PROFILES AND FUZZY K-NEAREST NEIGHBOR ALGORITHM FOR PROTEIN SECONDARY STRUCTURE PREDICTION

RAJKUMAR BONDUGULA, OGNEN DUZLEVSKI, AND DONG XU*

Digital Biology Laboratory, Department of Computer Science, University of Missouri-Columbia Columbia, MO 65211, USA

We introduce a new approach for predicting the secondary structure of proteins using profiles and the Fuzzy *K*-Nearest Neighbor algorithm. *K*-Nearest Neighbor methods give relatively better performance than Neural Networks or Hidden Markov models when the query protein has few homologs in the sequence database to build sequence profile. Although the traditional *K*-Nearest Neighbor algorithms are a good choice for this situation, one of the difficulties in utilizing these techniques is that all the labeled samples are given equal importance while deciding the secondary structure class of the protein residue and once a class has been assigned to a residue, there is no indication of its confidence in a particular class. In this paper, we propose a system based on the Fuzzy *K*-Nearest Neighbor Algorithm that addresses the above-mentioned issues and the system outperforms earlier *K*-Nearest neighbor methods that use multiple sequence alignments. We also introduce a new distance measure to calculate the distance between two protein sequences, a new method to assign membership values to the Nearest Neighbors in each of the *Helix*, *Strand* and *Coil* classes. We also propose a novel heuristic based filter to smoothen the prediction. Particularly attractive feature of our filter is that it does not require retraining when new structures are added to the database. We have achieved a sustained three-state overall accuracy of 75.75% with our system. The software is available upon request.

1 Introduction

The ability to predict the secondary structure of a protein from sequence alone is an important step in understanding the three dimensional structure of a protein and the function of a protein. Owing to the importance of protein secondary structure prediction, much attention has been given to this problem [4, 6-12, 14,16]. Of all the successful prediction methods, the most popular systems are based on Neural Network methods [16], Nearest Neighbor methods [7,10] and Hidden Markov Model methods [14]. Currently, the systems based on Neural Network methods are one of the most accurate of all prediction systems [16]. However, Neural Network methods have some drawbacks. Firstly, the black-box nature of Neural Networks makes it difficult to understand how the networks predict the structure. Secondly, the systems based on Neural Network methods and the Hidden Markov Models perform well if the query protein has many homologs in the database [6-7]. On the other hand, the prediction systems based on Nearest Neighbor methods do not suffer from any of the above-mentioned drawbacks [10]. Also, the Nearest Neighbors methods are sub-optimal methods and the 1-NN rule is bounded above by no more than twice the optimal Bayes error rate [3]. Albeit these advantages, conventional *K*-Nearest Neighbor algorithms have some drawbacks. Firstly, while

* Corresponding author. Dong Xu can be contacted at dong@cecs.missouri.edu.

assigning class membership values (i.e., the weights that represent the likelihood of different secondary structure types), atypical vectors and true representatives of the classes are given equal importance. Secondly, once the class has been assigned to a vector, there is no indication of the strength (significance) of membership to indicate how much the vector belongs to a particular class.

In this paper we propose a prediction system that is based on a generalized Nearest Neighbor method, the Fuzzy *K*-Nearest Neighbor (Fuzzy *K*-NN) method [3]. This method while retaining all the advantages of the (Crisp-) Nearest Neighbor methods, addresses all of its drawbacks. We use position specific scoring matrices (PSSMs) [13] of the query protein sequence as input to the prediction system. We also introduce a new distance measure to calculate the distance between two residues, a new method to assign class-membership values to the Nearest Neighbors and a novel heuristic based filter to smoothen the prediction. Particularly attractive feature of our filter is that it does not require retraining when new structures are added to the database. The mean Q_3 accuracy of our system on the widely adopted Rost and Sander benchmark [9], which contains 126 proteins with less then 25% sequence identity between each other, is 73.53%. The accuracy on the same 126 proteins is 75.75% when a larger custom database with 1372 proteins (also with less then 25% sequence identity between each other) is used for searching the homologous segments. This method is an integral part of a larger project under development to predict the three-dimensional structure of a protein based on the concept of mini-threading [15], as the outputs of our secondary structure prediction provide the structure segments and secondary structure variations to be used in mini-threading.

