**SETIT 2009**
5th International Conference: **S**ciences of **E**lectronic, **T**echnologies of **I**nformation and **T**elecommunications
March 22-26, 2009 – TUNISIA

# A System for Protein Classification Based on Protein 3D Structure

Kire TRIVODALIEV*, Slobodan KALAJDZISKI *and Danco DAVCEV*

* *University Ss. Cyril&Methodius, Faculty of Electrical Engineering and Information Technologies, Skopje, Macedonia*

**kiret@feit.ukim.edu.mk**
**skalaj@feit.ukim.edu.mk**
**etfdav@feit.ukim.edu.mk**

**Abstract:** The classification of proteins based on their structure plays an important role in the deduction or discovery of protein function. Furthermore, the large number of potential classes causes problems for many classification strategies, increasing the likelihood that the classifier will reach local optima while trying to distinguish between all of the possible structural categories. In this paper, we present an efficient system for protein classification by using 3D structure content representation. We use a 3D structure-based approach for the efficient classification of protein molecules. The method relies on descriptors extracted from the known protein structure. These descriptors integrate geometry-based and biological features of the protein. An ART neural network algorithm is introduced to achieve dimensionality reduction, thus improving the overall performance of the system. In this work, a hierarchical strategy, using Boosted C4.5 algorithm, is applied for structural classification based on the SCOP (Structural Classification of Proteins) hierarchy. The SCOP database was used to evaluate the effectiveness of the multi-level approach of this system.
**Key words:** Data mining, protein classification.

## INTRODUCTION

The structure of a protein molecule is the main factor which determines its chemical properties as well as its function, hence the 3D representation of a residue sequence and the way this sequence folds in the 3D space are very important. The 3D protein structures are stored in the world-wide repository Protein Data Bank (PDB) [1] which is the primary repository for experimentally determined proteins structures. With the technology innovation the number of 3D protein structures increases every day.

At present, the Structural Classification of Protein (SCOP) database [2], which is manually constructed by human experts, is believed to maintain the most accurate structural classification. Manual classification provides reliable results. However, it is labour intensive. The gap between protein holdings of PDB and SCOP databases continues to grow [3]. Hence, developing an efficient and accurate classifier of protein structures will have a vital impact on effectively classifying high-throughput newly-discovered structures.

In this paper we present a system for classification of protein molecules from the existing protein database. We use the PDB files to extract the information about the 3D structure of the protein. After proper positioning of the structures, the Spherical Trace Transform is applied to them to produce descriptor vectors, which are completely rotation invariant. We have applied the method given in [4] to extract geometry descriptor. Additionally, biological properties of the protein are taken as in [5], forming better integrated descriptor.

Protein 3D structure data are with high dimensionality and very immense, which could easily overwhelm the processing and storage capacity of a centralized database system. Unsupervised learning Artificial Neural Networks (ANNs) are able to discover both regularities and irregularities in the redundant input data. As a result of the dimensionality reduction obtained easily from the outputs of these algorithms, computation can be done faster and with lower costs. We have chosen the ART ANN models [6],[7] for dimensionality reduction. These models have both long- and short- term memory and are able to distinguish the two, which we consider to be an advantage because whenever a new protein is added to

the system the existing model is only adapted to the change introduced by that protein.

There are many algorithms used for protein classification as Naive Bayesian classifier, nearest neighbour classifier, decision trees and so on. Our approach is to use decision tree classification algorithm which will classify the structures according to the SCOP classification hierarchy. We believe a classification scheme should take advantage of such a natural hierarchy as SCOP. By using a multi-level classification strategy, one can cut down on the number of potential outcomes, which is useful when faced with noisy, real-world data that is not clearly separable. The proposed strategy should eliminate many of the issues that arise from a large multi-class decision problem [8],[9].

The proposed system architecture and research methods are given in section 2, experiments and evaluation results are given in section 3 and section 4 concludes the paper.

## 1. System architecture

In this paper we present a system for efficient classification of protein molecules based on structural features of the molecules stored in protein database. The general system architecture is shown on Figure 1.

