THE PROTEIN STRUCTURE PREDICTION MODULE OF THE PROT-GRID INFORMATION SYSTEM

Dimitrios Vogiatzis
University of Cyprus, Dept. of Computer Science, 1678, Nicosia, Cyprus
Email: dimitrv@cs.ucy.ac.cy

Dimitrios Frosyniotis
National Technical University of Greece, Dept. of Electrical and Electronic Engineering, Athens, Greece
Email: dfros@cslab.ntua.gr

George A. Papadopoulos
University of Cyprus, Dept. of Computer Science, 1678, Nicosia, Cyprus
Email: george@cs.ucy.ac.cy

Key words: Biological databases, secondary structure prediction, multiple classifiers.

Abstract: In this work, we describe the protein secondary structure prediction module of a distributed bio-informatics system. Protein databases contain over a million of sequenced proteins, however there is structuring information for at most 2% of that number. The challenge is to reliably predict the structure based on classifiers. Our contribution is the evaluation of architectures of multiple classifier systems on a standard dataset (CB-396) containing protein sequencing information. We compare the results of a single classifier system based on SVMs, as well as with our version of an SVM based adaBoost algorithm and a novel fuzzy multi-SVM classifier.

1 INTRODUCTION

Over the past years we have witnessed an increase in the number of biological databases, but it is mainly the tremendous increase of their size that points to data mining tools so as to extract information out of the stored data. The desirable functions of such a data mining tool, would include the automatic categorisation of the data and the discovery of patterns. In this paper we are mainly interested in *protein structure databases* and we present the module for protein secondary structure prediction. This module is part of the PROT(ein)-GRID, which is an information retrieval and distribution system for the bio-informatics sector, under co-development in our lab and supported by the European Union.

Proteins are very important to all living organisms. All the cellular chemical transformations are aided by proteins and much of the structure of the cell is actually proteins. Proteins are chains of an "alphabet" of 20 amino acids, varying in length from 50 to 3000, with an average length of 200.

While it has been relatively inexpensive to sequence proteins, i.e. to discover the amino acids they are made of, it is expensive to discover the way the amino acids fold in 3-D space. In other words, the discovery of the protein structure which is made through X-ray crystallography, nuclear magnetic resonance etc. is time consuming and demands highly trained personnel (Baxevanis and Landsman, 1998). Thus, there is a great disparity between the number of proteins that have been sequenced and and the number of proteins with known structure. For instance, the Protein Information Resource (PIR) (http://pir.georgetown.edu/pirwww/-aboutpir/aboutpir.html), contains 1,011,453 sequencing entries [1], while the Protein Data Bank (PDB) has only 18,616 structures [2]. The ratio is almost the same as it was 4 years ago (Baxevanis and Landsman, 1998). A comprehensive understanding of the functional role of proteins depends on their structure and it is extremely important in drug design.

Because of the aforementioned situation, there is the need to develop algorithms for the prediction of the 3-D structure based on sequencing data. The earliest attempt for secondary structure prediction with Artificial Neural Networks (ANNs) goes back to (Qian and Sejnowski, 1988). Recently, there has been work on secondary structure prediction with Support Vector Machines (SVMs) (Hua and Sun, 2001).

Among the classical methods for secondary structure prediction, we should mention PHD (Rost and Sander, 1993), DSC (King and Sternberg, 1996), NNSSP (Salamov and Solovyev, 1995) and PREDA-

[1] release 1.05, 9-Sep-2002
[2] Last Update: 03-Sep-2002

TOR (Frishman and Argos, 1995). The PHD method, is based on a 3 level neural network. DSC combines residue attributes, hydrophobicity and aminoacid position along with linear discrimination. The NNSSP is based on a scored nearest neighbour algorithm. Finally, the PREDATOR employs a pairwise alignment method, rather than using global multiple alignment. PREDATOR also uses propensities for α-helix, b-strand and coil derived form a nearest-neighbour approach. The homology search is another method for secondary structure prediction, and it is based on comparison of an unknown sequence with a database of known sequences. However, is has been noted that 80% of homologues may not be found by these search methods (Brenner et al., 1998).

