2014 IEEE 14th International Conference on Bioinformatics and Bioengineering

A Genetic algorithm for the selection of features used in the prediction of protein function

Larissa Fernandes Leijôto*, Thiago Assis de Oliveira Rodrigues*, Luis Enrique Zárate† and Cristiane Neri Nobre †
*Pontifical Catholic University of Minas Gerais
Departament of Computer Science, Belo Horizonte, Brazil 30535-901
Email: {larissa.leijoto, taorodrigues}@sga.pucminas.br
†Pontifical Catholic University of Minas Gerais
Post-Graduation Program in Informatics, Belo Horizonte, Brazil 31980-110
Email: {zarate, nobre}@pucminas.br

Abstract—**Proteins are macromolecules that have a high molecular weight, and make up, along with water, most of the composition of cells. The functions they perform are extremely important, such as the catalysis of biochemical reactions, cytoskeleton formation, and the transportation and storage of substances. With the completion of genome sequencing, protein discovery has been growing exponentially, and the laboratory methods for determining their functions have not been able to keep up with this growth. Due to this fact, it is necessary to develop methods to aid in this function discovery process. Thus, this work proposes a physical-chemical feature selection methodology calculated by means of the structures that compose the proteins. This stage has the goal of choosing a feature subset from all available features. A feature is considered relevant if it can be used by the machine to create a separation capability between the different protein classes. To select this subset, we proposed the use of a simple genetic algorithm. The results obtained with the proposed methodology were superior to those found in the literature, reaching a precision of 71% and a sensitivity of 68%.**

I. Introduction

The joint efforts of biology and computer science have created a new field called bioinformatics. It consists in the development of computational techniques that can infer and derive important and relevant predictions from data acquired by molecular biology [1].

Proteins are macromolecules that have high molecular weights, and along with water make up most of the composition of a cell, thus being responsible for performing the most important biological roles. Due do the large set of total protein functions, knowing what a protein does is fundamental to the development of many applications in biology [2]. Such applications range from the development of new drugs for treating illnesses to farming applications. After the completion of human genome sequencing, the rate of known proteins with unknown functions has been significantly increasing. The laboratory methods for the discovery of protein functions, such as x-ray diffraction, are costly and also very time-consuming, and are thus unable to keep up with protein discovery. Hence the need for computational techniques capable of helping in the discovery of protein functions, consequentially reducing the need for lab testing.

For this purpose, three approaches are commonly used. The first approach makes the prediction according to primary sequence similarity. This is a widely used approach due to the large amount of discovered sequences. However, it fails because the primary structure in the least preserved in the evolutionary aspect of structures. This means that proteins can have a high degree of similarity in their chains, but perform completely different functions. The second approach is related to the tertiary structure [3], which is far more preserved than the primary structure. The function of a protein is directly related to this structure. In spite of that, it has been observed that structural similarities do not always correspond to catalytic similarities. The third approach is based on the use of physical-chemical characteristics to represent amino acids present in the primary structure. They are calculated based on the interaction of all structures of a protein [4] [5].

Borro et al. [6] and Dias [7] have used the features found in Sting_DB [8], one of the largest physical-chemical, structural and biological protein feature databases. For the selection of the best features, statistical resources and data mining methods were used. However, it is not possible to guarantee that the features selected by these techniques are the best possible feature subset in the database. The goal of this work is to develop a methodology that, based on a genetic algorithm (GA), is able to select representative physical-chemical characteristics, given a protein set. We intend to predict their functions using machine learning. The learning technique of choice for the execution of this work was Support Vector Machine (SVM), largely used for solving similar problems [9] [10] [3] [11] [12]. It has good generalization capabilities, as well as a well-established theoretical foundation concerning mathematical and statistical concepts.

The remainder of this article is structured as follows. In Section II we detail the most relevant concepts used in this work, which are fundamental to its comprehension. The related work is described in Section III. Section IV details the proposed methodology. Results are presented in Section V. Finally, Section VI contains the conclusions and final considerations.

