Fuzzy k-Nearest Neighbor Method for Protein Secondary Structure Prediction and Its Parallel Implementation

Seung-Yeon Kim[1], Jaehyun Sim[2], and Julian Lee[3]

[1] Computer Aided Molecular Design Research Center, Soongsil University, Seoul 156-743, Korea
sykim@ssu.ac.kr
[2] School of Dentistry, Seoul National University, Seoul 110-749, Korea
jhsim@snu.ac.kr
[3] Department of Bioinformatics and Life Science, Soongsil University, Seoul 156-743, Korea
jul@ssu.ac.kr
http://bioinfo.ssu.ac.kr/~jul/jul_eng.htm

Abstract. Fuzzy k-nearest neighbor method is a generalization of nearest neighbor method, the simplest algorithm for pattern classification. One of the important areas for application of the pattern classification is the protein secondary structure prediction, an important topic in the field of bioinformatics. In this work, we develop a parallel algorithm for protein secondary structure prediction, based on the fuzzy k-nearest neighbor method, that uses evolutionary profile obtained from PSI-BLAST (Position Specific Iterative Basic Local Sequence Alignment Tool) as the feature vectors.

1 Introduction

Although the prediction of the three-dimensional structure of a protein from its amino acid sequence is one of the most important problems in bioinformatics [1,2,3,4], *ab initio* prediction of the tertiary structures based solely on sequence information has not been successful so far. For this reason, lots of research efforts have been made for the determination of the protein secondary structure [5,6,7,8,9,10,11,12,13,14,15,16], which can serve as an intermediate step toward determining its tertiary structure.

The most common definition of the secondary structure is based on Dictionary of Secondary Structure of Proteins (DSSP) [17] where the secondary structure is classified as eight states. By grouping these eights states into three classes Coil (C), Helix (H), and Extended (E), one obtains three state classification, which is more widely used. Therefore, the protein secondary structure prediction is a typical pattern classification problem, where one of the three possible states is assigned to each residue of the query protein.

D.-S. Huang, K. Li, and G.W. Irwin (Eds.): ICIC 2006, LNBI 4115, pp. 444–453, 2006.

The first step for solving such a problem is the feature extraction, where the important features of the data are extracted and expressed as a set of numbers, called feature vectors. The performance of the pattern classifier depends crucially on the judicious choice of the feature vectors. It has been shown that constructing feature vectors from the evolutionary profile obtained from PSI-BLAST (Position Specific Iterative Basic Local Alignment Search Tool) [18], a bioinformatics tool for the search of homologous protein sequences, gives better prediction results than other choices [6,16] (see Sect. 2.1).

Once an appropriate feature vector has been chosen, a classification algorithm is used to partition the feature space into disjoint regions with decision boundaries. The decision boundaries are determined using feature vectors of a reference sample with known classes, which are also called the reference dataset or training set. The class of a query data is then assigned depending on the region it belongs to. Various pattern classification algorithms such as artificial neural network or support vector machine have been used for the protein secondary structure prediction.

The k-nearest neighbor method is the simplest algorithm for the pattern classification. Moreover, it can be easily adapted for parallel computation. Although the k-nearest neighbor method has been used for the secondary structure prediction [11,12,14,15], the fuzzy variant of the algorithm [19] has never been used for the secondary structure prediction, although it has been used for the solvent accessibility prediction [20].

In this work, we develop a parallel algorithm for the protein secondary structure prediction, based on the fuzzy k-nearest neighbor method [19], where PSI-BLAST profiles are used as the feature vectors. As a test of our algorithm, we perform a benchmark test on EVA common set 1 consisting of 60 proteins [22].