Our contribution is in the design and evaluation of 2 different predictor models of the secondary structure of proteins, where the common substrate is the SVM. The models are based on multiple classifi er systems (Kittler and Roli, 2001) and we compare them to a single classifi er system, which is an SVM. The multiple classifi er systems we have used is the *AdaBoost* (Rätsch et al., 2000), where we developed an SVM version, and the *fuzzy multiSVM* an algorithm built by one the paper's authors (Frossyniotis and Stafylopatis, 2001).

The rationale of this approach is that a single classifi er, in spite of its quality, implements one hypothesis on the data, whereas multiple classifi ers test multiple hypothesis and vote on the best one. Normally, one expects performance gains in terms of prediction accuracy.

In Sect. 2 we present an overview of voting algorithms, as well as our approach. Sect. 3 contains the experimental settings and results. Finally, in Sect. 4 we present the conclusions and future directions. In the next sections, characters in bold denote vectors.

2 VOTING ALGORITHMS

It has been shown that a monolithic approach to challenging data mining problems is ineffective. Especially, in the domain of classifi cation, a multiclassifi er system can exhibit shorter training time and higher accuracy than a single classifi er. Furthermore, the multiple classifi er system, might be easier to understand and maintain (Bauer and Kohavi, 1999).

A multiclassifi er system combines the performance of multiple single classifi ers so as to reach a global decision. A major issue here is the way of combining those individual decisions, the simplest method is to follow the scheme of the *majority wins*, but this is not always optimal.

Another major issue, and this concerns mainly the training phase, is the method of splitting the original problem into subproblems, which are assigned to the single classifi ers. Essentially, there two schemes: the *ensemble learning* and the *modular approach*.

In the fi rst scheme, each classifi er is trained on the whole of the available data. Then the decision of a multiclassifi er system is not simply the decision of the best classifi er, but it takes advantage of the information that is provided by the other classifi ers. One of the most popular approaches is the boosting method, (Avnimelech and Intrator, 1999), whereby specialised classifi ers are serially constructed to focus on data points misclassifi ed in previous stages.

The second scheme, i.e. the modular approach, advocates the idea of using a set classifi ers which are trained independently (possibly in parallel) on the available training patterns, and combining their decisions to produce the fi nal decision of the system.

2.1 Our Approach

The voting algorithms we will be using (i.e. the *AdaBoost* and the *fuzzy multiSVM*) belong in the *ensemble* and *modular* categories respectively, according to the aforementioned scheme (see Fig. 1 and Fig. 2 for a general overview of these methods). Furthermore, there has been used a method of multiple classifi ers, where each classifi er is trained on the whole of the data set of protein sequences (Cuff and Barton, 1999). This is known as *Consensus* method, which belongs essentially in the modular category and it combines the decisions of PHD, NNSSP, DSC and PREDATOR methods, which have been mentioned in the introduction.

We believe that the AdaBoost and the fuzzy multiSVM have certain distinct advantages, when compared to Consensus. The fi rst, advantage of the fuzzy multiSVM is the shorter training time, since each classifi er can be trained on a subset of the data. The advantage of AdaBoost, is based on the existence of theoretical limits on the performance of classifi ers on unseen data (i.e test data), which is not the case for the Consensus method. Furthermore, almost all of the real life classifi er applications are highly nonlinear; splitting the data into subsets, can potentially "smooth" the non-linearities and thus to improve the accuracy of the classifi er. Another reason for preferring the AdaBoost over the Consensus method is based on the fact that each classifi er in AdaBoost, implements a hypothesis on the data. The hypotheses become gradually, more and more refi ned which is not the case for the Consensus.

2.2 AdaBoost Voting Algorithm

The *Ada*ptive *boo*sting algorithm seeks to boost the performance of a single classifi er (Rätsch et al., 2000). Initially a classifi er is trained on the whole data

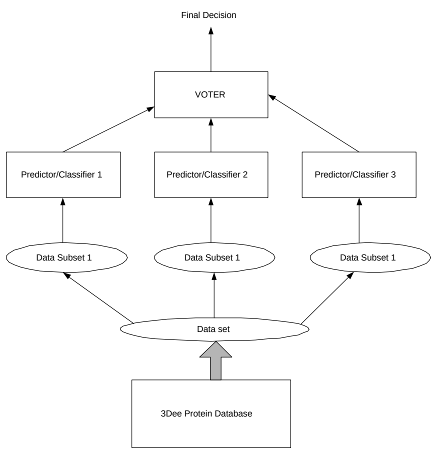

Figure 1: The fuzzy multiSVM method. Each predictor/classifier is trained on a region of the data set, which may partly overlap another region. For the final prediction, the classifiers are combined.

