Brill Academic Publishers
P.O. Box 9000, 2300 PA Leiden
The Netherlands

Lecture Series on Computer
and Computational Sciences
Volume 1, 2005, pp. 1-3

A protein classification engine based on stochastic finite state automata

F.E. Psomopoulos[1] and P.A. Mitkas[2]

Dept. Electrical and Computer Engineering,
Aristotle University of Thessaloniki,
GR-541 24 Thessaloniki, Greece

and

Informatics and Telematics Institute,
Centre for Research and Technology Hellas,
GR-570 01 Thessaloniki, Greece

Received 3 August, 2005; accepted in revised form 10 August, 2005

Abstract: Accurate protein classification is one of the major challenges in modern bioinformatics. Motifs that exist in the protein chain can make such a classification possible. A plethora of algorithms to address this problem have been proposed by both the artificial intelligence and the pattern recognition communities. In this paper, a data mining methodology for classification rules induction in proposed. Initially, expert – based protein families are processed to create a new hybrid set of families. Then, a prefix tree acceptor is created from the motifs in the protein chains, and subsequently transformed into a stochastic finite state automaton using the ALERGIA algorithm. Finally, an algorithm is presented for the extraction of classification rules from the automaton.

Keywords: protein classification, data mining, bioinformatics, motifs, finite state automata

Mathematics Subject Classification: 68T30, 68Q45, 68R10

PACS: 87.15.Cc, 87.14.Ee

1. Protein Classification

Protein classification is currently one of the most interesting problems in computational biology. The biological action of proteins is traditionally identified by time consuming and expensive in-vitro experiments. However, recent developments in bioinformatics have enabled the use of computational tools and techniques towards this end [1]. Clustering algorithms [2], artificial neural networks [3], decision trees [4, 5] and statistical models [6] are few of the methods currently employed. *Motifs* that can be found in a protein chain have provided a higher level of abstraction to the problem, since protein properties are mainly defined by them. Motifs can be either *patterns*, which are short aminoacid chains with a specific order, or *profiles*, which are computational representations of multiple sequence alignments derived by the use of hidden Markov models.

On the other hand, proteins can be assorted into families, each family containing proteins with similar functions. Those families can either be *expert–based*, meaning that they have been experimentally specified and their significance is biologically meaningful, or *computer–generated*, meaning that they have been created with the use of unsupervised protein classification algorithms. In the latter case the major disadvantage is that the protein classes will not necessarily have any biological meaning or significance. In the former, expert – based classes are often overlapping, thus adding to the complexity of the classification algorithms.

[1] E-mail: fpsom@auth.gr
[2] E-mail: mitkas@eng.auth.gr

2. Core Engine

In [7] a technique for extraction of classification rules using finite state automata was introduced, the classes being expert – based protein families. The technique is outlined in Figure 1. First, the training set is constructed from a set of known proteins. The next step is the creation of the Prefix Tree Acceptor (*PTA*) using the protein chains of the train set. Using the ALERGIA algorithm [8], the PTA is converted to an equivalent Stochastic Finite State Automaton (*SFSA*), in which every transition is associated with a probability. Since the SFSA is a generalized representation of the protein structure, information can be obtained directly from it. As a result, certain probabilities can be calculated, such as the probability of a protein chain to contain a certain motif or a specific subset of motifs. Using these probabilities, rules can be extracted to better describe the form of the protein chains. In this paper we extend this technique by introducing both a hybrid form of classification and a new algorithm for the extraction of classification rules from the automata.

