Computers in Biology and Medicine 36 (2006) 145–156

Computers in Biology and Medicine

www.intl.elsevierhealth.com/journals/cobm

Non-parametric classification of protein secondary structures

Elias Zintzaras[a,*], Nigel P. Brown[b], Axel Kowald[c]

[a]Department of Biomathematics, University of Thessaly School of Medicine, Papakyriazi 22, Larisa 41222, Greece
[b]Biomedical Informatics Unit, Imperial Cancer Research Fund, London, UK
[c]Max Planck Institute for Molecular Genetics, Berlin, Germany

Received 29 June 2004; accepted 18 October 2004

Abstract

Proteins were classified into their families using a classification tree method which is based on the coefficient of variations of physico-chemical and geometrical properties of the secondary structures of proteins. The tree method uses as splitting criterion the increase in purity when a node is split into two subnodes and the size of the tree is controlled by a threshold level for the improvement of the apparent misclassification rate (AMR) of the tree after each splitting step. The classification tree method seems effective in reproducing similar structural groupings as the method of dynamic programming. For comparison, we also used another two methods: neural networks and support vector machines. We could show that the presented classification tree method performs better in classifying proteins into their families. The presented algorithm might be suitable for a rapid preliminary classification of proteins into their corresponding families.

Keywords: Classification; Classification tree; Neural nets; Support vector machines; Protein structure; Physico-chemical properties; Geometrical properties; Biostatistics; Bioinformatics

1. Introduction

Proteins are macromolecular products of the cellular biosynthetic machinery and consist of a linear chain of amino acids, which is encoded by the sequence of nucleotides in the DNA. While currently the primary structure (amino acid sequence) of tens of thousands of proteins is known, the tertiary structure

[*] Corresponding author. Fax: +30 2410 565270.
E-mail address: zintza@med.uth.gr (E. Zintzaras).

0010-4825/$ - see front matter © 2004 Elsevier Ltd. All rights reserved.
doi:10.1016/j.compbiomed.2004.10.001

Table 1
The groups and families of the proteins used for the classification tree algorithm

Group	Family	Abbreviation
α-helix proteins	Lysozymes	Lys
	Calcium binding	Cal
	Hemerythrins	Hem
	Globins	Glo
	Cytochromes	Cyt
β-strand proteins	Antigen binding	Ant
	Copper binding	Cop
	Immunoglobulin heavy chains	ImH
	Immunoglobulin light chains	ImL
Serine proteases	Serine proteases	Ser
Mixed	Trypsin inhibitors	Try
	Glyceraldehyde phosphate dehydrogenases	Gly
	Subtilisins	Sub
	Periplasmic space binding proteins	Per
	Dinucleotide binding folds	Din

(three-dimensional folding) of only a few hundred sequences have been solved. The tertiary structure is built up of secondary structure elements whose main components are α-helices and β-strands which are connected via short loops of amino acids. The three-dimensional structure of a protein can be described by a set of variables which are functions of the angles and distances between the secondary structure elements [1].

Using data derived from these variables (see method) we used a classification tree method in order to classify and identify families of a set of 75 proteins with known structures (Table 1). The classification tree method is a non-parametric discrimination method and therefore it is a distribution-free method.

The results obtained by applying the classification tree method are compared with the findings of Orengo et al. [2], who used the dynamic programming method, an elaborate and a time-consuming method, to study the same set of proteins.

The results of the presented tree classification algorithm are also related to the results obtained after training a neural network on our data. The surge of interest in neural networks over the last years and their ability for generalised pattern recognition has also led to their application in predicting protein classes and secondary structures [3,4]. The comparison shows that under certain conditions the tree method gives considerably better results than the neural network.

In addition, the protein structures were classified using support vector machines (SVM) [5]. The tree method may be advantageous in classifying the proteins into their families.

