International Journal of Engineering and Technical Research (IJETR)
ISSN: 2321-0869 (O) 2454-4698 (P), Volume-5, Issue-2, June 2016

Parameter Tuning of a Genetic Algorithm devised to support automatic Protein Classification

Denise Fukumi Tsunoda, Alex Sebastião Constâncio

***Abstract*— Proteins are responsible for important processes in living organisms. Nowadays, the data about proteins is massive, but being able to observe an unknown protein and determine its function is still challenging. One form of understanding a protein is classifying it, since proteins in the same class or family share similar purposes. A tool named GAMBIT implements a Genetic Algorithm that analyses already classified proteins in order to find characteristic patterns and provide assistance to automatic classification. Genetic Algorithms work guided by many parameters that must be tuned for better results. A strategy of evaluating the results achieved from various configurations of those parameters is presented, as well as the one that was considered the best.**

***Index Terms*— Data Mining, Evolutionary Computation, Genetic algorithms, Genetic operators.**

I. Introduction

Proteins were discovery in 1838 by Berzelius and Mulder, and were named like that by the derivation of the Greek word *proteios* that means, "standing in front" (TANFORD and REYNOLDS, 2001). Berzelius justified the name by stating that such a molecular structure plays the leading role in animal nutrition, in order to emphasize its importance for life.

Proteins work as biological automata and are responsible for several functions in living organisms such as sustentation, regulation, increase of reaction speed and others. The building blocks of proteins are molecules named amino acids. There are 20 different kinds of amino acids present in proteins, which are made of a sequence of those. The molecular structure of a protein presents a spacial shape, but such a 3D layout may be abstracted and only the amino acids sequence is enough to understand the protein's function. The sequence of amino acids of a protein is also called protein´s polypeptide chain. Such a sequence is also often referred as the protein's primary structure and is inextricably linked to its function (LEHNINGER, NELSON and COX, 2012).

Proteins are organized in groups with similar or related functions. Those groups are sometimes named as classes or families. Being able to assign a class to a just discovered protein helps to predict or understand what is the function of such a protein (LEHNINGER, NELSON and COX, 2012; XU and DUNBRACK, 2012).

Genome-sequencing technology has produced a huge amount of data about proteins. However, there are a large number of proteins whose function is unknown. Hence, an active research area consists of predicting proteins' functions based on their primary sequences. Despite the existence of several methods that try to solve the protein function prediction problem, it still remains one of the main challenges in the current post-genomic era.

Many existing methods are being experimented to approach the problem of protein classification (which means, to assign an already known class to a still unclassified protein), such as Decision Trees, Neural Networks, Support Vector Machines (FARIA, FERREIRA and FALCÃO, 2009), Naïve Bayesian and Random Forest (IQBAL et al, 2014). Among them, there are the Evolutionary Computation algorithms and a particular kind of evolutionary method is the Genetic Algorithm (GA) (GOLDBERG 1983; 1989).

This paper presents a product named GAMBIT (Genetic Algorithm-based Motif Browsing and Identification Tool), that uses a GA particularly implemented to work on the so-called Protein Classification Problem (PCP).

Any GA uses many parameters that govern the way it behaves to cover the space of solutions and one step of the process of evaluating how well such an algorithm performs on its task is the parameter tuning. It's not different for GAMBIT.

It's important to stress that this paper focus on the particular step of **parameter tuning** and shows what are the parameters used by GAMBIT during a run and how they were discovered to maximize GAMBIT's results quality.

It´s not the intention of this paper to go deep in the design decisions or implementation details of the GA present in GAMBIT. Such content will be object of another paper. This one is organized as follows: section II presents the definitions of the problem that GAMBIT tackles (the Protein Classification Problem), section III presents a brief explanation on Genetic Algorithms, the technology used by GAMBIT, section IV presents the GAMBIT's approach to tackle the above mentioned problem, section V presents the datasets used during both training and testing phases, section VI presents the experiments methodology used during the parameter tuning of GAMBIT, section VII presents the quality criteria used when comparing the experiments results, section IIX presents the experiments and their results, section IX presents the conclusions gathered from the experiments performed and the last section presents the references.

