Protein secondary structure prediction using distance based classifiers

Ashish Ghosh *, Bijnan Parai

Machine Intelligence Unit, Indian Statistical Institute, 203 B.T. Road, Kolkata 700108, India

Abstract

De novo structure determination of proteins is a significant research issue of bioinformatics. Biochemical procedures for protein structure determination are costly. Use of different pattern classification techniques are proved to ease this task. In this article, the secondary structure prediction task has been mapped into a three-class problem of pattern classification, where the classes are *helix*, *sheet* and *coil*. Here we have made an attempt to analyze this secondary structure prediction problem using three distance based classifiers (minimum distance, *K*-nearest neighbor and fuzzy *K*-nearest neighbor). The only information about the proteins used is the primary structure (sequence of amino acids) itself. A matrix-based new representation of such categorical data is used to convert the sequence into real numbers. A comparative study among these classifiers has been made based on some standard classification performance measures. From this study, it is found that the simple minimum distance classifier performs better compared to others.

1. Introduction

Number of unknown proteins is increasing exponentially compared to the number of known protein structures [1] thereby widening the gap of protein sequence-structure mapping. Protein secondary structure prediction is a task for predicting the conformational state of each amino acid in a protein sequence, e.g., to predict whether a given amino acid is a part of *helix*, *sheet* or *coil* structure. Since the biochemical methods for protein structure determination is too expensive, some computational tools that can predict protein structures are needed to narrow this widening gap. Pattern classification methods are popular tools that help to perform this prediction job. Thus, design and development of such classifiers is one of the prime concerns of bioinformatics.

Advancement of *in vitro* techniques enables availability of primary structure information of thousands of proteins. The three-dimensional conformational state (tertiary/quarternary structure) of a protein is dependent on the primary structure to a large extent. Function of a protein depends on its final conformational state. So, the determination of protein structure is an important problem. This problem is computationally not tractable till now. So we move for protein structure prediction. Again, since reliable tertiary structure prediction is far beyond, an intermediate step is to predict the secondary structure which is a way to simplify the prediction task. Note that the secondary structure of a protein is very useful to understand its three-dimensional structure and its function [2].

Accurate prediction of secondary structure of proteins is one of the greatest challenges in protein structure analysis. The task of secondary structure prediction of proteins is synonymous with identification of types of secondary structural elements, i.e., *helices*, *sheets* and random *coils*. It should be noted that location of an amino acid in the protein sequence along with the nature of its neighboring amino acids affect the overall secondary structure. Prediction tools like classifiers attempt to investigate this apparent relationship between amino acids sequences and their structures.

Protein secondary structure prediction can be mapped to a standard pattern classification problem. Structural categories of proteins are considered as classes, whereas, structural and functional units of proteins (amino acids) are treated as patterns. Since, amino acids are considered to be patterns, they need to be represented by numerical features that are responsible for classification. The only information available in a sequence is the symbolic name of the amino acid itself along with its neighbors. The underlying principle of the prediction problem is that the secondary structure of a particular amino acid is highly correlated with its neighboring amino acids [3–5]. Protein secondary structure also depends on the local short ranged interactions between the neighboring amino acids (residues) [6].

Research has been conducted for more than 40 years on prediction of protein secondary structures. At present we have several classification approaches that predict the secondary structure with acceptable accuracy. Neural network [7] based methods include PHD [8], PHDpsi [5], PROFsec [9], SSPro2 [10], JNET [11] and PSIPRED [12]. Although neural network based methods give higher accuracy, they suffer from some drawbacks. Black-box nature of neural networks makes it difficult to view how the structures are actually predicted [2]. Neural based methods along with hidden Markov models (HMM) [13] perform well when many homologs of query protein are available. This goes against generalization of prediction. It may be noted that use of support vector machines (SVM) introduced by Vapnik [14] improves prediction accuracy effectively [1].