2. Methods

Various physico-chemical and geometrical measures have been suggested to classify proteins [6,2]. In this study, we use a set of eight variables per protein for the classification process. Seven of these variables

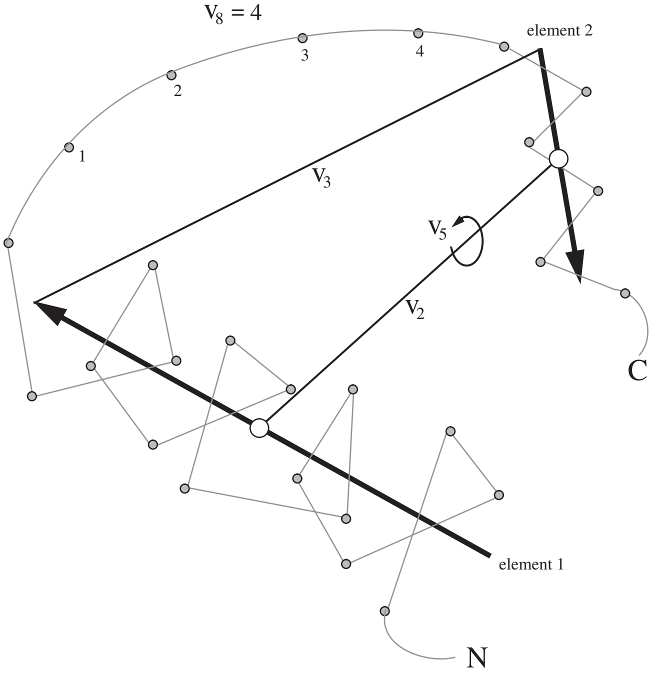

Fig. 1. Schematic representation of some of the variables used to describe protein structures. A fragment of a protein is represented by the dashed line running from the N-terminus to the C-terminus connecting some amino acid residues indicated by grey spots. Two secondary structure elements along this, an α-helix (element 1) and a β-strand (element 2), are abstracted as thick vectors with centres indicated by white spots. The intercentroid distance (v_2), distance between ends of secondary structures (v_3), intervening loop length (v_4) and intercentroid angle (v_5) are shown by the thin solid lines. For the classification tree algorithm these variables are summarised by calculating the coefficient of variation for the variables of a given protein.

are derived from measurements between secondary structures as defined by Orengo et al. [2]. We also added another variable which is a measure of the size of the protein and is defined as the number of possible interactions between secondary structures of the protein ($v_1 = n(n-1)/2$, n being the number of secondary structures). The following list summarises the variables used for classifying proteins (see also Fig. 1):

$v_1 =$ number of possible pairs of secondary structures in a protein.
$v_2 =$ distances between the centres of two secondary structures.
$v_3 =$ distances between the ends of secondary structures.
$v_4 =$ intercentroid tilt angles between each two structures.
$v_5 =$ intercentroid angles between each two structures.
$v_6 =$ hydrophobic moment vector angles.
$v_7 =$ centroid moment vector angles.
$v_8 =$ connecting loop lengths given in amino acid residues between secondary structures.

Because each protein consists of several secondary structures, a whole set of measurements corresponds to each variable. For example, protein 1CC5 consists of 10 secondary structures and therefore each of the variables v_2–v_8 has a number of observations equal to the number of secondary structures. In order to summarise the observations on each variable (v_2–v_8), it is necessary to use a summary statistic such

Table 2
The variables v'_1–v'_8 for a set of typical proteins are shown. Each protein is a representative of its family (see results)

Family	Protein	v'_1	v'_2	v'_3	v'_4	v'_5	v'_6	v'_7	v'_8
Lys	2LZ2	210	0.301	0.439	0.688	0.385	0.481	0.687	0.412
Cal	3CLN	153	0.454	0.497	0.705	0.490	0.448	0.721	0.538
Hem	2MHR	276	0.192	0.575	0.985	0.483	0.504	0.861	0.592
Glo	1HBS	105	0.323	0.414	0.764	0.390	0.447	0.727	0.397
Cyt	155C	45	0.339	0.661	0.683	0.363	0.423	0.614	0.296
Ant	2HLA	78	0.549	0.587	0.758	0.417	0.456	0.456	0.526
Cop	1PCY	120	0.348	0.677	0.587	0.486	0.471	0.665	0.416
ImH	2HFL	171	0.531	0.563	0.789	0.446	0.433	0.719	0.525
ImL	2MCP	231	0.554	0.575	0.815	0.438	0.442	0.693	0.525
Ser	2KAI	120	0.488	0.427	0.879	0.413	0.492	0.783	0.452
Try	2SEC	325	0.387	0.549	0.804	0.468	0.455	0.723	0.388
Gly	2GDI	36	0.460	0.499	0.783	0.427	0.430	0.708	0.457
Sub	1SBT	78	0.375	0.536	0.778	0.443	0.435	0.700	0.383
Per	2GBP	91	0.423	0.620	1.037	0.489	0.424	0.718	0.432
Din	4MDH	91	0.445	0.538	0.832	0.422	0.436	0.714	0.444

as the coefficient of variation (CV). The CV is defined as the standard deviation divided by the mean and it is independent of the units of the measurement. It can be used to summarise the distribution of each measured variable and is commonly used in biological sciences for distribution comparisons [7]. This statistic has already been used for classifying successfully a small set of proteins by Zintzaras et al. [8]. In the following analysis, the v_1 and the CVs of the variables v_2–v_8 were used for classifying the 75 protein structures (Table 2 shows these data for typical proteins of each family).