2 Methods

2.1 The Feature Vectors

In order to construct the feature vector for a protein residue, we first perform database search with PSI-BLAST [18]. PSI-BLAST then calculates the rate of substitution of each residue of the query protein to another amino acids. By multiplying appropriate normalization factors, taking logarithms, and rounding off to integer values, these numbers are converted to what is called the position specific scoring matrix, also called profile, a matrix of the size (protein length) $\times$ 20. This PSI-BLAST profile contains evolutionary information that cannot be obtained from the raw sequence only. For a protein residue whose secondary structure is to be predicted, one takes a window of size N_w centered around this residue, and uses the matrix of size $N_w \times 20$ as the feature vector to be input into the pattern classification algorithm (see Fig. 1). We use $N_w = 15$ in this work. The resulting feature vector is a $15 \times 20 = 300$ dimensional matrix. This feature vector is the same as the one used in previous works [6,16] based on other pattern classification methods.

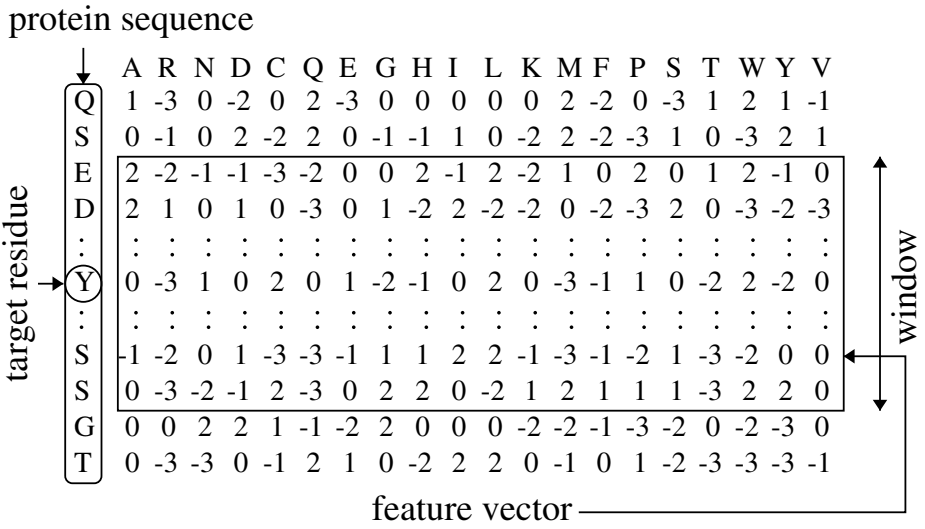

Fig. 1. The relation between PSI-BLAST profile and the feature vector of a residue. The feature vector corresponding to a target residue is constructed from the PSI-BLAST profile by considering a window of finite size (15 residues in this work) centered on the residue.

2.2 The Distance Measure

There are various ways of defining the distance between two feature vectors A and B, but in this work we use three methods, Euclidean, Cosine, and Correlation distances, defined as

$$D_{AB}(Euc) = \sum_{i=1}^{N_w} w_i \sum_j (P_{ij}(A) - P_{ij}(B))^2, \tag{1}$$

$$D_{AB}(Cos) = 1 - \sum_{i=1}^{N_w} w_i \frac{\sum_j P_{ij}(A) \cdot P_{ij}(B)}{\sqrt{\sum_p P_{ip}(A)^2 \sum_q P_{iq}(B)^2}}, \tag{2}$$

$$D_{AB}(Corr) = 1 - \sum_{i=1}^{N_w} w_i \frac{\sum_j (P_{ij}(A) - \bar{P}_i(A)) \cdot (P_{ij}(B) - \bar{P}_i(B))}{\sqrt{\sum_p (P_{ip}(A) - \bar{P}_i(A))^2 \sum_q (P_{iq}(B) - \bar{P}_i(B))^2}}, \tag{3}$$

respectively, where $P_{ij}(A)(i = 1, 2, \cdots, 15; j = 1, 2, \cdots, 20)$ is a component of the feature vector A, w_i a weight parameter, and

$$\bar{P}_i(A) \equiv \frac{1}{20} \sum_{j=1}^{20} P_{ij}(A).$$

Since we expect the profile elements for residues nearer to the target residue to be more important in determining the local environment of the target residue, we use weights $w_i = (8 - |8 - i|)^2$.

