discriminant algorithm on the same prokaryotic dataset as Reinhardt et al. and achieved a total accuracy of 87%. Hua and Sun (2001) constructed a prediction system using support vector machine (SVM)—a new machine learning method based on the statistical learning theory—on the same prokaryotic and eukaryotic datasets. The prediction accuracy of Hua et al has reached up to 91.4% for prokaryotic proteins and 79.4% for eukaryotic proteins.

In this paper, we developed a novel method for protein subcellular localization based on amino acid composition. We integrate boosting algorithm and probabilistic neural network into our prediction system. Boosting is a novel and powerful machine learning method for classification and regression. It can effectively convert a base or “weak” algorithm with accuracy just slightly better than random guessing into a strong classifier which can achieve high prediction accuracy. Boosting algorithm has been successfully applied in the field of pattern recognition, such as text classification (Schapire et al, 2002) and speech recognition (Rochery et al, 2002) et al. Probabilistic neural network (PNN) is an approach for classification problem. It overcomes some faults of the traditional back-propagation network. Here the probabilistic neural network was used as the base classifiers in boosting algorithm. The testing results show that the prediction accuracy has been improved with this novel method. In this paper, our method combining boosting and PNN is called Boost-PNN method.

## 2 Method and Database

### 2.1 Database

In this paper, we first chose the database generated by Reinhardt and Hubbard (1998), which is a commonly used subcellular localization prediction dataset, to test our new model. The sequences in this database were extracted from SWISSPORT 33.0 and subcellular location of each protein has been annotated. This set of sequences was filtered, only keeping those appeared complete and those had what appeared to be reliable location annotations. Transmembrane proteins were excluded because some reliable prediction methods for these proteins have already existed. Plant sequences were also removed for the sufficient difference of the composition. Finally, the filtered dataset included 997 prokaryotic proteins (688 cytoplasm, 107 extracellular and 202 periplasmic proteins) and 2427 eukaryotic proteins (684 cytoplasm, 325 extracellular, 321 mitochondrial, and 1097 nuclear proteins).

On the other hand, we newly constructed a much larger database to test our algorithm further. The new database we constructed included 8304 eukaryotic proteins in 8 subcellular locations: 1019 chloroplast proteins, 2387 cytoskeleton proteins, 595 extracellular proteins, 211 Golgi proteins, 133 lysosomal proteins, 644 mitochondria proteins, 3199 nuclear proteins and 116 perox proteins. All the proteins in this dataset were selected from SWISSPORT release 40 according to the similar filtering rule as Reinhardt and Hubbard’s dataset.

### 2.2 Boosting algorithm and AdaBoost

Boosting is a novel and powerful machine learning method for classification and regression. In addition, it is a general method for improving the accuracy of most learning algorithms. It can effectively convert a base or “weak” algorithm with accuracy just slightly better than random guessing into a strong classifier, which can

achieve arbitrarily low error rate given sufficient training data. The work process of a boosting algorithm is to repeatedly reweight the examples in the training set and rerunning the weak learning algorithm on those learning algorithm to concentrate on the hardest examples. The final decision is a weighted combination of the outputs of each weak classifier.

The first boosting algorithms were discovered by Schapire and Freund (1992). After that, they improved their algorithms and designed the so called AdaBoost algorithm (Freund et al, 1995, 1996), which has been shown to be very effective in many experiments. As the former boosting algorithms, AdaBoost is an adaptive algorithm to boost a sequence of classifiers, in which the weights are updated dynamically according to the errors in previous learning. AdaBoost is also a kind of large margin classifiers.

There are different versions of AdaBoost algorithms. In this paper, we adopted Freund and Schapire’s AdaBoost.M1 algorithm (Freund & Schapire, 1996). The algorithm is described as follows.