The classification tree method was applied to these variables as follows: if v'_1–v'_8 are the derived variables (the v_1 and the CVs of v_1–v_8, respectively) of each protein, the method first sorts the individual proteins according to each variable. It then finds for each variable the point that results in the best split, i.e. the split which results in the purest subsets. After choosing the variable with the overall best possible split the data set is divided into two subsets. Finally, in a recursive manner, the subsets are split according to the same procedure as the initial data set.

The partition algorithm is represented as a tree. The node t is split into two subnodes t_l and t_r in such a way that the subnodes are purer than the original node. The proportion of individual proteins in t that go to t_l and t_r are p_l and p_r, respectively. A node is pure if it contains only individuals from one class. The goodness of a split, s, is defined by the increase in purity $D(s)$ and is given by

$$D(s) = I(t) - [p_\mathrm{l} I(t_\mathrm{l}) + p_\mathrm{r} I(t_\mathrm{r})], \tag{1}$$

where $I(t) = 1 - \sum_{i=1}^{c} [\pi(i/t)]^2$ is the Gini index of diversity. $\pi(i/t)$ is the proportion of individuals of class i in node t and c is the number of different classes [1].

The measure of performance of the classification tree is the resubstitution estimate (R) of the true misclassification rate. This is the estimate of the proportion of misclassified individuals by using the resubstitution method. Using the resubstitution method, a tree is constructed with the data of the 75 proteins as a training set and we see how the tree classifies each protein [9].

The tree is growing in the following way: each time after a split classes are assigned to the new nodes using the majority rule and the AMR of the tree is calculated. If the split improves the AMR by a certain percentage the split is performed, otherwise the node becomes a terminal node [1,8].

Orengo et al. [2] have developed a method for classifying protein families. The method first recognises related proteins by sequence similarity and subsequently performs detailed structural comparison to establish a set of unique fold families. The proteins are structurally compared using dynamic programming which obtains the optimal alignment between every pair of protein structures [10].

The neural networks were constructed using the Stuttgart neural network simulator (SNNS) [11]. The nets constructed possess eight input units (corresponding to the eight variables), two hidden layers with ten and six units and four, respectively, 15 output units. The standard backpropagation algorithm has been used for learning with a learning parameter, η, of 0.2. The variables v'_1–v'_8 were used for input. For this purpose v'_1 was normalised before training. For learning the data set of 75 proteins has been split into a training set of 50 proteins and a validation set of 25 proteins. Every ten training cycles learning was interrupted and the error on the validation set was calculated. Learning was stopped in the minimum of the validation set error. At this point the net generalises best. If learning is continued, overtraining occurs and the performance of the net on the whole data decreases, despite the fact that the error on the training set still gets smaller.

For pruning, the method of finding the "non-contributing units" has been used. Again, standard backpropagation was used as learning function with 1000 cycles for the first training and 100 cycles for each retraining. Only pruning of input nodes was allowed, no hidden units were pruned.

Another classification technique that has become popular in recent years is support vector machines (SVMs) [5]. The 75 proteins of our study represent a multi-class problem, since they can either be assigned to four groups or fifteen families. Originally, SVMs could only be applied to binary clustering problems, but in recent years variants have been developed, that can also cope with multi-class problems. We used libSVM 2.6 [12] for this purpose. As a pre-processing step, the eight variables (v'_1–v'_8) were scaled to the range of -1 to 1 with a libSVM utility program and were then used for classification.

3. Results

The 75 proteins belong to four groups and each group consists of families; in total there are fifteen families. The groups are predominantly α proteins, predominantly β proteins, serine proteases and mixed proteins (Table 1).

