Prediction of protein secondary structure using large margin nearest neighbour classification

Wei Yang
Biocomputing Research Centre,
School of Computer Science and Technology,
Harbin Institute of Technology,
Harbin 150001, China
and
School of Computer and Information Engineering,
HeNan University,
Kaifeng, 475004, China
E-mail: yang0sun@gmail.com

Kuanquan Wang*
and Wangmeng Zuo
Biocomputing Research Centre,
School of Computer Science and Technology,
Harbin Institute of Technology,
Harbin 150001, China
E-mail: wangkq@hit.edu.cn
E-mail: cswnmzuo@gmail.com
*Corresponding author

Abstract: In this paper, we introduce a novel method for protein secondary structure prediction by using Position-Specific Scoring Matrices (PSSM) profiles and Large Margin Nearest Neighbour (LMNN) classification. Since the PSSM profiles are not specifically designed for protein secondary structure prediction, the traditional nearest neighbour method could not achieve satisfactory prediction accuracy. To address this problem, we first use a LMNN model to learn a Mahalanobis distance metric for nearest neighbour classification. Then, an energy-based rule is invoked to assign secondary structure. Tests show that the proposed method obtains better prediction accuracy when compared with previous nearest neighbour methods.

Keywords: nearest neighbour; distance metric; large margin; protein secondary structure prediction.


Biographical notes: Wei Yang is a PhD student in the Biocomputing Research Centre at the Harbin Institute of Technology. His research interests are in the areas of machine learning, pattern recognition and bioinformatics.
1 Introduction

Protein secondary structure prediction plays an important role in structural biology. Accurate protein secondary structure prediction not only helps in understanding the function and the three-dimensional structure of a protein, but also is valuable in determining subcellular locations and improving the sensitivity of fold recognition methods (Montgomerie et al., 2006). Moreover, protein secondary structure prediction has become a routine part of protein analyses and annotation (Rost et al., 2004; Pollastri et al., 2007; Podtelezhnikov and Wild, 2009).

Dictionary of Secondary Structure of Proteins (DSSP) (Kabsch and Sander, 1983) classifies the secondary structure into eight categories (H (alpha helix), G (310 helix), I (pi helix), E (beta strand), B (isolated beta-bridge), T (turn), S (bend) and – (rest)). According to the convention of Evaluation of automatic structure prediction servers (Rost and Eyrich, 2001), these eight categories are reduced into three states: H, G and I to helix (H); B and E to strand (E); other categories to coil (C). Therefore, the protein secondary structure prediction can be considered as a typical classification problem, where each residue of a protein sequence is assigned a conformational state, H, E or C.

Various algorithms have been developed for protein secondary structure prediction. Early methods were mainly based on statistics and physico-chemical properties, such as GGBSM (Gascuel and Golmard, 1988). However, these algorithms usually cannot achieve three-state per-residue accuracy of 70% and thus lack practicability. It was not until 1993 that the prediction accuracy of protein secondary structure obtained the first breakthrough by combining neural networks with multiple sequence alignments derived from homology-derived structures of proteins (HSSP) database (Rost and Sander, 1993). Subsequently, Jones (1999) further improved prediction accuracy by incorporating PSSM obtained using the PSI-BLAST program into neural networks. As a result, many successful protein secondary structure prediction systems, such as Jpred (Cole et al., 2008), PROTEUS (Montgomerie et al., 2006), Porter (Pollastri and McLysaght, 2005), PSIPRED (McGuffin et al., 2000) and PHDpsi (Przybylski and Rost, 2002), are based on neural networks. Furthermore, using PSSM matrix, other classification algorithms, such as multiple predictor fusion (Palopoli et al., 2009), K-nearest neighbour (Joo et al., 2004),
Prediction of protein secondary structure

support vector machine (Karypis, 2006; Chang et al., 2008; Nguyen and Rajapakse, 2007; Kountouris and Hirst, 2009) and hidden Markov model (Malekpour et al., 2009; Madera et al., 2010), have also shown good prediction performance.