II. The Protein Classification Problem

Proteins are organized in families or classes, which are groups of proteins with similar functions (XU and DUNBRACK, 2012). Being able to assign a class to a newly discovered protein helps to understand its function in a certain organism. The problem is to know which class is that.

The primary structure of a protein is the simple sequence of its containing amino acids. The amino acids may be

Denise Fukumi Tsunoda, DECIGI / PPGCGTI, Federal of Paraná University, Curitiba, Brazil, +55 41 33604191.
Alex Sebastião Constâncio, PPGCGTI, Federal of Paraná University, Curitiba, Brazil, +55 41 33604191.

represented by letters (LEHNINGER, NELSON e COX, 2012), so a primary structure of a protein is represented by a string of those letters, each one being an amino acid. This is usually referred in literature as Amino-Acid Sequence (AAS). Under this conception, one example of a not so big protein could be **RLQQWRKAALVLNASRRFRYTLDLK**. This protein is named **2M73** and its class is known as **HYDROLASE**. It is a particular kind of protein, an enzyme, and because that it also follows an enzyme classification system, under which its class is **EC.3.6.3.8**.

In terms of proteins, similar function means similar structure (LEHNINGER, NELSON e COX, 2012). One approach to try to classify a certain protein into a class is to find pieces of its primary structure and check which class is characterized by those pieces.

Computationally speaking, the problem is to identify which substrings characterize each class and then try to match those patterns in a still unclassified protein. This corresponds to use what is called a sequential model representation of the proteins, rather than a discrete model one (NANNI, LUMINI and BRAHNAM, 2014).

In many occasions, several researchers used the so-called alignment matrices to identify the presence of AAS in proteins. However, recent works have preferred to not use such matrices in order to avoid error propagation problems (FARIA, FERREIRA and FALCÃO, 2009).

Also, the uses of the alignment matrices, like PAM and BLOSUM have a severe impact in performance, what was another reason to avoid those (IQBAL et al, 2014).

III. Genetic Algorithms

Like other evolutionary algorithms, a GA is a search and optimization method inspired by the principle of natural selection in biological evolution (MITCHELL, 1997).

GAMBIT implements a GA especially designed to work on the PCP. As any GA, GAMBIT works by transforming a number of candidate solutions (not a single one) from a state where they do not solve the problem to a state where they are considered good enough to work as a solution, under some criteria. Among the various highly enhanced candidates, at least one is expected to have the qualities to solve the original problem. As a GA, GAMBIT does not pursue the optimal solution, but a solution with an acceptable level of quality.

As any GA, GAMBIT starts by producing a random set of candidates, in GA vernacular, a population of individuals. The next step is to transform the original candidates into better ones (evolution), by applying what are called genetic operators and then producing a second population.

Since the process starts from one population and produces a second one from it, it's said that the second population is the next generation of the original one. The same process is repeated, producing generation after generation until a certain stop criterion is reached, like a specific number of generations.

The genetic operators are used to select individuals in one generation and evolve them to produce the following generation. Typically, the operations are **mutation** (that simply randomly changes an individual into another), **crossover** (that takes two individuals and mix their parts together to produce another pair or individuals) and **reproduction** (an individual is just send from one generation to the next, unchanged). Which operator is to be applied over a given individual during the evolution of a population is determined by probabilities established as GA parameters.

The individual that is selected and submitted to a genetic operator in order to produce another is called an **ancestor** and the newly created one, its **offspring**.

After a new generation is produced, it is necessary to measure its evolution to ensure that at least a few individuals really evolved towards the optimal solution. When evaluating and guiding the evolution process, the GA uses a **fitness function**. Such a function measures the quality of an individual and produces a grade to it. For instance, an individual with no quality at all will evaluate to 0, while the optimum solution will evaluate to 1. If an individual in one generation (ancestor) has a fitness of 0.324 and after a mutation it becomes another individual (offspring) with its fitness measured as 0.456, the genetic operation resulted in an improvement of the individual, getting it closer to the optimal solution (evolution).

