# An Evolutionary Approach for Motif Discovery and Transmembrane Protein Classification

Denise F. Tsunoda[1], Heitor S. Lopes[1], and Alex A. Freitas[2]

[1] Lab. Bioinformática, Centro Federal Educ. Tecnol. do Paraná, Curitiba, Brazil
dtsunoda@brturbo.com, hslopes@cpgei.cefetpr.br
[2] Computing Laboratory, University of Kent, Canterbury, UK
A.A.Freitas@ukc.ac.uk

**Abstract.** Proteins can be grouped into families according to their biological functions. This paper presents a system, named GAMBIT, which discovers motifs (particular sequences of amino acids) that occur very often in proteins of a given family but rarely occur in proteins of other families. These motifs are used to classify unknown proteins, that is, to predict their function by analyzing the primary structure. To search for motifs in proteins, we developed a GA with specially tailored operators for the problem. GAMBIT was compared with MEME, a web tool for finding motifs in the TransMembrane Protein DataBase. Motifs found by both methods were used to build a decision tree and classification rules, using, respectively, C4.5 and Prism algorithms. Motifs found by GAMBIT led to significantly better results, when compared with those found by MEME, using both classification algorithms.

## 1 Introduction

After unveiling the DNA sequence of an organism, researchers turn to the laborious task of annotation. Afterwards, the proteome of the organism is seen as one of the main products of genome sequencing projects. In recent years researchers have witnessed an exponential growth of biological databases, thanks to the many genome sequencing projects in the world.

Proteins are essential for life since they are responsible for most functions in an organism, such as: transport of small molecules (e.g., hemoglobin), regulation (e.g., insulin), sustentation (e.g., collagen), increase of reaction speed (e.g., enzymes) and others. Biological organisms have thousands of different types of proteins, which are constituted basically of amino acids linked in linear chains through peptide connections. Active intra-molecular forces cause the proteins to assume specific three-dimensional shapes that are directly related to their biological functions [8]. Proteins are grouped into super families, families and subfamilies according to their biological function. The classification of proteins is an important task for the molecular biologist, and, ultimately, it is aimed to identify the function of the protein.

There are several protein databases available, for instance, Swiss-Prot and Protein Data Bank (PDB) [1]. In this work we used the TMPDB (TransMem-

F. Rothlauf et al. (Eds.): EvoWorkshops 2005, LNCS 3449, pp. 105–114, 2005.

brane Protein DataBase) [13],[7],[6], a transmembrane subset extracted from some public databases that contains information about the primary structure of 302 transmembrane proteins. The choice for this subset was due to the extremely important functions that these proteins plays in life as pumps, channels, receptors, catalyzers, energy transducers, etc., and have been reported recently to share approximately 20-30% of genes in a whole genome. The transmembrane protein molecules are difficult to crystallize due to their amphiphilic characteristics – they present hydrophobic transmembrane segments (TMSs) but also hydrophilic loops.

The protein-classification problem (PCP) is a very important research area in bioinformatics. As mentioned before, the many genome sequencing projects have been unveiling a growing number of gene products whose function is unknown or barely estimated by homology techniques. The prediction of protein function has been done basically in two ways: prediction of the protein structure and then prediction of function from the structure, or else, classifying proteins into functional families and supposing that similar sequences will have similar functions. Notwithstanding, most proteins share similar structures (in particular, considering the primary structure), since many of them have a common evolutionary origin [11]. Common structures may be characteristic of a given family of proteins but, on the other hand, unrelated families can also share common structures. This two-fold characteristic makes the PCP a challenging problem, for which many methods have been suggested; see, for instance [5],[9],[10],[14],[15].

This paper reports the development and application of a computational tool, named GAMBIT (Genetic Algorithm-based Motif Browsing and Identification Tool), specially devised for the automatic discovery of motifs (short sequences of amino acids). This tool is based on genetic algorithms and uses as input only the information about the primary structure of proteins. The system finds variable-length motifs that occur very often in proteins of a given class (family) but rarely occur in proteins of other classes. Those discovered motifs can be further used to discriminate families of (known) proteins and for the automatic classification of unknown proteins. That is, using the motifs discovered by the proposed system, one can estimate function of an unknown protein by analysing only its primary structure.