As can be seen the descriptor generation process of the system extracts the geometrical and biological features of the protein molecule, thus forming the real valued descriptor. The dimensionality of the descriptor is 416 features for the geometrical part, and 34 features for the biological part, thus giving the dimensionality of 450 in the resulting descriptor for one protein.

When classifying the proteins according to SCOP classification, and using knowledge discovery in data techniques, the dimensionality of the descriptor is crucial factor. The number of 450 features in the descriptor is very high, thus the process of classification can be very slow. We apply dimension reduction of the descriptor by using ART ANN. The dimensionality of the descriptor is reduced from 450 to 88 features (from 416 features to 54 features for the geometrical part and from 34 features to 34 features for the biological part).

The Boosted C4.5 decision tree algorithm is applied to the datasets produced on the basis of the new descriptor. We have built trees for class, fold and superfamily levels in the SCOP classification database. These classification trees can be further used for assigning the appropriate SCOP level for new protein entries in the database.

There are two phases in protein classification: offline and online phase, which are shown on Figure 1.

The offline phase refers to the process of generating descriptors for all proteins in the database and then training the models for the dimensionality reduction and classification modules of the system.

The online phase refers to the process of classification of a protein that has an unknown SCOP hierarchy. For this protein the corresponding descriptor is extracted at first, which is then used in the previously trained models for dimensionality reduction and classification. In result the computed SCOP hierarchy is assigned to the protein.

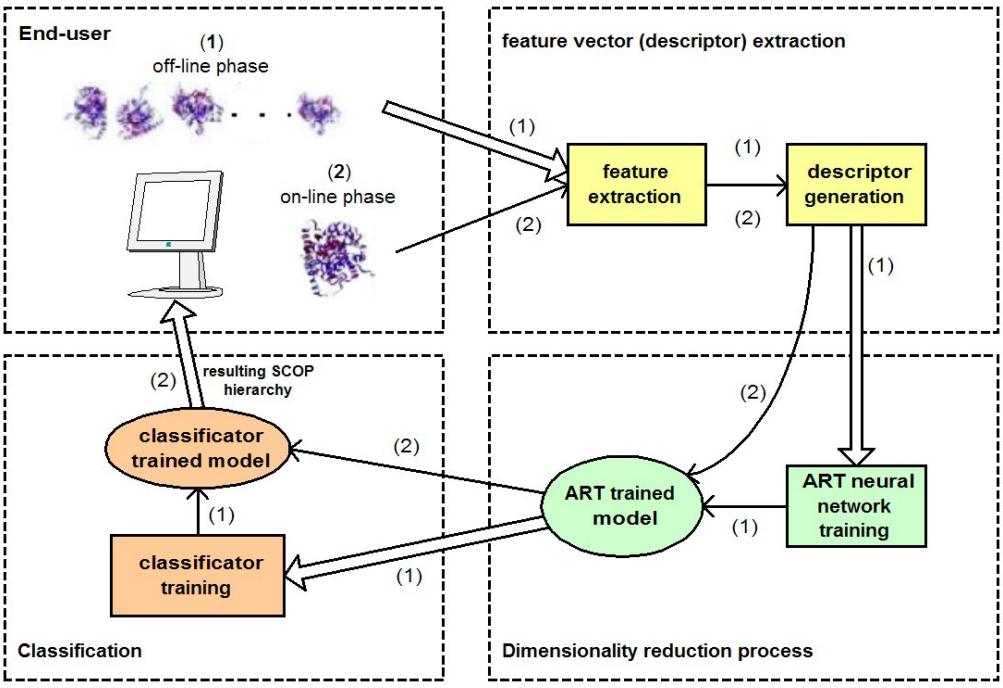


**Figure 1.** *Protein classification system architecture*

## 2. Research methods

### 2.1. Protein Descriptor Extraction

The information about protein structure is stored in PDB files. They contain information about their 3D structure and their biological properties. For each atom of the protein, the coordinates of the origin are presented and also information about the type of the atom. Information about the amino acid sequence, secondary structure elements, and some other futures are also contained in the PDB file.