Nearest-neighbor classifiers have been used mainly for predicting one category of secondary structures, e.g., *beta turn* [15]. In the present study, we have used the *K*-nearest neighbor method (a generalization of nearest neighbor) to handle the three-class secondary structure prediction problem. Conventional *K*-nearest neighbor method also has some drawbacks. It gives equal importance to all the classes. This difficulty can be reduced by using fuzzy *K*-nearest neighbor [16] method. In this investigation, we used both fuzzy *K*-nearest method and minimum distance classifiers for predicting secondary structures.

For all classifiers, we need to represent an amino acid as a numerical pattern. A stretch of amino acid sequence tends to attain a particular secondary structure depending on what kind of amino acids are present in that stretch, their properties and mode of arrangement. This is also affected by the other amino acids present in the chain (i.e., the total amino acid chain [17]). Thus, the ‘‘length of the stretch’’ is very important for prediction purpose. To make life simple, many methods examine a slice of sequence window and assume that the central amino acid in that window will adopt a conformation that is determined by side groups of all the amino acids present in that window. We exploit this idea to represent an amino acid as a pattern, i.e., an amino acid is represented as a pattern whose feature values come from the neighbors of the amino acid.

Using this new representation for patterns (amino acids) we investigated the problem of secondary structure prediction using distance based classifiers.

2. Classifiers for structure prediction

We have studied three different distance based classifiers for protein structure analysis. The required data used in classification process is collected from the standard non-homologous protein data set. An amino acid is represented as a pattern (discussed in Section 2.6) whose feature values come from the neighbors of the amino acid. Accuracy of different types of classifiers depends on classification principle as well as characteristics of patterns. For example, if the classes are linearly separable, then use of minimum distance classifier may be a wise decision, whereas, it is not useful if the classes are linearly non-separable. In that case *K*-nearest neighbor classifier can produce better results. In the following section, we discuss the working principles of the classifiers being used.

2.1. Minimum distance classifier

Minimum distance classifier basically generates the decision plane between two classes which is the perpendicular bisector of the plane between center points of the classes. A pattern is assigned a class label based on the closeness with respect to the class representatives. The representative of a class is assumed to be the mean point of all patterns belonging to that class. For an n-class problem, n numbers of means are determined from the training set. Let μ_i and μ_j represent the mean of class i and j, respectively. Now for each new pattern to be classified, distance is calculated from each of these mean points. Although different distance measures can be used for this, e.g., Euclidian distance, city block distance, Mahalanobis distance, etc; in this paper we used square of the Euclidean distance measure. This can be expressed as

$$d_k = \|(X - \mu_k)(X - \mu_k)^T\|. \tag{1}$$

where X is the new pattern vector, $X = [x1, x2, x3, \cdots, xn]$ and μ_k stands for mean of kth class. $\mu_k = [m1, m2, m3, \cdots, mn]$.

This classifier is very low cost in terms of both time and memory requirement. It gives sound results when the distribution of samples is regular, round or oval shaped and the classes are linearly separable.

2.2. K-nearest neighbor method

In most pattern recognition applications, the assumption of distribution of patterns is the prime suspect. Practically, the common parametric forms rarely fit the densities that actually encountered. In such cases, we can examine with non-parametric procedures that can be used with arbitrary distributions and without knowing the assumption of the form and their underlying densities. K-nearest neighbor is one such method that overlooks the probability estimation and go directly to decision function.

We begin by letting $D^n = \{X_1, X_2, \ldots, X_n\}$ denote a set of n labeled prototype and letting $X' \in D^n$ be the nearest to a test point X. Then (nearest-neighbor rule) for classifying X is to assign it the label associated with X' [18]. K-nearest-neighbor rule is an extension of the nearest-neighbor rule. A pattern X is classified by assigning it the label of most frequently encountered class from among K nearest samples. Thus, decision is made by majority voting contributed by its nearest K neighbors.

2.3. Fuzzy k-nearest neighbor

One of the problems encountered in K-nearest neighbor method is that each sample vector is considered equally important in assigning class label to unknown pattern irrespective of its actual degree of belonging to each class. To reduce this problem fuzzy techniques are incorporated with classical k-nearest neighbor rule.