1. Initialize  $t = 1$ , labels  $D_i^{(t)} = 1/n, i = 1, 2, \dots, n$ .
2. Train a base classifier using distribution  $D^{(t)} = \{D_i^{(t)} \mid i = 1, 2, \dots, n\}$ .
3. Get the hypothesis (decision function) of the base classifier. By minimizing the error  $\mathbf{e}^{(t)} = \sum_{i: h'(x_i) \neq y_i} D_i^{(t)}$  (if  $\mathbf{e}^{(t)} > 1/2$ , then set  $t=t-1$  and abort loop).

where the hypothesis:  $h_t = X \rightarrow Y = \{1, \dots, k\}$

4. Choose  $\mathbf{a}^{(t)} = \frac{1}{2} \ln\left(\frac{1-\mathbf{e}^{(t)}}{\mathbf{e}^{(t)}}\right)$ .
5. Update  $D^{(t)}$ :

$$D_i^{(t+1)} = \frac{D_i^{(t)} \exp(-\mathbf{a}^{(t)} y_i h^{(t)}(x_i))}{Z^{(t)}},$$

where  $Z^{(t)}$  is a normalization factor, which is chosen so that  $D^{(t+1)}$  will be a probabilistic distribution.

6. Repeat steps 2-5 for T times.
7. Final hypothesis: the final hypothesis is given by a boosted classifier which is a linear combination of individual locally optimal SVMs.

$$H_{final}(x) = \arg \max_{y \in Y} \left( \sum_{t=1}^T \mathbf{a}^{(t)} h^{(t)}(x) \right).$$

A direct way to use to probability distribution  $D^{(t)}$  over the training samples when training the base classifier is to construct boosting set, which is a subset of all training samples. The boosting set can be created by sampling samples from the original training set according to the probability distribution  $D^{(t)}$ . Then it will be used as the training dataset for the base classifier in a particular iteration.

The base classifier can be any kind of classifier algorithms, such as decision trees, neural networks, et al.

In this paper, we use probabilistic neural network as the base classifier in boosting process.

### 2.3 Probabilistic neural network

The probabilistic neural network (PNN) was first introduced by Specht (1990). The original probabilistic model was designed in order to solve some faults of the traditional back-propagation neural network, such as the long train time and the false minimum problem. A PNN is a feed-forward, 4 layers network for solving classification problems. It is based on the well-established statistical principles derived from bayes decision rule and non-parametric kernel based estimators of probability density functions.

Consider a pattern vector  $x$  with  $m$  dimensions in a mutli-classification problem. The bayes decision rule implies that  $x$  belong to class  $k$  if and only if

$$h_k l_k f_k(x) > h_i l_i f_i(x), \text{ for all } i \neq k \quad (2)$$

where  $h_i$  and  $h_k$  are the priori probability of occurrence of patterns from class  $i$  and class  $k$ ;  $l_i$  are the loss function associated with the decision as misclassifying the vector as other classes when it belong to  $i$ ,  $l_k$  has the similar definition as  $l_i$ ;  $f_i(x)$  and  $f_k(x)$  are the probability density functions for categories  $i$  and  $k$ . Often the priori probabilities are known or can be estimated accurately, and the loss function  $s$  require subjective evaluation. In many situations, the loss functions and the probabilities can be considered equally. Therefore, the key to use the decision rule given by (2) is to estimate the probability density functions from the training samples.

The probabilistic neural network learns to approximate the probability density function of the training samples. More precisely, the PNN is interpreted as a function which approximates the probability density of the underlying samples' distribution. A nonparametric estimate method known as Parzen Window (Parzen, 1962) is used to construct the class-dependent probability density functions for each class required by bayes rule. If the  $j$ th training pattern for the  $i$ th class is  $x_j$ , then the Parzen estimate of the probability density function for the  $i$ th class is

$$F_i(x) = \frac{1}{(2p)^{m/2} \mathbf{s}^m n} \sum_{j=1}^n \exp \left[ -\frac{(x-x_j)^T(x-x_j)}{2\mathbf{s}^2} \right] \quad (3)$$

