A Parallel Data Mining Methodology for Protein Function Prediction Utilizing Finite State Automata

Christos N. Gkekas
Student Member IEEE
Department of Electrical and Computer Engineering
Aristotle University of Thessaloniki
email: chggr@ee.auth.gr

Fotis E. Psomopoulos
Member IEEE
Department of Electrical and Computer Engineering
Aristotle University of Thessaloniki
email: fpsom@danae.ee.auth.gr

Pericles A. Mitkas
Senior Member IEEE
Department of Electrical and Computer Engineering
Aristotle University of Thessaloniki
email: mitkas@eng.auth.gr

***Abstract*—One of the most important challenges in modern bioinformatics is the accurate prediction of the functional behaviour of proteins. The strong correlation that exists between the properties of a protein and its motif sequence makes such a prediction possible. In this paper a novel parallel methodology for protein function prediction will be presented. Data mining techniques are employed in order to construct a model for each Gene Ontology term, based on data generated from already annotated protein sequences. In order to predict the annotation of an unknown protein, its motif sequence is run through each GO term model, producing similarity scores for every term. Although it has been experimentally proven that this process is efficient, it unfortunately requires heavy processor resources. In order to address this issue, a parallel application has been implemented and tested using the EGEE Grid infrastructure.**

I. Introduction

Proteins are large organic compounds consisting of amino acids arranged in a linear chain and joined together by peptide bonds. The sequence of amino acids in a protein is defined by a gene and encoded in the genetic code of every living organism. Proteins participate in every process withing cells and thus they are essential parts of all organisms. Some of them are enzymes that catalyze biochemical reactions, whereas others have various structural or mechanical functions. Moreover, a lot of proteins are involved in cell signaling, cell adhesion, immune responses and the cell cycle. Finally, it is also very common for proteins to work together in order to achieve a particular function, and they often associate to form stable complexes.

A protein family, on the other hand, is a group of evolutionarily and/or functionally related proteins. To the best of our knowledge, there is an ongoing worldwide effort to organize proteins into families and describe their component domains and motifs. Reliable identification of protein families is key to phylogenetic analysis, functional annotation and the exploration of diversity of protein function in a given phylogenetic branch. It is a fact that all proteins in the same family share strong structure similarities and therefore exhibit similar behavior. Until recently, the biological properties of proteins had to be experimentally determined by means of costly and time-consuming in vitro methods. Bioinformatics on the other hand uses computational methods to address such problems. The algorithmic means for establishing protein families on a large scale are based on a notion of similarity. Most of the time the only similarity we have access to is sequence similarity.

One of the most recent tools for protein function annotation is the Gene Ontology Project [1]. This project aims at providing a controlled vocabulary to describe gene and gene product attributes in organisms. In order to assign Gene Ontology terms to new non-annotated protein sequences, they have to be either processed directly in a lab or characterised through similarity to already annotated sequences. At the moment, the amino acid sequence of more than 1.000.000 proteins has been obtained. On the contrary, the properties and functions of only 4% of these proteins are known. Therefore, the need for a systematic way to derive clues for the properties of a protein by inspecting its amino acid sequence is obvious.

In this paper a novel methodology will be presented, which uses the motifs present in already annotated protein sequences in order to model the corresponding Gene Ontology terms. The models created can then be used to predict the annotation of new protein sequences. In more detail, the motif sequence of the protein under examination is extracted and run through all available Gene Ontology term models. This process generates similarity scores, which constitute an accurate prediction of the protein's annotation.

The main drawback of this methodology is that it requires a substantial amount of computational time to complete. It has been shown experimentally that the execution time needed to process the entire dataset on a single processor is prohibitively long. In order to address this problem, we have implemented it both as a standalone and as a grid-based application. The grid-based application utilizes the MPI [2] library for communication between distinct processes and employs the EGEE Grid [3] Infrastructure in order to shorten the execution time. Moreover, the Grid provides for the seamless integration of the training process and the actual model evaluation by allowing the concurrent retraining of Gene Ontology models from different input sources or experts and the use of the existing ones. This methodology can be easily generalized to produce models for different protein classification schemes, such as SCOP [4] families etc.