Fuzzy K-NN introduced by Keller [16] addresses the aforementioned problem. Basically a fuzzy algorithm uses the fuzzy class membership value of samples and produces fuzzy class labels. Fuzzy concepts can be applied on three stages of a classifier, i.e., input stage, decision rule generation stage and output stage. The membership value of a pattern to a particular class denotes the level of assurance for being in that class.

The basis of fuzzy K-NN algorithm is to assign class membership to an unknown pattern as a function of that pattern's distance from its K-nearest neighbors and those neighbors' memberships in possible classes [16]. We find K-nearest neighbors of each sample X from the labeled sample set. Now, let $u_{i,j}$ be the membership value of the jth neighbor of pattern X to the ith class. The predicted membership value $u_i(X)$ of X to class i can be calculated as

$$u_i(X) = \frac{\sum_{j=1}^{K} u_{i,j} \times (1/\|d(X,X_j)^{2/m-1}\|)}{\sum_{j=1}^{K}(1/\|d(X,X_j)^{2/m-1}\|)}. \tag{2}$$

As evident from the above Eq. (2), degree of belonging of X to different classes are influenced by inverse distances from its nearest neighbors and their class membership. Inverse distance reflects that lesser the distance of a pattern from the representatives of a class, its degree of belonging to that class is more, and vice versa. $u_{i,j}$

is a $K \times C$ matrix, where C is the number of classes. Here the variable m, called as fuzzifier, determines how heavily the distance is weighted when calculating each of the neighbor's contribution to the membership value.

2.4. Data set

An effective prediction tool should have the capability of handling extreme non-homologous data. We used three data sets to test our algorithms. One of them is standard protein data set used by Rost and Sander [8], referred as RS126 set. This set consists of 126 non-homologous amino acid sequences; which means no two sequences in this data set share more than 25% sequence similarity. RS126 contains a total of 23,346 amino acids. Average sequence length of all the proteins in this set is 185. Another larger data set used in our algorithm is CB396 [11]. It was constructed by Cuff and Barton. CB396 has 396 number of non-redundant proteins whose average sequence length is 157. Total number of amino acids in this set is 57,996. The third data set contains 87 proteins which are more or less dissimilar having 22,031 amino acids.

We used the standard DSSP labels for the training samples [11]. DSSP distinguishes the amino acids into eight classes according to their secondary structures as, H (α-helix), G (3_{10}-helix), I (π-helix), E (β-strand), B (isolated β-bridge), T (turn), S (bend) and – (rest). These eight structural class can be reduced to three using two different reduction method; (i) H, G and $I \rightarrow H$; E and $B \rightarrow E$; rest are C, and (ii) H and $G \rightarrow H$; E and $B \rightarrow E$ and all other states as C [19]. We adopted the first one for simplicity.

2.5. Training and validation sets

The data set we considered is divided into two parts, one part is used as training set, and the rest is for testing. The classification is considered better which gives better result taking less amount of training data. Varying percentage of data from the total data set is used for training. As the percentage of training data increases, prediction accuracy also grows accordingly. But after a certain amount of training data, the rise in prediction becomes slower. We conducted experiments taking 10–30% data for training.

2.6. Data representation

Alphabetical characters represent the amino acids. Classifiers are not capable to handle this type of data. To make it accessible to the classifiers, we need to convert the alphabetic characters into some numeric values that is meaningful and holds biological significance. Information about individual amino acids are encoded using *unary encoding* scheme [20]. Class label of a single amino acid depends not only on itself, but depends on the effect of its neighboring amino acids also. To take care of this effect of neighbors we consider a window of length W while determining the feature values of the central amino acid. Thus, $W/2$ residues remain on either side of the central amino acid. To have equal number of neighbors in both the sides of the central residue, W is taken as odd.