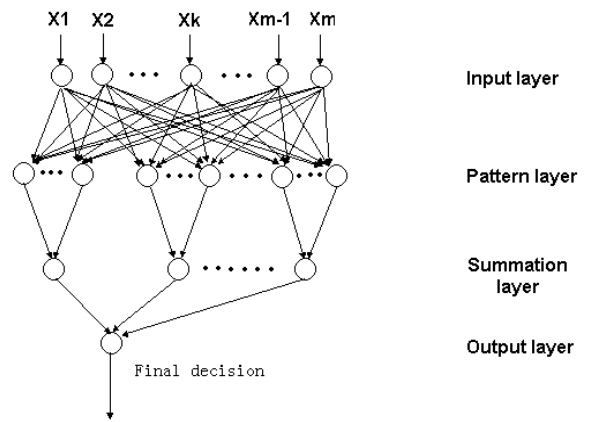
where  $n$  is the number of training samples,  $m$  is the dimension of patterns and  $\mathbf{s}$  is the "smooth parameter" which should be adjusted experimentally.

The architecture of PNN including 4 layers is shown in Figure 1. The first layer, the input layer, is consisted with a number of merely distributional units that supply the same input values to all of the pattern units in the pattern layer. The unit number in this layer is the same as the dimension of input samples. The second layer is called the pattern layer, which is consisted of the so called radial basis neurons equal to the number of training samples.

The weights of this layer are set to the transpose of the matrix formed from the total number of training pairs. The input to each radial basis neurons of this layer is the vector distance of the between the weight vector  $w_i$  and the input vector  $x$ , multiplied by bias  $b$ . The outputs of the radial basis neurons are derived by the function:

$$Y = \exp(-n^2) \quad (4)$$

where  $n = b \cdot \|w - x\|$ ,  $\|\bullet\|$  denote the Euclidean distance. Each bias  $b$  in the pattern layer is set to  $\sqrt{-\log(0.5)} / \text{Spread}$ . This gives the radial basis functions that cross 0.5 at a weighted input of  $\pm \text{Spread}$ . The parameter  $\text{Spread}$  determines the width of an area in the input space to which each neuron responds. A larger  $\text{Spread}$  lead to a larger area around the input vector, where the radial basis function respond with significant output.



**Figure 1:** Architecture of the Probabilistic Neural Network (PNN). The structure of a PNN is consisted of 4 layers: input layer, pattern layer, summation layer, output layer.

The summation units simply sum the inputs from the pattern units that correspond to the category from which the training pattern was selected. Therefore, the unit number in this layer is the same as the category number of the training samples. The final layer, the output layer, is consisted with units which produce integral outputs corresponding with the highest probability density function value.

Any reader who wants to learn more about probabilistic neural network could read Y. Freund and R. Schapire's paper (1995, 1996) or P. Wasserman's book (1993).

### 2.4 Cross-validation and algorithm criterion

In this paper we used 5-fold cross validation for testing our method. In the process of a  $k$ -fold cross validation, the entire sample set was divided into  $k$  subsets with equal sizes randomly. In each turn, one subset was used as the testing set and the other  $k-1$  subsets were combined to train the Boost-PNN method. The final prediction result was generated by average the results in each turn.

The total prediction accuracy, the accuracy in each location and the Matthew's Correlation Coefficient (MCC) were used to assess of the prediction result.

Denote  $M_{ij}$  as the number of proteins observed in location  $i$  and predicted in location  $j$ , then the total number of proteins observed in state  $i$  is

$$obs_i = \sum_{j=1}^k M_{ij}, \text{ where } k \text{ is the number class. The total}$$

$$pre_i = \sum_{j=1}^k M_{ji}.$$

The total prediction accuracy and the prediction accuracy in location  $i$  are defined as follows:

$$Total\_Accuracy = \frac{\sum_{i=1}^k M_{ii}}{N} \quad (7)$$

$$Accuracy(i) = \frac{M_{ii}}{obs(i)} = \frac{M_{ii}}{\sum_{j=1}^k M_{ij}} \quad (8)$$

Matthew's Correlation Coefficient (MCC) is defined as follows:

$$MMC_i = \frac{p_i \cdot n_i - u_i \cdot o_i}{\sqrt{(p_i + u_i)(p_i + o_i)(n_i + u_i)(n_i + o_i)}} \quad (9)$$

$$p_i = M_{ii} \quad n_i = \sum_{j \neq i} \sum_{k \neq i} M_{jk}$$

$$o_i = \sum_{j \neq i} M_{ji} \quad u_i = \sum_{j \neq i} M_{ij}$$

where  $p_i$  is the number of correctly predicted sequences in location  $i$ ,  $n_i$  is the number of correctly predicted sequences do not in location  $i$ ,  $u_i$  is the number of under-predicted sequences and  $o_i$  is the number of over-predicted sequences.