2.3 The Reference Dataset

In order to construct the reference dataset consists of representative protein chains without bias, we utilize the ASTRAL SCOP database, where the protein chains are hierarchically classified into structural families, and representative proteins are selected for each of them. In particular, we used ASTRAL SCOP (version 1.63) chain-select-95 subset and chain-select-90 subset [21]. We then clustered these sequences with BLASTCLUST (NCBI BLAST 2.2.5, http://www.ncbi.nlm.nih.gov/BLAST/) and selected the representative chain for each cluster, in order to remove additional homologies. The resulting reference dataset consists of 4362 non-redundant proteins (905684 feature vectors) that have less than 25% sequence identity with each other.

2.4 Fuzzy k-Nearest Neighbor Method

In the simplest version of the fuzzy k-nearest neighbor (FKNN) method [19], the fuzzy class membership $u_s(\mathbf{x})$ to the class s is assigned to the query data $\mathbf{x}$ according to the following equation:

$$u_s(\mathbf{x}) = \frac{\sum_{sec(j)=s} D_j^{-2/(m-1)}}{\sum_{j=1}^{k} D_j^{-2/(m-1)}} \tag{4}$$

where the summation of j in the numerator is restricted to those belonging to the class s, m is a fuzzy strength parameter, which determines how heavily the distance is weighted when calculating each neighbor's contribution to the membership value, k is the number of nearest neighbors, and c is the number of classes, and D_j is the distance between the feature vector of the query data $\mathbf{x}$ and the feature vector of its j-th nearest reference data $\mathbf{x}(j)$.

The advantage of the fuzzy k-nearest neighbor algorithm over the standard k-nearest neighbor method is quite clear. The fuzzy class membership $u_s(x)$ can be considered as the estimate of the probability that the query data belongs to class i, and provides us with more information than a definite prediction of the class for the query data. Moreover, the reference samples which are closer to the query data are given more weights, and an optimal value of m can be chosen along with that for k, in contrast to the standard k-nearest neighbor method with fixed value of $2/(m-1) = 0$. In fact, the optimal value of k and m are found from the leave-one-out cross-validation procedure (see below), and the resulting value for $2/(m-1)$ is indeed nonzero.

The optimal values of k and m were determined by leave-one-out cross validation test, where the prediction was performed for one of the chains in the reference dataset, using the remaining 4361 chains as the reference dataset, procedure being repeated for each of the 4362 chains. The optimal values of k and

m are determined as the ones yielding the maximum average value of Q_3 score, which is define as:

$$Q_3 \equiv 100\% \times \frac{N_{corr}}{N} \tag{5}$$

with N and N_{corr} being the total number of residues of the query protein, and the total number of correctly predicted residues, respectively.

The optimal value of m turns out to be 1.29, and that of k is 85 when using Euclidean and Correlation distances, and 70 when using Cosine.

2.5 The Parallel Implementation

The FKNN method can be easily adapted for parallel computation. In the parallel implementation, the computational load is shared between computational nodes, resulting in drastic increase in computational speed. The advantage of the parallel program in terms of computational time can also be seen from Fig. 2 (see Results and Discussions). To elaborate on the parallel algorithm, each of the nodes is assigned a distinct subset of the feature vectors in the reference dataset, and each member of this set is compared with query vector, and k_{nn} of them with the smallest distance from the query vector are chosen. The numbers of the feature vectors assigned to the nodes are all equal up to roundoff error, so that the loads are balanced. The 0-th node, which we call the master, performs the job of collecting k_{nn} candidates of nearest neighbors from each of the nodes. It then sorts these $N_{nodes} \times k_{nn}$ indices with respect to the distance D to select the final k_{nn} nearest neighbors. The master then produces the final output. The pseudo-code for the parallel algorithm is given in algorithm 1., along with the sub-algorithms 2., 3., and 4..