Since the total number of amino acids is twenty, they are considered as a frame which consists of the symbolic representations of all the 20 amino acids. Each of the amino acids in a window is compared with this frame. This comparison results a 1 if there is a match, otherwise it is 0. Thus, a 0/1 vector of length 20 is generated for each amino acid of the window, having a single 1. For example, alanine, the first amino acid of the frame is represented as →(1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0). Thus, for a window, a matrix

$$M_{20\times W} = \begin{pmatrix} 0 & 0 & 1 & 0 & \cdots & 0 \\ 0 & 0 & 0 & 0 & \cdots & 0 \\ 1 & 0 & 0 & 1 & \cdots & 0 \\ 0 & 0 & 0 & 0 & \cdots & 1 \\ \vdots & \vdots & \vdots & \vdots & \cdots & \vdots \\ 0 & 1 & 0 & 0 & \cdots & 0 \end{pmatrix}$$

Fig. 1. Pattern representation for amino acids.

$M_{20\times W}$ is generated. This matrix represents a pattern for the amino acid present at the center. Such a window is now slided over the whole sequence. To generate patterns for the terminal amino acids of a sequence, zeros are padded to fit the window. This matrix representation of a pattern is shown in Fig. 1.

Class labels are represented by 1, 2, and 3; 1 representing helix class (H), 2 beta sheet(E) class and 3 for the rest.

3. Performance measures

To measure the performance of predictive methods, there exists some standard statistical scoring techniques. All these measures are used depending on problems. The most frequently used measures can be derived from the following scalar quantities which are directly available from the classifiers:

$tp_\alpha =$ number of correctly predicted α residues.
$tp_\beta =$ number of correctly predicted β residues.
$tp_{\text{coil}} =$ number of correctly predicted *coil* residues.
$tp = tp_\alpha + tp_\beta + tp_\beta =$ is correctly predicted residues in total.
$tn_\alpha =$ number of correctly classified non-α residues.
$tn_\beta =$ number of correctly classified non-β residues.
$tn_{\text{coil}} =$ number of correctly classified non-*coil* residues.
$tn = tn_\alpha + tn_\beta + tn_{\text{coil}}$ is the sum of correctly predicted non-α, non-β and non-*coil* residues.
$fp_\alpha =$ number of incorrectly classified α residues.
$fp_\beta =$ number of incorrectly classified β residues.
$fp_{\text{coil}} =$ number of incorrectly classified *coil* residues.
$fp = fp_\alpha + fp_\beta + fp_{\text{coil}}$ is the sum of incorrectly classified α, β and *coil* residues.
$fn_\alpha =$ number of incorrectly classified non-α residues.
$fn_\beta =$ number of incorrectly classified non-β residues.
$fn_{\text{coil}} =$ number of incorrectly classified non-*coil* residues.
and $fn = fn_\alpha + fn_\beta + fn_{\text{coil}}$, where fn indicates false negative alpha, beta and coil residue prediction.

The most commonly used overall performance measure is Q_{total} [8] as described in Eq. (3). It indicates the ratio of correctly predicted and incorrectly predicted residues. It shows the overall accuracy of the classifier.

$$Q_{\text{total}} = \frac{tp + tn}{tp + tn + fp + fn} \times 100. \tag{3}$$

Another way of measuring the sensitivity of prediction performance is Q_{pred} [8] as described in Eq. (4). It is the fraction of correctly predicted residues among predicted residues.

$$Q_{\text{pred}} = \frac{tp}{tp + fp} \times 100. \tag{4}$$

Selectivity of prediction performance can be measured using Q_{obs}, which is the fraction of correctly predicted residues among the observed residues [8] described in Eq. (5).

$$Q_{\text{obs}} = \frac{tp}{tp + fn} \times 100. \tag{5}$$

To retain the flavor of both sensitivity and selectivity, we can use MCC (Matthews Correlational Coefficient) score [21] given by the following equation.