### 3 Results

#### 3.1 Prediction result and comparison

The prediction results from the Boost-PNN method were compared with that of other subcellular localization prediction methods. Reinhardt and Hubbard's dataset (1998) was also tested with neural network method, the markov chain model (Yuan, 1999), the covariant discriminant algorithm (Chou, 1999), and the SVM method (Hua & Sun, 2001). All these methods except the markov chain model are based on amino acid composition alone. The prediction results for eukaryotic and prokaryotic proteins were summarized in Table 1 and Table 2, respectively. The results of the covariant discriminant algorithm, the markov chain model and the SVM method were obtained by the jackknife test while the results of the neural network method and the Boost-PNN method were obtained with cross validation test.

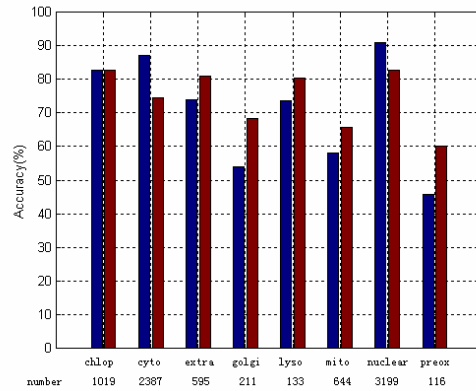
The results showed that the total accuracy of the

Boost-PNN method was 15.4% higher than that of the neural network method and 2.0% higher than that of the SVM method for eukaryotic proteins. For cytoplasm and nuclear sequences, its prediction accuracies were 27% and 16.8% higher than the neural network method, and 5.1% and 1.4% higher than the SVM method. The results of MCC for each subcellular location were also shown in Table 1. For the prokaryotic sequences, the total accuracy using our method was about 11.8% higher than neural network, 6.3% higher than covariant discriminant algorithm and 1.4% higher than the SVM method. The accuracy of cytoplasm sequences reached up to 98.9%, which is 18.9% higher than the neural network method.

The comparison between the Boost-PNN method and the SVM method was exciting. The SVM method is widely considered as the most powerful single classifier algorithm. Except the accuracy of extracellular, all other prediction results using the Boost-PNN methods had improvement compared with the SVM method. The improvement of MCC is more significant than the improvement of accuracies. With the Boost-PNN method, the MCC of every subcellular location were obviously higher than the corresponding one from the SVM method.

Our method was also compared with the markov chain model, which was based on the full sequence information including the order information while the Boost-PNN method was based only on the amino acid composition. The total accuracy using our method was 8.4% higher for eukaryotic proteins and 3.7% higher for prokaryotic proteins. For the eukaryotic and prokaryotic proteins, the MCC of each subcellular location using our method is significant higher than the corresponding one from the markov chain model.

For our new data with 8304 proteins and 8 subcellular locations, the total accuracy achieved 83.2%. The values of accuracy and MCC for each subcellular location were shown in Figure 2. The location with more proteins usually has higher accuracy than those with fewer proteins. This implicates that sufficient training data will help to improve the prediction accuracy.



**Figure 2:** The prediction result of our new dataset with 8304 proteins and 8 subcellular locations. The blue bar represents the accuracy of a particular subcellular location and the red bar represents the MCC value of the corresponding subcellular location. The number under the figure is the protein number of the corresponding subcellular location. The result is based on 5-fold cross validation. The boosting iteration number  $T=50$ ,