3.1. Classification trees

We applied the tree method to the data of the 75 and when a 6% improvement in the AMR was used as threshold a tree with four terminal nodes resulted (Fig. 2). The misclassification rate of the tree is $R=48\%$. Most proteins from the same groups are segregated in the four terminal nodes. The only discrepancies are the cytochromes (Cyt), which are mixed with the serine proteases, and the mixed family and the copper and calcium binding proteins (Cop, Cal) as well as the hemerythrins (Hem), which too are mixed with proteins from the mixed family group. However, overall the tree method based on the proposed eight variables was able to identify the four main groups of proteins.

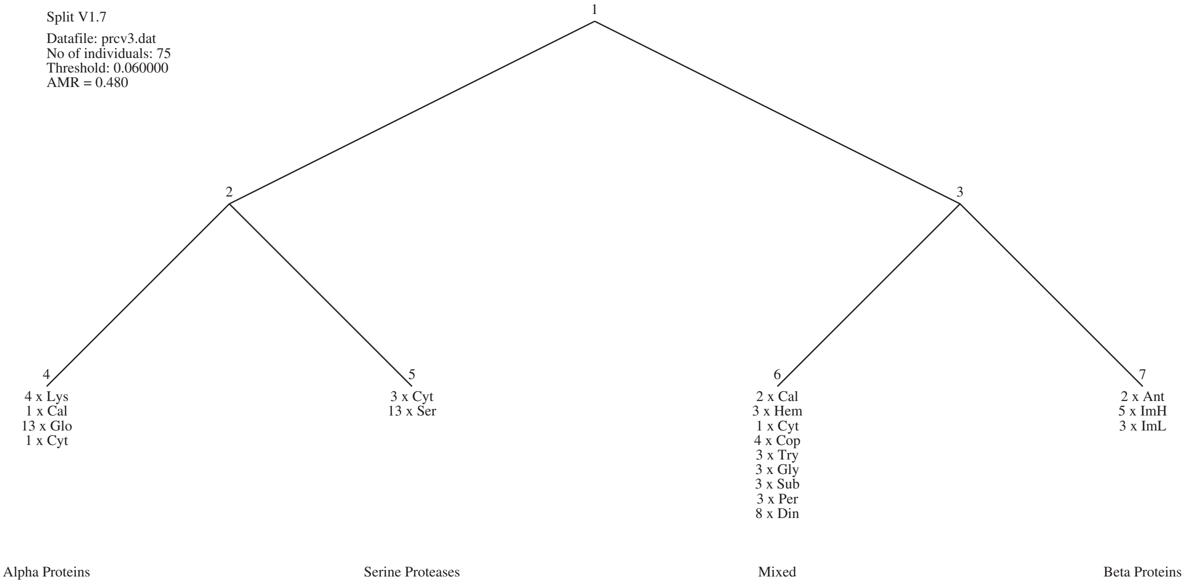

Fig. 2. The tree obtained for the four major groups of proteins (see Table 1). The threshold for splitting a node was based on a 6% improvement of the AMR. At each terminal node the number of individuals from each protein family is shown. In terminal node 5 for instance there are three members of the cytochromes (3*X*Cyt) and thirteen members of the family of serine proteases (13*X*Ser).

When the threshold value for improvement is reduced, the size of the tree increases. The maximum tree consists of terminal nodes equal to the number of proteins, i.e. 75. By varying this threshold a tree results with terminal nodes that represents the protein families.

When a 2% improvement for the AMR is used as threshold a tree with 14 terminal nodes is generated and the misclassification rate of the whole tree is $R = 12\%$ (Fig. 3). Two cytochromes are mixed with the lysozymes (node 8) and the copper binding proteins (node 17). In addition one calcium binding protein is mixed with the lysozymes (node 8). The method has also problems to separate subtilisins and dinucleotide binding proteins (nodes 22 and 26). And finally the immunoglobins heavy and light chains are segregated together in one terminal node (node 15).

In general, the tree method together with the eight proposed variables (v'_1–v'_8) classified the proteins into their families. The most used variables for growing the tree are v'_1 and v'_2 with three splits each. But also v'_3, v'_5 and v'_7 are important with two splits each. However, two variables (v'_6 and v'_8) were not involved in the construction of the tree at all.

The dynamic programming method of Orengo et al. [2] has classified the 75 proteins to their families successfully; however, the method has not been applied for classifying the proteins to their groups.