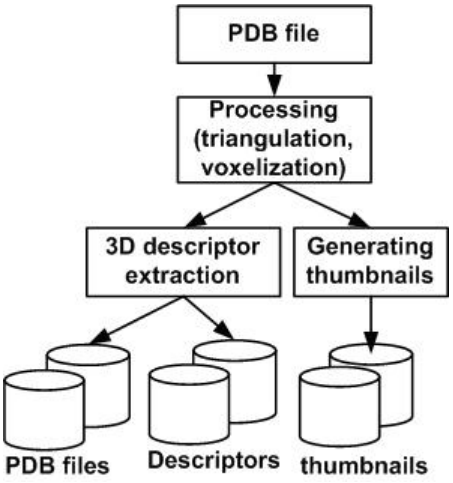


**Figure 2.** *Process of protein descriptor extraction*

As can be seen from the figure 2, since the exact 3D position of each atom and its radius are known (according to PDB file), it may be represented by a triangulated sphere. The protein is comprised of a set of spheres, along with the corresponding vertices and the connections among them. To provide translation and scale invariance, the center of mass is calculated and the protein is translated so the new center of mass is at the origin. The distance $d_{max}$ between the new origin and the most distant vertex is computed and protein is scaled, so $d_{max}$ =1.

After triangulation, we perform voxelization. Voxelization transforms the continuous 3D-space, into the discrete 3D voxel space. The voxelization proceeds in three steps: discretization, sampling, and storing. Discretization divides the continuous 3D-space into voxels. With sampling, depending on positions of the polygons of a 3D-mesh model, to each voxel $v_{abc}$, a value is attributed. Usually, $v_{abc}$ is a scalar value, and we used real voxel grid, where $v_{abc}$ is equal to the fraction of the total surface area S of the mesh which is inside the region $\mu_{abc}$ (1).

$$\upsilon_{abc} = \frac{area\{\mu_{abc} \cap I\}}{S}, \quad 0 \le a,b,c \le N-1. \tag{1}$$

Each triangle $T_j$ of a model is subdivided into $p_j^2$ coincident triangles each of which has the surface area equal to $\delta = S_j / p_j^2$, where $S_j$ is the area of $T_j$ . If all vertices of the triangle $T_j$ lie in the same cuboid region $\mu_{abc}$, then we set $p_j$ = 1, otherwise we use (2) to determine the value of $p_j$.

$$p_j = \left\lceil \sqrt{p_{\min} \frac{S_j}{S}} \right\rceil \tag{2}$$

For each newly obtained triangle, the center of gravity G is computed, and the voxel $\mu_{abc}$ is determined. Finally, the attribute $v_{abc}$ is incremented by δ. The quality of approximation is set by the parameter $p_{min}$.

The information contained in a voxel grid can be processed further to obtain both correlated information and more compact representation of voxel attributes as a feature. We applied the 3D Discrete Fourier Transform (3D-DFT) to obtain a spectral domain feature vector which also provides rotation invariance of the descriptor.

A 3D-array of complex numbers F = [$f_{abc}$] is transformed into another 3D-array by (3).

$$f'_{pqs} = \frac{1}{\sqrt{MNP}} \sum_{a=0}^{M-1} \sum_{b=0}^{N-1} \sum_{c=0}^{P-1} f_{abc} e^{-2\pi j(ap/M + bq/N + cs/P)} \tag{3}$$

Since we apply the 3D-DFT to a voxel grid with real-valued attributes, we shift the indices so that (a;b;c) is translated into (a–M/2; b–N/2; c–P/2). Let M=N=P and we introduce the abbreviation (4).

$$\upsilon'_{a-M/2,b-N/2,c-P/2} \equiv \upsilon_{abc} \tag{4}$$

Thus, the origin (0;0;0) is shifted to (N/2;N/2;N/2), and we adjust with (5).

$$f'_{pqs} = \frac{1}{\sqrt{N^3}} \sum_{a=-N/2}^{N/2-1} \sum_{b=-N/2}^{N/2-1} \sum_{c=-N/2}^{N/2-1} \upsilon'_{abc} e^{-2\pi j(ap+bq+cs)/N} \tag{5}$$

We take magnitudes of low-frequency coefficients as components of the vector. Since the 3D-DFT input is a real-valued array, the symmetry is present among obtained coefficients, so the feature vector is formed from all non-symmetrical coefficients (6).

$$1 \le |p| + |q| + |s| \le k \le N/2 \qquad (6)$$

We normalize $f'_{pqs}$ by dividing by $|f'_{000}|$. Then, we form the feature vector by the scaled values of $f'_{pqs}$. This vector presents geometrical properties of the protein, and consists of 416 real valued features.