II. Background

A. Protein

The chemical reactions that take place within amino acids form a protein, and determine its three-dimensional organization. To understand the properties of a protein, it is necessary to describe how the amino acids are formed. Their structure is

978-1-4799-7502-0/14 $31.00 © 2014 IEEE
DOI 10.1109/BIBE.2014.42

composed by a central carbon atom, called alpha carbon, which can be attached to four groups: the amine group (NH2), the carboxyl group (COOH), a hydrogen atom or a chain called the side-chain, which is where amino acids differ from each other [13] [14] [1]. Proteins have four levels of organization: 1) The *Primary Structure* is the sequence of amino acids along its chain; 2) the *Secondary Structure* consists of the special relationships among amino acids that are close to the primary structure. In proteins, the basic secondary structure units are: alpha helices and beta sheets; 3) the *Tertiary Structure* are the way in which the atoms of a polypeptide chain are organized in a three-dimensional space; 4) the *Quaternary Structure* are the interactions between the many amino acid chains present in a protein.

B. Support Vector Machine

Currently, there are many available machine learning techniques. Among them, SVM, proposed by Vapnik [15] has been standing out the most. Machine learning techniques seek to minimize the probability of erroneously classifying still-undiscovered patterns by means of a probability distribution [16]. SVM maps the training samples into a higher-dimensional space, assuming that for high dimensions in space every problem becomes separable. From this premise, it finds the support vectors on the margins of a hyperplane, which distinguish different characteristics by maximizing the distance between classes, reducing the problem to a decision among classes. In Table I the main kernels used in the SVM classifying process are presented. They have the goal of finding the hyperplane that best separated the data. In this work we used LibSvm [17], which is a library that uses support vector machines in pattern classification and regression. It supports many operating systems and has implementations in different languages. It can also easily integrate with the WEKA environment [18], which was also used in this work.

Table I. COMMONLY USED SVM KERNELS

Function Type	Equation	Parameters
Linear	$X_i^T X_j$	-
Polynomial	$(\gamma X_i^T X_j + r)^d$	r, d
Gaussian(RBF)	$exp(\gamma \|X_i - X_j\|^2)$	γ
Sigmoid	$\tanh(\gamma X_i^T X_j + r)$	γ, r

Source: Hsu et al. [17]

C. Feature Selection

Feature selection is an important preprocessing step, because attributes that will serve as inputs to the learning algorithm are derived from it. The stage has the goal of choosing an attribute subset from all available attributes. An attribute is deemed relevant if the machine can use it to create a separation capacity between the different classes. Among the algorithms used for feature selection, we can cite exponential algorithms, which perform an exhaustive search in the solution set to determine the best possible solution. This is not a viable method, because computational time increases exponentially. Another technique is sequential selection, such as *forward selection* and *backward elimination* [19]. Its disadvantage is disregarding interactions between characteristics. Two evolutionary algorithms are part of the random search methods: genetic algorithms [20] and *particle swarm optimization* (PSO) [21]. The advantage of randomized algorithms in comparison with sequential methods is that the former consider characteristic interactions, and the latter do not.

D. Genetic Algorithm

The genetic algorithm (GA), initially proposed by Holland [22], is based on Darwin's theory of evolution and belongs to the evolutionary algorithms group. These algorithms start from an initial population, in which an individual is associated to a potential solution in the global solution set. Each individual has a *fitness* value, which determines how well-adapted to its environment such individual is, as well as its chances of survival. After a fitness-based selection process, the individuals chosen to remain in the population are then recombined using the crossover and mutation genetic operators. From then on the process repeats itself, hoping to obtain increasingly better fitness results for each generated population. The use of GA is highly justifiable for the problem proposed in this work due to its ability to generate representative samples from the solution set after few iterations.

III. Related Work

Dobson and Doig [23] have proposed a methodology for enzyme prediction using their structural data. Their method aims to classify the enzymes into one of six superfamilies, based on a group of the protein's structural attributes. The average precision value for the analyzed classes was 35%, using SVM.

Borro et al. [6] used the parameter selection process to increase the precision of their protein classification model. In order to select the best parameters, they used statistical and data mining resources, such as data correlation analysis and association rule mining. The *Discrete Cosine Transform* (DCT) was used to circumvent the size problem for the feature vectors, used as input in the classifier. Another problem was the imbalance of the used classes, since the largest class had 161 proteins and the smallest class had only 18. The *sampling with replacement* statistical resource was used to try and solve this problem. With the proposed methodology, precision levels of about 53,9% were obtained, using Bayesian networks.