So, the general behavior of a GA is to evolve a population over generations by randomly selecting individuals from one population, operate over them with mutation, crossover and reproduction (again randomly selected), produce a new generation and measure the evolution process with the fitness function, to start over again from this newly generated population. When an individual reaches the threshold of an acceptable fitness (a minimum quality), a solution is found and the process terminates.

IV. GAMBIT'S APPROACH

GAMBIT aims to find a number of amino acid sequences, from now on named **motifs**, also named as **features** in literature (IQBAL et al, 2014), that highly characterize a certain class but does not the others. For instance, suppose the motif **AABADCGGAB** (an amino acid sequence) exists in all the proteins of EC.1 class, but absolutely in none of the other five classes (EC.2 to EC.6, explained later). This would be an absolutely discriminating motif, able to be used to determine if a so far unknown protein should or not be classified as EC.1.

Unfortunately, experience shows that there is not a single motif with such a discrimination power. That's why GAMBIT looks for a set of motifs to characterize a certain protein class instead a single one.

GAMBIT builds and evolves populations of motifs (motifs are the GA individuals) and calculates their quality by using a function to evaluate its discrimination rate (the fitness function). The absolutely discriminating motif, if it existed, would have its fitness evaluated to 1.

The fitness function used by GAMBIT is given by the presented equation where:

F(i,m) is the relative frequency rate of the motif ***m*** in the ***i***-th class; the relative frequency rate is given by dividing the count of proteins that present the motif by the total count of the proteins in the considered class;

n is the total count of classes;

k is the total count of classes that has at least one occurrence of the motif ***m***;

j is the index of all complementary classes to the ***i***-th class.

$$Fitness(i,m) = F(i,m) \bullet \left(1 - \frac{\sum_{j=1, j\neq i}^{n} F(j,m)}{k-1}\right)$$

The interpretation of such a function becomes easy with an example. Suppose the motif LGA occurs in 30% of the proteins in class EC.3, 12% in class EC.1, 7% in class EC.6 and 0% in the other three classes (EC.2, EC.4 and EC.5). When calculating the fitness for the assignment of the motif to EC.3, the calculations result in ***F(3,LGA)* = 0.3** and the average of all other frequencies (***F(1,LGA)=0.12*** and ***F(6,LGA)=0.07***) result in **0.095** ((0.12 + 0.07) / 2). In the formula, those values give *Fitness*(3,LGA)=0.3(1-0.095)=0.2715. So, the fitness of motif LGA, when it is considered as a characterizing motif for EC.3, is **0.2715**.

GAMBIT promotes the evolution of the populations along generations and, at each one, selects the individuals (motifs) that better discriminate a certain protein class among the others (highest fitness for a certain class). At the end, GAMBIT associates a number of motifs to each of the existing protein classes in the experiment.

Hence, what GAMBIT produces is a set of motifs that, when considered together, should characterize the protein class to which it is associated.

Sometimes, a certain motif presents a non-zero fitness for more than one class. In such a case, the motif is assigned to the class where it computes the higher fitness. This approach means that GAMBIT is a single-label (or single-class) classifier, that uses the one-against-all labeling scheme (WANG and YAO, 2012).

V. THE INPUT DATASET

GAMBIT works by using a number of already classified proteins as input in order to discover which set of motifs better discriminates their classes.

There are several protein databases available for public access like PDB (*Protein Data Bank*), PIR (*Protein Information Resource*), SCOP (*Structural Classification of Proteins Database*) and SWISS-PROT (ABOLA, SUSSMAN, PRILUSKY *et al*, 1997; BERMAN, WESTBROOK, FENG *et al*, 2000).

PDB was selected because it provides not only the primary structure of proteins, but also secondary and tertiary structure annotations (XU and DUNBRACK, 2012), which are out of the scope of this paper. However, future research will include the use of the additional data present in PDB in more advanced versions of GAMBIT, so PDB was considered the best choice even at this time of the work.

PDB presents a large (hundreds of thousands) and continuously increasing amount of proteins. For the purposes of GAMBIT, not all of those proteins were used, but only a particular group, namely, the enzymes.