$$\text{MCC} = \frac{(tp \times tn) - (fp \times fn)}{\sqrt{(tp + tn)(tp + fn)(tn + fp)(tn + fn)}}. \tag{6}$$

4. Results

Experiments were conducted for all the three data sets and three classifiers. Since results were similar for all the data sets, here we present results for only one data set (namely the RS126). Percentage of training data was

Table 1
Comparison of classifiers with varying window size

W	Minimum distance				K-NN				Fuzzy K-NN			
	Q_{total}	Q_{pred}	Q_{obs}	MCC	Q_{total}	Q_{pred}	Q_{obs}	MCC	Q_{total}	Q_{pred}	Q_{obs}	MCC
3	55.7724	55.7018	38.4511	0.1062	52.1112	49.4035	33.5447	0.0093	54.4585	53.7803	36.9258	0.0753
5	56.9716	57.0607	39.6584	0.1302	52.2133	47.9506	32.4478	−0.0056	53.1135	52.3252	35.5908	0.0499
7	57.0353	57.2061	39.7849	0.1326	51.0907	45.6113	30.3534	−0.0404	54.4074	53.2863	36.5985	0.0699
9	57.4818	57.4526	40.0969	0.1388	49.3558	41.8242	26.1312	−0.0963	54.2544	52.2831	35.9589	0.0576
11	59.2167	59.5357	41.9672	0.1750	49.8533	42.1776	26.3296	−0.0867	53.0806	49.9429	34.0686	0.0229
13	58.6682	59.1503	41.5336	0.1669	47.9753	38.7144	22.6438	−0.1469	52.4429	49.4126	33.6091	0.0121
15	58.5917	59.0554	41.4436	0.1651	46.2687	35.3711	17.7595	−0.1993	52.5458	48.0073	32.4178	−0.002
17	57.9411	58.3364	40.7572	0.1519	45.0682	33.6211	15.9912	−0.2313	51.0142	45.4923	29.5811	−0.0436

chosen randomly in the range 10–30%. Since the finding was similar, here we put results of 30% training data. In Table 1, we provide the classification performance (using the measures discussed in Section 3) for the three classifiers. The window size W is varied in the range 3, 5, ... , 17. From Table 1 we notice that window size 11 gives the best Q_{total} accuracy of 59.2167% for minimum distance classifier. The MCC score for the same is 0.1750 which is better than others. Further increment in W does not affect the accuracy significantly, sometimes it decreasing instead.

For K-NN and fuzzy K-NN methods, we varied two parameters to determine their effect in prediction performance. One of them is number of neighbors, K and the second one is corresponding window size W. Taking initial window size as three, we increased its value. As the value of W increases, K-NN and fuzzy K-NN produce relatively poor results. In this case K-NN and fuzzy K-NN performs better for small window size. Window 3 gives the best result in our experiment for both K-NN (MCC score 0.0093) and fuzzy K-NN (MCC score 0.0753).

We have also experimented with the number of neighbors, K for K-NN and fuzzy K-NN classifier. We considered a wide range of K-values and depict the result in Table 2. From Table 2, we see that gradual increase of the number of neighbors gives improvement in accuracy. But large value of K deteriorates the performance. Thus, we put results for window size three in Table 2. Fuzzy K-NN produced comparatively better results with maximum accuracy as 53.78% Q_{total} score. In this algorithm, we set the value of the fuzzifier, m to 2.4. Taking the values of m with $m = 1.1, 1.2, \dots, 2.8, 3$ did not affect the performance of the algorithm significantly.

We have compared our approaches to an existing neural network based secondary structure prediction method [22]. Implementation of this neural method has been done using the same framework and data set. We constructed a multilayer perceptron model with single hidden layer having 15 number of nodes. Output layer contains three nodes corresponding to each of the structural classes. Some considerable findings from the MLP are compared with minimum distance classifier results has been depicted in the following Table 3.

It may be noticed that minimum distance gives better results compared to MLP. The important factor is time requirement for minimum distance is negligible compared to MLP. Thus, minimum distance is comparatively performs better both in terms of time of execution and accuracy.