## 2 Methodology

### 2.1 Data Preprocessing

The version of the TMPDB used in this work was #6.3, from November 2003. A TMPDB file uses the same format as Swiss-Prot and it has information about the primary sequence of a protein. For the purposes of this work we used only the following fields: ID (identification code in other databases), ME (membrane in which the protein exists) and SQ (Sequence header and its length, followed by the amino acids sequence).

The TMPDB contributors [6] have collected 1,074 articles reporting TM topology, by using MEDLINE search using keywords "transmembrane" and "topol-

ogy" and they found 895 articles. By searching the web directly without using MEDLINE they found 46 articles, and by searching for Swiss-Prot and TrEMBL entries whose RP line contains the annotations of "X-ray crystallography", "structure by neutron diffraction", "structure by electron cryomicroscopy"', "structure by NMR" or "topology", they found 133 articles. After the validation of each article, they extracted the 302 experimentally-characterized transmembrane proteins. To obtain the complete sequence, they made a cross-reference to public databases (using the protein name or the partial sequences). Finally, by combining the information contained in the articles and other information of the public databases, they constructed TMPDB.

The transmembrane proteins are distributed across 25 classes. In this work, aiming to have statistically meaningful results, we used only 6 classes, those with 10 or more proteins. The number of proteins in each class was: 144 proteins in class Inner Membrane (IM), 64 in class Plasma Membrane (PM), 22 in class Mitochondrial Inner Membrane (MM), 10 in class Chloroplast Thylakoid Membrane (CM), 25 in class Endoplasmic Reticulum Membrane (EM) and 16 in class Outer Membrane (OM). Therefore, we used 281 out of 302 proteins of TMPDB, and this data set is available at *http://bioinfo.cpgei.cefetpr.br/en/softwares.htm.*

### 2.2 Encoding and Fitness Function

Genetic Algorithms (GA) were used in this work due to its ability to perform adaptive, powerful and robust searches. Besides, their intrinsic parallelism allows the simultaneous exploration of different regions of the search space. The use of GA for real-world problems encompasses two important definitions: the encoding scheme of an individual and the fitness function. In the implemented system, individuals represent a single motif, that is, a variable-length string of characters, over the alphabet used for encoding the 20 standard amino acids [8].

Recall that our goal is to find a sequence of amino acids (motif) with a high discriminatory power – i.e., a pattern that occurs in most proteins of a given class and occurs in few or no proteins of all other classes. Therefore, this pattern can be characteristic of a given family, allowing it to be discriminated from all others – the essence of classification.

In order to discriminate an individual, we developed a special fitness function that is computed as follows. Given a motif found by the GA, for each class $i$, $i$=1,...,6 (for the transmembrane dataset used in this work), the relative frequency of occurrence of the motif in that class is computed. This is simply the number of proteins of the $i$-th class where the motif occurs anywhere in the protein´s sequence divided by the number of proteins in the $i$-th class. Next, for each class $i$, a measure of the ability of the motif to discriminate between class $i$ and the other classes is given by the equation (1):

$$Disc_i = F_i . \left( 1 - \frac{\sum_{j=1, j \neq i}^{n} F_j}{k-1} \right) \quad (1)$$

where $F_i$ is the relative frequency of the motif in the $i$-th class, $n$ is the number of classes ($n = 6$ in this work), and $k$ is the number of classes that contain at least

one protein whose primary sequence contains the given motif. The rightmost term of the formula simply computes the average relative frequency of the motif in all classes $j \neq i$ containing at least one occurence of the motif. This term is subtracted from 1, so that the term between brackets is to be maximized. Similarly, the value of $F_i$ is also to be maximized. Therefore, a high value of $Disc_i$ means that the motif occurs very often in class $i$ but rarely in the remaining classes. If $k = 1$, in order to avoid division by zero in equation (1), the fraction in the formula is considered to collapse to zero, so that the term between brackets collapses to 1 and $Disc_i$ collapses to $F_i$. This reflects the desirable case where the motif occours only in class $i$ (and in no class $j$, $j \neq i$), so that the motif quality depends only on $F_i$.