Experiments were performed over a set of 37,597 enzymes extracted from PDB. Enzymes present a hierarchical classification system, the Enzyme Commission Number, or simply EC Number. In the first level there are six classes (EC.1, EC.2, EC.3, EC.4, EC.5 and EC.6, named respectively Oxidoreductase, Transferase, Hydrolase, Lyase, Isomerase and Ligase) (MCDONALD, BOYCE, TIPTON, 2001), each of those is subdivided until the fourth level (for instance, the already presented enzyme **2M73** is a **HYDROLASE** also classified as **EC.3.6.3.8**).

GAMBIT uses a particular input file (the ".data" file) that is directly generated from items present in PDB. Such an input file was generated from the 20150102 snapshot of PDB (produced at January 2nd, 2015), freely available and found at ftp://snapshots.rcsb.org/20150102/pub/pdb/data/structures/, which was the most up to date one when the experiments took place.

GAMBIT operates on the primary structure only, but PDB files contain much more on each protein, what required a preprocessing phase. Such a preparation process consisted in a) downloading all the ".gz" files (each one containing a single protein structure), b) extracting the ".ent" file, c) scanning the ".ent" file to extract the primary structure, protein class and d) saving a single file that contains all the proteins found (GAMBIT ".data" file), described as the amino acid sequence, the protein's name and the protein's class.

A PDB ".ent" file is a text file in which each line is a record. There are many kinds of records and they are identified by a tag. After that tag, the content of each line has a specific meaning. For GAMBIT's purposes, only the records tagged as HEADER (provides the name and class of the protein), COMPND (provides the EC class, if the protein is an enzyme) and ATOM (provides each of the amino acids present in the protein) are important.

VI. EXPERIMENTS METHODOLOGY

A typical GA uses a number of parameters that have deep influence over its performance and effectivity. Such parameters are sensible to particular circumstances of the data when the search is being performed. Hence, the first part of the experiment was to find a good (ideally the best) set of those parameters.

GAMBIT uses the following parameters to control the process of looking for discriminating motifs:

- **Hill Climbing probability**, that controls the evolution pressure and establishes the minimum amount of individuals that must increase in fitness during each generation;
- **Mutation and Crossover probability**, that control the amount of each kind of genetic operator happen during the evolution of a generation;
- **Reproduction probability**, determines how many individuals in a generation are replicated unchanged to the next;
- **Selection rate**, that uses a stochastic tournment to select which individuals of a population will be subjected to the evolution process; this rate represents a percentile of the population with the higher fitness from where the evolution candidates will be selected;
- **Population size**, that stablishes the number of individuals in each generation;
- **Motif count per class**, that determines how many motifs will be used to characterize each class; only the ones with higher fitness are considered, even when more exist;
- **Number of generations**, that establishes how many generations will occurbefore the process stops.

A methodology to experiment and measure the influence of those parameters was devised and applied, consisting in a set of experiments and results comparisons. As already mentioned, only enzymes were considered in this process.

The protein base was divided in two mutually excluding parts: the one used to perform a process of training corresponds to 2/3 of all enzymes (25,084) and is called the **training set**. The remaining 1/3 (12,513) was dedicated to a testing process and is named the **test set**.

The training set (25,084 proteins, 2/3 of all the proteins) was submitted to GAMBIT that resulted in a number of motifs assigned to each class. Those motifs were then used to perform a classification process of the proteins in the test set (12,513 proteins, 1/3 of all the proteins). The evaluation of the classification performance over the test set was used as comparison factor among the various parameter combinations, in order to find the best one.

Only the first level of the EC classification was used in these experiments, so all the enzymes are distributed in six classes. The count of the proteins in each class of each set is presented in Table 1.

Table 1: Protein classes distributions in training and test datasets

Class	Training	Test	Total
EC.1 – Oxidoreductase	4,394	2,178	6,572
EC.2 – Transferase	7,796	3,886	11,682
EC.3 – Hydrolase	9,009	4,504	13,513
EC.4 – Lyase	1,912	959	2,871
EC.5 – Isomerase	1,138	567	1,705
EC.6 – Ligase	835	419	1,254
Total	25,084	12,513	37,597

Table 1 shows that classes are unbalanced, presenting very different sizes among themselves. The size difference between EC.6 (1,254 proteins) and EC.3 (13,513 proteins) is remarkable.