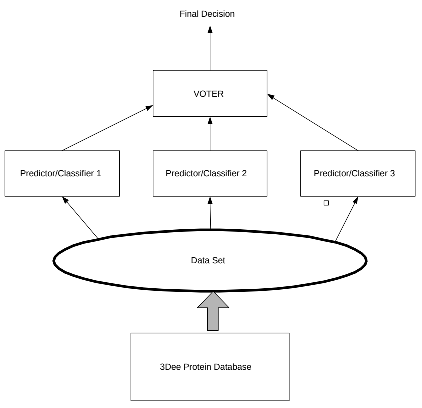

Figure 2: The AdaBoost method. Each predictor/classifier is trained on the whole of the data set. The final prediction involves participation of each classifier according to an importance factor.

Table 1: The AdaBoost algorithm for Support Vector Machines

Input A data set (**X**,Y), where **X** contains N multidimensional training data and Y are the corresponding categories.

Loop Let $\mathbf{w}$ be a vector of the weights of the input samples, and cc (mc) be the set of indexes of the correctly classified (misclassified) samples

1. train classifier C_i : $|C_i(\mathbf{X}') - Y| < d$, where d is a small positive constant. X' stems from X, different for every repetition because SVMs are deterministic. To create X' we sample the data of X according to their "importance" —to be defined later; and we add some uniform noise. Initially, $X' = X$.
2. $e_i = \frac{1}{n} \sum_{C_i(x_i) \neq y_i} \mathbf{w}_{x_i}$, sums the weights of misclassified samples.
3. $b_i = \log \frac{1-e_i}{e_i}$
4. $\mathbf{w}_{cc} = \frac{\mathbf{w}_{cc}}{2(1-e_i)}$, decreases the weights of correctly classified samples.
5. $\mathbf{w}_{mc} = \frac{\mathbf{w}_{mc}}{2e_i}$, increases the weights of misclassified samples

Output Each C_i classifier votes according to b_i, which denotes its "importance".

set, then the "importance" of the classifier can defined which, is high when the training error is small. Naturally, some of the training examples are misclassified by the current classifier, thus the next step is to mark the misclassified samples (actually it attaches a high weight to misclassified samples and low to the correctly classified ones). After this, the adaBoost algorithm will deploy another classifier, which will focus mostly, but not exclusively, on the misclassified samples. In the end, there will a series of classifiers, each with its own importance factor. The final decision is based on a voting scheme, where all classifiers vote according to their importance. Theoretical studies of the AdaBoost algorithm show that when each of the individual classifiers performs a little better than random, then the training error drops exponentially fast.

For the purpose of this work we have developed our own version of a multiclass AdaBoost algorithm. The learners upon which it is based are Support Vectors Machines (see Table 1).

2.3 The Fuzzy multiSVM algorithm

The fuzzy multiSVM is based on both supervised and unsupervised learning. According to the principle of *"divide-and-conquer"*, the input space is partitioned into overlapping subspaces and Support Vector Machines (SVMs) are subsequently used to solve the respective classification subtasks. Finally, the decisions of the individual SVMs are appropriately combined to obtain the final classification decision. We used the Fuzzy c-means (FCM) method for input space partitioning and we considered a scheme for combining the decisions of the SVMs based on a probabilistic interpretation.

In the present work we partition the original training data set into subsets and subsequently we use individual classifiers for solving the respective learning subtasks. A key feature of the method is that the training subsets represent non-disjoint regions that result from input space clustering. Thus, SVMs are assigned to overlapping regions from the beginning and acquire their specialisation through training with data sets that are representative of the regions. This partitioning approach produces a set of correlated "specialised" classifiers which attack a complex problem by applying the divide-and-conquer principle.