2 Methods and Materials

First the PSSM profiles of both the query protein and the proteins in the database are calculated with the PSI-BLAST [13]. In the database, the DSSP standard [2] of eight secondary structures are reduced to the CASP (*http://predictioncenter.llnl.gov*) standard of three-state secondary structures as follows: $\{H, G, I\} \rightarrow H$, $\{E, B\} \rightarrow E$, and $\{C, T, S\} \rightarrow C$. The approach in *K*-Nearest Neighbor algorithm is to predict the secondary structure state of the central residue of a sliding window of size W (usually an odd number), based on the secondary structure states of the homologous segments from the database proteins (proteins with known three-dimensional structures) [17]. In order to find the homologous segments, a distance measure 'd' has to be defined. We introduce a simple 'position-weighted absolute distance measure' that can be defined as follows:

$$d = \max\left\{1, \left(\sum_{i=1}^{20} \sum_{j=1}^{W} \left| \left(p_{ij}^{Query} - p_{ij}^{Database} \right) \min\left\{ j, \left(W + 1 - j \right) \right\} \right| \right) \right\} \qquad (1)$$

Where, W is the window size, p_{ij} is the profile score of the j^{th} position in the window corresponding to the i^{th} amino acid. The symbol 'max' represents the maximum while the symbol 'min' represents the minimum. The expression is designed such that the position in the center of the window gets the maximum weight and the importance decreases as we proceed towards the edges to reflect the fact that the farther an amino acid is from the central residue, the less influential it is on the secondary structure of the central residue. The weighing function used with a profile corresponding to window of size 7 is illustrated in Figure 1.

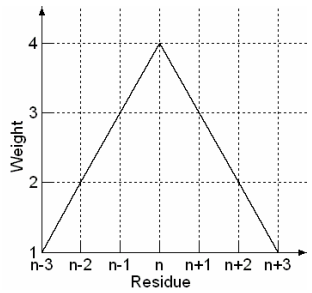


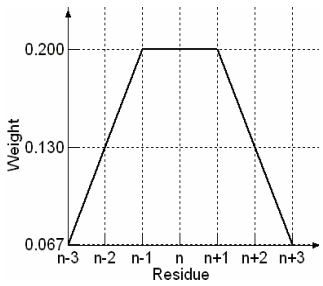
Figure 1. The profile-weighing function. The function is designed such that the central residue gets the maximum weight while the weight decreases as we proceed towards the edges.

The confidence with which a residue belongs to particular class is represented by a membership value that lies in [0,1]. Membership value of '1' in a particular class indicates that the residue belongs to that class and value of '0' indicates that the residue does not belong to that class. In order to predict the secondary structure of the n^{th} residue, the distances between profile corresponding to the window centered on the n^{th} residue and the profiles corresponding to the windows of the proteins in the database are calculated. For each residue in the query sequence, the K nearest windows and their corresponding profiles are retained. The membership value of current residue in each of the *Helix*, *Strand* and *Coil* classes is calculated from the membership values of the retained K-Nearest Neighbors. In the following section, we describe the procedure to assign the membership values to the retained neighbors and in section 2.2 we explain the procedure to calculate the membership values of the current residue from the membership values of the neighbors.

2.1 Membership assignment to the neighbors

There are many ways to assign the membership values to the neighbors. The simplest method is to assign the neighbor to the class to which the central residue belongs. For example, if the secondary structure of the central residue is in *Helix* then the neighbor will have a membership of '1' in *Helix* class and '0' in other classes. This scheme was

used in the earlier work that used *K*-Nearest Neighbor algorithm [10]. Another method is to assign membership value in a particular class based on the percentage of neighbors in that class weighted by the inverse of the distance from the current residue [6]. The first method ignores the fact that secondary structure of central residue is also influenced by the secondary structure state of its neighboring residues. We introduce a new membership assignment method that will take into consideration the fact that the neighboring residues play an influential role in secondary structure state of the central residue (but not as influential as the central residue itself). Also, our method guarantees that neighbors that lie on intersection of two secondary structures (for example, a *Helix* and a *Loop*) do not have full membership values in either class so that atypical neighbors do not contribute much to the classification of the residues while true representatives contribute relatively more. An example of the membership weighing function is illustrated in Figure 2 (a). The procedure to calculate the membership values for a typical neighbor when $W = 7$ is illustrated in Figure 2 (b).