Dias [7] used SVM in the prediction of protein functions. To represent the protein's amino acids, physical-chemical characteristics contained in Sting_DB were used, as previously done in [6]. Along with these characteristics, data from GO (Gene Onthology) were also used. To extract relevant features from the proteins that were used as inputs in the neural network, DCT was used. After this stage, twenty-three local binary classifiers were created, able to confirm or refute a specific function. Afterwards, a global classifier that gathers all predetermined functions was created. With the employed methodology, precision levels of 98% and sensitivity levels of 93% were obtained.

Huang and Wang [20] proposed the use of a genetic algorithm to simultaneously optimize feature selection and the parameters for the SVM kernel. For this purpose they used an evaluation function based on SVM accuracy. The experiments were conducted with 11 databases, available in the *University of California, Irvine* (UCI) repository. It is noteworthy that none of the database used belong to the context of prediction of protein function. Finally, the authors compared their parameter optimization solution with *Grid Search*, a method commonly used in the literature for this purpose. The proposed approach was able to significantly improve accuracy, but on the other hand its processing time is considerably larger.

IV. Methodology

One of the main difficulties in the protein classification process is determining how amino acids will be coded, since SVM does not accept nominal variables. This methodology has the goal of determining by means of a genetic algorithm which characteristics are the most suitable for enzyme classification in their respective classes. Figure 1 shows a flowchart of the proposed algorithm, and the following subsections describe its stages.

A. Database and Feature Extraction

The protein set used in this work was the same one used by Dobson [23] e Borro et al. [6]. This set was used so that results from our work can be accurately compared to theirs. Proteins were extracted from *Protein Data Bank* (PDB) [24], the largest and most complete protein repository in existence. Table II presents the amount of enzymes used in this work. *Enzymes* are catalyst proteins, that is, they accelerate the rate of biochemical reactions. The International Union of Biochemistry and Molecular Biology (IUBMB) developed a classification system that divides enzymes in six classes, which can be also seen in Table II.

Table II. Enzyme classes with their respective functions

EC	Class	Function	Amount
1	Oxidoreductase	Electron transfer reactions	76
2	Transferases	Group transfer between two molecules	120
3	Hydrolases	Hydrolysis reactions of many covalent bonds	161
4	Lyases	Breaking of covalent bonds and removal of water, ammonia and carbon gas molecules	60
5	Isomerases	Modification of a single molecule, without partitioning another	57
6	Ligases	Reactions of new molecule formation by joining two other molecules	18

Sting_DB, put together by the Computational Biology laboratory of Embrapa Brazil, has a varied set of features extracted from all the structures that compose a protein. In this work we used 338 features from one of the modules included in this repository, called *Java Protein Dossier* [25]. Features extracted from Sting_DB can be grouped in:

- **Evolutionary** features are calculated from changes in proteins, that is, how much their sequences have evolved along time.
- **Interatomic contacts** are calculated from the contact between atoms present in each residue of a protein.
- **Physical-chemical** features are obtained from attractions exerted by the many types of connections between amino acids.
- **Geometrical structures** are calculated from the three-dimensional structure of the protein.
- **Surface** features are calculated from cavities in the surface of a protein to which ligands bind themselves.

B. Chromosome design

In a GA, an individual represents a possible solution to the problem in question, so it is necessary to find the best way to represent them. In Figure 2, we present an example of an individual used in the algorithm we propose. This individual is represented by a vector composed by 11 integer variables, where f can receive values from 1 to N, where N is the number of features contained in the database. Each of the values that compose the vector represents a feature found in the files extracted from Sting DB. Among the 338 features available in the database, only 11 were considered, once we compare our study with the one proposed by Borro et al, which used this amount.

$f1$	$f2$	$f3$	$f4$	$f5$	$f6$	$f7$	$f8$	$f9$	$f10$	$f11$

Figure 2. Chromosome design

C. Normalization

Before the SVM training process, some additional stages are necessary for the preprocessing of selected features. Data normalization has the goal of avoiding the prevalence of attributes with bigger interval values over those with smaller interval values. Another purpose of data normalization is to avoid great difficulties during the numerical calculations carried out by the kernel chosen for the SVM. Each feature is normalized, where X is the original value of the feature, and max and min are the largest and smallest value of the aforementioned feature, respectively.