Next, we address the issue of *data set partitioning* based on unsupervised learning with the fuzzy c-means method, the *training sets generation* and the *combination of decisions* for the individuals SVMs

2.3.1 Partitioning of the Data Set

Consider a data set D having N patterns $\mathbf{x^i}$ where $\mathbf{x^i} \in R^d,\ i = 1, \ldots, N$. The first stage of the proposed classification technique consists in partitioning the original data set $D = \{\mathbf{x^1}, \ldots, \mathbf{x^N}\}$ using clustering techniques to identify natural groupings. As a result of clustering with the Fuzzy c-means algorithm (FCM) (Bezdek, 1981), a number of training subsets $D_1, D_2, \ldots, D_M$ are generated from the set D. Each data sample belongs to all clusters with a membership degree. The main drawbacks of this algorithm is that its performance depends on the initial cluster centers and that the number of clusters is predefined by the user. Therefore, it is required to run the FCM algorithm several times, each time with a different number of clusters to discover the number of clusters that is optimal.

2.3.2 Training Sets Generation

Following fuzzy clustering we can specify the degree (varying between 0 and 1) with which a data point belongs to each cluster. Let $\mathbf{x}$ be an input data point with its corresponding membership degree u_j to cluster j. To create M non-disjoint training sets corresponding to the M clusters we perform the following steps for each data point $\mathbf{x}$:

1. If $u_J \geq u_j\ ,\ \forall\ j = 1, \ldots, M$, then the data point $\mathbf{x}$ is assigned to the training set D_J.
2. For every $j = 1, \ldots, M\ ,\ j \neq J$, a random number q is generated according to a uniform distribution in the interval $(0, 1)$ and the data point $\mathbf{x}$ is assigned to the training set D_j if $q < u_j$.

Therefore, the data point $\mathbf{x}$ is assigned deterministically to the training set corresponding to the cluster with maximum membership for that point and is assigned probabilistically to each of the remaining training sets (with probability equal to the degree of membership to the respective cluster).

We can also observe a degree of overlapping between the training sets, as some patterns belong to two or more training sets simultaneously. The correlation between the data sets has a beneficial impact increasing the robustness of the multiSVM classification system.

In what concerns the classification modules of the proposed multiclassifier system, the primary idea is to train an SVM for each group of patterns D_j generated through the partitioning of the original data set D. In this sense, each classifier learns a subspace of the problem domain and becomes a "local expert" for the corresponding subdomain.

As already mentioned, an important advantage of the multiclassifier method is that the training of each SVM can be done separately and in parallel. Thus, in the case of a parallel implementation, the total training time of the system equals the worst training time achieved among the SVMs. It must be noted that this total training time cannot be greater than the training time of a single SVM classifier dealing with the entire training set.

2.3.3 Combination of Decisions

As described above, the original training data set D is partitioned into M (non-disjoint) subsets, and M classifiers are trained, one for each subset. Consider a new input vector $\mathbf{x}$ which belongs to one of c classes. Given the vector $\mathbf{x}$, a class label $C_j\ , j = 1, \ldots, M$, is produced by each SVM_j, and the membership degree u_j of the vector $\mathbf{x}$ to the respective cluster j is computed. To obtain the final classification decision, the decisions of the individual SVMs are combined in a probabilistic way.

To obtain the classification of $\mathbf{x}$ we compute the probability $P(k \mid \mathbf{x})$, $(k = 1, \ldots, c)$, that pattern $\mathbf{x}$ belongs to class k and select the class C with the maximum $P(C \mid \mathbf{x})$ as the final decision following the Bayes rule. The probability $P(k \mid \mathbf{x})$ is computed as follows:

$$P(k \mid \mathbf{x}) = \sum_{j=1}^{M} u_j I(C_j = k) \qquad (1)$$

Here $I(z)$ is an indicator function, i.e. $I(z) = 1$ if $z =$ true, otherwise $I(z) = 0$. The above equation states that the class probability $P(k \mid \mathbf{x})$ results as the sum of the weights u_j of the classifi ers that suggest class k. It is easy to check that $\sum_{k=1}^{c} P(k \mid \mathbf{x}) = 1$.