Algorithm 1. parallel FKNN algorithm for the protein secondary structure prediction

1: k_{nn} = Number of nearest neighbors (constant)
2: N_{nodes} = Total number of computing nodes
3: $Rank$ = The number of this node, a number between 0 and $N_{nodes} - 1$
4: Construct the feature vector for each residue of the query protein {algorithm2.}
5: Construct the feature vector for each residue in the database {algorithm3.}
6: $st = Rank * N_f/N_{nodes} + 1$
7: $ed = (Rank + 1) * N_f/N_{nodes}$ {st and ed is the starting and ending number of feature vectors the current node will look into. This is to divide computational load between nodes.}
8: **for** $j_q = 1$ to L_q **do**
9: Calculate the probabilities $prob(j_q, s)$ of the residue q being in each of the conformational state s (=C,H,E) {algorithm4.}
10: The predicted secondary structure $S(j_q) = (s$ that maximizes prob(j_q, s))
11: print out j_q, $S(j_q)$, and $prob(j_q, s)$ (s = C,H,E)
12: **end for**

Algorithm 2. Constructing feature vectors for each residue of the query protein

Read in query profile
L_q = Length of the query sequence
for $j_q = 1$ to L_q **do**
  Construct matrix $\mathbf{P}_q(j_q)$ of size 15×20, centered around the residue j_q, from the query profile
end for

Algorithm 3. Constructing feature vectors for each residue in the database

$N_f = 0$ {N_f will be the total number of feature vectors in the reference dataset, equal to the total number of residues in the dataset}
N_p = Number of protein chains in the reference dataset
Read in profiles in the reference dataset (database profiles)
for $i = 1$ to N_p **do**
  $L(i)$ = Length of the i-th protein chain
  for $j = 1$ to $L(i)$ **do**
    $N_f \Leftarrow N_f + 1$
    Construct matrix $\mathbf{P}_{DB}(N_f)$ of size 15×20, centered around the residue j of the i-th protein, from the database profile
  end for
end for

3 Results and Discussions

The benchmark test was performed on EVA common set 1 consisting of 60 proteins [22] and RS126 set consisting of 126 non-homologous protein [5], with the optimal values of m and k determined by the leave-one-out cross-validation on the reference dataset derived from ASTRAL SCOP (see Methods). The performance on EVA common set 1 was compared with three neural network based prediction methods, PSIPRED (v2.3) [6], PROFking (v1.0) [7], and SABLE (v2.0) [8], and the performance on RS126 set was compared with two methods based on support Vector Machine (SVM), SVM freq [9] and SVMpsi [10].

In addition to Q_3 score (see section 2.4), two additional performance scores, SOV score [23] and three state correlation coefficient (Corr(3)) [24], are used for the assessment of performance. The average values and the standard errors of these scores for the performance on EVA common set 1, of the fuzzy k-nearest method with various distance measures, and the other three methods, are displayed in Table 1. The results of the test on RS126 set are shown on Table 2.

We see that in both of these test, the performance is best when the Correlation distance measure is used. We see that in the first test, average performance scores are lower than those of PSIPRED and SABLE, but higher than PROFking. However, considering the magnitudes of the standard error, these differences are not drastic, and we may say that the performances are more or less comparable to other methods. Also, the actual performances of the prediction algorithms depend on their versions and the set of proteins used for the test, and it should

Algorithm 4. Calculating the fuzzy membership of a query residue to each of the secondary structural class