Additionally, characteristic attributes of the primary and secondary structure of the protein molecules are extracted, forming attribute-based descriptor vectors. This part of the descriptor consists of 34 real valued features appended to the end of the geometry based descriptor. More specifically, concerning the primary structure, the ratio of the amino acids' occurrences, the hydrophobic amino acids ratio and the ratio of the helix types' occurrences in a protein are calculated. Concerning the secondary structure, the number of Helices, Sheets and Turns in a protein are also calculated. These features and the weights assigned to them are listed in Table 1.

| **Secondary structure features** | **Weight** |
|---|---|
| Number of HELICES | 1 % |
| Number of SHEETS | 1 % |
| Number of TURNS | 1 % |
| **Primary structure features** | **Weight** |
| Hydrophobic residue ratio | 6 % |
| Helix type | 1 % |
| Residue ratio | 90 % |

**Table 1.** *Structural features and their weights.*

### 2.2. ART Algorithm

ART networks develop stable recognition codes by self-organization in response to arbitrary sequences of input patterns. They were designed to solve the so called stability-plasticity dilemma: how to continue to learn from new events without forgetting previously learned information. ART networks model several features such as robustness to variations in intensity), detection of signals mixed with noise, and both short- and long-term memory to accommodate variable rates of change in the environment. There are several variations of ART-based networks: ART1 (three-layer network with binary inputs), Fuzzy ART (with analog inputs, representing neuro-fuzzy hybrids which inherit all key features of ART), their supervised versions ARTMAP and FuzzyARTMAP and many others.

In Figure 3 typical representation of an ART Artificial Neural Network is given. Winning F2 category nodes are selected by the attentional subsystem. Category search is controlled by the orienting subsystem. If the degree of category match at the F1 layer is lower than the so called vigilance level ρ, a reset signal will be triggered, which will deactivate the current winning F2 node for the period of presentation of the current input. An ART network is built up of three layers: the input layer (F0), the comparison layer (F1) and the recognition layer (F2) with N, N and M neurons, respectively.

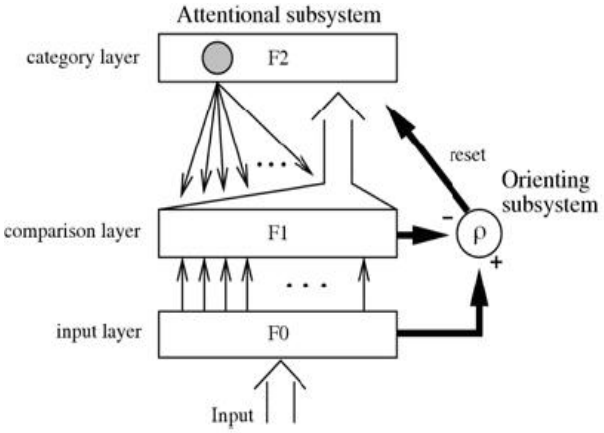


**Figure 3.** *Architecture of the ART network*

The input layer stores the input pattern, and each neuron in the input layer is connected to its corresponding node in the comparison layer via one-to-one, non-modifiable links. Nodes in the F2 layer represent input categories. The F1 and F2 layers interact with each other through weighted bottom- up and top-down connections that are modified when the network learns. There are additional gain control signals in the network (not shown in Figure 3) that regulate its operation, but those will not be detailed here. The learning process of the network can be described as follows: At each presentation of a non-zero binary input pattern x ($x_j \in \{0, 1\}; j = 1, 2, \ldots, N$), the network attempts to classify it into one of its existing categories based on its similarity to the stored prototype of each category node. More precisely, for each node $i$ in the F2 layer, the bottom-up activation $T_i$ is calculated, which can be expressed as