The strategy was to establish a set of values to all the parameters used by GAMBIT and execute it. Only one parameter is tested at a time. For instance, when testing for Hill Climbing, all the other parameters were fixed to certain values and experiments using 10%, 40% and 70% took place. For each value, GAMBIT was executed three times.

For each time, the resulting motifs were used to generate a WEKA input file (an ".arff" file), which was submitted to WEKA J48 method. The results provided by J48 at each time were used to calculate an average performance measure and the corresponding standard deviation.

The average performances of each set of parameters were compared together and the set of parameters with higher performance was kept for the next set of tests that would focus on some other parameter.

The same process was repeated for each of the parameters until all were covered and measured, what produced what was considered the best values for all the parameters of GAMBIT. WEKA J48 method is an implementation of C4.5 classification algorithm (QUINLAN, 2014), that produces a decision tree derivated from the evaluation of the input file (WANG and YAO, 2012). After generating the decision tree, WEKA validates it by calculating the hit rate and then provides a confusion matrix. For GAMBIT purposes, J48 was configured to run a 10-fold cross validation test (KOHAVI, 1995).

When generating the WEKA input file, GAMBIT considers each of the motifs assigned to a class as an attribute, that can be valued either 0 or 1. For each protein, the set of attributes that are supposed to characterize its class receive the value 1 and the others the value 0. Each protein generates one line in the ".arff" file consisting of values for those attributes for the protein's class. That is the file processed by J48.

Finally, the process is made more reliable by assigning a new random seed when producing the initial generation of each run.

VII. QUALITY CRITERIA

Since the idea is to find a good set of parameters to use in future experiments over the test set, it's a need to determine some kind of performance comparison so it will be possible to determine which settings provide better results and therefore should be kept.

Such a performance factor (named from now on as Performance or simply **Pf**) is calculated as the product of Specificity (**Sp**) by Sensitiveness (**Sn**) (LOPES, 1996). Hence, experiments with higher Pf values as results determine the parameter set to be used in the next experiments. Pf is preferred over hit rate because it better fits distributions of unbalanced classes (HAND, 1997).

Sp and Sn are calculated directly from the Confusion Matrix delivered by WEKA, in the execution report generated by J48. Since WEKA also gives the hit rate (HRate), which is preserved and presented in all result tables, for a comparison with Pf.

VIII. EXPERIMENTS FOR PARAMETER TUNING

The parameter tuning had to start with some initial set of parameter values. The initial values for the GAMBIT parameters were set as follows: a) Hill Climbing probability: **10%**; both mutation and crossover probabilities: **45%**; Reproduction probability: **10%**; Selection percentile (GAMBIT uses the stochastic tournament): **1%** (of the population size); Maximum count of proteins per class: **10**; Population size: **500**; Generations count: **100**.

The tables with results show the parameter values tested, the processing time (column **Time**) from GAMBIT - in hours and minutes - Specificity (**Sp**) and Sensitiveness (**Sn**) in percentiles, the Performance (**Pf**) and the hit rate (**HRate**) given by WEKA. The latter was included since it is usually found in literature, but in this work the focus is maximizing the **Pf**, as already explained.

The first parameter to be tested and tuned was the Hill Climbing (HB) probability. The values established for this parameter were 10%, 40% and 70%.

As explained, the idea is to choose the value that results in the highest Pf, but here there is an exception. The table 2 shows that 10% resulted in Pf=**45.95 ± 0.05**, while 40% resulted in Pf=**46.12 ± 0.09**. It's also a technical tie, since the values, when including the standard deviation, almost touch each other. However, the processing time is considerably higher for 40% and even more for 70%.

The conclusion was that all this cost did not provide a proportional gain in results. This parameter in particular does

not significantly affect the quality of the motifs discovered, but does affect the running time in a dramatic level.