for s = C,H,E **do**
  membership$(j_q, s) = 0$
end for
for $m_{DB} = st$ to ed **do**
  $D(j_q, m_{DB})$ = Distance between $\mathbf{P}_q(j_q)$ and $\mathbf{P}_{DB}(m_{DB})$
end for
Sort indices of the feature vectors the current node is examining, with respect to $D(j_q, m_{DB})$, in descending order.
if *Rank* == 0 **then** {This node is the master, so collect the results and re-sorts them, and print the final output}
  indx() $\Leftarrow$ save indices of k_{nn} nearest neighbors among the feature vectors examined by the master
  dscore() $\Leftarrow$ save distances of k_{nn} nearest neighbors among the feature vectors examined by the master
  for $i = 1$ to $N_{nodes} - 1$ **do**
    Receive indices and distances of k_{nn} nearest neighbors among the feature vectors examined by the i-th node
    indx() $\Leftarrow$ add indices of k_{nn} nearest neighbors among the feature vectors examined by the i-th node
    dscore() $\Leftarrow$ add distances of k_{nn} nearest neighbors among the feature vectors examined by the i-th node
  end for
else
  Send indices and distances of k_{nn} nearest neighbors among the feature vectors examined by the i-th node to the master
end if
if *Rank* == 0 **then**
  Sort indices with respect to **dscore**() {The collection consists of $N_{nodes} \times k_{nn}$ results, so master must sort them again to select k_{nn} nearest neighbors}
  for $j_{DB} = 1, k_{nn}$ **do** {Calculate the fuzzy membership from k_{nn} nearest neighbors}
    $s(j_{DB})$ = secondary structural class corresponding to the j_{DB}-th feature vector
    membership$(j_q, s(j_{DB})) \Leftarrow$ membership$(j_q, s(j_{DB}))$+ fuzzy membership calculated from $D(j_q, j_{DB})$
  end for
  for s = C,H,E **do**
    prob(j_q, s) = membership$(j_q, s) / \sum_{s' \in \{C,H,E\}}$ membership(j_q, s')
  end for
end if

be emphasized that the result is not to be considered as an extensive test of these methods.

Since the programs based on SVM are not available for public use, we quote the values from the literature [9,10]. The values of performance measures not

Table 1. Average scores of secondary structure prediction on EVA common set 1, using fuzzy k-nearest neighbor (FKNN) method with Euclidean (Euclid), Cosine (Cos), and Correlation (Corr) distance measures. The average scores are given also for three other methods for comparison. The numbers in the parentheses are the standard errors.

	Q3	SOV	Corr(3)
FKNN(Euclid)	70.9 (1.8)	64.5 (2.3)	0.495 (0.024)
FKNN(Cos)	70.9 (1.8)	64.5 (2.3)	0.499 (0.034)
FKNN(Corr)	71.8 (1.9)	67.9 (2.4)	0.527 (0.026)
PSIPRED	75.1 (1.8)	75.3 (2.4)	0.557 (0.024)
PROFking	67.2 (2.3)	64.3 (2.8)	0.463 (0.029)
SABLE	75.6 (1.5)	73.1 (2.5)	0.532 (0.029)

Table 2. Average scores of secondary structure prediction on RS126 set, using fuzzy k-nearest neighbor (FKNN) method with Euclidean (Euclid), Cosine (Cos), and Correlation (Corr) distance measures. The average scores are given also for two other methods based on SVM, for comparison.

	Q3	SOV	Corr(3)
FKNN(Euclid)	88.6	83.1	0.791
FKNN(Cos)	88.6	83.1	0.744
FKNN(Corr)	89.0	84.0	0.796
SVMfreq	75.1	-	-
SVMpsi	76.1	72.0	-

reported in the references are omitted. We see that the fuzzy k-nearest neighbor method also shows good performance when compared with SVM-based methods.

The parallel code was implemented in mpi C, and run on 32 Intel Xeon processors. For 60 proteins in the EVA set, for the Euclidean, cosine, and correlation distance measures, respectively, the calculation took 47, 58, and 60 minutes of wall clock time, defined as the time elapsed between the start and end of the program.