3 EXPERIMENTS

There are data sets, which have been derived from protein databases and they serve the purpose of testbeds. These sets are used for training a predictor, and they contain the primary structure of a protein, as well as the secondary structure. Normally, one uses a part of such a set for training and another part for testing the predictor's accuracy on unseen data. One of the oldest and widely used is the RS126 set (Rost and Sander, 1993), which contains 126 proteins. However, it has been noted that this set does not qualify for secondary structure prediction (Cuff and Barton, 1999).

For the purpose of our experiments we have used the CB396 set (Cuff and Barton, 1999), which contains sequencing information from 396 proteins, as well as secondary structure information for every aminoacid. The CB396 has been produced from the 3Dee database of structural domain defi nitions of proteins (Siddiqui et al., 2001). The same authors, have also produced the CB512 data set from the same database, which contains 512 proteins, however we choose not to use it because of the current computational limitations.

3.1 Data Preprocessing

The CB396 contains 396 protein sequences, with an average aminoacid length of 156. Each aminoacid belongs to one of the eight following folding categories:

H	a-helix	B	isolated β-bridge
G	3_{10} -helix	T	urn
I	π-helix	S	bend
E	β strand	-	rest

This information has been used to train with supervised learning the three classifi ers that we have described. Since the length of proteins, in terms of aminoacids varies, we have used a window of length 9. The window concept implies that the neighbouring aminoacids play a certain role in the current aminoacid. For instance given the following subsequence of a protein (upper line), we wish to train a classifi er to predict the secondary structure of the middle aminoacid (lower line).

Aminoacid seq.:	ISFH $\boxed{\text{S}}$ GYSG
Folding:	T

Furthermore, we have adopted the unary coding for each aminoacid, which produces an input vector of 21*9=189 dimensions. There are 20 aminoacids plus another symbol to denote the left or right end of protein. Finally, as it customary in the literature we have mapped the eight folding categories into three, following the DSSP standard (Cuff and Barton, 1999)

H,G	$\rightarrow$	h (elix)
E, B	$\rightarrow$	e (strand)
other	$\rightarrow$	c (oil)

We have also tried to reduce the number of dimensions, by the application of Principal Component Analysis, but the predictive accuracy of the classifi ers slightly dropped, which is not acceptable at this point.

3.2 Results

We have compared three predictive models for secondary structure prediction based on supervised learning. The fi rst model was a Support Vector Machine, which is a single predictor. The second one is the AdaBoost algorithm, which is a method based on multiple hypotheses about the data which gradually become more and more refi ne. Finally, the third model (fuzzy multiSVM) is based on both unsupervised and supervised learning methods. To build the classifi cation system, fi rst the original training set is divided into overlapping subsets by applying a clustering technique. Then, an individual SVM is trained on every defi ned subset. To obtain the classifi cation of a new pattern, the decisions of the SVMs are appropriately combined.

In order to assess the results we have adopted the average Q_3 measure, which has been used extensively and it is defi ned as follows,

$$Q_3 = \sum_{i=h,e,c} \frac{predicted_i}{total_i} \tag{2}$$

The following results are averages over 5 cases, where the training and testing sets are chosen randomly from a pool of 61,895 vectors. Each training set comprises 15,000 vectors, and the rest is used for testing. In the following table we report on the predicting ability of the classifi ers, and we provide certain parameters that we have found to be optimal.

For, the single SVM we have used the "osu SVM" (Ma and Ahalt,), whereas for the other two algorithms we have written code on matlab on top of the single SVM code. The training time for a single SVM was about 30min (Pentium IV, 1500Mhz, Windows 2000), considering that AdaBoost SVM reached its best result with 5 hypotheses, we reach a number of 150min. This was not enough, for we provided averages over 5 different experiments, which amounts to 750 minutes. Then, we had to perform the same experiments with at least four different parameter sets, which amount to 3000min. Similar was the training time for the fuzzy multiSVM, where there is an additional clustering phase, but it is counterbalanced with smaller training sets for each SVM.