Considering all that, the value HC=10% was chosen as the best, resulting in Pf=**45.95 ± 0.05%**

Table 2: Hill Climbing probability tests and results

HC	Sn (%)	Sp (%)	Pf (%)	HRate (%)	Time
10%	**72.94 ± 0.05**	**62.99 ± 0,05**	**45.95 ± 0.05**	**72.94 ± 0.02**	**0:52**
40%	73.34 ± 0.19	62.89 ± 0.03	46.12 ± 0.09	74.76 ± 0.19	2:24
70%	73.31 ± 0.01	62.90 ± 0.01	46.11 ± 0.01	73.31 ± 0.01	3:42

Then, crossover (**Cross**) and mutation (**Mut**) probabilities were tested. Those probabilities together sum 0.9. The remaining 0.1 is reserved to the reproduction operation. So, during the processing of a certain generation (in order to produce its offspring), there is 90% of probability to occur a mutation or a crossover and 10% to occur a simple reproduction. The selected individual is directly inserted in the next generation without any changes. Table 3 presents all the combinations tested. The best result was achieved with Mut=**10%**, Cross=**80%**, with Pf=**46,54 ± 0.06%**.

Table 3: Crossover and Mutation tests and results

Mut	Cross	Sn (%)	Sp (%)	Pf (%)	HRate (%)	Time
30%	60%	73.08 ± 0.03	62.93 ± 0.05	45.99 ± 0.16	73.08 ± 0.53	0:57
20%	70%	72.90 ± 0.26	62.97 ± 0.04	45.90 ± 0.14	72.90 ± 0.26	1:00
10%	**80%**	**74.17 ± 0.13**	**62.75 ± 0.03**	**46.54 ± 0.06**	**74.17 ± 0.13**	**1:09**
60%	30%	73.11 ± 0.12	62.93 ± 0.03	46.01 ± 0.06	73.11 ± 0.12	0:48
70%	20%	73.43 ± 0.07	62.88 ± 0.02	46.17 ± 0.03	73.77 ± 0.49	0:49
80%	10%	73.44 ± 0.17	62.87 ± 0.04	46.18 ± 0.08	73.46 ± 0.18	0:49

The influence of the population size (Pop) and the number of generations (Gen) were also investigated. Populations with 200, 500 and 1000 individuals were tested against experiments with 50, 100 and 250 generations. The best result was achieved with Gen=**100**, Pop=**500**, what resulted in Pf=**46.54 ± 0.06%**, as visible in table 4.

Table 4: Population size and number of generations tests and results

Gen	Pop	Sn (%)	Sp (%)	Pf (%)	HRate (%)	Time
50	1000	73.12 ± 0.26	62.97 ± 0.07	46.02 ± 0.12	73.12 ± 0.26	2:21
100	**500**	**74.17 ± 0.13**	**62.75 ± 0.03**	**46.54 ± 0.06**	**74.17 ± 0.13**	**1:09**
250	200	73.29 ± 0.21	62.94 ± 0.08	46.13 ± 0.16	73.29 ± 0.34	0:48

Finaly, the percentile of the population that was used for tournament selection (Sel) was tested with 1%, 3%, 5% and 7%. The best result was Sel=**1%**, what resulted in Pf=**46.54 ± 0.06%**, as presented in table 5.

Table 5: Tournment rate tests and results

Sel	Sn (%)	Sp (%)	Pf (%)	HRate (%)	Time
1%	**74,17 ± 0,13**	**62,75 ± 0,03**	**46,54 ± 0,06**	**74,17 ± 0,13**	**1:09**
3%	73,28 ± 0,33	62,94 ± 0,02	46,13 ± 0,18	73,28 ± 0,33	3:19
5%	73,00 ± 0,14	63,00 ± 0,05	45,99 ± 0,05	73,00 ± 0,14	5:06
7%	73,15 ± 0,06	62,93 ± 0,01	46,03 ± 0,03	73,15 ± 0,06	6:50

After the evaluation of the parameters directly related to the genetic algorithm implemented by GAMBIT, an evaluation of the influence of the number of motifs (Mot) associated to each protein class was performed. WEKA was provided with 5, 10, 15 and 20 motifs and 10 gave the best result, Pf=**46.54 ± 0.06%**, as presented in table 6.