Several $K$-nearest neighbour methods, such as NNSSP (Salamov and Solovyev, 1995) and PREDICT (Joo et al., 2004), have also been proposed for protein secondary structure prediction. By far, most state-of-the-art protein secondary structure prediction methods are based on PSSM profiles. However, PSSM profiles are obtained by using the PSI-BLAST program, which is not designed for protein secondary structure prediction. Thus, if we could learn a set of features from PSSM profiles that are specifically designed for protein structure prediction, the prediction accuracy would be expected to be further improved. In this paper, using the nearest neighbour framework, we investigate the feasibility and effectiveness of this strategy. If the feature extraction is linear and the classifier is $K$-nearest neighbour, the task would be to learn a transform matrix, which is equivalent to a distance metric learning problem. By analysing and comparing several distance metric learning approaches, we use the LMNN (Weinberger and Saul, 2009) classification to learn the optimal Mahalanobis distance metric for protein secondary structure prediction.

In this paper, we present a novel nearest neighbour method for protein secondary structure prediction. Our method first constructs the feature vector based on the PSSM matrix. Then, the LMNN model is used for distance metric learning to obtain a linear transform matrix. Finally, we employ an energy-based rule to assign secondary structure. When compared with the previous nearest neighbour methods, the proposed method shows significant improvement. Furthermore, the prediction performance of our method is comparable with that of the state-of-the-art methods.

In Section 2.1, the datasets used in our work are introduced. In Section 2.2, the LMNN model is briefly reviewed. In Section 2.3, we give two protein secondary structure assignment rules. Section 3 describes the performance measures. The results and discussion are presented in Section 4. Section 5 concludes our work by summarising our main contributions.

2 Materials and methods

2.1 Datasets

Two public datasets were employed to evaluate our method. The first is the RS126 dataset (Rost and Sander, 1993), which was constructed by Rost and Sander and contains 126 protein chains. It is non-redundant, i.e., no two protein chains in the dataset have more than 25% sequence identity over a length of more than 80 residues. The second is the CB513 dataset (Cuff and Barton, 2000), which was constructed by Cuff and Barton and contains 513 non-redundant protein chains evaluated with the more stringent SD score than the percentage identity. The PSSM matrices corresponding to the protein chains in the datasets were generated using PSI-BLAST with three iterations against the NCBI nr database, which was filtered to remove low-complexity regions, transmembrane regions and coiled-coil segments. Each PSSM matrix is composed of $n \times 20$ integers ranging from $-9$ to 13, where $n$ is the length of the protein chain, 20 corresponds to 20 amino acids. The feature vector for each residue is constructed by considering $w$ consecutive residues centred on the target residue. We use $w = 11$ in this work. Since
each residue corresponds to 20 integer elements in PSSM matrix, the resulting feature is a 220 dimensional vector. Besides, for the residues on the N terminal and the C terminal, ‘empty’ feature attributes are filled with zeros. Using the example protein \textit{latp}, Figure 1 illustrates how to process protein sequence.

Figure 1  Flow chart illustrates how to process protein sequence

2.2 The LMNN model

The goal of distance metric learning is to derive a more discriminative distance metric for protein secondary structure predication from a set of labelled examples. To this end, several distance metric learning approaches, such as neighbourhood components analysis (Goldberger et al., 2005) and relevant component analysis (Bar-Hillel et al., 2005), have been developed. However, the objective functions of these approaches are non-convex and thus prone to suffer from local minima. Different from the methods mentioned earlier, the recently proposed LMNN classification model formulated the distance metric learning problem in $K$-nearest neighbour classification as a convex semi-definite programming, which implies that the globally optimal solution can be obtained from the training data. Therefore, we adopt it to learn a distance metric for protein secondary structure prediction.
Prediction of protein secondary structure

Given a training set \( T = \{(x_1,y_1), \ldots, (x_n,y_n)\} \), where \( x_i \) is a feature vector, \( y_i \) is the class label and \( n \) is the number of samples, LMNN model attempts to learn a linear transformation such that each sample and its \( K \) nearest neighbours share the same label whereas examples with different labels are separated by a large margin. At the beginning of learning, LMNN model requires to assign target neighbours for each sample in the training set. The target neighbours of a sample are the sets of samples that have the same class label and are expected to be closest to the sample. In our work, the first three nearest neighbours with the same label under Euclidean distance are considered as the target neighbours. Specially, the notation \( j \rightarrow i \) is used to indicate \( x_j \) is a target neighbour of \( x_i \).