3.2. Neural nets

We also used neural nets for the classification of proteins. In particular, we used the Stuttgart Neural Network Simulator (SNNS) to construct and train two networks to classify the proteins into their groups and families, respectively (for detail on how the nets were trained see methods). We used a training and

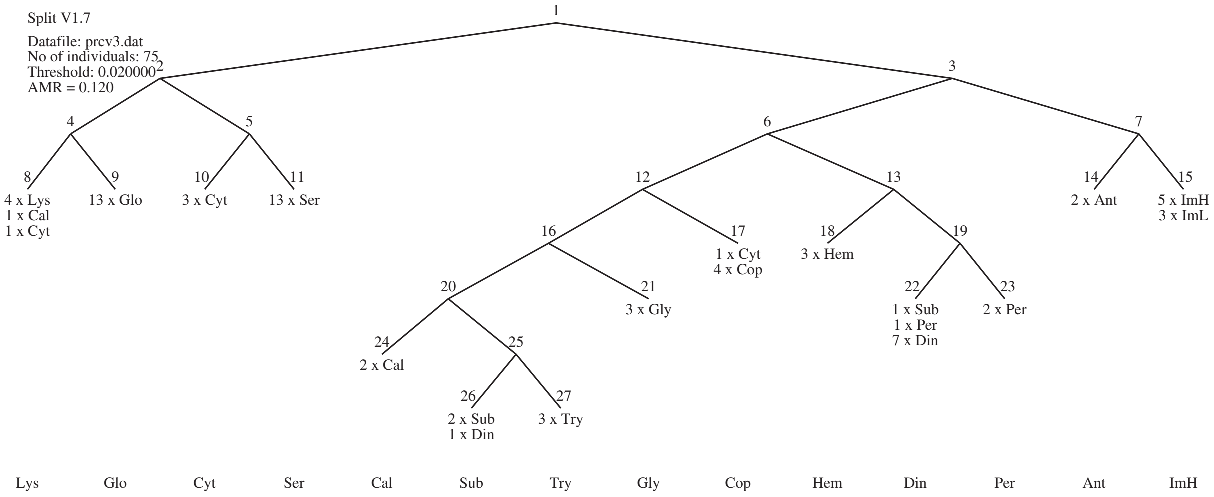

Fig. 3. Using a splitting threshold of 2% a much more detailed tree of the protein groups and families can be obtained. Below each terminal node the family is shown to which the node has been assigned according to the majority rule.

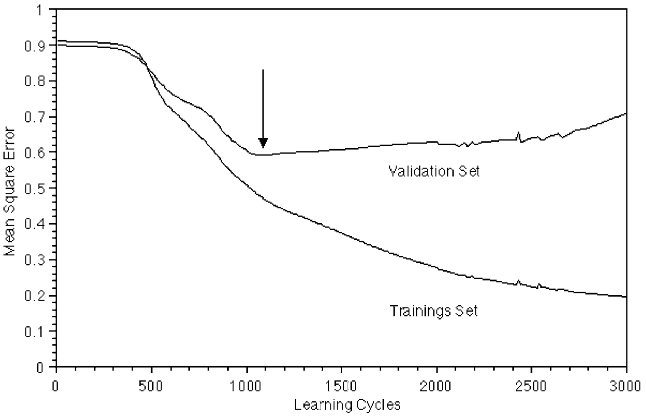

Fig. 4. Development of the mean square error (MSE) for the validation and the trainings set during the learning phase of the net which was used to classify protein families. The arrow points to the minimum of the MSE of the validation set which it reaches after 1080 learning cycles. For details on training parameters see methods.

a validation set to decide on the optimal number of learning cycles. Fig. 4 shows the results of a learning session for the net which was trained to identify individual protein families. The mean square errors (MSE) of the training set as well as the validation set keep decreasing until cycle 1080, where the error for the validation set reaches a minimum and begins increasing again. This is the point when overtraining starts and the training of the net has to be stopped.

After training the two nets (for group and family identification) to this optimum the ability of the nets to classify the 75 proteins was tested. After presenting a specific input pattern (v'_1–v'_8) the output unit with the highest value decided on group or family membership. The results are summarised in Table 3.

Table 3
Classification results of the 75 proteins for classification trees, neural nets and support vector machines. For group classification the proteins were assigned to four different groups and for family classification the proteins were assigned to 15 different families (see Table 1)

	Classification tree	Neural net	SVM
Group misclassification	13/75	2/75	8/75
Family misclassification	9/75	26/75	24/75

For group classification, the neural net method resulted in less misclassifications while the classification tree method performed much better when the proteins were classified into families.