$$T_i = \frac{|w_i \cap x|}{\beta + |w_i|} \quad i = 1,\ldots,M \qquad (7)$$

where $|\cdot|$ is the norm operator (for a vector $u$ it is: $|u| \equiv \sum_{j=1}^{N} u_j$), $w_i$ is the (binary) weight vector or prototype of category $i$, and $\beta > 0$ is a parameter. Then the F2 node $I$ that has the highest bottom-up activation, i.e. $T_I = \max\{T_i \mid i = 1, \ldots, M\}$, is selected (realizing so called winner-takes-all competition). The weight vector of the winning node ($w_I$) will then be compared to the current input at the comparison layer. If they are similar enough, i.e. if they satisfy the matching condition:

$$\frac{|w_I \cap x|}{|x|} \ge \rho \qquad (8)$$

where $\rho$ is a system parameter called vigilance ($0 < \rho \leq 1$), then the F2 node $I$ will capture the current input and the network learns by modifying $w_i$:

$$w_I^{new} = \eta(w_I^{old} \bigcap x) + (1-\eta)w_I^{old} \qquad (9)$$

where $\eta$ is the learning rate ($0 < \eta \leq 1$) (the case when $\eta$=1 is called "fast learning"). All other weights in the network remain unchanged.

If, however, the stored prototype $w_I$ does not match the input sufficiently, i.e. if the condition (8) is not met, the winning F2 node will be reset (by activating the reset signal in Figure 3) for the period of presentation of the current input. Then another F2 node (or category) is selected with the highest $T_i$, whose prototype will be matched against the input, and so on. This "hypothesis-testing" cycle is repeated until the network either finds a stored category whose prototype matches the input well enough, or allocates a new F2 node in which case learning takes place according to (9).

The number of developed categories can be controlled by setting the vigilance $\rho$: the higher the vigilance level, the larger number of more specific categories will be created. At its extreme, if $\rho$ = 1, the network will create a new category for every unique input pattern.

When using the ART algorithm as a middle layer in our system we only use the pure 3D features of the protein, that is the first 416 attributes of the protein 3D descriptor, as input to the ART neural network. In this way we reduce the dimensionality of the vector and reduce the time and resources needed for training of the classification algorithm.

### 2.3. Classification Strategy

The C4.5 classification algorithm uses the concept of entropy as follows. Suppose that we have a candidate split S, which partitions the training data set T into several subsets T1, T2, . . . , Tk . The mean information requirement can then be calculated as the weighted sum of the entropies for the individual subsets. We then define our information gain to be the increase in information produced by partitioning the training data T according to this candidate split S. At each decision node, C4.5 chooses the optimal split to be the split that has the greatest information gain.

The boosting method combines multiple models by explicitly seeking models that complement one another. Boosting encourages new models to become experts for instances handled incorrectly by earlier ones. Additionally boosting weights a model's contribution by its performance.

Figure 4 depicts the classification strategy. The general idea is to first classify proteins at the class level, grouping them based on global features. Once this partitioning is complete, the resulting subsets (each subset corresponds to one class) we subdivide further, classifying each protein by fold. The intent of this step is to improve accuracy by using a increasingly fine grained classification model, separating the data based on more local, fold-specific attributes than a typical decision tree that is forced to distinguish between all possible classes. The same step is applied for the superfamily level.