$$X' = \frac{X - max}{max - min} \tag{1}$$

D. Discrete Cosine Transform

Proteins have different amounts of amino acids, so if amino acids were coded sequentially, the SVM input vectors would have different sizes. Thus, it is necessary to use some technique to circumvent this situation, because SVM only accepts feature vectors of equal dimension. To solve the dimension problem, we used DCT [26], whose formula is presented in Equation 2. DCT is a feature extraction technique that transforms data in the time domain into data in the frequency domain. In this process, frequencies are sorted in descending order, where the first frequencies are the ones that store the most relevant information from the data set. High frequencies are considered to be noise.

$$C_k = \alpha_k \sum_{n=0}^{N-1} X_n \cos\left[\frac{\pi}{N}\left(n + \frac{1}{2}\right)k\right], n > 0 \tag{2}$$

where $\alpha_k = \frac{1}{\sqrt{N}} for, k = 0$ e $\alpha_k = \sqrt{\frac{2}{N}} for, k = 1...N$

E. Selection of low-frequency coefficients

To select the best amount of coefficients to represent the protein, we carried out experiments with the same characteristics used by Borro et al. [6] and Dias [7]. It can be observed in Figure 3 that precision rates increase as the amount of selected coefficients increases; conversely, sensitivity rates decrease. Since both metrics are important, it was necessary to choose a coefficient amount that was able to yield the best balance

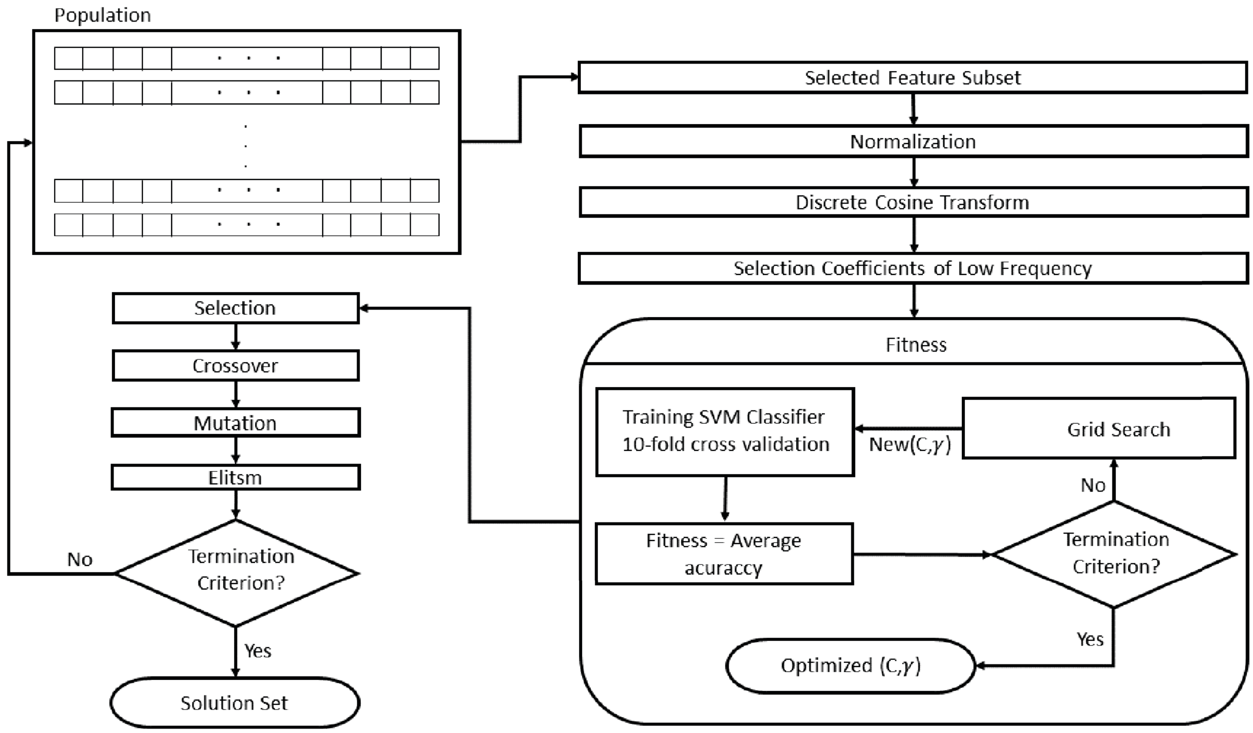

Figure 1. Proposed Methodology

between them. Therefore, the chosen coefficient amount was 75.