CB396 data set		
	Testing (avg)	Std. Deviation
single-SVM	61.9%	0.19
SVM AdaBoost	5 hypotheses (SVMs)	
	63.4%	0.20
fuzzy multiSVM	8 clusters	
	65.3%	0.17

4 CONCLUSIONS

We have described and evaluated the protein secondary structure prediction module of a larger bioinformatics retrieval system. The other modules which are to follow, refer to integration of a variety of information sources to aid researchers in biology. Among the extensions, there will be the automatic construction of relationships between proteins and relevant scientific literature. All this will be developed in a Grid environment, as it is dictated by the requirements of the IST *PROT-GRID* project.

The focus has been on secondary structure prediction based on sequencing information. To this end with have designed two algorithms the AdaBoost SVM and the fuzzy multiSVM and evaluated them against a single SVM. The results of the multiclassifier methods (i.e. in AdaBoost SVM and fuzzy multiSVM) are better than the ones of the single classifier, which verifies our initial assumption.

The adaBoost has been implemented for the multi class problem, whereas most of its public domain implementations are for two classes only. Furthermore, because SVMs are deterministic predictors, on the testing of different hypotheses we had to sample a subset of the training data. There are theoretical results that suggest that adaBoost methods perform better than simple modular methods (such as the *Consesus* algorithm, which is mentioned in the introduction). More specifically it has been shown that the training error drops exponentially fast in the case of adaBoost algorithm. In addition, the adaBoost performs a balanced reduction of the *bias error* and *variance error*, which is not the case for the Consensus method. The bias error stems from the fact that we have not found the optimal solution and the variance expresses the dependence of the solution on the current training set.

The fuzzy multiSVM implemented in this work is quite general allowing the implementation and testing of other techniques both in the clustering and the classification module. One of the advantages of fuzzy multiSVM is the straightforward parallel implementation of the training phase. When compared to the *Consensus* method mentioned in the introduction, we get small training subsets for each classifier, which is translated into performance gains. Generally speaking, the adaBoost and fuzzy multiSVM will exhibit lower learning time and improved prediction capability when compared to monolithic methods (such as the single SVM and the other methods we have mentioned in the introduction). To emphasize this, imagine that we have used just 396 proteins out of 18,000 available, which produced 65,895 training samples. If we were to use a significant percentage of the 18,000, the size of the training set would be unfeasible for monolithic methods.

Another point is that in the literature, the prediction is better than ours, but they have not been tested on the same data, thus the comparison is meaningless at this point, the interested reader is advised to consult (Cuff and Barton, 1999). Actually, our data set was restricted so that the prediction may be computationally feasible. We believe that the results could be improved with a larger training set, this is based on the observation that for the single Support Vector Machine (i.e. the first type of the classifiers), the number of support vectors was very big. It is known, that the support vectors denote the borders of the classes. For instance, $(9.780,\ 7.576,\ 9.557)$ vectors were used to separate classes h , e and c from each other, whereas the total number of training vectors, was 15,000. This suggests that there are barely sufficient data to train the classifiers, the same reasoning can be extended to the multi classifier cases.

One of the issues that arise, when using ANNs or SVMs as predictors of the a data set, is that they are black boxes. What this means, is that the knowledge is stored in arithmetic form (weights in ANNs and Lagrange coefficients in SVMs). This cannot be communicated to humans nor it can be easily extended as it is the case in a rule based system. However, there has been considerable progress in techniques that allow us to look into the ANNs (Tickle et al., 1998) or SVMs (Núñez et al., 2002). This entails the extraction of knowledge in the form of if-then rules.

A rather technical problem that we faced and which renders the comparison of similar methods difficult if not unfeasible, is that the protein data in biological databases are classified in 8 categories, however the predictive model is built on 3 categories, by an appropriate mapping. However, the exact mapping is not always mentioned in the relevant papers.

In a future work we plan to tackle two issues. First, are the technical matters that refer to the predictive models that we employed. Second, is the application of the our method into larger problems. This means the application to larger set of proteins, so as to achieve higher accuracy. We also intend to move in the field of *tertiary structure prediction*.