Table 2
Result for different K value

K	K-NN				Fuzzy K-NN			
	Q_{total}	Q_{pred}	Q_{obs}	MCC	Q_{total}	Q_{pred}	Q_{obs}	MCC
5	47.7612	45.7065	30.2978	−0.0644	52.175	51.9429	35.114	0.0377
7	49.1007	47.0597	31.4474	−0.0394	53.1063	52.3897	35.6891	0.0497
9	52.1112	49.4035	33.5447	0.0093	54.4585	53.7803	36.9258	0.0753
11	52.2133	47.9506	32.4478	−0.0056	54.4585	53.7803	36.9258	0.0753
13	52.9787	49.6797	33.8721	0.01928	54.2161	53.2982	36.5719	0.0683
15	52.5195	48.7916	33.1049	0.0060	53.5655	52.9752	36.1605	0.0595
17	52.6215	48.5855	32.9742	0.0044	54.1651	53.1481	36.4661	0.06612
19	53.5145	49.2763	33.6102	0.0190	53.8462	53.3896	36.496	0.06640
21	53.3486	48.775	33.0922	0.0135	53.8079	53.1141	36.3362	0.0632

Table 3
Result comparison with MLP and minimum distance

W	MLP				Minimum distance			
	Q_{total}	Q_{pred}	Q_{obs}	MCC	Q_{total}	Q_{pred}	Q_{obs}	MCC
11	51.78	47.3309	31.2489	−0.0182	59.2167	59.5357	41.9672	0.1750
13	52.1848	48.2814	31.9922	−0.0043	58.6682	59.1503	41.5336	0.1669
15	51.5657	47.5639	31.2566	−0.0180	58.5917	59.0554	41.4436	0.1651
17	51.9467	47.3998	31.0883	−0.0172	57.9411	58.3364	40.7572	0.1519

In *unary encoding* scheme, length of sliding window plays a very important role to have better prediction accuracy. Length of the sliding window, W implies the consideration of neighboring information. The residues situated nearer to the target amino acid has more influence in determining its structure rather than the further residues. Window size W controls this influence and so determination of the optimal window length is important. If we take the size too small, it may lose important classification information and lead to low prediction accuracy, whereas, larger value of W may suffer from incorporating unnecessary noise. From Table 1, we see that a value of W that gives better performance for a classifier, does not guarantee the same for other classifiers. That means different window size for different classifiers help to achieve optimal performance.

5. Conclusions and discussion

In this article, we have made an attempt to map the protein secondary structure prediction problem as pattern classification problem and used three different low cost pattern classification techniques for solving it. We used minimum distance, K-NN and fuzzy K-NN classifiers, among which minimum distance produced the best results for window size 11. Between the other two, fuzzy K-NN gave better performance with window size 3. We also varied the value of K, better results were achieved in the range 19–21. Experiments are conducted using different percentage of training sets and the findings were similar.

The main problem in protein secondary structure prediction is that, the data cannot be used directly to classifiers. An efficient technique is needed that can convert these categorical data to acceptable numerical forms. The techniques available now are not too effective to perform this job. As described in Section 2.6, the amino acid in the central position of a window has the maximum effect to retain its secondary structure along with its corresponding neighbors. Present encoding scheme fails to impose any weight factor to the central residue that can reflect its strength compared to its neighbors. An unique encoding scheme that can provide the positional information of each residue as well as its corresponding weight may be helpful to make a good representation of protein data.

Another problem for this prediction problem is the selection of ideal training data. In classification of protein secondary structure, it is very hard to find out good representatives of a class. In future we will do some studies on this.

All the results provided in Tables 1 and 2 are the direct outcomes of the classifiers. No post-processing technique is used to improve this performance. Some smoothing filters may be used here to remove certain breaker residues into different classes.

Finally, we observe in this study that some patterns in a particular protein are classified correctly by a classifier better than the other classifiers. Keeping this in mind, our further aim is to build ensemble of classifiers to predict the class label of residues using the classifier that favors it.