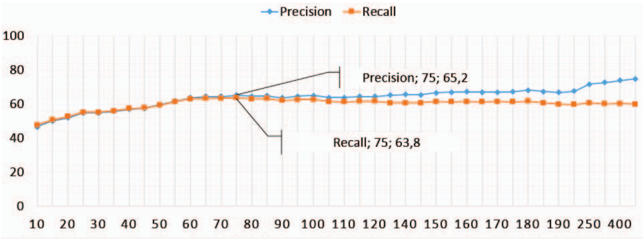

Figure 3. Average precision and sensitivity per number of selected coefficients

F. Fitness function

The fitness function determines how adequate a solution is for solving a problem. In this work the goal was maximizing the average hit rate of protein classification. To achieve this, we used the cross-validation process and the *grid search* heuristic.

The most used kernel, which is also the default kernel for the LibSvm library, is RBF (*Radial Basic Function*). With it, it is possible to solve multi-class problems, through mapping features into a space of higher dimensions. RBF has a parameter that can be varied to obtain better learning rates for the classifier. The γ (gamma) parameter has the goal of finding the Gaussian curve that best separates the instances. Another parameter to be adjusted is C (cost), which is common to all kernels. It determines the penalty for the function, a type of tolerance to errors inherent to a classification problem.

1) Grid Search: Grid search seeks to optimize the classification through the execution of SVM and enables results obtained with the parameter adjustments to be analyzed, testing exponential sequences for them.

2) Cross Validation: In this work, the parameter optimization process was used along with cross-validation, which is a statistical technique for partitioning the test and training sets. To partition the sets, we used $k = 10$. With this, the base is divided into k subsets, where $k - 1$ subsets are used in the model construction process, during which training is conducted, and the remaining subset is used for validating. This process is repeated k times, and each time a different test set is used. This way, the technique seeks to optimize machine learning so it can learn as much as possible in order to generalize the model and thus predict the behavior of data for future inputs.

G. Genetic Operators

An important part in assuring the efficacy of an AG is the adequate choice of mutation and crossover operators. With the changes in encoding or for specific problem types, the traditional operators are ineffective.

1) Selection: The chosen selection method was Tournament. In it, k individuals compete for their permanence in the population, with the individual possessing the best fitness value gaining the right to stay in the population.

2) Crossover: Due to encoding restrictions, it was necessary to use a special type of crossover, called *Partially Mapped Crossover* (PMX) [27] [28]. It consists of a two-point crossover where values are mapped to assure there are no repetitions between the variables of an individual. Variable repetition during the selection process is completely undesirable, since the duplication or repetition of a feature adds no value to the classification process. The chosen crossover rate was 0.65.

3) Mutation: In the proposed algorithm, simple mutation was used. In it, some individuals are randomly chosen and their variables are randomly altered. This operator has the goal of maintaining population variability. A mutation rate of 0.01 was used.

4) Elitism: Elitism has the goal of keeping the best solution produced in each generation. By using it, we assure that the most fit individuals will always participate in the production of new population members.

H. Solution

When the algorithm terminates, a feature subset is obtained. They are the features that enabled the individual to be best adapted to its environment, that is, that yielded the best accuracy in the classification process.

I. Performance Measurements

To evaluate the best feature set obtained with the proposed methodology, three metrics were used: Accuracy, Precision and Sensitivity. To understand these metrics, the following concepts are fundamental: *True Positive* (TP): the amount of proteins that were correctly classified to the class in question;*False Positive* (FP): the amount of proteins that did not belong to the given class, but were classified as part of the class;*False Negative* (FN): the amount of proteins in the analyzed class that were wrongfully classified; *True Negative* (TN): the amount of proteins belonging to other classes, classified to the class in question.

- **Accuracy** is the rate of test instances correctly classified by the classifier

$$\mathit{Accuracy} = \frac{TP+TN}{TP+TN+FP+FN} \tag{3}$$

- **Precision** is the rate of instances classified to a class that actually belonged to said class

$$\mathit{Precision} \; = \frac{TP}{TP+FP} \tag{4}$$

- **Sensitivity** is the rate of class instances that were actually classified as belonging to said class.

$$Sensitivity = \frac{TP}{TP+FN} \tag{5}$$

- **F-Measure** is the harmonic mean between precision and sensitivity

$$F - Measure = \frac{2TP}{2TP+FP+FN} \tag{6}$$

V. Results and Discussion

This section has the goal of presenting the results given with the proposed methodology. The results were compared with Borro et al. [6] due to the fact of the results present in other related works were insufficient to realize the necessary comparisons. For instance, Dobson and Doig [23] presented only the rate of accuracy. It is also important to highlight the results given by Borro et al. which were better than the other results found in related works. Borro et al. used the same the database that was used in the present work.