Let \( x' = Lx \) be a linear transformation that needs to be learned. Under this linear transformation, the squared distance between two samples \( x_i \) and \( x_j \) can be denoted by:

\[
D_L(x_i, x_j) = \|L(x_i - x_j)\|^2 = (x_i - x_j)^T L^T L (x_i - x_j).
\]

To minimise the distance between a sample and its target neighbours, the transformation matrix \( L \) should be chosen to minimise

\[
\epsilon_1(L) = \sum_{i,j=1} \|L(x_i - x_j)\|^2.
\]

Moreover, \( L \) should satisfy that the samples with different labels have large distances. To address this, a new term, *impostor*, is introduced. Given a sample \( x_i \) and its target neighbour \( x_j \), an impostor is any sample \( x_l \) that has different label with \( x_i \) and satisfies the constraint

\[
\|L(x_i - x_j)\|^2 \leq \|L(x_i - x_l)\|^2 + 1.
\]

To minimise the number of impostors, Weinberger and Saul (2009) suggested minimising:

\[
\epsilon_2(L) = \sum_{i,j=1} \sum_l (1 - y_{il}) \left[ 1 + \|L(x_i - x_j)\|^2 - \|L(x_i - x_l)\|^2 \right],
\]

where \( y_{il} = 1 \) if and only if \( y_i = y_l \), otherwise \( y_{il} = 0 \); the function \( [z]_+ = \max(z,0) \). In particular, when \( D(x_i, x_l) \geq D(x_i, x_j) + 1 \), the hinge loss will be zero and thus makes no contribution to the total sum. By further combining \( \epsilon_1(L) \) and \( \epsilon_2(L) \) into a single term, the total loss function is given by \( \epsilon(L) = \epsilon_1(L) + \epsilon_2(L) \). To address the non-convexity of the function \( \epsilon(L) \), a positive semi-definite matrix \( M = L^T L \) is introduced. By further introducing nonnegative slack variable \( \xi_{il} \) to represent the hinge loss, the minimisation of the loss function \( \epsilon(L) \) can be formulated as a convex semi-definite programming problem:

\[
\begin{align*}
\min & \quad \sum_{i,j=1} (x_i - x_j)^T M(x_i - x_j) + \sum_{i,j=1} \sum_l (1 - y_{il}) \xi_{il} \\
\text{s.t.} & \quad (x_i - x_j)^T M(x_i - x_j) - (x_i - x_j)^T M(x_i - x_j) \geq 1 - \xi_{il} \\
& \quad \xi_{il} \geq 0 \\
& \quad M \succeq 0.
\end{align*}
\]
This semi-definite programming can be solved efficiently by using the sub-gradient descent algorithm. The detailed description of the algorithm can be found in Weinberger and Saul (2009). The linear transform matrix \( L \) can be obtained by decomposing the matrix \( M \). Furthermore, note that when the matrix \( L \) in equation (1) is substituted by the matrix \( M \), the resulting distance is the Mahalanobis distance. Therefore, the goal of LMNN model can be considered as learning an optimal Mahalanobis distance.

Using a sample \( x_i \) with three target neighbours, Figure 2 illustrates the sample’s behaviour in the LMNN model. The solid squares denote the target neighbours and the impostors are represented by the solid circle. Arrows indicate the gradient directions on distance arising from the optimisation of objective loss function. During the training process, the target neighbours become close to \( x_i \), at the same time as the impostors become farther to \( x_i \). This leads to the robust LMNN classification.

**Figure 2** The sample’s behaviour in the LMNN model

### 2.3 Protein secondary structure prediction

After the linear transform matrix \( L \) is obtained, the fuzzy \( K \)-nearest neighbour rule and the energy-based rule are used to assign protein secondary structure. In the fuzzy \( K \)-nearest neighbour rule, the squared distance in equation (1) is first used to determine \( K \)-nearest neighbours of the test sample. Then, class memberships are assigned to the test sample \( x_t \) according to the following equation:

\[
  u_s(x_t) = \frac{\sum_{j=1}^{K} u_s(x^{(j)}) \|L(x_t - x^{(j)})\|^{-2(m-1)}}{\sum_{j=1}^{K} \|L(x_t - x^{(j)})\|^{-2(m-1)}}, \quad s = H, E, C
\]

where \( m \) is a fuzzy strength parameter, which determines how heavily the distance is weighted when calculating each neighbour’s contribution to the membership value. Also, \( x^{(j)} \) is the \( j \)th nearest neighbour of the test sample \( x_t \), \( u_s(x^{(j)}) \) is the indicator function to denote whether the membership value of the \( j \)th neighbour is the class \( s \), which is 1 if \( x^{(j)} \) belongs to the class \( s \), and 0 otherwise. After the calculation of memberships is completed, the class with the maximum membership value is assigned to the test sample.