(a)

AA SS	G H	A H	G C	A C	N C	P E	R E
PW	0.067	0.13	0.2	0.2	0.2	0.13	0.067
H	1	1					
E						1	1
C			1	1	1		

AA: Amino acid **SS**: Secondary Structure
PW: Position Weight

H = 0.067x1 + 0.133x1 = 0.20
E = 0.133x1 + 0.067x1 = 0.20
C = 0.200x1 + 0.200x1 +0.200x1 = 0.60

(b)

Figure 2. Calculation of membership value. (a) The weighing function used in assigning the membership values to the neighbors for a window size of 7. The secondary structures of residues in the center get more weight than the ones in the edges. (b) Procedure to illustrate the calculation of the membership values for a typical neighbor (top row) from its corresponding secondary structure (second row) in each of the *Helix*, *Strand* and *Coil* classes. The membership value in each class is proportional to number of residues in each class weighted by value of its position (third row) in the window.

2.2 The Fuzzy K-Nearest neighbor algorithm

The secondary structure state of the center residue in the window can be predicted from membership values of the neighbors in each class with the Fuzzy *K*-Nearest Neighbor algorithm. The following algorithm adopted and modified from [3] represents the procedure to calculate the membership values of the current residue from the neighbors is presented. Let $P = \{r_1, r_2, r_3, \ldots, r_l\}$ represent a protein with l residues. Each residue r has *K*-Nearest neighbors that are found using the window centered on r. Also, let u_{ij} be the

membership in the i^{th} class ($i \in \{Helix, Strand, Coil\}$) of the j^{th} neighbor. For each r, the predicted membership value u_i in class i can be calculated as follows:

BEGIN

Initialize $i = 1$.

DO UNTIL (r assigned membership in all classes)

Compute $u_i(r)$ using

$$u_i(r) = \frac{\sum_{j=1}^{K} u_{ij}\left(1/\left\|d(r,r_j)^{2/m-1}\right\|\right)}{\sum_{j=1}^{K}\left(1/\left\|d(r,r_j)^{2/m-1}\right\|\right)} \quad (2)$$

Increment i.

END DO UNTIL

END

It can be noticed from the algorithm that the membership values are inversely proportional to the distance between the (window centered at) current residue r and the (window centered at) neighbor r_j. The way in which the sample residues are assigned the class membership values, plays a vital role in the performance of the algorithm. The variable m is called fuzzifier [3]. The fuzzifier determines how the membership value varies with the distance. If m is set to 3, then the membership value of the residue is proportional to the inverse of the distance from the neighbor. If m is set to 2, then the membership value is proportional to the inverse of the square of the distance and so on.

Fuzzy K-Nearest Neighbor algorithms have two main advantages over the traditional (crisp) K-Nearest Neighbor algorithms. Firstly, while determining the class of the current residue, the algorithm is capable of taking into consideration the ambiguous nature of the neighbors if any. The algorithm has been designed such that these ambiguous neighbors do not play a crucial role in the classification of the current residue. The second advantage is that residues are assigned a membership value in each class rather than binary decision of '*belongs to*' or '*does not belong to*'. The advantage of such assignment is that these membership values act as strength or confidence with which the current residue belongs to a particular class. These membership values enable us to filter (explained in section 2.3) the output efficiently. For example, a residue has membership value of 0.9 in class *Helix*, 0.05 in the other two (*Strand* and *Coil*) classes; the residue can be assigned to the class *Helix* with high confidence. If the membership values are 0.5, 0.45 and 0.05 in classes *Helix*, *Strand* and *Coil* respectively, it is unlikely that the residue belongs to the class *Coil*.

Formatted

2.3 Filtering the output

In a basic setting, each residue can be assigned to class in which it has the maximum membership value. However, it has become a common practice to use a filter to smoothen the predicted output [10,11,12]. The filters eliminate unrealistic structures from the prediction, such as an isolated helix residue. We have designed a filter that has increased the overall accuracy by ~2%. The same filter can be used with any database as it is totally heuristic based and does not require any training. We explain the details of the filter with the memberships calculated by our algorithm for predicting the structure for a 63-residue protein chain *1cse_i*. The filter can be broken into the following six stages.