To find out if the neural network method predominantly uses the same input variables as the tree method, we used a pruning algorithm provided by SNNS to successively prune the least important input node. Fig. 5 shows the result after four pruning steps. The order in which the input nodes were pruned is I5 (v'_5), I6 (v'_6), I7 (v'_7) and I8 (v'_8).

3.3. Support vector machines

For a further comparison we used the SVM package libSVM of Chang and Lin [12]. The protein data were scaled and then classified using the radial basis function (RBF) kernel and a five-fold cross validation. The parameters gamma and C fine tune the behaviour of the radial basis function. Gamma influences the width of the RBF and controls the generalization vs. overfitting behaviour. C is a cost factor associated with misclassified examples. To find optimal values for both parameters, the 2D parameter space was sampled at regular intervals (on a logarithmic scale) and a contour map of the resulting cross validation accuracy is generated. As can be seen in Fig. 6, the highest accuracy for family classification of 68% was achieved for $C = 2^{6.8} = 111.4$ and $\gamma = 2^{-3.3} = 0.1$. This translates to a misclassification rate of 24/75 (Table 3) for the classification of protein families. For the classification of groups the misclassification rate was 8/75 with optimal $C = 32$ and $\gamma = 0.5$.

4. Discussion

The results indicate that the derived variables (CVs) may be used for classifying proteins into their families. The classification tree has some advantages over other conventional multivariate methods [9] like linear discriminant analysis, because it allows to see the structure of the data at each growing stage of the algorithm. Using a large threshold of 6% (which gives rise to few terminal nodes) the four major groups of proteins could be identified (Fig. 2). Using a smaller value of 2% the classification process reproduced most of the individual protein families (Fig. 3).

It is not surprising that the most polluted group in Fig. 2 is the mixed family group. As the name indicates, this group contains a mixture of families which do not share such clearly defined properties as the α-helix or β-strand groups. Under these considerations the classification results are surprisingly good and indicate that other common properties have been used by the tree algorithm. Although the classification into individual family using a lower threshold is not perfect (Fig. 3), there are only relatively few exceptions. In most cases families were grouped together, which belong to the same group (nodes

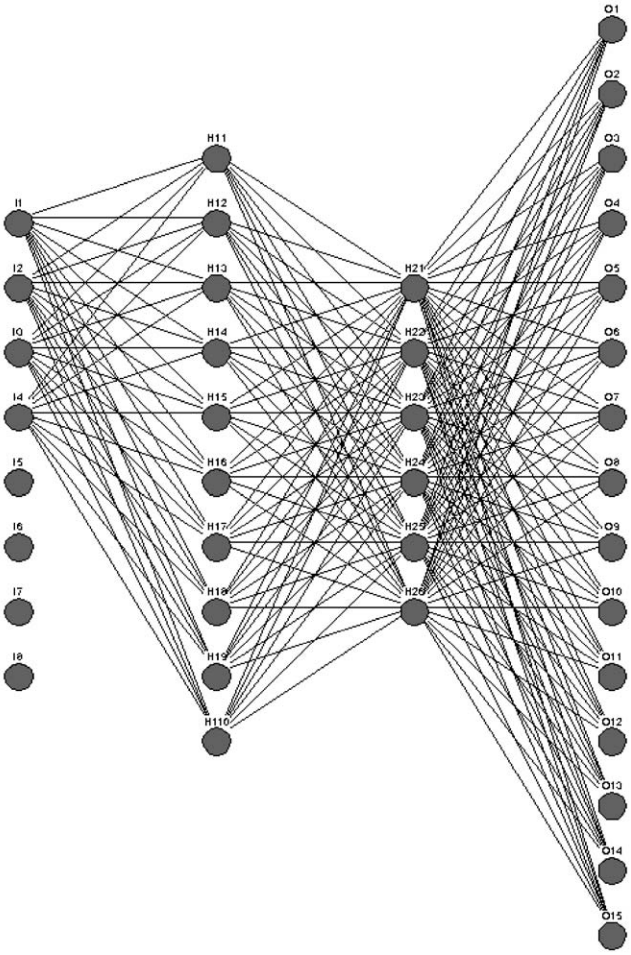

Fig. 5. Diagram of the neural network which classified protein families after several pruning steps of input nodes. The net consists of an input layer with 8 units (I1–I8), two hidden layers with ten and six units (H11–H110 and H21–H26) and an output layer with 15 units (O1–O15). Initially the network is completely connected, but after pruning of the input units from I5 to I8 their links are no longer shown.