In the energy-based rule (Weinberger and Saul, 2009), the classification of a test sample \( x_t \) is done by considering it as an extra training example and calculating the loss function similar to that in LMNN model for every possible class label. To compute the loss function, the target neighbours of the test sample under different hypothetical labels...
should be determined in advance. For each hypothetical label $y_t$, the first $K$ nearest training samples with the same label determined by the squared distance in equation (1) are considered as the target neighbours of the test sample. Note that the determination of target neighbours of the test sample depends on the linear transform matrix $L$, which is different from that of training samples. Given the test sample $x_t$ and its hypothetical label $y_t$, the corresponding loss function is defined as:

$$
E_p(y_t) = \sum_{i \neq t} D_L(x_t, x_i) + \sum_{j \neq t} (1 - y_t)[1 + D_L(x_t, x_j) - D_L(x_t, x_j)] + \sum_{i,j \neq t} (1 - y_t)[1 + D_L(x_t, x_j) - D_L(x_t, x_j)].
$$

Since a smaller loss function means a more reliable label, the class with the minimal loss value is assigned to the test sample.

Moreover, to avoid the prediction results contain incorrect secondary structure content (HEH, EHE and CHC), we perform a smoothing operation to convert HEH, EHE and CHC into HHH, EEE and CCC, respectively.

### 3 Performance measures

Three widely used measures, $Q_3$, Segment Overlap Measure (SOV) (Zemla et al., 1999) and the Matthews correlation coefficient (Matthews, 1975), were employed to evaluate the performance of the proposed method. $Q_3$, a measure of the overall three-state per-residue prediction accuracy, denotes the percentage of residues whose secondary structures are correctly predicted:

$$
Q_3(\%) = \frac{N_c}{N} \times 100
$$

where $N_c$ is the total number of correctly predicted residues and $N$ is the total number of observed residues. The SOV is a structurally more meaningful measure because it can efficiently capture the structurally important features including secondary structure segments. Its definition is complicated and the reader can refer to Zemla et al. (1999) for more detailed explanations.

The Matthews correlation coefficient (Matthews, 1975) is generally regarded as a balanced measure, which can be used even though the classes are of very different sizes. It returns a value between $-1$ and $+1$. A coefficient of +1 represents a perfect prediction and −1 a completely inverse prediction. For each of secondary structure state H, E and C, the Matthews correlation coefficient is given by

$$
C_i = \frac{p_i n_i - u_i o_i \sqrt{(p_i + u_i)(p_i + o_i)(n_i + u_i)(n_i + o_i)}}{(p_i + u_i)(p_i + o_i)(n_i + u_i)(n_i + o_i)}
$$

where $p_i$, $n_i$, $u_i$ and $o_i$ are defined as follows:

- $p_i$: the number of residues that are of state $i$ and are correctly predicted to be state $i$.
- $n_i$: the number of residues that are of state $i$ and are not correctly predicted to be state $i$.
- $u_i$: the number of residues that are not of state $i$ and are correctly predicted to be state $i$.
- $o_i$: the number of residues that are not of state $i$ and are not predicted to be state $i$.
4 Results and discussion

In this work, seven-fold cross validation on the two datasets RS126 and CB513 was performed to assess the proposed method. In the seven-fold cross validation, each dataset is divided into seven subsets with approximately equal numbers of protein chains. Of the seven subsets, a subset is chosen as the validation data for testing, and the remaining six subsets are used as training data. The cross-validation process is then repeated seven times, with each of the seven subsets used exactly once as the validation data. In particular, the samples with 'empty’ feature attributes are only used for testing since they could not represent the real local environment of the target residue.