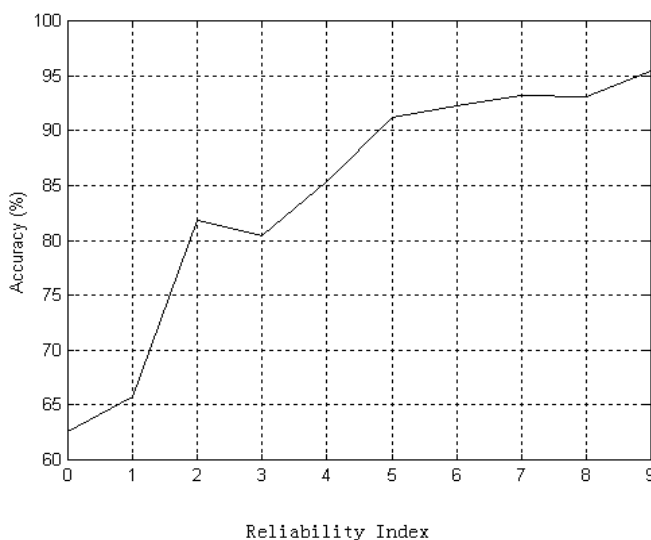
$Spread=0.025$ , the size of the boosting set is 20% of the total training set.

### 3.2 Reliability index

It is important to know the prediction reliability when using machine learning approaches for the predicting the protein subcellular localization. One of the common used definitions of reliability index is the difference between the largest and the second largest output value<sup>8,10</sup>. The formal formula of RI can be described as follows:

$$RI = \begin{cases} \text{INTEGER}(\text{diff}/0.55) & \text{if } 0 \leq \text{diff} \leq 4.95 \\ 9 & \text{if } \text{diff} > 4.95 \end{cases}$$

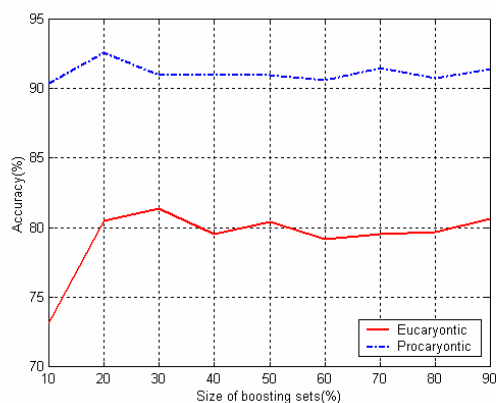
The RI assignment provides useful information about the level of certainty in the prediction for a certain protein sequence. The statistical relation between the value of RI and the prediction accuracy for eukaryotic proteins is shown in Figure 3. Similar curves can be obtained for prokaryotic cases (data not shown).



**Figure 3.** Expected prediction accuracy with a reliability index equal to a given value.

## 4 Discussion and conclusion

### 4.1 Parameter selection



**Figure 4.** Prediction accuracy with different size of the boosting set.

There are three parameters needed to be optimized with experiments. One is the boosting iteration number  $T$ . The second one is  $Spread$ , the so called “smooth parameter” in probabilistic neural network. In our experiments, we found the accuracy didn’t have significant change when the value of  $T$  is larger than 15. When the value of  $Spread$  is between 0.01 and 0.05, the final accuracy has no significant differences. This reflects that boosting algorithm is rather robust to the parameter mutation of the base classifiers. The third parameter need to be adjusted is the size of the boosting set. We have done experiments with different boosting set sizes for a large range from 10% to 90% of the total training samples. For both eukaryotic and prokaryotic proteins, the accuracy performance of the Boosting-PNN method is not very sensitive to the boosting set size for a large range from 10% to 90%. The prediction accuracy under different sizes of boosting sets is shown in Figure 4.

### 4.2 Further work

There are several ways to further improve the prediction performance. One way is to combine other complementary methods. Mitochondrial proteins were still not well predicted (61.4%) by the Boost-PNN method, although the accuracy has been higher than that of all other prediction methods based only on amino acid composition. About 22% of the mitochondrial proteins were misclassified into cytoplasmic. This reflects that it is difficult to discriminate the mitochondrial proteins from cytoplasmic proteins due to the similar amino acid compositions between the cytoplasmic and mitochondrial sequences. Those methods based on sorting signals can effectively recognize mitochondrial proteins, but the prediction accuracy is sensitive to the errors in the N-terminal sequence (Hua & Sun, 2001). In contrary, the performance of methods based on amino acid composition is robust to the errors in N-terminal sequence. Therefore, we believe that a combination of complementary methods will improve the prediction accuracy.