References

[1] Sujun Hua, Zhirong Sun, A novel method of protein secondary structure prediction with high segment overlap measure: support vector machine approach, Journal of Molecular Biology 308 (2001) 397–407.

[2] Rajkumar Bondugula, Ognen Duzlevski, Dong Xu, Profiles and fuzzy k nearest neighbor algorithm for protein secondary structure prediction, in: Proceedings of APBC 2005, The Third Asia-Pacific Bioinformatics Conference, Singapore, January, 17–21, 2005.

[3] A.G. Szent-Gyorgyi, C. Cohen, Role of proline in polypeptide chain configuration of proteins, Science 126 (1957) 697–698.

[4] Burkhard Rost, Review: Protein secondary structure prediction continues to rise, Journal of Structural Biology 134 (2001) 204–218.

[5] D. Przybylski, Burkhard Rost, Alignments grow, secondary structure prediction improves, Proteins 46 (2002) 197–205.

[6] Gavin E. Crooks, Steven E. Brenner, Protein secondary structure: entropy, correlations and prediction, Bioinformatics 20 (2004) 1603–1611.

[7] Ning Qian, Terrence J. Sejnowski, Predicting the secondary structure of globular proteins using neural network models, Journal of Molecular Biology 202 (1988) 865–884.

[8] Burkhard Rost, Chris Sander, Prediction of protein secondary structure at better than 70% accuracy, Journal of Molecular Biology 232 (1993) 559–584.

[9] Kuang Lin, Victor A. Simossis, William R. Taylor, Jaap Heringa, A simple and fast secondary structure prediction method using hidden neural networks, Bioinformatics 21 (2) (2005) 152–159.

[10] G. Pollastri, D. Przybylski, Burkhard Rost, P. Baldi, Improving the prediction of protein secondary structure in three and eight classes using recurrent neural networks and profiles, Proteins 47 (2002) 228–235.

[11] J.A. Cuff, G.J. Barton, Application of multiple sequence alignment profiles to improve protein secondary structure prediction, Proteins 40 (2000) 502–511.

[12] David. T. Jones, Protein secondary structure prediction based on position-specific scoring matrices, Journal of Molecular Biology 292 (1999) 195–202.

[13] K. Karplus, C. Barrett, R. Hughey, Hidden markov models for detecting remote protein homologies, Bioinformatics 14 (1998) 846–856.

[14] Vladimir Vapnik, The Nature of Statistical Learning Theory, Springer-Verlang, New York, 1995.

[15] Saejoon Kim, Protein β-turn prediction using nearest-neighbor method, Bioinformatics 20 (2004) 40–44.

[16] J.M. Keller, M.R. Gray, J.A. Givens Jr., A fuzzy *k*-nearest neighbor algorithm, IEEE Transaction on SMC SMC-15 (4) (1985) 580–585.

[17] J. Garnier, D.J. Osguthorpe, B. Robson, Analysis of the accuracy and implications of simple methods for predicting the secondary structure of globular proteins, Journal of Molecular Biology 120 (1978) 97–120.

[18] Richard O. Duda, Peter E. Hart, David G. Stork, Pattern Classification, John Wiley, New York.

[19] J.A. Cuff, G.J. Barton, Evaluation and improvement of multiple sequence methods for protein secondary structure prediction, Proteins: Structure Function and Genetics 34 (1999) 508–519.

[20] S.K. Riis, A. Krogh, Improving prediction of protein secondary structure using structured neural networks and multiple sequence alignments, Journal of Computational Biology 292 (3) (1996) 163–183.

[21] B.W. Matthews, Comparison of the predicted and observed secondary structure of t4 phage lysozyme, Biochimica et Biophysica Acta 405 (1975) 442–451.

[22] John-Marc Chandonia, Martin Karplusk, Neural networks for secondary structure and structural class predictions, Protein Science 4 (1995) 275–285.