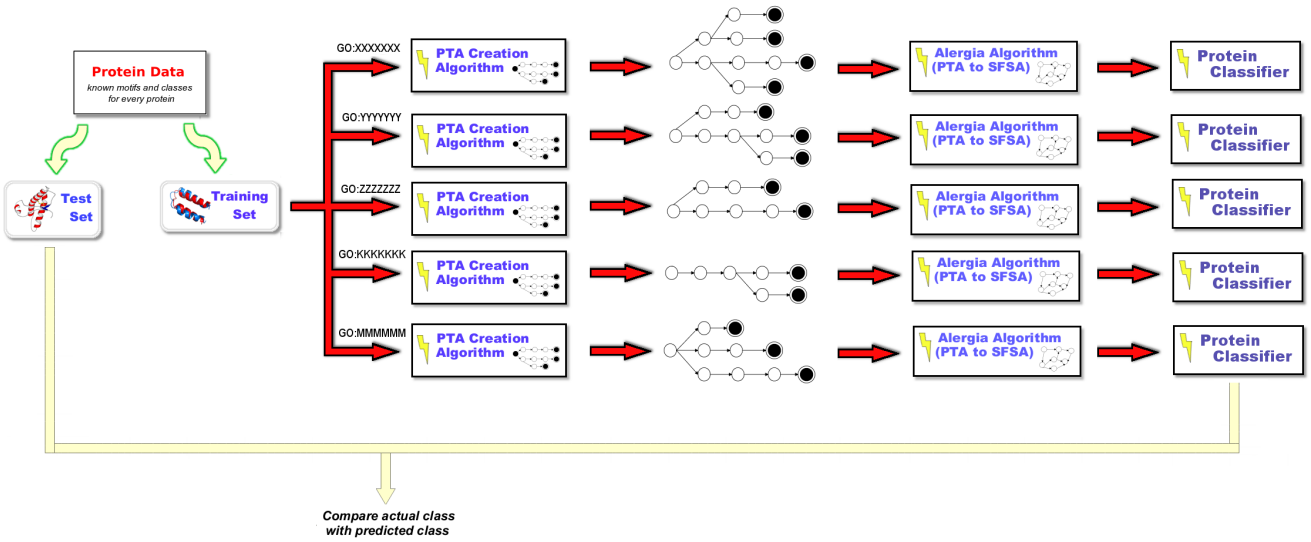

Fig. 1. The methodology outline.

The results obtained so far seem to be rather encouraging. The application was executed on available clusters using from 4 to 32 processors in various experiment configurations. In all cases the accuracy of the results was very high and the overall execution time was satisfactory.

The main difference of our methodology compared to algorithms proposed by both the artificial intelligence community and the pattern recognition community is that it utilizes Finite State Automata to solve the problem of protein function prediction. On the contrary, other algorithms use statistical models [5], neural networks [6], or decision trees [7], [8]. Moreover, there are also some algorithms in the literature that use Finite State Automata to address the aforementioned problem [9], [10], [11]. The main difference of those algorithms to the one presented in this paper is that it allows the creation of Gene Ontology term models and the classification of proteins to be performed in parallel. Due to its parallel nature, the algorithm can be deployed over computer clusters and shorten the execution time significantly.

The remainder of this paper is structured as follows: Section II is entitled "Methodology Outline" and contains a brief outline of the proposed methodology. In Section III: "Data Preprocessing" a detailed description of the preprocessing of protein data will be provided. Next, in Section III, "Training Process", the training process and the algorithms used in it will be illustrated in full detail. Furthermore, in Section IV: "Experiments", the results of the experiments conducted will be presented.Finally, in Section V: "Conlusions", useful conclusions about the accuracy of the methodology will be drawn and our plans for future work will be introduced.

II. Methodology Outline

The first step of the methodology presented in this paper is to obtain the motifs present in already annotated protein sequences. This is performed using the UNIPROT [12] code of each protein and the InterProScan [14] tool, taking into consideration every available sequence database (e.g. PRODOM, PROFILE, PFAM, etc). Through this process the initial protein data set is constructed. This data set contains for every protein not only its motif sequence, but also the Gene Ontology terms that have been assigned to it. The acquired protein data is then divided into two sets: the Training Set (DS_{train}) and the Test Set (DS_{test}). It is very important to note that these two sets are disjoint:

$$DS_{train} \cap DS_{test} = \emptyset$$

As the name implies, the Training Set is provided as input to the data mining algorithms the methodology uses in order to create a model for every Gene Ontology term. On the contrary, the Test Set is used in order to test the methodology's accuracy and effectiveness.

In the next step, the Training Set is further devided into small protein data sets. In more detail, every one of these small protein data sets consists only of protein sequences that have been annotated with a certain Gene Ontology term. Therefore, for every available Gene Ontology term, its corresponding data set is constructed. This protein data set is part of the initial Training Set and will be later used as input to data mining algorithms in order to derive a model for every Gene Ontology term.