The performance comparison of the proposed method under different distance metrics and the assignment rules is shown in Table 1. From Table 1, we can see that the Mahalanobis distance is significantly better than the Euclidean distance on both datasets. In particular, the \( Q_3 \) and SOV prediction accuracies achieved by the former are 4.75% and 4.32% higher than those of the latter on the CB513 dataset, respectively. This result indicates that distance metric learning is critical to \( K \)-nearest neighbour classification. Based on the learned Mahalanobis distance, we also investigate the impact of the two assignment rules to the prediction accuracies of secondary structure prediction. Compared with the fuzzy \( K \)-nearest neighbour rule, it can be observed from Table 1 that the energy-based rule obtains better prediction accuracies, which can be explained by the fact that the energy-based rule could alleviate the influence of unbalanced class distribution. Moreover, by comparing the prediction results on the RS126 dataset with those on the CB513 dataset, we can see that the increase in the size of the training set generally enhances the prediction accuracies, which is consistent with previous studies.

### Table 1: Effect of distance metric and assignment rule on the performance of the proposed method

<table>
<thead>
<tr>
<th>Method</th>
<th>RS126 dataset</th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>( Q_3 )</td>
<td>SOV</td>
<td>( C_{id} )</td>
</tr>
<tr>
<td>Fuzzy KNN + Euclidean</td>
<td>69.68</td>
<td>66.75</td>
<td>0.62</td>
<td>0.47</td>
</tr>
<tr>
<td>Fuzzy KNN + Mahalanobis</td>
<td>72.54</td>
<td>69.56</td>
<td>0.66</td>
<td>0.52</td>
</tr>
<tr>
<td>Energy + Mahalanobis</td>
<td>75.09</td>
<td>72.18</td>
<td>0.69</td>
<td>0.58</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Method</th>
<th>CB513 Dataset</th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>( Q_3 )</td>
<td>SOV</td>
<td>( C_{id} )</td>
</tr>
<tr>
<td>Fuzzy KNN + Euclidean</td>
<td>69.15</td>
<td>70.19</td>
<td>0.58</td>
<td>0.47</td>
</tr>
<tr>
<td>Fuzzy KNN + Mahalanobis</td>
<td>73.90</td>
<td>74.51</td>
<td>0.66</td>
<td>0.56</td>
</tr>
<tr>
<td>Energy + Mahalanobis</td>
<td>75.44</td>
<td>75.66</td>
<td>0.69</td>
<td>0.59</td>
</tr>
</tbody>
</table>

The optimal values of the parameters \( m \) and \( K \) are determined by seven-fold cross validation. In particular, we found that the fuzzy strength parameter \( m \) has only little influence on the prediction accuracies of the proposed methods and its value is set to 2 in our work. For the three combination strategies, Fuzzy KNN + Euclidean, Fuzzy KNN + Mahalanobis and Energy + Mahalanobis, their corresponding parameters \( K \) are set to (75, 80, 6) and (240, 80, 20) on the RS126 and CB513 datasets, respectively.

Since the nearest neighbour methods, e.g., NNSSP (Salamov and Solovyev, 1995) and PREDICT (Joo et al., 2004), and support vector machine methods, e.g., SVMfreq (Hua and Sun, 2001), SVMpsi (Kim and Park, 2003) and YASSPP (Karypis, 2006), have
been tested on the RS126 or CB513 datasets in a similar experimental setup, the direct comparison of our method with them can be made. Tables 2 and 3 show the performance comparison of our method against the methods mentioned above. From these tables, it can be observed that the prediction accuracies of our method are higher than those of SVMfreq, NNSSP and PREDICT. In particular, the SOV score of our method is 5.96% higher than that of PREDICT on the CB513 dataset. Actually, PREDICT is a cascaded method using PSSM profiles and performs two-layer nearest neighbour calculation, sequence-to-structure and structure-to-structure, for a prediction. If only the first layer’s prediction results is considered, the $Q_3$ and SOV accuracies of PREDICT are 70.9% and 65.4% on the CB513 dataset, respectively, which are significantly lower than those of our method. By using the two-layer cascaded scheme, YASSPP shows better performance when compared with our method. Therefore, if the structure-to-structure prediction is introduced, the performance of our method may be further improved. Moreover, it should be noted that the prediction accuracies of our method are comparable with those of SVMpsi. Although the $Q_3$ accuracy of the proposed method is lower than that of SVMpsi, the SOV accuracy of SVMpsi is lower than that of the proposed method. One possible explanation is that our method does not use the samples with ‘empty’ feature attributes in training and the smooth post-processing is performed to avoid incorrect secondary structure content.