8,15,22,26). Only once, families of different groups were mixed when cytochromes and copper binding proteins were assigned to the same terminal node (node 17).

The analysis of the variables used for the thirteen splits in Fig. 3 shows that v_1' and v_2' are used most often (three times), but one dominant variable does not exist. Instead it is more interesting that v_6' and v_8' have not been used at all. It seems that the hydrophobic moment vector angles (v_6') and the connecting loop length (v_8') are not relevant for the classification process.

The described method cannot only be used for classification, but also to predict the family of a new protein. For this purpose the variables v_1'–v_8' have to be measured for the new protein. The protein can then be *dropped* into the tree which has been constructed with well-known proteins. The algorithm then uses the splitting variables and values calculated for the construction of the tree and assigns the

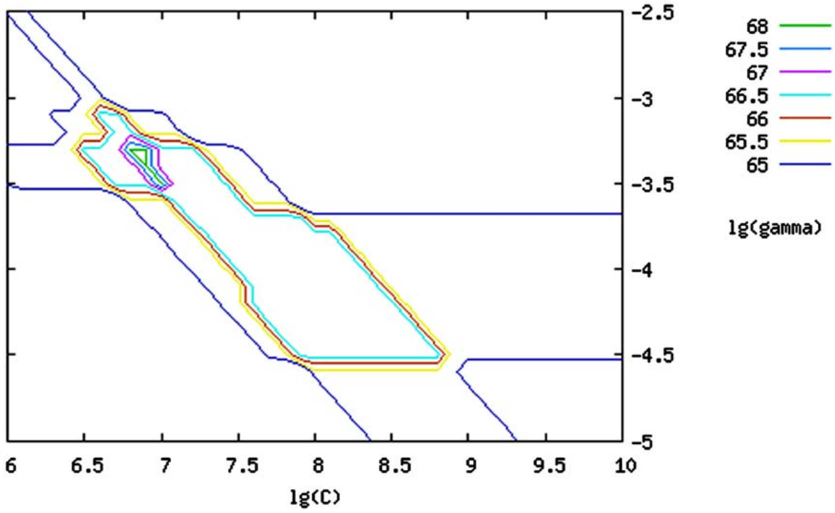

Fig. 6. Contour plot of the accuracy achieved by SVMs for the classification of the protein families. The radial basis function, that was used as kernel, is controlled by the parameters gamma and C. To find optimal values, the parameter space was sampled with five-fold cross validation and the axes of the contour plot are shown on a logarithmic scale of base 2.

unknown protein to a terminal node. The family assigned to this node is the predicted family of the new protein.

The dynamic programming method classifies successfully the proteins into their families, as it was anticipated, since it is a method based on a detailed topological analysis of the protein structures. However, the method can be used only on powerful machines, it is complicated and time consuming.

Using the Stuttgart Neural Network Simulator we constructed two feedforward nets and compared their ability to classify the proteins of our data set with the results of the classification tree method. When the neural nets were used to identify the four major groups (α-helix, β-strands, serine proteases and mixed) there were only two misclassifications in contrast to 13 for the tree method. However, the situation was reversed when the aim was to identify the 15 different families. The tree method had only nine false classifications while there were 26 for the neural network. As can be seen from Fig. 2, the most mistakes of the tree method were in assigning proteins to the mixed family group (ten out of 30 are assigned erroneously). As mentioned earlier, the mixed family group is a very heterogeneous group which makes it difficult for the tree method to identify. The bad performance of the neural net in classifying individual protein families is probably caused by the small size of our data set. Considering that the data set had to be split (for creating a training and validation set) and two neural nets had to be constructed and trained, it seems that especially for small data sets the presented classification tree method is superior and simpler to use compared to the neural network approach.

In Fig. 5, the results after four successive rounds of input node pruning are shown. The pruning algorithm provided by SNNS removes the least important node and retrains the net afterwards. The input nodes were removed in the order I5 (v'_5), I6 (v'_6), I7 (v'_7) and I8 (v'_8). Although difficult to compare it seems that for both methods the derived variables v'_1–v'_4 are more relevant than the variables v'_5–v'_8.

Support vector machines performed very badly in classifying proteins into their families. The family results are similar to the high misclassification rate of neural nets. While neural nets performed significantly better than the tree method in classifying proteins into their groups, the results of SVMs are in between the misclassification rate of classification trees and neural nets. For the classification of families, the tree method outperformed both, SVMs and neural nets.