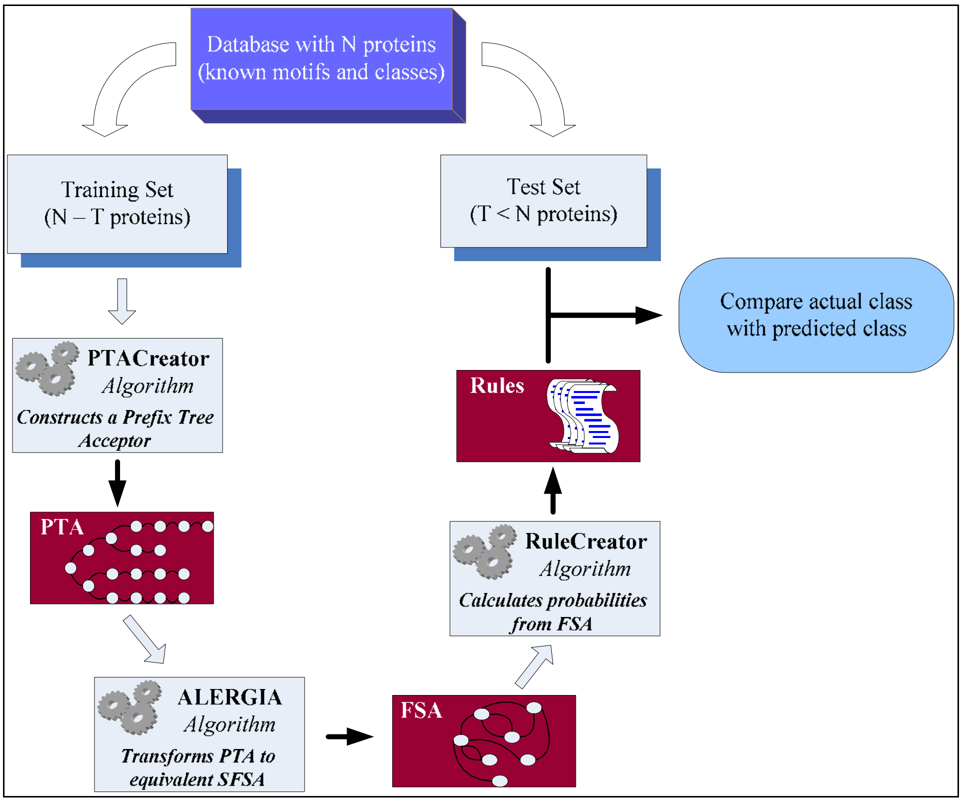
Database with N proteins (known motifs and classes)
Training Set (N – T proteins)
Test Set (T < N proteins)
Compare actual class with predicted class
PTACreator Algorithm Constructs a Prefix Tree Acceptor
Rules
PTA
RuleCreator Algorithm Calculates probabilities from FSA
ALERGIA Algorithm Transforms PTA to equivalent SFSA
FSA

Figure 1: FSA based technique for protein classification rules induction

3. Methodology Outline

The proposed methodology for rule induction consists of three sequential parts: a) Preprocessing, b) Model Creation and c) Rule Extraction.

During the Preprocessing phase, all the protein classes in the training set are recombined to create *Virtual Classes*. These constitute a hybrid form of the original partitioning, where the overlapping areas between classes are assigned to new classes, as shown in Figure 2. In this case, the classes that arise still maintain their biological meaning, but they also represent more closely the proteins they contain. On an algorithmic level, this step creates disjoint sets of proteins, which can be independently processed for classification rules.

For each ensuing Virtual Class, during the Model Creation phase, a SFSA is created using the algorithm described in [7], modified as follows: before the construction of the PTA, the motif list is transformed in order to move the most frequently appearing motifs at the beginning of each list. This transformation ensures that the order of appearance of the motifs in the lists will have no impact on the classification rules.

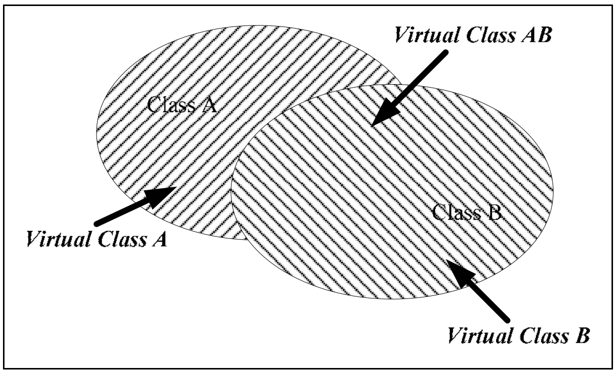

Figure 2: Virtual protein classes

The final step in the methodology is the extraction of classification rules from the SFSA. This is accomplished using the *Max_Probability_Path* algorithm, presented in Box 1. This algorithm is a modification of the shortest path: it finds the maximum probability path from the starting vertex to any other vertex in the SFSA. However, only the paths to the terminating vertices are important due to the fact that they define the conditions necessary for a protein to belong to a class. For each vertex the algorithm stores its predecessor, thus allowing tracing the path from the starting vertex to any other vertex in the SFSA.

```
algorithm Max_Probability_Path
input:
   A: Stochastic Finite State Automaton
output:
   p[]          : probability to reach each vertex from the starting vertex
   predecessor[]: the previous vertex following the max probability path
begin
   s: starting vertex in A
   for each vertex u ∈ A
      p[u] = 0;
   end for
   p[s] = 1;
   predecessor[s] = null;
   Q = Queue with all vertices in A;

   while(Q NOT empty)
      u = first vertex in Q;
      for (each vertex v that connects with u)
         if (p[u]*probability(u, v) > p[v])
            p[v] = p[u]*probability(u, v);
            remove vertex v from Q;
            predecessor[v] = u;
         end if
      end for
   end while

end algorithm
```

Box 1: Max_Probability_Path algorithm

The concept behind the algorithm is that the probability of any path from the starting node is equal to the product of the probabilities of the edges that comprise the path. A classification rule is derived from a path simply by reading the edge “types”, i.e. the motifs that exist in the path. The probability of the path is a quantitative measure of the interest of the rule: the higher the probability of the path the stronger the rule will be.