1. The membership curve is smoothed, i.e., the membership value of any given residue is equal to the sum of 50% of its own membership, 25% of the membership of the each of previous and next residues. This step will eliminate abrupt changes membership values of the residues due to lack of homologous segments in the database. The effect of smoothing the membership on *Helices* is depicted in Figure 3(a).
2. We incorporate the global information in to the filter. We first calculate the mean global propensity (the average membership of all the residues in the query protein) in each class (H_{Mean}, S_{Mean} and C_{Mean}). We find and mark all regions in which there are at least four contiguous residues that have the membership values greater than H_{Mean} and discard the rest as noise for Helix. A similar procedure is repeated for *Strand* class except that the marked region should have at least three contiguous residues that have membership values greater than S_{Mean}. The process for marking *Helix* segments is depicted in Figure 3(b).
3. The conflict between the *Helix* and *Coil,* and the conflict between the *Strand* and *Coil* are resolved. In order to resolve the conflict, first the overlap areas where both the *Helix* and *Coil* are marked are found. In overlap region, if the average *Helix* propensity is at least 50% of average *Coil* propensity, then the *Helix* is retained, else discarded. Similar procedure is repeated with *Strand* regions.
4. The conflict between the all the retained *Helices* and the *Strand* structures is resolved. The average propensities of both the structures are compared in the overlap region. The structure with a higher average propensity value is retained and the other is discarded. Figure 3 (c) illustrates the conflict between the *Helix* and *Strand* structures (residues 7-8).
5. There is chance that a structure might have been missed as some continuous residue segments are discarded in the previous stages. In order to prevent this situation, the structure discovered so far is compared with the structure without any filter. If there is *Helix* or *Strand* segment is overlooked by former and discovered by later, then the segment is simply copied.

6. We used the filters proposed by King and Sternberg in [12] and add additional four filters that are designed using visual examination. The additional filters we included are as following: `[E,H,E]→[E,E,E], [H,E,H]→[H,H,H], [H,C,H]→[H,H,H],[E,C,E]→[E,E,E].` For example, all occurrences of structure segment `EHE` will be replaced by structure segment `EEE`. Finally, the unrealistic structures like *Helix* with 2 resides, and *Strands* with one residue are filtered out. The resulting structure is the filtered structure.

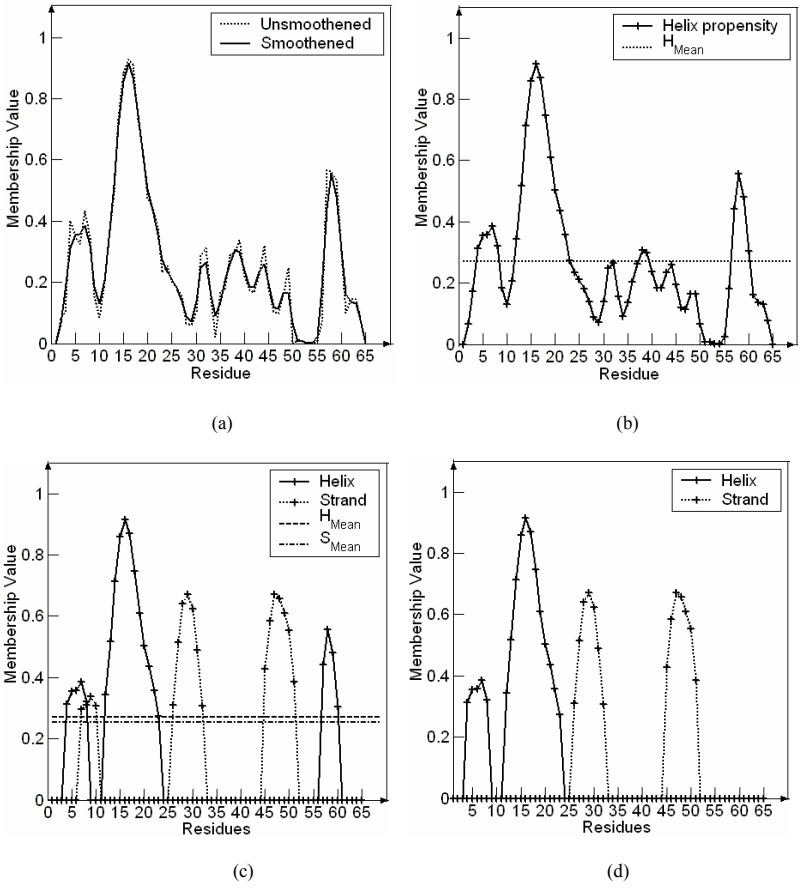


Figure 3. Filtering procedures after the initial secondary structure prediction. (a) The plots of *Helix* propensities before and after smoothening. (b) Residue segments 4-8, 13-20 and 57-60 are all marked as possible *Helix* structures. (c) The plot of the structures that remain after stage 3 filtration. The conflict between the retained *Helix* and *Sheet* structures for residues 7 and 8 can be observed. (d) The final membership plot of *1cse_i* after the six-stage filtration.

```
Unfiltered CCCCCHCCCCCHHHHHHHHCCCCCCEEEEECCCCCCCCCCCCCEEEEEECCCCCCHHHCCCCC
Target     CCCHHHCCCCHHHHHHHHHHCCCCEEEEEECCCCEECCCCCCEEEEEEECCCCEECCCCEEC
Filtered   CCCHHHHCCCCHHHHHHHHHCCCCCEEEEEECCCCCCCCCCCCEEEEEECCCCCCCCCCCCC
```

Figure 4. The predicted structure, the target structure and the filtered structure of the protein chain *1cse_i*.