During the experiments, 7 executions of the genetic algorithm were carried out in order to try and find the best feature set. The number of generations and individuals was limited to 50 and 10, respectively. This limitation is due to the fact that the time required to run the algorithm increases significantly as these parameters are increased. After the completion of the executions, the feature set with the best accuracy was selected. The SVM parameters selected by *grid search* for this feature set were C = 32.0 and γ = 4.8828125E-4.

The features selected by the GA are described below:

- ***3DEntropyINT(8)*** number of residues in the same chain used to calculate the relative entropy, then divided by the volume of the sphere.
- ***3DEntropyLHAsw(9,7)*** entropy relative to amino acids found in the domain of radius equal to 7, and using a sliding window with 9 residues.
- ***ACCC*** residue solvent accessibility calculated with joined proteins.
- ***ACCI*** residue solvent accessibility calculated for each isolated protein.
- ***ContactsEnergyAllsw(true,3)*** average energy for all residue contacts and 2 of its neighbors, also considering contacts with water molecules.
- ***Curvature*** average residue curvature in protein complexes.
- ***DensityCAsw(3,6)*** density calculated with the sphere centered in Cα, radius equal to 6, using a sliding window with 3 residues.
- ***DensityCAsw(3,6)*** density calculated with the sphere centered in Cα, radius equal to 4, using a sliding window with 5 residues.
- ***DensityCAsw(3,6)*** density calculated with the sphere centered in Cα, radius equal to 5, using a sliding window with 9 residues.
- ***DensityLHAsw(3,4)*** density calculated with the sphere centered in the LHA (last atom in the side chain except Hydrogen atoms), radius equal to 4, using a sliding window with 3 residues.
- ***DistanceCG*** represents the distance between the Cα of each residue and the center of mass of the chain (barycenter).

Table III presents the precision and sensitivity obtained for each class and the average for all classes, using the features selected by the genetic algorithm. Aiming to improve the predictions and to compare them with other methodologies proposed in the literature, we added the amino acid frequency for each protein to the feature vector. The results of such addition are also shown in Table III. It can be observed that the insertion of the frequency caused average precision and sensitivity levels to be improved by 1% and 6% respectively.

Table III. Results without and with adding amino acid frequencies

	Without Frequency		With Frequency	
	Precision	Sensitivity	Precision	Sensitivity
Oxidoreductase	0.68	0.58	0.74	0.66
Transferases	0.60	0.66	0.62	0.73
Hydrolases	0.65	0.76	0.77	0.76
Lyases	0.73	0.55	0.62	0.60
Isomerases	0.75	0.67	0.76	0.70
Ligases	0.82	0.50	0.79	0.61
Mean	0.70	0.62	0.71	0.68

The results for the evaluated metrics were obtained from data contained in the Confusion Matrix, shown in Table IV. The matrix represents the number of correct classifications overlapped with the number of predicted classifications for each class. By analyzing the confusion matrix, we can observe that some instances were classified into different classes. A possible explanation is due to the imbalance between classes. Most of the false positives occurred for the two predominant classes, which are Hydrolases and Transferases, respectively.

Table IV. Confusion Matrix

Real Classes	Classes assigned by the classifier					
	Oxi	Tra	Hyd	Lya	Iso	Lig
Oxidoreductase **(Oxi)**	50	9	11	1	4	1
Transferases **(Tra)**	6	87	13	10	3	1
Hydrolases **(Hyd)**	5	26	122	6	1	1
Lyases **(Lya)**	1	14	7	36	2	0
Isomerases **(Iso)**	6	3	5	3	40	0
Ligases **(Lig)**	0	1	1	2	3	11

Table V presents the results obtained for the metric presented in the ”Performance Measure” subsection. The average sensitivity rate was 68%. Again, a possible explanation for this fact is the imbalance between classes. Instances tend to be classified into the prevailing classes, due to the large number of samples from these classes the classifier used to learn from.