The protein data sets created during the previous step are then processed indepedently in parallel. For every one of them, a Prefix Tree Acceptor is constructed using the motif sequence of the proteins in the set. Utilizing the Alergia algorithm [13], this PTA is consequently transformed into a more generalized Stochastic Finite State Automaton (SFSA), which is considered to be a model for the corresponding Gene Ontology term. Therefore, through this parallel process, one model for each Gene Ontology term is derived.

The final step of this methodology involves the utilization of the previously constructed Gene Ontology models in order to predict the annotation of an unknown protein. For this purpose, the motif sequence of the protein under examination is extracted and run through every Gene Ontology term model. This process produces similarity scores for every Gene Ontology term, which altogether constitute an accurate prediction of the protein's annotation. Moreover, the classification efficiency of this methodology can be obtained by applying this methodology to the Test Set and then comparing the predicted annotation with the actual one.

III. Data Preprocessing

The initial Protein Data Set that will be used as input to the methodology consists of a great number of protein files. Each one of them holds the following information: the motif sequence of the protein and its annotation. These files are actually XML files that were obtained using the InterProScan [14] tool, taking into consideration every available sequence database, such as PROSITE [15], PFAM [16], and PRINTS [17] etc.

In genetics, a sequence motif is an amino-acid sequence pattern that has, or is conjectured to have, a biological significance. The occurence of motifs in the protein chain enables scientists to predict protein functionality. Motifs are thought to be the main factors determining the behavior of the protein. At this point it must be noted that there is not always a direct association between motifs and properties because the final properties of the protein are a function of many motifs, with some overpowering or amplifying others. Protein motifs can be divided into two main categories: patterns and profiles. Patterns constitute the simplest form of motifs and they are widespread amino-acid sequences. On the other hand profiles are Hidden Markov Models (HMMs). The main difference between patterns and profiles is that a search for the first returns a binary TRUE/FALSE value whereas a search for the second returns a similarity score.

The initial Protein Data Set is subsequently divided into two disjoint sets: the Training Set and the Test Set. This process is performed in a totally random manner, so that the homogeny between the initial Protein Data Set and the produced data sets can be preserved. The Training and Test Sets must be homogenous, which means that all Gene Ontology terms have to be equally represented. If the split of the initial Protein Data Set is not performed in a totally random manner, then some Gene Ontology terms will be overrepresented, while others will be misrepresented. This will render the models created for the misrepresented Gene Ontology terms deficient and therefore the methodology's accuracy will be severely limited.

The Training Set and the Test set have to be disjoint. This term means that a protein file cannot at the same time belong to both of these sets. It is important to ensure that these two sets are disjoint, because otherwise the accuracy of the proposed methodology cannot be precisely measured. Moreover, it is senseless to try to classify a protein which has already been utilized in order to derive the models used in the classification procedure. Such an experiment would definately produce excellent results, but those results would overestimate the accuracy and effectiveness of the algorithm, thus leading to false conclusions.

The Training Set is utilized in order to create the Gene Ontology term models. For the construction of every Gene Ontology term model, only the proteins that have been annotated with the specific Gene Ontology term must be taken under consideration. Therefore, the Training Set is further divided into multiple data sets. Each one of these data sets contains proteins that have one common characteristic: they are annotated with the same Gene Ontology term. Data mining algorithms will be applied to these data sets in order to derive the required Gene Ontology term models. Due to the fact that one protein can be annotated with more than one Gene Ontology term, each protein can be part of more than one data sets.

The Test Set is used to test the accuracy of the models created by the methodology. Once the Gene Ontology term models have been successfully constructed, they can be used to classify new proteins. In more detail, for every protein in the Test Set, its motif sequence is extracted and run through each Gene Ontology term model, thus producing similarity scores for every term. These scores consitute a prediction of the protein's annotation. By applying thresholds it is possible to obtain only those Gene Ontology terms that have the highest probability of being correct. If the predicted annotation for every protein in the Test Set is compared to the actual one, statistics about the accuracy of the methodology can be generated and used to draw valuable conclusions.

IV. Training Process

The training process accepts as input the data set for every Gene Ontology term and employs data mining algorithms in order to derive a model for each one. At first, all proteins in the data set are used to construct a Prefix Tree Acceptor (PTA). This PTA is later transformed into a more generalized Finite State Automaton (FSA) by merging some of its states that comply to certain criteria. The produced FSA constitutes a model for the corresponding Gene Ontology term.