Once the value of $Disc_i$ has been computed for all $n$ classes ($i = 1, ..., n$), the individual is associated with the class having the largest value of $Disc_i$. In other words, the motif represented by the individual is considered as a characteristic pattern for proteins of the class with the largest value of $Disc_i$. The proposed fitness function is normalized in the range [0..1], making the interpretation of results somewhat easier, since 1 is the best possible value, meaning maximum discrimination.

### 2.3 Selection Method and Genetic Operators

In this work, the selection method used was the well-known stochastic tournament (with tournament size $k \geq 2$). The usual one-point crossover operator is stochastically applied with a predefined probability, using two individuals of the selected pool. Since the length of the chromosome is variable, the traditional concept of crossover point was slightly modified and adapted to our individual representation. The crossover point is a percentage (of the length of the individual) that defines the starting point from where the crossover breaks the string. The same percentage is used for both parents. For instance, if the percentage is 80%, the rightmost 20% of the amino acids contained in the parents are crossed-over.

As usual, the mutation operator is used to further explore the search space and to avoid unrecoverable loss of genetic material that leads to premature convergence to some local minima. Due to the specific purpose of our system, we devised four different types of mutation (herein, sub-operations), as follows:

1. Left-adding: one randomly generated character (corresponding to an amino acid) is added to the left of the motif.
2. Right-adding: one randomly generated character (corresponding to an amino acid) is added to the right of the motif.
3. Random-changing: all the amino acids from a randomly selected starting point up to the end of the motif are changed, except the first and the last position.
4. Cutting-out: it removes a single character from the amino acid sequence. The removal position is randomly generated.

The mutation probability is a user-defined parameter, as usual in GA. Once the system has decided to do a mutation, all sub-operations have the same probability of being chosen, in a random fashion.

Both crossover and mutation operators are also "hill-climbing-based operators" because they are implemented in such a way that a new individual is immediately evaluated after it has been created and, if its fitness is lower than the parent's fitness, the parent (rather than the child) is copied to the next generation. This procedure does not increase significantly the computational cost and makes the evolutionary process faster in terms of number of generations necessary for convergence, since the generated offspring will be always better than their parents (or will not be generated otherwise). Hence, after a genetic operator is selected according to a given probability, it can be applied in the usual way (inserting the children in the new population regardless of their fitness) or as a hill-climbing-based operator. This choice is done probabilistically according to another user-defined parameter – hill climbing-based operator rate.

The expansion operator is a new operator specifically designed for the motif discovery and protein classification problem. This operator starts by accessing the first protein of the class associated with the individual (this class was determined during the computation of the fitness) and locating the position, in that protein, where the individual's amino acid sequence occurs. Then, it tentatively adds the immediately preceding amino acid (in the protein) to the individual's amino acid sequence (candidate motif). The relative frequency of the individual's amino acid sequence in that class is recomputed. If the new relative frequency is equal to or higher than the previous relative frequency, the just-added amino acid is effectively added to the motif. This operation corresponds to expansion of the individual's genotype. This process is iteratively repeated until the relative frequency becomes lower than the previous one. At this point the above-described expansion process is applied to the amino acid immediately subsequent in the protein. Finally, the whole process (expansion to the left and expansion to the right) is repeated for all the other proteins of the class associated with the individual. Note that this is a computationally expensive operator, but our preliminary experiments have shown that it effectively leads to motifs with a higher predictive power for protein classification.

### 2.4 Running Parameters

The implemented GA has several parameters and many preliminary runs were done to adjust these parameters. This task was done using an enzyme dataset with 6 classes and 100 enzimes per class. These results will be published in [14]. In these runs the expansion operator was always turned on, and those tests produced the following optimal values of parameters: number of generations = 300, population size = 200, hill-climbing-based operator rate = 10%, tournament size = 1%; crossover probability = 20%; mutation probability = 70%. The hill climbing-based operator rate is low to avoid losing population diversity and to prevent a premature convergence.

A conventional GA returns, as its result, the best individual (the one with highest fitness) found during the run. However, in our system the desired result is not a single individual, but rather, a set of individuals. That is, each individual represents a single amino acid sequence (motif), associated with a single class, and this kind of pattern will be used further to classify proteins. Therefore, it is necessary to discover many patterns, associated with as many different classes as possible during the GA search. In each generation, after the fitnesses of all individuals have been computed, some high-quality motifs for each class are saved in a separated file, called the set of discovered patterns - SDP. In fact, the individuals representing those patterns still remain in the population; only a copy of them is saved into SDP. The criterion to select these individuals is their fitness – only those with fitness greater than a user-defined minimum quality threshold will be saved. This procedure results in the discovery of many motifs, associated with different classes, as desired. However, special care is taken to prevent adding motifs that are substrings of other motifs already in the SDP.