Table 6: Number of motifs influence on J48 classifier tests and results

Mot	Sn (%)	Sp (%)	Pf (%)	HRate (%)	Time
5	69.09 ± 0.03	63.63 ± 0.03	43.96 ± 0.03	69.09 ± 0.03	1:12
10	**74.17 ± 0.13**	**62.75 ± 0.03**	**46.54 ± 0.06**	**74.17 ± 0.13**	**1:09**
15	74.00 ± 0.26	62.73 ± 0.07	46.42 ± 0.13	74.00 ± 0.26	1:12
20	74.08 5± 0.19	62.74 ± 0.03	46.48 ± 0.10	73.99 ± 0.20	1:12

After all the experiments and their results collection and compilation, the parameter set that produced the best results from GAMBIT when evaluating 25,084 enzymes, distributed in 6 classes (EC.1 to EC.6) is: Hill Climbing probability: **10%**; mutation probability: **10%**; crossover probability: **80%**; selection probability: **10%**; population size: **500 individuals**; number of generations: **100**; tournament percentile: **1%**; motif amount to each class: **10**.

IX. CONCLUSIONS

The best results for the enzyme dataset with J48 (C4.5 implementation) and 10-fold cross validation methods (performance=**46.54 ± 0.06** and hit rate=**74.17 ± 0.13**) suggest that the proposed evolutionary computation method is effective in finding predictive features (motifs) for protein classification, but still required enhancements.

The Hill Climbing probability does not produce any enhancements in terms of quality of the motifs, but severely slows down the evolution process. Higher values for this parameter increase the pressure for evolution, requiring a larger amount of individuals with higher fitness at each generation. In order to get closer to that amount, the GA works for longer before giving up.

The evolution process resulted in better motifs when only 10% of the operations where mutation, 80% were crossover and 10% was replication. This happens because high rate of mutation produces many non-existing motifs, since it's a random search. On the other hand, crossover uses parts of a pair of individuals with certain quality to produce other two individuals, eventually combining two motifs with lower quality to produce others with higher quality. 10% of the motifs were simply replicated to the next generation, giving them the opportunity to be part of the evolution process again.

Experiments with a population with 500 individuals that evolved for 100 generations produced better results. Smaller populations take less time to evolve, even for a longer chain of generations, but the results are worse. The opposite situation (larger populations for fewer generations) did not produce better motifs either, but took more time to run.

The selection rate with higher values did not provide better results, but increased the execution time. The conclusion is

that a bigger portion of candidates does not produce better offspring, only the best ones seem to really produce quality motifs.

The amount of motifs found to characterize each class has influence, but only until a certain level. Experiments shown those more than ten motifs do not provide any better results. This phenomenon is related to the quality of the individuals. For instance, certain experiments produced 20 motifs for each class, but only the best ten of them have a significant discriminating power. The others had non-zero fitness, but they did not help to characterize a protein class.

All the motifs found during all the experiments were exactly three amino acids long. Motifs with this size are not hard to be found in many proteins, but they occur in more than one class. However, each class is characterized by several motifs (ten in the experiments with the best performance) combined in a decision tree.

In the experiments with the best performance, the decision trees generated by J48 had in average 2905 nodes, with 1453 leaves, after pruning. Those tree test for combinations of presence and absence of the motifs in a given protein and J48 uses that to predict the class of such a protein.

The discriminating power of a single motif (its fitness) is not taken in account during the evaluations performed by J48, since the attributes in the input file for WEKA just establish presence of absence of the motifs.

One possible approach for future research could be assigning more than one class to a single motif and design a fitness function that evaluates the discrimination power by using the intersection of the protein sets involved.

Considering the generated motifs have a fitness which value varies between zero and one, maybe some kind of fuzzy system evaluation could provide an interesting prediction heuristic.

When working specifically with enzymes, another possibility for future work could be a hierarchical evaluation and evolution process, since enzymes are already organized in classes and subclasses.

Yet another possible course of action could be work with motifs that accept wildcards, giving them more flexibility instead of being required to occur in absolutely identical form in proteins.

GAMBIT was conceived in a modular design and can be adapted to accommodate such variations. The experiments used more than 37000 proteins and were successful to predict the class of an unclassified protein in around 75% of the times. This suggests that GAMBIT presents a promising approach to the Protein Classification Problem, but still requires some improvements in order to increase its prediction power.