Table V. Metrics used for the evaluation of the best set

	Precision	Sensitivity	Accuracy	F-measure
Oxidoreductase	0.74	0.66	0.71	0.69
Transferases	0.62	0.73	0.64	0.67
Hydrolases	0.77	0.76	0.76	0.76
Lyases	0.62	0.60	0.62	0.61
Isomerases	0.76	0.70	0.74	0.73
Ligases	0.79	0.61	0.72	0.69
Mean	0.71	0.68	0.70	0.69

The average obtained precision was 71%, most of which coming from the Ligase class. This happened because, among the 14 instances classified into this class, 11 were correctly classified. The classes with smallest precision rates were Transferase and Lyase, with a precision rate of 62%. For Transferase, only 87 of the 140 instances classified into the class actually belonged to it. For Lyases, the 62% precision rates can be attributed to the significant number of instances from the prevailing classes mistakenly classified as being Lyases. This can, once more, be attributed to the imbalance between classes. Since the number of instances of the Lyase class given to the classifier was smaller, it was not able to learn enough to distinguish Lyases from other prevailing classes. Concerning accuracy, we obtained an average rate of 70%.

Figure 4 presents a comparison of the precision obtained with the proposed methodology and the methodology used by Borro et al. [6]. It can be seen that the proposed methodology was able to substantially improve upon average precision, with an increase of 17.5%.

In Figure 5 we present a comparison of the sensitivity metric obtained by each methodology. An improvement of 22,1% was obtained using the features selected by our proposed methodology.

In Figure 6 the comparison results for the *F-measure* metric are displayed. The results for the features identified by our proposed methodology were the best for all classes. An average improvement of 22.8% in comparison with the methodology by Borro et al. [6] was obtained.

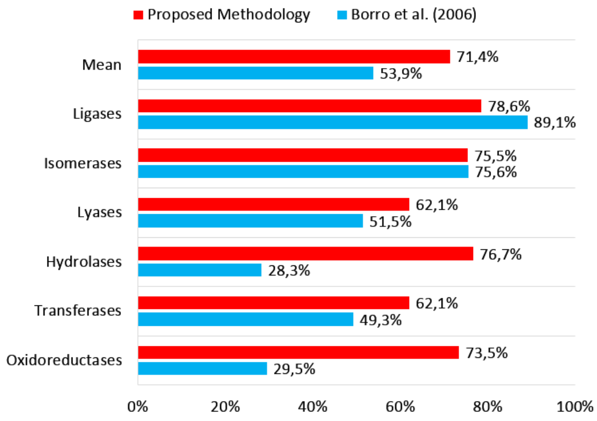

Figure 4. Precisions obtained by the proposed methodology and by Borro et al. (2006)

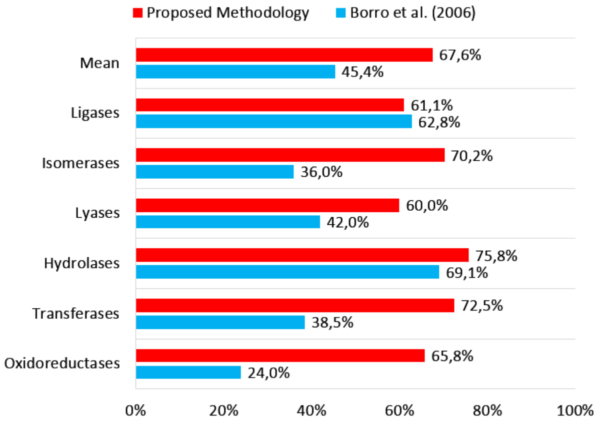

Figure 5. Sensitivity obtained by the proposed methodology and by Borro et al. (2006)

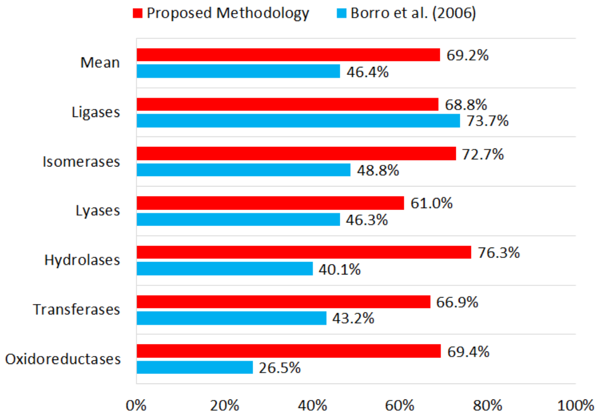

Figure 6. F-measure obtained by the proposed methodology and by Borro et al. (2006)