The second way in future work is to incorporate other informative features, including gene expression profile and regulatory pathway information. Drawid and Gerstein<sup>19</sup> have localized all the yeast proteins using a Bayesian system integrating features in the whole genome expression data. Some information fusion technologies, such as the Meta learning methods<sup>20</sup> may be used to combine the information from different datasets.

## 5 Conclusion

In this paper, a novel method for protein subcellular localization prediction is presented. This is the first attempt to apply the boosting algorithm in the field of genome sequence analysis and protein function prediction. This new approach provides superior prediction performance compared with existing algorithms based on amino acid composition. It is also a complementary method to other existing methods based on sorting signals. In conclusion, we have developed a powerful and effective method for protein subcellular localization

prediction. It is anticipated that this prediction method would be a useful tool for large-scale genome function analysis.

## 6 Acknowledgements

The author would like to thank Prof. A.Reinhardt for providing the dataset. This work was supported by a National Nature Science Grant (No. 19947006) and The National High Technology Research and Develop Program of China (863 Program).

## 7 References

- Andreas L. Prodromidis, Philip Chan and Salvatore J. Stolfo, "Meta-Learning in Distributed Data Mining Systems: Issues and Approaches", Book on "Advances of Distributed Data Mining", editors Hillol Kargupta and Philip Chan, AAAI press, 2000.
- Cedano, J., Aloy, P., Perez-Pons, J. A., and Querol, E. Relation between amino acid composition and cellular location of proteins. *J. Mol. Boil.*, 1997, **266**: 594-600.
- Chou, K.C. and Elord, D. Protein subcellular location prediction. *Protein Eng.*, 1999, **12**: 107-118.
- Drawid, A., and Gerstein, M. (2000) A Bayesian system integrating expression data with sequence patterns for localizing proteins: comprehensive application to the Yeast genome. *J. Mol. Boil.* **301**, 1059-1075.
- Emanuelsson, O., Nielsen, H., Brunak, S. and von Heijne, G. Predicting subcellular localization of proteins based on their N-terminal amino acid sequence. *J. Mol. Boil.*, 2000, **300**:1005-1016.
- Freund, Y. and Schapire, R.. A decision-theoretic generalization of on-line learning and an application to boosting. Unpublished manuscript. An extended abstract appeared in Computational Learning Theory: Second European Conference, *EuroCOLT'95*, 1995, 23-37, Springer-Verlag.
- Freund, R. and Schapire, R.. Experiments with a new boosting algorithm. *In Proc of 13<sup>th</sup> Intl. Conf. On Machine Learning*, 1996
- Frishman, D., Mironov, A. and Gelfand, M. Start of bacterial genes: estimating the reliability of computer prediction. *Gene*, 1999, **234**:257-265
- Hua, S.J. and Sun, Z.R Support vector machine approach for protein subcellular localization prediction. *Bioinformatics*, 2001, **17**: 721-728.
- Matthew, B. W. Comparison of predicted and observed secondary structure of T4 phage lysozyme. *Biochim. Biophys. Acta*, 1975, **405**:442-451
- Nakai, K. Protein sorting signals and prediction of subcellular localization. *Advances in Protein Chemistry*, 2000, **54**: 277-344.
- Nakashima, H., and Nishikawa, K. Discrimination of intracellular and extracellular proteins using amino acid composition and residue-pair frequencies. *J. Mol. Boil.*, 1994, **238**: 54-61.
- Nielsen, H., Engelbrecht, J., Brunak, S. and von Heijne, G. A neural network method for identification of prokaryotic and eukaryotic signal peptides and prediction of their cleavage sites. *Int. J. Neural Sys.*, 1997, **8**: 581-599.
- Nielsen, H., Brunak, S. and von Heijne, G. Machine learning approaches for the prediction of signal peptides and other protein sorting signals. *Protein Eng.*, 1999, **12**: 3-9.
- Parzen, E. On estimation of a probability density function and mode. *Annals of Mathematical Statistics*, 1962, **33**:1065-1076
- Reinhardt, A. and Hubbard, T. Using neural networks for prediction of the subcellular location of proteins. *Nucl. Acids Res.*, 1998, **26**: 2230-2236.
- Robert E. Schapire. The strength of weak learnability. *Machine learning*, 1993, 5(2):197-227
- Rochery, M., Schapire, R., Rahim, M., Gupta, H., Riccardi, G., Bangalore, S., Alshawi, H. and Douglas, S.. Combining prior knowledge and boosting for call classification in spoken language dialogue. *In International Conference on Accoustics, Speech and Signal Processing*, 2002.
- Ross, J. Quinlan. C4.5: Programs for Machine Learning. *Morgan Kaufmann*, 1993
- Schapire, R.E and Singer, Y.. BoosTexter: A boosting-based system for text categorization. *Machine Learning*, 2002, 39(2/3):135-168
- Specht, D. Probabilistic Neural Networks. *Neural networks*, 1990, **3**:109-118
- Wasserman, P.D. *Advanced methods in neural computing*. pp33-55. 1993
- Yuan, Z. Prediction of protein subcellular locations using Markov chain models. *FEBS Letters*, 1999, **451**, 23-26.