For the technical matters, first we must tackle the problem of dimensions, in all of our tests the number of dimensions is 189, this number was produced considering a window of length 9, larger windows will produce even more dimensions. Thus, we must apply dimensionality reduction methods, such as non-linear Principal Component Analysis and possibly the Independent Component Analysis, since the linear Component Analysis seems to worsen the accuracy of our predictors. The use of the window implies that only the neighbourhood of an aminoacid affects its local folding. However, there is enough evidence that there are non-local interactions among the aminoacids; in this context locality is to be meant in terms of the primary structure. One way to tackle this problem is to employ recurrent neural networks, where the prediction depends on the history of the inputs.

REFERENCES

Avnimelech, R. and Intrator, N. (1999). Boosted mixture of experts: an ensemble learning scheme. *Neural computation*, 11(2):483–497.

Bauer, E. and Kohavi, R. (1999). An empirical comparison of voting classifi cation algorithms: Bagging, boosting, and variants. *Machine Learning*, 36:105–142.

Baxevanis, A. and Landsman, D. (1998). Predictive methods using protein sequences. In Baxevanis, A. and Ouellette, B., editors, *Bioinformatics, A Practical Guide to the Analysis of Genes and Proteins*, chapter 11, pages 246–267. Wiley-Interscience.

Bezdek, J. (1981). *Pattern Recognition with Fuzzy Objective Function Algorithms*. Plenum Press, New York.

Brenner, S., Chothia, C., and Hubbard, T. (1998). Assessing sequence comparison methods with reliable structurally idenfi fi ed distant evolutionary relationships. *Proc. Nat. Acd. Sci.*, pages 6073–6087.

Cuff, J. and Barton, G. (1999). Evaluation and improvement of multiple sequence methods for protein secondary structure prediction. *Proteins: Structure, Function and Genetics*, 34:508–519. [http://www.compbio.dundee.ac.uk/~www-jpred/data/].

Frishman, D. and Argos, P. (1995). Knowledge-based secondary structure assignment. *Proteins: Struct. Funct. Genet.*, 23:566–579.

Frossyniotis, D. and Stafylopatis, A. (2001). A Multi-SVM Classifi cation System. In *Proceedings of the Second International Workshop on Multiple Classifier Systems (MCS 2001)*, LNCS 2096, pages 198–207, Cambridge, UK. Springer.

Hua, S. and Sun, Z. (2001). A novel method of protein secondary structure prediction with segment overlap measure: Support vector machine approach. *Journal of Molecular Biology*, 308:397–407.

King, R. and Sternberg, M. (1996). Identifi cation and application of the concepts important for accurate and reliable protein secondary structure prediction. *Protein Science*, 5:2298–2310.

Kittler, J. and Roli, F., editors (2001). *Multiple Classifier Systems*. Springer Verlag.

Ma, J. and Ahalt, S. OSU SVM Classifi er Matlab Toolbox (ver 2.00). [http://eewww.eng.ohio-state.edu/maj/osu_svm/].

Núñez, H., Angulo, C., and Catala, A. (2002). Rule extraction from support vector machines. In *ESANN'2002 proceedings, European Symposium on Artificial Neural Networks*, pages 107–112.

Qian, N. and Sejnowski, T. (1988). Predicting the secondary structure of globular proteins using neural network models. *Journal of Molecular Biology*, 202:865–884.

Rätsch, G., Onoda, T., and Müller, K.-R. (2000). Soft Margins for AdaBoost. *Machine Learning*. [http://www.kernel-machines.org].

Rost, B. and Sander, C. (1993). Prediction of secondary structure at better than 70% accuracy. *J. Mol. Biol.*, 232:584–599.

Salamov, A. and Solovyev, V. (1995). Prediction of protein secondary structure by combining nearest neighbor algorithms and multiple sequence alignments. *Journal of Molecular Biology*, 247:11–15.

Siddiqui, A., Dengler, U., and Barton, G. (2001). 3Dee: A database of protein structural domains. *Bioinformatics*, 16:200–201. [http://www.compbio.dundee.ac.uk/3Dee/].

Tickle, A., Andrews, R., Golea, M., and Dietrich, J. (1998). The truth will come to light: Directions and challenges in extracting the knowledge embedded withing trained artifi cial neural networks. *IEEE Transactions on Neural Networks*, 9:1057–1068.