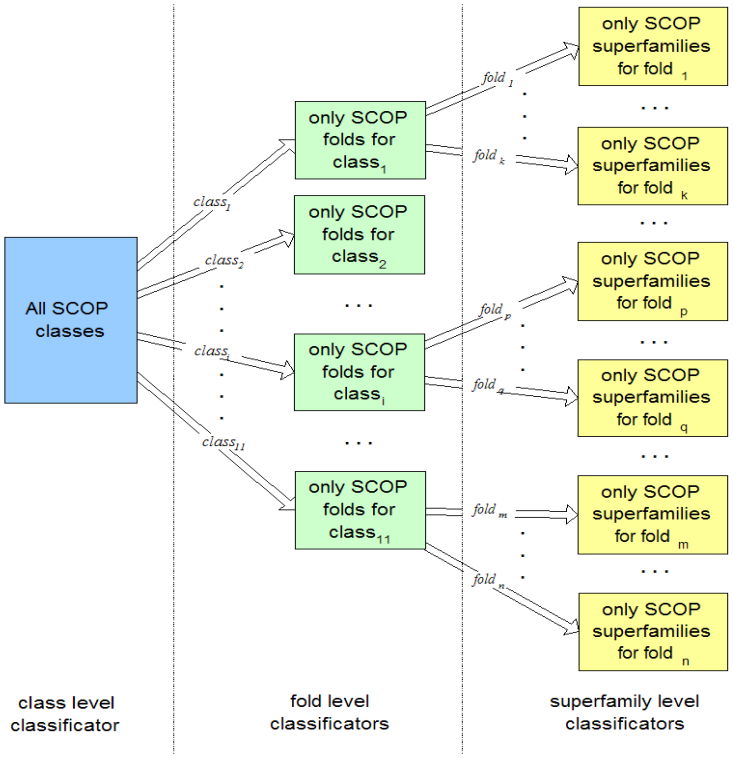


**Figure 4.** *Multi-level classification strategy*

## 3. Experiments and Evaluation Results

We have implemented a system for protein classification. The system is built on Microsoft Visual Studio.NET 2005, while the data is stored in a SQL Server 2005 database. The database contains 6873 proteins and is a representative sample of the SCOP database in which each SCOP hierarchy is represented in approximately the same proportion as in the whole SCOP database.

We have performed several experiments using different datasets for training the classification algorithm, but also a comparison of the classification performance of the system with and without using structural features in the protein descriptor and with and without using the ART algorithm as a middle layer.

When using the ART algorithm as a middle layer the vigilance parameter is set to 10%. This value was obtained empirically as the optimal taking into account the classification accuracy and complexity. Using this parameter we reduce the descriptor vector length from 416 (this refers only to the pure 3D features) to 56. Using lower values produces big information loss, and higher values increase the dimensionality both of which are unjustifiable for the system performance.

The Boosted C4.5 algorithm is evaluated using 10 fold cross-validation. The minimum number of objects per leaf, which reflects in pre-pruning, is set to 2. The confidence factor, which reflects in post-pruning, is set to 0.25.

| Hierarchy level | Training set size | Correctly classified | Correctly classified (%) | Correctly classified (%) (using ART) |
|---|---|---|---|---|
| *Class* | 6873 | 2770 | 45.30% | 47.04% |
| *Fold* | 6873 | 1710 | 29.88% | 30.76% |
| *Super Family* | 6873 | 3300 | 53.01% | 55.9% |

**Table 2.** Classification results on protein descriptor without biological features, with and without using ART

Table 2 shows the results when using a training set consisted of 6873 proteins, i.e. the whole database. The protein descriptor is consisted of 416 attributes representing its 3D structure, but without the additional structural features. The results indicate that using only the pure 3D structure information is not enough to perform a precise classification. When using the ART as a middle layer the classification precision increases, that is due to the fact that more information can be taken into account when the decision tree splits on a given attribute. Also the proportion of attributes (in terms of the whole vector) present in the C4.5 tree increases. Considering the building of the decision trees and their evaluation, when using ART, the time needed is reduced by 3 times.

| Hierarchy level | Training set size | Correctly classified | Correctly classified (%) | Correctly classified (%) (using ART) |
|---|---|---|---|---|
| *Class* | 869 | 752 | 87.54% | 89.64% |
| *Fold* | 869 | 556 | 68.98% | 73.19% |
| *Super Family* | 869 | 758 | 88.23% | 93.09% |

**Table 3.** *Classification results on protein descriptor with biological features, with and without using ART*

Table 3 shows the results when using a training set consisted of 869 proteins. The dataset is resampled from the whole database with the corresponding class attribute represented in approximately the same proportions as in the whole database. The protein descriptor is consisted of 450 attributes, including the additional structural features. We use a smaller dataset because the time needed for classification when ART is not used, exceeds several hours. With the additional information integrated in the descriptor we get much higher precision, that is especially evident in the case of Fold Hierarchy level classification. That is due to the fact that the additional attributes bring more suitable information for the aim of classifying a protein in a SCOP class hierarchy. Once again the usage of ART as a middle layer, outperforms the standard classification both in precision and in time.