An overview of the complete methodology is presented in Figure 3.

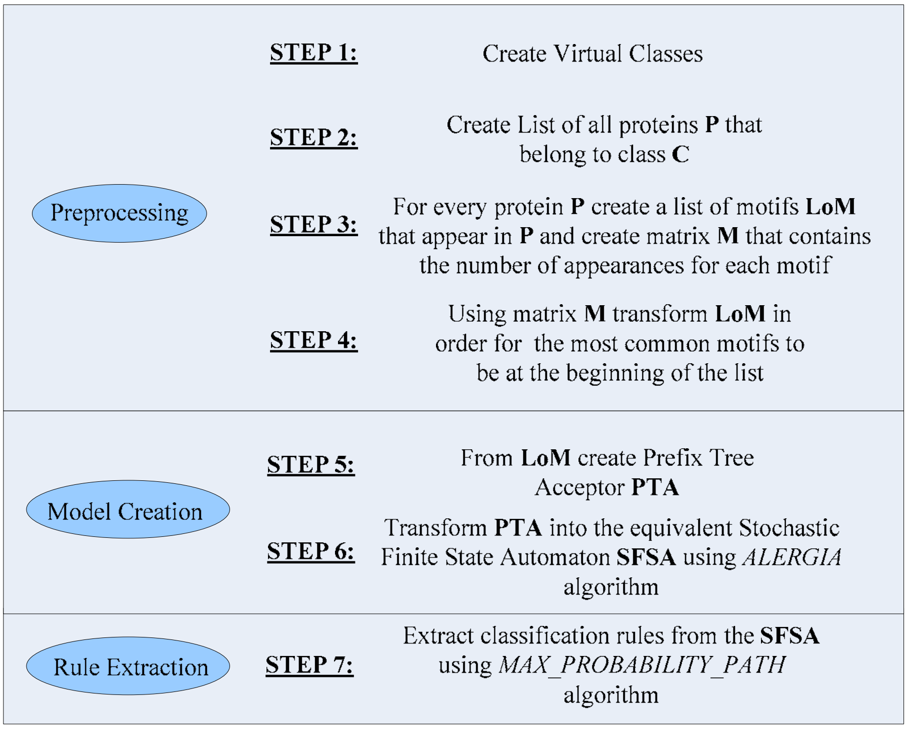
Preprocessing
STEP 1: Create Virtual Classes
STEP 2: Create List of all proteins **P** that belong to class **C**
STEP 3: For every protein **P** create a list of motifs **LoM** that appear in **P** and create matrix **M** that contains the number of appearances for each motif
STEP 4: Using matrix **M** transform **LoM** in order for the most common motifs to be at the beginning of the list
Model Creation
STEP 5: From **LoM** create Prefix Tree Acceptor **PTA**
STEP 6: Transform **PTA** into the equivalent Stochastic Finite State Automaton **SFSA** using *ALERGIA* algorithm
Rule Extraction
STEP 7: Extract classification rules from the **SFSA** using *MAX_PROBABILITY_PATH* algorithm

Figure 3: Methodology Outline

References

[1] P.F. Baldi, S. Brunak, *Bioinformatics: A Machine Learning Approach*, The MIT Press, Cambridge, MA, 2001.

[2] C. Makris, Y. Panagis, K. Perdikuri, E. Theodoridis, A. Tsakalidis, Algorithms for Protein Clustering, In Proceeding of the 2nd International Greek Biotechnology Forum, pp 38-44, July 1-3, 2005.

[3] C. Bishop, *Neural Networks for Pattern Recognition*, Oxford University Press, New York, 1995.

[4] D. Wang, X. Wang, V. Honavar, D. Dobbs, Data-driven generation of decision trees for motif-based assignment of protein sequences to functional families, *In: Proceedings of the Atlantic Symposium on Computational Biology*, Genome Information Systems & Technology, 2001.

[5] J.R. Quinlan, *Programs for Machine Learning*, Morgan Kaufmann, San Mateo, CA, 1992.

[6] R. Duad, P. Hart, *Pattern Classification and Scene Analysis*, Wiley, New York, 1973.

[7] F. Psomopoulos, S. Diplaris, P.A. Mitkas, A Finite State Automata Based Technique for Protein Classification Rules Induction, *In Proceedings of the Second European Workshop on Data Mining and Text Mining for Bioinformatics*, pp. 54-60, ECML/PKDD 2004, Pisa, Italy, September 20-24, 2004.

[8] R.C. Carrasco, J. Oncina, Learning Stochastic Regular Grammar by means of a State Merging Method, *In Proceedings of the Second International Colloquium on Grammatical Inference (ICGI '94)*, Alicante, Spain, Lecture Notes in Artificial Intelligence, pp 139-152, Springer – Verlag, 1994.