The final membership plot of *1cse_i* after the six-stage filtration is presented in Figure 3 (d). The predicted structure, the actual structure and the filtered structure of *1cse_i* are presented in Figure 4.

3 Results

In this section we discuss the performance of our algorithm and compare our results with the reported results of the NNSSP [10] that used multiple sequence alignments and *K*-Nearest Neighbor algorithm on the Rost and Sander 126-protein benchmark. We also compare the performance of our algorithm with PSIPRED [16], currently one of the best prediction systems (*http://predictioncenter.llnl.gov*) that use PSSM profiles and Neural Networks.

To compare the performance of our algorithm with NNSSP, we use the jack-knife approach to test the performance of our algorithm i.e., when one of the protein is used as a query protein, the profiles that correspond to remaining 125 proteins are used to search for homologous segments. For window size (W) = 7 and 20-Nearest Neighbors (K), we have achieved a sustained overall three-state accuracy (Q_3) of 73.53%. Our per-residue accuracies [8] in each of the three states (*Helix*, *Strand* and *Coil*) and corresponding Matthews Correlation Coefficients (MCC) [1] are as follows: Q_{Helix} = 65.03%, MCC_{Helix} = 0.64; Q_{Sheet} = 60.94%, MCC_{Sheet} = 0.60; Q_{Coil} = 83.86%, MCC_{Coil} = 0.51. The results of our method and NNSSP are compared in Table 1.

Table 1. Performance comparison of our method with NNSSP tested on the same dataset of 126 proteins. Per-residue measures: Q_3, number of correctly predicted residues in all three states (*Helix*, *Strands* and *Coil*) divided by total number of residues. Q_{Helix}, number of correctly predicted residues in *Helix* divided by total number of residues in *Helix*. Q_{Strand}, number of residues in *Strand* correctly predicted divided by total number of residues in *Strand*.

	Q_3(%)	Q_{Helix}(%)	MCC_{Helix}	Q_{Strand}(%)	MCC_{Strand}
NNSSP	71.80	74.70	0.63	53.50	0.50
NNSSP + Filter	72.20	72.40	0.64	52.20	0.50
Our method	71.67	58.81	0.59	51.93	0.57
Our method + Filter	73.53	65.03	0.64	60.94	0.60

The performance of our algorithm cannot be directly compared with the performance PSIPRED directly as the distributed version of PSIPRED uses a much larger training set.

To make comparison relatively fair, we prepared a custom database with 1372 proteins that contains Rost and Sander 126-proteins and subset of proteins from PDB-select25 December 2003 list (*http://homepages.fh-giessen.de/~hg12640/pdbselect/*). We excluded the low-resolution structures (>2Å) and proteins shorter than 41 residues while preparing the database. Also, the sequence identity is less than 25% between any two proteins in the database. We predicted the structure of the Rost and Sander 126-proteins using the jack-knife approach using the larger database. We attained a sustained mean Q_3 accuracy of 75.75% while PSIPRED's reported [16] accuracy is 76.5%. As more protein structures will be solved and our method will be further improved, we expect that our performance may be as good as PSIPRED if not better.

4 Discussion

Our method uses a simple procedure. We first calculate the profiles (PSSMs) of the proteins in Rost and Sander database with the help of PSI-BLAST [13] using the *nr*-database (*ftp://ftp.ncbi.nlm.nih.gov/blast/db/*). We use a sliding window size of 7 (i.e., the segment used to predict the structure of n^{th} residue contains the residues n-3, n-2, n-1, n, n+1, n+2 and n+3) to calculate the neighbors. We then assign the membership values to neighbors in the three classes {*Helix*, *Strand*, *Coil*} using the procedure depicted in Figure 2 (b). Once the neighbors for each residue in the query protein are assigned membership values in various classes, the membership values of the residues in query protein are calculated using the Fuzzy *K*-Nearest Neighbor algorithm (section 2.2). The predicted structures are then smoothened using the filter described in section 2.3.