| Hierarchy level | Tree size | Number of leaves | Tree size (using ART) | Number of leaves (using ART) |
|---|---|---|---|---|
| *Class* | 118 | 66 | 105 | 60 |
| *Fold (flat)* | 242 | 123 | 225 | 98 |
| *Fold (multi-level)* | 35 | 18 | 32 | 15 |
| *Super Family (flat)* | 418 | 240 | 380 | 207 |
| *Super Family (multi-level)* | 30 | 16 | 28 | 14 |

**Table 4.** *Average decision tree size and number of leaves for each of the classification variations tested*

Using a separate decision tree for each level of the SCOP hierarchy predicting all possible outcomes for the level is referred to as flat classification. Table 4

clearly shows the advantage of using a multi-level classification scheme compared to a flat classification scheme. Smaller decision trees have higher accuracy than larger trees. Locally optimizing information tends to produce small, shallow, accurate trees. By using the multi-level scheme we overcome the large decision trees problems like noise, fragmentation and subtree replication. When ART is applied the trees are even smaller and more accurate.

## 4. Conclusion

We have presented a system for protein molecules classification by using information both about their 3D structure and biological properties. We have applied the voxel-based method for generating geometry descriptor. Additionally, characteristic attributes of the primary and secondary structure of the protein molecules were extracted, forming attribute-based descriptor vectors.

The dimensionality of the produced descriptors can be crucial for classification purposes. Therefore we have applied the ART algorithm for reducing the dimensionality of the descriptors, thus improving the performance of the system in both precision and computation time.

The multi-level strategy we present here is meant to be general. It can be applied to any domain where the data falls into a natural hierarchy (or one where such a hierarchy can be readily deduced). In addition, any classification strategy can be used at the different levels of the hierarchy. If a certain method is found to be more effective at one stage, it can be used there and replaced with something else at the others.

The SCOP database, which provides a hierarchical structural classification level of the proteins, was used to evaluate the classification. The results show that our system can achieve high precision for some levels of the SCOP hierarchy (over 93% for the Super Family level). Even the lowest result of approximately 70% precision for the Fold level is satisfactory.

Our future work will be concentrated on increasing the precision of the classification by using hierarchical multi-label classification decision trees that will be able to classify the whole SCOP hierarchy at once. Also, we will investigate new 3D descriptors and incorporate additional characteristics in the descriptors.

## REFERENCES

[1] H.M. Berman, J. Westbrook, Z. Feng G. Gilliland, T.N. Bhat, H. Weissig, I.N. Shindyalov, and P.E. Bourne, The Protein Data Bank, *Nucleic Acids Research, vol. 28*, 2000, 235-242.

[2] A. G. Murzin, S. E. Brenner, T. Hubbard, C. Chothia, SCOP: a structural classification of proteins database for the investigation of sequences and structures. *J. Mol. Biol. 247*, 1995, 536-540.

[3] A. Andreeva, D. Howorth, J.-M. Chandonia, S.E. Brenner, T.J.P. Hubbard, C. Chothia, A.G. Murzin, Data growth and its impact on the SCOP database: new developments, *Nucleic Acids Research, vol. 36*, 2008, 419-425.

[4] D. V. Vranic, 3D Model Retrieval (Ph.D. Thesis, University of Leipzig, 2004)

[5] P. Daras, D. Zarpalas, A. Axenopoulos, D. Tzovaras, M. G. Strintzis, Three-Dimensional Shape-Structure Comparison Method for Protein Classification, *IEEE/ACM Transactions on computational biology and bioinformatics, Vol. 3, No. 3*, 2006, 193-207.

[6] S. Grossberg, *Adaptive Resonance Theory in Encyclopedia Of Cognitive Science* (Macmillan Reference Ltd, 2000).

[7] G.A. Carpenter, S. Grossberg, D.B. Rosen , Fuzzy ART: Fast stable learning and categorization of analog patterns by an adaptive resonance system, *Neural Networks, vol. 4*, 1991, 759-771.

[8] S. K. Murthy, Automatic construction of decision trees from data: A multi-disciplinary survey, *DMKD*, vol. 2, no. 4, pp. 345–389, 1998.

[9] L. A. Breslow and D. W. Aha, Simplifying decision trees: A survey, NCARAI, Tech. Rep. AOC-96-014, 1996.