REFERENCES

[1] TANFORD, C and REYNOLDS, J – **Nature's robots**: a history of proteins, USA: Oxford University Press, 312p, 2001.

[2] LEHNINGER, A. L.; NELSON, D. L.; COX, M. M. **Principles of Biochemistry**. 6. ed. Macmillan, 2012.

[3] XU, Q.; DUNBRACK, R. L. Assignment of protein sequences to existing domain and family classification systems: Pfam and the PDB. **Bioinformatics**, v. 28, n. 21, p. 2763, 11, 2012.

[4] FARIA, D.; FERREIRA, A. E. N.; FALCÃO, A. O. Enzyme classification with peptide programs: a comparative study. **BMC Bioinformatics**, England, v. 10, p. 231-231, 2009.

[5] IQBAL, M. J. et al. Efficient feature selection and classification of protein sequence data in bioinformatics. **The Scientific World Journal**, United States, v. 2014, p. 173869-173869, 2014.

[6] GOLDBERG, D. E. **Computer-Aided Pipeline Operation Using Genetic Algorithms and Rule Learning**. PhD Thesis, University of Michigan, 1983.

[7] GOLDBERG, D. E. **Genetic Algorithms in Search, Optimization & Machine Learning**. Reading, MA: Addison-Wesley, 1989.

[8] NANNI, L.; LUMINI, A.; BRAHNAM, S. An empirical study of different approaches for protein classification. **The Scientific World Journal**, United States, 2014.

[9] MITCHELL, M. **An Introduction to Genetic Algorithms.** Cambridge: MIT Press, 1997.

[10] WANG, S.; YAO, X. Multiclass Imbalance Problems: Analysis and Potential Solutions. **IEEE Transactions on Systems**, Man & Cybernetics: Part B, v. 42, n. 4, p. 1119, 08, 2012.

[11] ABOLA, E. E.; SUSSMAN, J. L.; PRILUSKY, J., MANNING, N. O. Protein data bank archives of three-dimensional macromolecular structures. **Methods in Enzymology**, San Diego: Academic Press, v. 277, 1997.

[12] BERMAN, H. M.; WESTBROOK, J.; FENG, Z., GILLILAND, G.; BHAT, T. N., WEISSIG, H.; SHINDYALOV, I. N.; BOURNE, P. E. The protein data bank. **Nucleic Acids Research**, v. 28, p. 235-242, 2000.

[13] MCDONALD, A. G.; BOYCE, S.; TIPTON, K. F. **Enzyme Classification and Nomenclature**. eLs , 2001

[14] QUINLAN, J. R. **C4.5**: programs for machine learning. Elsevier, 2014.

[15] KOHAVI, R. **A Study of Cross-Validation and Bootstrap for Accuracy Estimation and Model Selection**. Proceeding of International Joint Conference on Artificial Intelligence (IJCAI), 1995

[16] LOPES, H. S. **Analogia e Aprendizado Evolucionário**: uma Aplicação em Diagnóstico Clínico. Tese (Doutorado em Sistemas de Informação), Departamento de Engenharia Elétrica, Universidade Federal de Santa Catarina, 1996, 159 p.

[17] HAND, D. J. **Construction and Assessment of Classification Rules**. New York: John Wiley & Sons, 1997.

Denise Fukumi Tsunoda has a BS in Computer Science from the Federal University of Paraná (1992), a master's degree (1996) and Ph.D. (2004) in Electrical Engineering and Industrial Informatics at the Federal Technological University of Paraná. She is currently a professor at the Federal University of Paraná in the course of Information Management, Department of Science and Information Management and coordinator of the Master's program in Science, Management and Information Technology.

Alex Sebastião Constâncio has a BS in Computer Science from the Federal University of Paraná (1992) and has more than 25 years of experience as a software engineering professional and IT management. At this moment, he works as an IT analyst in the Center of Electronic Computing at Federal University of Paraná. Additionally, he is currently member of the Master's program in Science, Management and Information Technology and performs researches in machine learning and text mining.