We have experimented with a wide range of values for the parameters in both the algorithm and the filter. We tried with various window sizes (W = 5, 7, …, 21) and found W = 7 produces the most accurate results. We have also experimented by varying the number of neighbors (K= 10, 20, 30, 40, 50). The performance of the algorithm was optimal when K = 20. Finally, in the Fuzzy *K*-Nearest Neighbor algorithm, we set the value of the fuzzifier (m) to 2. Changing the value of the fuzzifier (m = 1.1, …, 2, 3…) did not affect the performance of the algorithm significantly. We also, experimented with various weighing functions for both profile weighing and membership value assignments to neighbors. The functions presented in section 2 produced the optimal results.

Acknowledgments

The authors thank Dr. James Keller for useful insight and discussion. This work was supported by the Laboratory Directed Research and Development Program of Oak Ridge National Laboratory, under Contract DE-AC05-00OR22725, managed by UT-Battelle, LLC. It was also supported by the US Department of Energy's Genomes to Life program (http://www.doegenomestolife.org) under project, "Carbon Sequestration in *Synechococcus* Sp.: From Molecular Machines to Hierarchical Modeling" (*www.genomes-to-life.org*). This research used supercomputer resources of the Center for

Formatted

Computational Sciences at Oak Ridge National Laboratory, which is supported by the Office of Science of the Department of Energy.

References

1. B.W. Matthews. Comparison of the predicted and observed secondary structure of T4 phage lysozyme. Biochim.Biophysics.Acta.405:402-451
2. W. Kabsch and C. Sander. Dictionary of Protein Secondary Structure: Pattern Recognition of Hydrogen-bonded and Geometrical Features. Biopolymers, 22(12): 2577-637, 1983.
3. J. M. Keller, M. R. Gray and J. A. Givens, Jr. A fuzzy K-Nearest Neighbor Algorithm. IEEE Trans. on SMC, Volume SMC-15, No.4, 1985.
4. N. Qian and T. J. Sejnowski. Predicting the Secondary Structure of Globular Proteins Using Neural Networks. J.Mol. Biol, 202:865-884, 1988.
5. S. F. Altschul, W. Gish, W. Miller, E. W. Myers and D. J. Lipman. Basic local alignment search tool. J. Mol. Biol. 215:403-410, 1990.
6. X. Zhang, J. P. Mesirov and D.L Waltz. Hybrid system for Protein Secondary Structure Prediction. J. Mol. Biol., 225:1049-1063, 1992.
7. Tau-Mu Yi and E. S. Lander. Protein Secondary Structure Prediction using Nearest-Neighbor Methods. J. Mol. Biol., 232:1117-1129, 1993.
8. B. Rost and C. Sander. Prediction of Protein Secondary Structure at Better than 70% Accuracy. J. Mol. Biol., 232:584-599, 1993.
9. B. Rost and C. Sander. Combining Evolutionary Information and Neural Networks to Predict Protein Secondary Structure. Proteins: Structure, Function and Genetics 19:55-72, 1994.
10. A. A. Salamov and V. V. Solovyev. Prediction of Protein Secondary Structure by Combining Nearest-neighbor Algorithm and Multiple Sequence Alignments. J. Mol. Biol., 247:11-15, 1995.
11. J. M. Chandonia and M. Karplus. Neural networks for secondary structure and structural class predictions. Protein Science, 4:275-285, 1995.
12. R. D. King and M. J. E. Sternberg. Identification application of the concepts important for the accurate and reliable protein secondary structure prediction. Protein Science, 5:2298-2310, 1996.
13. S. F. Altschul, T. L. Madden, A. A. Schäffer, J. Zhang, Z. Zhang, W. Miller, and D. J. Lipman. Gapped BLAST and PSI-BLAST: a new generation of protein database search programs. Nucleic Acids Res. 25:3389-3402, 1997.
14. K. Karplus, C. Barrett and R. Hughey. Hidden Markov models for detecting remote protein homologies. Bioinformatics, 14:846-856, 1998.
15. C. Bystroff and D. Baker. Prediction of local structure in proteins using a library of sequence-structure motifs. J. Mol. Biol., 281:565-577, 1999.
16. D.T. Jones. Protein secondary structure prediction based on position-specific scoring matrices. J. Mol. Biol., 292:195-202, 1999.
17. H.M. Berman, J. Westbrook, Z. Feng, G. Gilliland, T.N. Bhat, H. Weissig, I.N. Shindyalov, P.E. Bourne. The Protein Data Bank. Nucleic Acids Research, 28:235-242, 2000.