The classification tree method may be advantageous to support vector machine analysis and to neural networks when a small data set (e.g. 75 individuals) is considered. The classification tree method could provide a fast and preliminary analysis, and a more detailed topological analysis can afterwards be provided by more elaborate methods such as the dynamic programming method Orengo et al. [2].

5. Summary

In this paper, a classification tree method for classifying proteins into their groups and families, based on physico-chemical and geometrical measures of their secondary structures is described. The results were compared with the findings of three other methods: dynamic programming, neural networks and support vector machines. The tree method produces similar structural groupings with the dynamic programming, and may perform better than neural networks and support vector machines in classifying proteins into their families.

Acknowledgements

We thank Dr. W.R. Taylor, Dr. D.T. Jones and Prof. T.B. Kirkwood for useful comments and discussion.

References

[1] L. Breiman, J.H. Friedman, R.A. Olson, C.J. Stone, Classification and Regression Trees, Wadsworth, Belmont, California, 1984.

[2] C.A. Orengo, N.P. Brown, W.R. Taylor, Fast structure alignment for protein data bank searching, Proteins Struct. Function Genetics 14 (1992) 139–167.

[3] M. Reczko, H. Bohrc, The DEF data base of sequence based protein fold class predictions, Nucleic Acids Research 22 (1994) 3616–3619.

[4] P. Stolorz, A. Lapedes, Y. Xia, Predicting Protein Secondary Structure Using Neural Net and Statistical Methods, Los Alamos Preprint, LA-UR-91-15, 1991.

[5] N. Vapnik, The Nature of Statistical Learning Theory, Springer, Berlin, 1995.

[6] C.A. Orengo, T. Flores, W.R. Taylor, J.M. Thorton, Identification and classification of protein fold families, Protein Eng. 6 (1993) 485–500.

[7] R. Mead, R.N. Curnow, Statistical Methods for Agriculture and Experimental Biology, Chapman & Hall, London, 1998.

[8] E. Zintzaras, N.P. Brown, A. Kowald, Growing a classification tree using the apparent misclassification rate, Comp. Appl. Biosci. 10 (1994) 263–271.

[9] W.J. Krzanowski, Principles of Multivariate Analysis, Clarendon Press, Oxford, 1988.

[10] D.T. Jones, W.R. Taylor, J.M. Thorton, A new approach to protein fold recognition, Nature 358 (1992) 86–89.

[11] A. Zell, N. Mache, T. Sommer, T. Korb, Proceedings Applications of Neural Networks Conference, SPIE, Aerospace Sensing International Symposium, Florida, Orlando, vol. 1469, 1991, pp. 708–719.

[12] C.C. Chang, C.J. Lin, A library for support vector machines. Software available at http://www.csie.ntu.edu.tw/∼cjlin/libsvm, 2001.

Elias Zintzaras holds a B.Sc. in Mathematics (University of Thessaloniki, 1988), an M.Sc. in Biometrics (University of Reading, 1990) and a Ph.D. in Statistical and Informatics Modelling in the Biosciences (British Medical Research Council and Eotvos University, 1998). He was a research fellow in Biometry at the British MRC National Institute for Medical Research (1990–1993), Biostatistical consultant at the Hellenic Organization for Medicines (1994–2000) and Researcher in Bioinformatics at the Greek National Agricultural Research Foundation (2001–2002). He is an Assistant Professor in Biomathematics-Biometrics at the University of Thessaly School of Medicine. His current research activities focus on computational biology, genetic epidemiology and mathematical modelling in evolutionary genetics.

Nigel Brown holds a Ph.D. in Protein Structure (National Institute for Medical Research, London, 1992), an M.Sc. in Information Systems Engineering (South Bank Polytechnic, London, 1986) and a B.Sc. in Genetics (Liverpool University, 1982). His current research activities focus on protein sequence and structure analysis, genome analysis, computational biology using object orientation, logic programming and database techniques.

Axel Kowald holds an M.Sc. in Biochemistry (Free University Berlin, 1987) and a Ph.D. in Mathematical Biology (National Institute for Medical Research, London, 1992). After a postdoctoral position at the University of Manchester (1993–1995) he spent a year as research fellow at the Institute for Advanced Studies in Budapest (1996–1997). From 1997 until 2001 he was research scientist at the Humboldt University of Berlin and is currently in the Kinetic Modelling Group of the Max Planck Institute for Molecular Genetics in Berlin. His current research interests focus on the mathematical modelling of processes involved in the biology of ageing and systems biology.