The key characteristic of the training process is its parallel nature. As it is clearly illustrated on Fig. 1, the construction of each Gene Ontology term model is completely independent from the construction of the other ones. Therefore the whole training process can be easily parallelized, allowing for further reduction of the execution time. The implemented application creates one master process and many slave processes. The master process is used to coordinate the behavior of all slave processes and assign jobs to them accordingly. On the other hand, every slave process has the obligation to listen to the master process, receive jobs, execute them and return the results back to the master process. This scema is very flexible and is able to take advantage of all available CPUs in order to complete the task as fast as possible.

A. PTA Construction

The first step of training process is the construction of a Prefix Tree Acceptor using all protein sequences found in the data set. In more detail, for every protein in the data set, its motif sequence is extracted and inserted into the PTA. After all sequences have been taken into consideration, the PTA is considered to be complete and the methodology advances to the second step: FSA construction.

The function of the PTA construction algorithm can be more clearly illustrated using an example. The set of proteins that will be used comprizes four proteins with motif sequences 'xyx', 'yxx', 'xxy' and 'xyy'. At the beginning, a start state is created, which is shown with black color at the left side of Fig. 2. Based on the motif sequence of the first protein, the first branch of the PTA is constructed. This branch consists of the start state and transisions between states according to the motif sequence of the protein that was inserted. The final state of the branch is an end state. Using the same procedure as above, the second branch can be attached to the PTA according to the motif sequence of the second protein on the data set. The insertion of the third motif sequence is slightly more complex: there is already a transision from the start state using motif x. So the algorithm uses this transision and a new branch is added as shown in Fig. 2. Finally, while inserting the motif sequence ot the fourth protein, the algorithm realizes that there are already two transisions using motifs x and y. Therefore it uses these transisions and adds only one transision with motif y and, of course, the end state. This is the end of the PTA construction using the set of these four proteins.

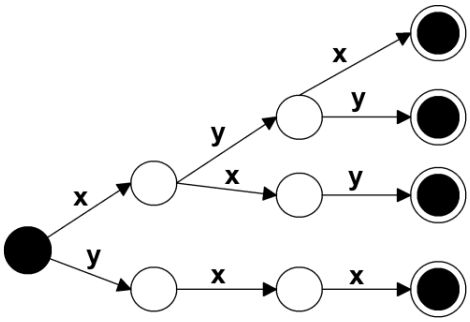

Fig. 2. An elementary PTA.

It is intresting to point out that the constructed PTA does not depend on the order by which the motif sequences where considered. If this order is changed in any way, the same PTA will be generated.

As shown in Fig. 1, the PTA creation procedure has been implemented in parallel. The Training Set consists of multiple XML files that contain protein data. These files are utilized by parallel instances of the PTA creation algorithm in order to construct a series of PTAs, one for every Gene Ontology term. Due to the parallel nature of this process, the PTAs can be delivered by the application in a very short period of time.

B. FSA Construction

The PTA created in the previous step will now be transformed into a more generalized FSA. This transformation is performed using the Alergia algorithm proposed by R.C. Carrasco and J. Oncina [13]. This algorithm is used in order to merge all statistically equivalent states. Two states are considered to be statistically equivalent when they have the same transition probabilities for every symbol and their next states are also equivalent. This brief description of the Alergia algorithm actually implies that it is a recursive algorithm. Therefore one may expect that the execution time will grow exponentially. Experimentally though it has been proven that the time complexity is linear and mainly depends on the size of the data set used.

In order to calculated the probabilities required by the Alergia algorithm, for every state the number of motif sequences that have arrived and ended there must be known. In addition to that, for every transition the number of motif sequences that have used it must be known. All these parameters can be easily calculated during the construction of each PTA. A more detailed description of the PTA to FSA transformation can be found in [11].

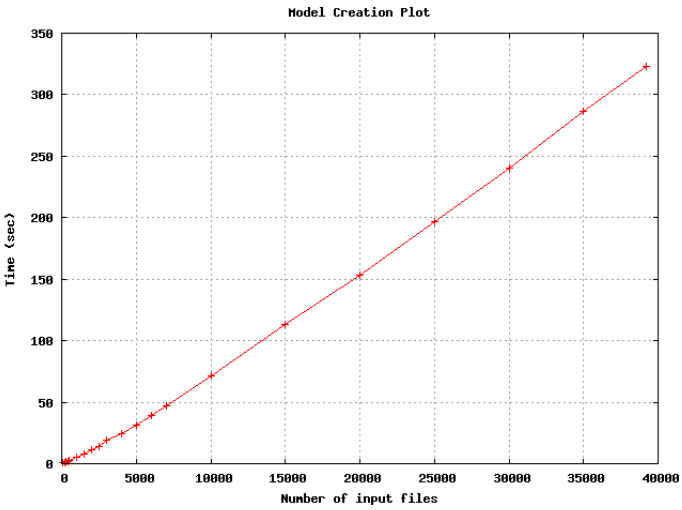

Fig. 3. Model creation plot.