## 3 Computational Experiments and Results

Using the data described in Section 2.1, motifs were discovered using two different tools: GAMBIT and MEME (Multiple EM for Motif Elicitation) [2]. MEME is a well-known freely-available web tool supported by the San Diego Supercomputer Center (*http://meme.sdsc.edu/meme/website/intro.html*). MEME essentially uses statistical modeling techniques to automatically choose the best width and description for each motif. In our experiments, we used all default parameters of MEME, except the number of motifs, set to 15.

After running GAMBIT and MEME, the top fifteen motifs discovered by each of those tools were set aside as designated results for each of those tools. The goodness of a motif was measured by its class-discrimination ability, as defined in equation 1. Recall that both GAMBIT and MEME are intended to discover motifs in sets of sequences and are not designed as classification tools. Hence, in order to evaluate the effectiveness of the discovered motifs in predicting the functional class of proteins, we have used the discovered motifs as predictor attributes in two classification algorithms available in WEKA (Waikato Environment for Knowledge Analysis) [16] , version 3.4.3. WEKA is a well-known Java-based data mining toolkit freely-available on the internet (*http://www.cs.waikato.ac.nz/ml/weka*).

The two classification algorithms used in the experiments were J4.8 (the WEKA implementation of the very well-known C4.5 decision tree induction algorithm [12]) and Prism [4], a rule induction algorithm that discovers classification rules directly from the data, without producing a decision tree. In our experiments, we used the default parameters of both J4.8 and Prism.

The predictive accuracies obtained by J4.8 and Prism were measured using a well-known 3-fold cross validation procedure [16], as follows. The data set was partitioned into 3 mutually-exclusive and exhaustive partitions. In the i-th iteration of the cross-validation procedure, $i$=1,2,3, the $i$-th partition was used

as the test set and the other two partitions were grouped and used as the training set. In each of the 3 iterations, first GAMBIT and MEME were used to discover motifs from the training set. Then, as mentioned earlier, those motifs were used as predictor attributes in J4.8 and Prism, which were also run on the training set. Each motif was used as a binary attribute, indicating whether or not the motif occurred in a given protein (training example).

Note that each of the two classification algorithms, J4.8 and Prism, was run twice: one run used motifs discovered by GAMBIT, and the other run used motifs discovered by MEME. This produced four classification models – two decision trees produced by J4.8 and two rule sets produced by Prism. Finally, the four classification models were evaluated on the test set – which was never accessed during training – in order to measure the predictive accuracy (generalization ability) of the discovered classification models. This procedure was carried out 3 times (corresponding to the 3 iterations of the cross-validation procedure), and the reported results are the average of the accuracy rate on the test set across the 3 iterations.

Figure 1 shows the decision tree generated by J4.8 and Table 1 shows the rules generated by Prism. Due to space limitations, both Figure 1 and Table 1 show only the classification models built from the motifs discovered by GAMBIT. In Figure 1, each internal node tests for the presence (1) or absence (0) of an attribute (a motif). Similarly, in Table 1 the conditions in the rule antecedents refer to the presence or absence (indicated by a "not" operator) of motifs. The predicted classes – represented in the leaf nodes of the decision tree and in the consequents of the rules – are the membrane classes defined in Section 2.1. For instance, the top-right part of the decision tree in Figure 1 corresponds to the rule: IF motif GHL is absent (0) AND motif AQS is present (1) THEN class = PM (Plasma Membrane).

Although there are many ways to measure classification accuracy (see, for instance, [3],[16]), in this work, the final performance was measured using the accuracy rate. The average accuracy rates (on the test set) computed by the cross-validation procedure were: 73.4% using J4.8 with motifs found by GAMBIT, 58.0% using J4.8 with motifs found by MEME, 99% using Prism with motifs found by GAMBIT, and 65.4% using Prism with motifs found by MEME.