<table>
<thead>
<tr>
<th>Method</th>
<th>$Q_3$</th>
<th>SOV</th>
<th>$C_{IH}$</th>
<th>$C_E$</th>
<th>$C_C$</th>
</tr>
</thead>
<tbody>
<tr>
<td>SVMfreq</td>
<td>71.60</td>
<td>–</td>
<td>0.62</td>
<td>0.52</td>
<td>0.51</td>
</tr>
<tr>
<td>SVMpsi</td>
<td>76.10</td>
<td>72.00</td>
<td>–</td>
<td>–</td>
<td>–</td>
</tr>
<tr>
<td>YASSPP</td>
<td>77.91</td>
<td>72.81</td>
<td>0.71</td>
<td>0.63</td>
<td>0.57</td>
</tr>
<tr>
<td>NNSSP</td>
<td>71.20</td>
<td>–</td>
<td>0.62</td>
<td>0.49</td>
<td>–</td>
</tr>
<tr>
<td>Our method</td>
<td>75.09</td>
<td>72.18</td>
<td>0.69</td>
<td>0.58</td>
<td>0.56</td>
</tr>
</tbody>
</table>

Entries marked with ‘–’ indicate that the data could not be obtained from literatures.

Table 3

<table>
<thead>
<tr>
<th>Method</th>
<th>$Q_3$</th>
<th>SOV</th>
<th>$C_{IH}$</th>
<th>$C_E$</th>
<th>$C_C$</th>
</tr>
</thead>
<tbody>
<tr>
<td>SVMfreq</td>
<td>73.50</td>
<td>–</td>
<td>0.65</td>
<td>0.53</td>
<td>0.54</td>
</tr>
<tr>
<td>SVMpsi</td>
<td>76.60</td>
<td>73.50</td>
<td>0.68</td>
<td>0.60</td>
<td>0.56</td>
</tr>
<tr>
<td>YASSPP</td>
<td>77.83</td>
<td>75.05</td>
<td>0.71</td>
<td>0.64</td>
<td>0.58</td>
</tr>
<tr>
<td>PREDICT (first layer)</td>
<td>70.90</td>
<td>65.40</td>
<td>0.48</td>
<td>0.46</td>
<td>0.47</td>
</tr>
<tr>
<td>PREDICT (second layer)</td>
<td>73.50</td>
<td>69.70</td>
<td>0.55</td>
<td>0.49</td>
<td>0.50</td>
</tr>
<tr>
<td>Our method</td>
<td>75.44</td>
<td>75.66</td>
<td>0.69</td>
<td>0.59</td>
<td>0.57</td>
</tr>
</tbody>
</table>

The histograms for the $Q_3$ and the SOV scores of our method on the RS126 and CB513 datasets are shown in Figure 3. One can see that the $Q_3$ and the SOV scores of most proteins are mainly located between 60% and 90%. For the low scoring outlier proteins ($Q_3$ or SOV scores are small than 60%), we find that there exist many short strand and
helix fragments in their secondary structures and most of these fragments are incorrectly predicted by our method. Therefore, how to efficiently predict the short strand and helix fragments should play an important role in the improvement of the proposed method. In the future work, we will concentrate on addressing this problem.

Figure 3  Histograms of prediction accuracies for the proposed method on the RS126 and CB513 datasets: (a) the distribution of $Q_3$ scores on the RS126 dataset; (b) the distribution of SOV scores on the RS126 dataset; (c) the distribution of $Q_3$ scores on the CB513 dataset and (d) the distribution of SOV scores on the CB513 dataset (see online version for colours)
5 Conclusions

In this paper, based on distance metric learning, we develop a novel K-nearest neighbour protein secondary structure prediction method. Tests on the RS126 and CB513 datasets show that the prediction accuracies of the proposed method are significantly better than those of previous nearest neighbour methods such as NNSSP and PREDICT. The improvement is mainly due to the use of distance metric learning technique and the energy-based classification rule. Furthermore, the proposed method shows the comparable performance with the state-of-the-art methods. To the best of our knowledge, this is the first time that the supervised distance metric learning has been introduced into the field of protein secondary structure prediction. In the future, we will further improve the proposed method by introducing structure-to-structure prediction and using multimetric distance learning (Weinberger and Saul, 2009). Besides, we will develop appropriate distance learning methods for other bioinformatics applications.

Acknowledgements

The authors acknowledge the many helpful suggestions of two anonymous reviewers. This work is supported in part by the National Natural Science Foundation of China under Nos. 60872099 and 60902099.

References