| Location       | Neural network | Markov model       |      | SVM                |      | Boost-PNN          |                    |
|----------------|----------------|--------------------|------|--------------------|------|--------------------|--------------------|
|                | Accuracy (%)   | Accuracy (%)       | MCC  | Accuracy(%)        | MCC  | Accuracy(%)        | MCC                |
| Cytoplasmic    | 55             | 78.1               | 0.60 | 76.9               | 0.64 | <b><u>82.0</u></b> | <b><u>0.71</u></b> |
| Extracellular  | 75             | 62.2               | 0.63 | <b><u>80.0</u></b> | 0.78 | 74.8               | <b><u>0.81</u></b> |
| Mitochondria   | 61             | <b><u>69.2</u></b> | 0.53 | 56.7               | 0.58 | 61.4               | <b><u>0.60</u></b> |
| Nuclear        | 72             | 74.1               | 0.68 | 87.4               | 0.75 | <b><u>88.8</u></b> | <b><u>0.80</u></b> |
| Total accuracy | 66             | 73.0               | --   | 79.4               | --   | <b><u>81.4</u></b> | --                 |

**Table 1:** The comparisons of different methods for the eukaryotic sequences. The result of neural network model and Boosting-PNN model are given by cross validation. The Markov model and SVM results were given by the jackknife (leave one out cross validation). Here, the boosting number T=30, *Spread*=0.025. The size of boosting set is 30% of the total training set.

| Location       | Neural network   | Covariant discrimination | Markov model |      | SVM          |      | Boost-PNN          |                    |
|----------------|------------------|--------------------------|--------------|------|--------------|------|--------------------|--------------------|
|                | Accuracy (%)     | Accuracy (%)             | Accuracy (%) | MCC  | Accuracy (%) | MCC  | Accuracy (%)       | MCC                |
| Cytoplasmic    | 80               | 91.6                     | 93.6         | 0.83 | 97.5         | 0.86 | <b><u>98.9</u></b> | <b><u>0.90</u></b> |
| Extracellular  | 77               | 80.4                     | 77.6         | 0.77 | 75.7         | 0.77 | <b><u>80.4</u></b> | <b><u>0.81</u></b> |
| Periplasmic    | <b><u>85</u></b> | 72.7                     | 79.7         | 0.69 | 78.7         | 0.78 | 78.7               | <b><u>0.81</u></b> |
| Total accuracy | 81               | 86.5                     | 89.1         | --   | 91.4         | --   | <b><u>92.8</u></b> | --                 |

**Table 2 :** The comparisons of different methods for the prokaryotic sequences. The result of neural network model and Boosting-PNN model are given by cross validation. The Markov model and SVM result were given by the jackknife (leave one out cross validation). Here, the boosting number T=30, *Spread*=0.03. The size of boosting set is 20% of the total training set.