Therefore, the motifs found by GAMBIT were clearly much more effective in predicting protein class than the motifs found by MEME, in both the classification algorithms used in the experiment (J4.8 and Prism).

Note that Prism obtained considerably better results than J4.8. A likely explanation for this result is that Prism is more flexible, in the sense that it can select only one relevant value of an attribute (motif) – either its presence or its absence. By contrast, J4.8 has to select both values of an attribute (motif) – both "1" (presence) and "0" (absence) – to be included in the tree (in different branches coming out from the same parent). In this kind of data set, intuitively the presence of a motif is a more relevant attribute value than its absence, which gives an advantage to the more flexible rule representation of Prism. Indeed, out of the 20 rule conditions in Table 1, only 4 refer to the absence of a motif (using

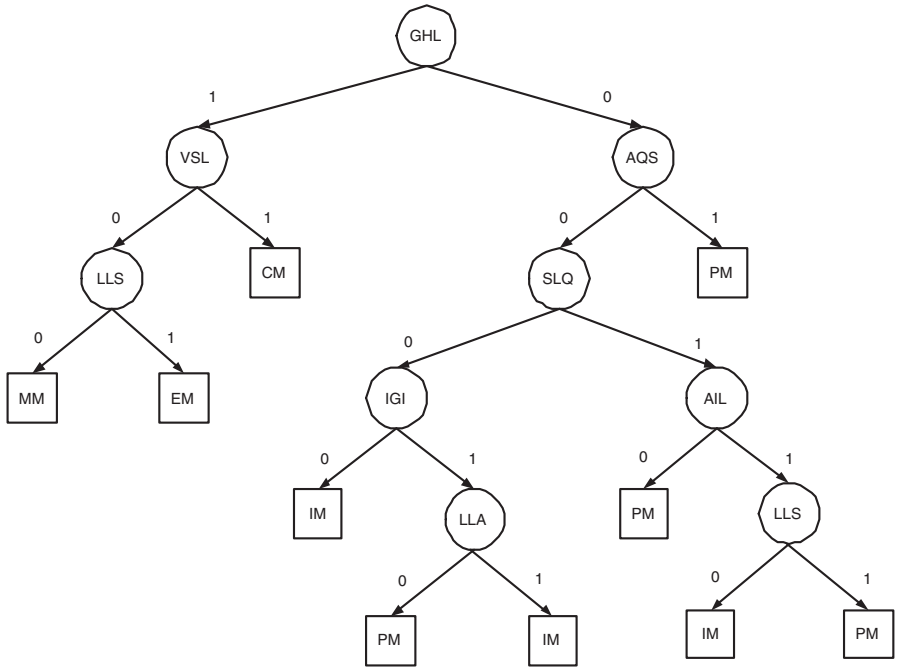


**Fig. 1.** Decision tree constructed by J48 using motifs found by GAMBIT

**Table 1.** Subset of the best rules found by PRISM using motifs found by GAMBIT

| |
|---|
| If (SRR) then IM |
| If (SNN) then IM |
| If (APML) then IM |
| If (MNNM) then IM |
| If (EWR) then PM |
| If (LIG and VLG and SLK) then PM |
| If (LWK and not(MKK)) then MM |
| If (RGYWQE) then CM |
| If (VTV and GFV and not(TN)and not(LWA)) then EM |
| If (VDY and DGD) then OM |
| If (DPT and LID and not(GDI)) then OM |

the operator "not"). The other 16 conditions refer to the presence of a motif. In addition, note that the class OM does not appear in the decision tree of Figure 1, which is a clear disadvantage of that classification model. Finally, note also that the decision tree of Figure 1 uses only short motifs (with 3 amino acids), whereas the rules in Table 1 have a somewhat wider diversity of motif size: two rules use motifs with four amino acids, and one rule uses a motif with six amino acids (a motif produced by the expansion operator).

## 4 Conclusions and Future Work

We described a system based on a Genetic Algorithm specifically designed for motif discovery, aiming to classify unknown-class proteins. The system was evaluated using a transmembrane protein dataset.

The genetic operators of GAMBIT, specifically designed for the PCP, have played an important role in the positive results achieved, since they allowed the GA to obtain motifs with high discriminatory power.