In Fig. 3 a diagram of the time needed to construct all Gene Ontology term models for various Training Set sizes is presented. The x axis indicates the number of files in the Training Set that were used as input to the PTA constuction algorithm and the y axis represents the time needed to fully construct the models. Of cource, the model construction procedure involves both the PTA creation and the PTA to FSA transformation. A series of experiments was conducted using only one processor and the mean execution time for each Training Set size is presented. This plot provides experimental evidence for what was previously stated that the time needed to construct all Gene Ontology term models varies linearly according to the size of the Training Set.

Another equally important deduction is that the training

process can be executed prety fast: for approximatelly 40.000 files in the Training set, only 322 seconds or 5.4 minutes were required in order to construct all Gene Ontology term models. Therefore one can assume that there is no need to use multiple processors for this part of the methodology, exactly due to the fact that the training process is very time effective. In fact, using multiple processors can make the situation worse instead of improving it, because a substantial amount of time is needed for the communication (i.e. exchange of messages) between the various processes.

V. Experiments

The final stage of the proposed methodology is the utilization of all previously generated Gene Ontology term models in order to assign Gene Ontology terms to new non-annotated protein sequences. This is the point where proteins from the test set are used in order to obtain the accuracy of the methodology.

The accuracy can be defined as the number of proteins whose annotation was correctly predicted over the total number of proteins in the test set. In more detail, if **A**, **B**, **C** and **D** are the Gene Ontology terms that constitute the annotation of a protein in the Test Set and the methodology predicts that **B**, **C**, **D**, **E** and **F** are the most possible terms, then the accuracy is defined as the number of correctly predicted terms over the total number of terms predicted. In this case, 3/5 = 0.6 or 60%. One other way to calculate accuracy is to divide the number of correctly predicted terms over the total number of terms of the protein. In this case the result would be 3/4 = 0.75 = 75%. The difference between those two methods of calculating accuracy is that in the second case no missclassification penalty exists. Therefore the first way of calculating accuracy is preferred:

$$Accuracy = \frac{Correct\ Terms\ Predicted}{All\ Terms\ Predicted}$$

The overal accuracy of the methodology can be estimated if one adds up the accuracy that was calculated for every protein and divides the result with the total number of proteins in the Training Set. Namely:

$$Total\ Accuracy = \frac{\sum_{i=0}^{N} Accuracy\ for\ each\ protein}{N}$$

where N is the total number of proteins in the Training Set.

The first phase of the classification procedure is to acquire the motif sequence of the protein under examination. Next, all possible permutations of this motif sequence are evaluated. If a motif sequence consists of *n* motifs, then the number of permutations will be *n!*. For every Gene Ontology term model, every one of the previously evaluated permutations is run through it and a probability value is calculated. A detailed description and mathematical proof of the methodology used to calculate the probability that a certain motif sequence appears inside a Finite State Automaton can be found in [18]. At the end, the permutation with the maximum probability is selected and presented to the user. If all permutations generate probability values that are bellow a certain threshold, then the algorithm rejects the specific Gene Ontology term.

For every prediction the methodology makes, there exist two very important parameters that indicate its quality. Using thresholds for these parameters, one can obtain only strong and interesting predictions. The first parameter is called "support" and it is calculated if one divides the number of proteins in the training set whose motif sequence matches the motif sequence of the protein under examination to the total number of proteins in the Training Set. Therefore:

$$Support = \frac{Maching\ Proteins}{Total\ Proteins}$$

It is obvious that the support parameter is a percentage value and that it actually counts the frequency of the appearance of a specific pattern in the motif sequence of all proteins in the Training Set. The second parameter is "confidence" and it is calculated if one divides the number of proteins in the Training Set whose motif sequence matches the motif sequence of the protein under examination and they are annotated with the same GO term to the total number of proteins in the training set that are annotated with the specific GO term. Thus:

$$Confidence = \frac{Maching\ Proteins\ with\ same\ GO}{Total\ Proteins\ with\ same\ GO}$$

This parameter is also a percentage value and it can be though of as the number of instances this prediction is correct expressed as a portion of all instances it applies to. It is actually a measure of the prediction's weight and therefore it can be used to refine the results, keeping only those that have adequate weight.