The advantage of the parallel algorithm we introduced in this work is that the communication between computational nodes are kept to a minimal level. In fact, the most of the computations are performed by each of the nodes independently, and the communication occurs only at the end of such computations, and only between the king and slaves, when the master collects the results from the slaves and sorts them again to predict the secondary structure. In order to examine the parallel efficiency, we repeated the computation for EVA common set 1 using the correlation distance measure for different number of CPUs in order to obtain the response curve in Fig. 2. In the figure, the inverse of the time is plotted against the number of CPUs involved in the computation, in order to show the dependence of the computational speed on the number of CPUs. The result shows that, although the dependence is not exactly linear, the scalability is reasonably good, demonstrating the advantage of parallel computation over serial version.

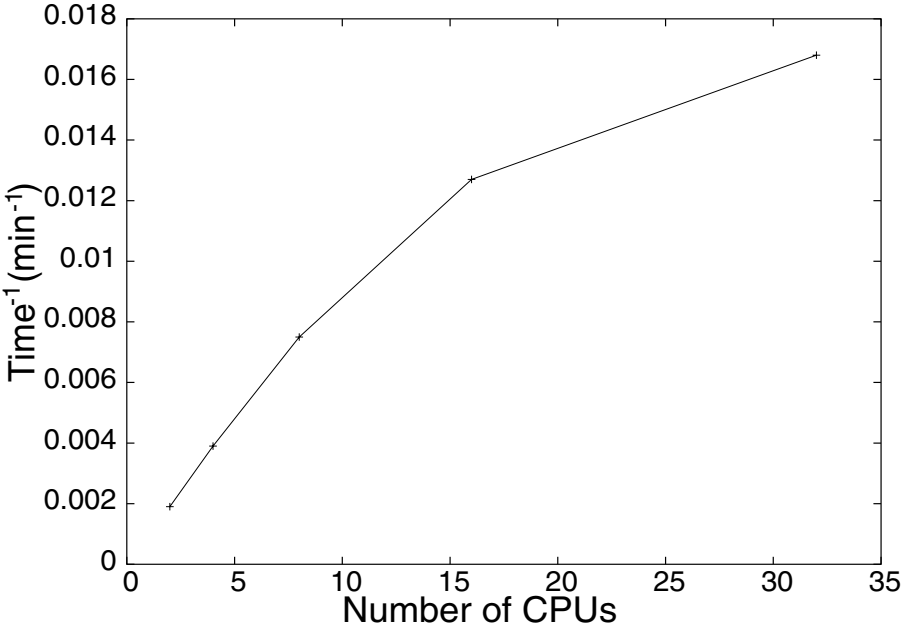

Fig. 2. The inverse of wall time in $\min^{-1}$ (*vertical axis*) plotted against the number of CPUs used for the computation (*horizontal axis*). The curve shows excellent scalability of the parallel FKNN algorithm, due to minimal amount of communication between CPUs.

Acknowledgement

This work was supported by the Korean Research Foundation Grant funded by the Korean Government (MOEHRD) (KRF-2005-005-J01101).

References

1. Kryshtafovych, A., Venclovas, C., Fidelis, K., Moult, J.: Progress over the First Decade of CASP Experiments. Proteins. vo. 61 (2005) 225–236
2. Lee, J., Kim, S.-Y., Joo, K., Kim, I., Lee, J.: Prediction of Protein Tertiary Structure using PROFESY, a Novel Method Based on Fragment Assembly and Conformational Space Annealing. Proteins. vol. 56 (2004) 704–714
3. Lee, J., Kim, S.-Y., Lee, J.: Protein Structure Prediction Based on Fragment Assembly and Parameter Optimization. Biophys. Chem. vol. 115 (2005) 209–214
4. Lee, J., Kim, S.-Y., Lee, J.: Protein Structure Prediction Based on Fragment Assembly and Beta-strand Pairing Energy Function. J. Korean Phys. Soc. vol. 46 (2005) 707–712
5. Rost, B., Sander, C.: Prediction of Secondary Structure at Better than 70% Accuracy. J. Mol. Biol. vol. 232 (1993) 584–599
6. Jones, D.: Protein Secondary Structure Prediction Based on Position-specific Scoring Matrices. J. Mol. Biol. vol. 292 (1999) 195–202