Comparing results obtained by GAMBIT with MEME, it can be seen that the latter did not find good motifs to discriminate one class from the others. On the other hand, this is a remarkable characteristic of GAMBIT, an innate ability accomplished by its fitness function. It is a matter of fact that MEME was not projected for the same purpose as GAMBIT but, to the best of our knowledge, it is the tool that most closely can be compared with our system. In short, MEME discovers motifs in a group of proteins, while GAMBIT discovers motifs that discriminate a group of proteins from another.

Using the discovered motifs found by both systems, the J48 and Prism algorithms generated comprehensive classifiers, useful to biologists. It is possible that those discovered motifs are related to known specific secondary or tertiary structures (this investigation was left to future work).

Finding groups of amino acids that uniquely characterize protein families is a very important issue in molecular biology. Results for the transmembrane dataset using GAMBIT and WEKA strongly suggest the efficiency of the method to find motifs capable of discriminating between groups or proteins, offering a feasible solution to the PCP.

Future work includes more exhaustive tests of the GA control parameters for fine-tuning and development of biologically-inspired genetic operators. We intend to improve GAMBIT so as to find motifs based on regular expressions. Also, it is intended to apply this system to find motifs for classification of other protein families of biological interest.

## Acknowledgments

This work was partially supported by a research grant from the Brazilian National Research Council – CNPQ (350053/02-0 and 475049/03-9).

## References

1. Abola EE, Sussman JL, Prilusky J and Manning NO (1997) Protein data bank archives of three-dimensional macromolecular structures. *Meth Enzymol* 277:556-571
2. Bailey TL and Elkan C (1994) Fitting a mixture model by expectation maximization to discover motifs in biopolymers. *Proc. 2nd Int.Conf. on Intelligent Systems for Molecular Biology*, pp. 28-36

3. Bojarczuk CC, Lopes HS, Freitas AA (2004) A constrained-syntax genetic programming system for discovering classification rules: application to medical data sets. *Artif Intell Med* 30(1):27-48
4. Cendrowska J (1987) Prism: an algorithm for inducing modular rules. *Int J Man-Mach Stud* 27:349-370
5. Hanke J, Beckmann G, Bork P and Reich JG (1996) Self-organizing hierarchic networks for pattern recognition in protein sequence. *Protein Sci* 5(1):72-82
6. Ikeda M, Arai M, Lao DM and Shimizu T (2002) Transmembrane topology prediction methods: a re-assessment and improvement by a consensus method using a dataset of experimentally-characterized transmembrane topologies. *In Silico Biol* 2:19-33
7. Kihara D, Shimizu T and Kanehisa M (1998) Prediction of membrane proteins based on classification of transmembrane segments. *Protein Eng* 11:961-970
8. Lehninger AL, Nelson DL and Cox MM (1998) *Principles of Biochemistry.* $2^{nd}$ ed. Worth Publishers, New York, pp. 134-137
9. Manning AM, Brass A, Goble CA, and Keane JA (1997) Clustering techniques in biological sequence analysis. In: *Proc.*$1^{st}$ *Europ. Symp. on Principles of Data Mining and Knowledge Discovery*, pp. 315-322
10. Mathura VS, Schein CH and Braun W (2003) Identifying property based sequence motifs in protein families and superfamilies: application to DNase-1 related endonucleases. *Bioinformatics* 19(11):1381-1390
11. Murzin AG, Brenner SE, Hubbard T and Chothia C (1995) A structural classification of proteins database for the investigation of sequences and structures. *J Mol Biol* 247:536-40
12. Quinlan JR (1993) *C4.5: Programs for Machine Learning.* Morgan Kaufmann Publishers, San Francisco
13. Shimizu T and Nakai K (1994) Construction of a membrane protein database and an evaluation of several prediction methods of transmembrane segments. In *Proc. Genome Informatics Workshop*, pp. 148-149
14. Tsunoda DF and Lopes HS (2005) Automatic motif discovery in an enzyme database using a genetic algorithm-based approach. To appear.
15. Weinert WR and Lopes HS (2004) Neural networks for protein classification. *Appl. Bioinformatics* 3:41-48
16. Witten IH and Frank E (2000) *Data Mining: Practical Machine Learning Tools and Techniques with Java Implementations.* Morgan Kaufmann, San Francisco