A safisfying value for accuracy can be obtain if one imposes certain thresholds on the values of the aforementioned parameters. These thresholds filter the results of the classification procedure and at the end only strong and interesting predictions are preserved. It has been shown experimentally that the optimum threshold values are:

Probability threshold = 0.1
Support threshold = 0
Confidence threshold = 0.1

All experiments conducted indicate that these threshold values produce the most accurate results. Therefore in the remainder of this paper, the usage of these values is assumed.

In order to validate the correctness of the proposed methodology and obtain a general picture of its efficiency and accuracy, a series of experiments were conducted. In the following paragraphs, the experimental procedure will be explained and the results of the experiments will be presented. These experiments involved the usage of Data Sets with sizes of 10.000, 20.000, 30.000 and 39.200 protein files, which were divided into Training and Test Sets according to the following ratios (Training Set / Test Set): 90/10, 20/80, 30/70, 40/60, 60/40, 70/30, 80/20 and 90/10. Moreover, a variable number number of CPUs was used (1, 4, 8, 16 and 32 CPUs).

VI. Conclusions

In this paper, a parallel data mining methodology for protein function prediction was presented. This methodology applies data mining techniques to protein data from already annotated

protein sequences in order to construct a model for every Gene Ontology term. This model is actually a Finite State Automaton, which can then be used in order to predict the annotation of new non-annotated protein sequences. The parallel nature of this methodology has allowed the creation of an MPI-enabled application that can be easily deployed over a Computer Grid, leading to great reduction of the execution time.

The experiments conducted using the EGEE Grid environment as a source of multiple CPUs clearly indicate that the methodology is highly efficient and accurate. Moreover, the utilization of many processors has reduced the execution time substantially.

REFERENCES

[1] The Gene Ontology Project (http://www.geneontology.org)
[2] Message Passing Interface (http://www-unix.mcs.anl.gov/mpi/)
[3] Enabling Grids for E-sciencE (http://www.eu-egee.org)
[4] Structural Classification of Proteins (http://scop.mrc-lmb.cam.ac.uk/scop/)
[5] R. Duad, P. Hart, *Pattern Classification and Scene Analysis*, Wiley, New York, 1973.
[6] C. Bishop, *Neural Networks for Pattern Recognition*, Oxford University Press, New York, 1995.
[7] D. Wang, X. Wang, V. Honavar, D. Dobbs, Data-driven generation of decision trees for motif-based assignment of protein sequnces to functional families, In: *Proceedings of the Atlantic Symposium on Computational Biology*, Genome Information Systems & Technology, 2001.
[8] J. R. Quilan, *Programs for Machine Learning*, Morgan Kaufmann, San Mateo, CA, 1992.
[9] F. Psomopoulos, S. Diplaris, P. A. Mitkas, A Finite State Automata Based Technique for Protein Classification Rules Induction, In: *Proceedings of the Second European Conference on Data Mining and Text Mining in Bioinformatics*, ECML/PKDD, 2004
[10] F. E. Psomopoulos, P. A. Mitkas, A protein classification engine based on stochastic finite state automata, *Lecture Series on Computer and Computational Sciences*, Vol. 1, 2005.
[11] F. E. Psomopoulos, P. A. Mitkas, PROTEAS: A Finite State Automata based data mining algorithm for rule extraction in protein classification, *5th Hellenic Data Management Symposium*, 2006
[12] Uniprot - Universal Protein Resource (http://www.ebi.uniprot.org)
[13] R. C. Carrasco, J. Oncina, *Learning stochastic regular grammar by means of state merging method*, Proc. The Second International Colloquium on Grammatical Inference (ICGI '94), Alicante, Spain, Lecture Notes in Artificial Intelligence LNAI 862, pp. 139 - 152, Springer - Verlag, 1994.
[14] InterProScan Sequence Search (http://www.ebi.ac.uk/InterProScan)
[15] PROSITE (http://ca.expasy.org/prosite/)
[16] Pfam (http://pfam.sanger.ac.uk/)
[17] PRINTS (http://www.bioinf.manchester.ac.uk/dbbrowser/PRINTS/)
[18] P. Hingston, "Using Finite State Automata for Sequence Mining", 25th Australian Computer Science Conference, Melbourne, Australia, 105-110, 2002.