7. Ouali, M., King, R.: Cascaded Multiple Classifiers for Secondary Structure Prediction. Protein Science. vol. 9 (1999) 1162–1176
8. Adamczak, R., Porollo, A., Meller, J.: Combining Prediction of Secondary Structure and Solvent accessibility in proteins. Proteins. vol. 59 (2005) 467–475
9. Hua, S., Sun, Z.: A Novel Method of Protein Secondary Structure Prediction with High Segment Overlap Measure: Support Vector Machine Approach. J. Mol. Biol. vol. 308 (2001) 397–407
10. Kim, K., Park, H.: Protein Secondary Structure Prediction based on improved Support Vector Machines Approach. Protein Eng. vol. 16 (2003) 553–560
11. Joo, K., Lee, J., Kim, S.-Y., I., Kim, Lee, S.J., Lee, J.: Profile-based Nearest Neighbor Method for Pattern Recognition. J. Korean Phys. Soc. vol. 44 (2004) 599–604
12. Joo, K., Kim, I., Lee, J., Kim, S.-Y., Lee, S.J., Lee, J.: Prediction of the Secondary Structure of Proteins Using PREDICT, a Nearest Neighbor Method on Pattern Space. J. Korean Phys. Soc. vol. 45 (2004) 1441–1449
13. Pollastri, G., McLysaght, A. Porter: a new, Accurate Server for Protein Secondary Structure Prediction. Bioinformatics vol. 21 (2004) 1719–1720
14. Jiang, F.: Prediction of Protein Secondary Structure with a Reliability Score Estimated by Local Sequence Clustering. Protein Eng. vol. 16 (2003) 651–657
15. Salamov A. A., Solovyev V. V.: Protein Secondary Structure Prediction Using Local Alignments. J. Mol. Biol. vol. 268 (1997) 31–35
16. Kim, H., Park, H.: Prediction of Protein Relative Solvent Accessibility with Support Vector Machines and Long-range Interaction 3D Local Descriptor. Proteins. vol. 54 (2004) 557–562
17. Kabsch, W., Sander, C.: Dictionary of Protein Secondary Structure: Pattern Recognition of Hydrogen-bonded and Geometrical Features. Biopolymers vol. 22 (1983) 2577–2637
18. Altschul, S.F., Madden, T.L., Schaffer, A.A., Zhang, J., Zhang, Z., Miller, W., Lipman, D.J.: Gapped BLAST and PSI-BLAST: a New Generation of Protein Database Search Programs. Nucleic Acids Res. vol. 25 (1997) 3389–3402
19. Keller, J. M., Gray, R., Givens, J. A. : A Fuzzy k-nearest Neighbor Algorithm. IEEE Trans. Systems Man Cybernet. vol. 15 (1985) 580–585.
20. Sim, J. H., Kim, S.-Y., Lee, J.: Prediction of Protein Solvent Accessibility Using Fuzzy k-Nearest Neighbor Method. Bioinformatics vol. 21 (2005) 2844–2849.
21. Brenner, S.E., Koehl, P., Levitt, M.: The ASTRAL Compendium for Protein Structure and Sequence Analysis. Nucleic Acids Res. vol. 28 (2000) 254–256
22. Koh, I. Y., Eyrich, V., Marti-Renom, M. A., Przybylski, D., Madhusudhan, M. S., Eswar, N., Grana, O., Pazos, F., Valencia, A., Sali, A., Rost, B.: EVA: Evaluation of Protein Structure Prediction Servers. Nucleic Acids Res. vol. 31 (2003) 3311–3315
23. Zemla, A., Venclovas, C., Fidelis, K., Rost, B.: A Modified Definition of Sov, a Segment-Based Measurement for Protein Secondary Structure Prediction Assessment. Proteins. vol. 34 (1999) 220–223
24. Gorodkin, J.: Comparing two K-category Assignment by a K-category Correlation Coefficient. Comput. Biol. and Chem. vol. 28 (2004) 367–374