VI. Conclusion

As future work, we propose the development of a parallel and distributed GA. Such a GA would enable us to increase the solution space encompassed by the GA. The parallelism also would allow tests with different parameter used in the GA and an analyse more detail about this parameters. Thus, also would be possible to compare the feature selection and with no feature selection, testing effectiveness of the methodology proposed in this paper. Another important issue is the fitness function for the GA. In this work we sought to maximize only accuracy, which is why a simple GA implementation was able to fulfill our needs. It is desirable to also maximize sensitivity and precision in addition to accuracy. To accomplish this goal, a multi-objective GA can be used, in which it is possible to perform more than one optimization simultaneously.

Acknowledgment

The authors acknowledge the financial support received from the Foundation for Research Support of Minas Gerais State-FAPEMIG, Brazil(Project grant APQ-01565-12). This research was developed with support from the National Supercomputing Center of the Federal University of Rio Grande do Sul, Brazil.

References

[1] A. M. Lesk, *Introduction to Bioinformatics*, Artmed, Ed. Oxford University Press, 2005.

[2] G. Pandey, V. Kumar, and M. Steinbach, "Computational approaches for protein function prediction: A survey," Department of Computer Science and Engineering University of Minnesota, Tech. Rep., October 2006.

[3] M. T. A. Shamim, M. Anwaruddin, and H. Nagarajaram, "Support vector machine-based classification of protein folds using the structural properties of amino acid residues and amino acid residue pairs," *Bioinformatics*, vol. 23, p. 3320–3327, 2007.

[4] A. Yang, R. Li, W. Zhu, and G. Yue, "A novel method for protein function prediction based on sequence numerical features," *MATCH Communication in Mathematical and in Computer Chemistry*, vol. 67, pp. 833–843, 2012.

[5] B. J. Lee, J. Y. Lee, H. G. Lee, and K. H. Ryu, "Classification of enzyme function from protein sequence based on feature representation," in *Bioinformatics and Bioengineering, 2007. BIBE 2007. Proceedings of the 7th IEEE International Conference on*, Oct 2007, pp. 741–747.

[6] L. C. Borro, S. R. de Medeiros Oliveira, M. E. B. yamagishi, A. L. Mancini, J. G. Jardine, I. Mazoni, E. H. do Santos, R. H. Higa, P. R. K. Falcão, and G. Neshich, "Predictiong enzyme class from protein structure using bayesian classification," *Genetic and Molecular Research*, vol. 1, pp. 193–202, 2006.

[7] U. M. Dias, "Predição da função das proteínas sem alinhamento usando máquinas de vetor de suporte," Master's thesis, Universidade Federal de Alagoas, Março 2007.

[8] A. L. Mancini, R. H. Higa, A. Oliveira, F. Dominiquini, P. R. Kuser, M. E. B. Yamagishi, R. C. Togawa, and G. Neshich, "Sting contacts: a web-based application for identification and analysis of amino acid contacts within protein structure and across protein interfaces," *Bioinformatics*, vol. 20, pp. 2145–2147, 2004.

[9] L. Y. Han, C. Z. Cai, Z. L. Ji, Z. W. Cao, J. Cui, and Y. Z. Chen, "Predicting functional family of novel enzymes irrespective of sequence similarity: a statistical learning approach," *Nucleic*, vol. 32, p. 6437–6444, 2004.

[10] C. Cai, W. Wang, L. Sun, and Y. Chen, "Protein function classification via support vector machine approach," *Mathematical Biosciences*, vol. 185, p. 111–122, 2003.

[11] L. Lua, Z. Qian, Y.-D. Cai, and Y. Li, "Ecs: An automatic enzyme classifier based on functional domain composition," *Computational Biology and Chemistry*, vol. 31, no. 3, p. 226–232, jun 2007.

[12] C. Cai, L. Han, Z. Ji, and Y. Chen, "Enzyme family classification by support vector machines," *Proteins: Structure, Function, and Bioinformatics*, vol. 55, pp. 66–76, 2004.

[13] J. Setubal and J. Meidanis, *Introduction to computational Molecular Biology*. Pws, 1997.

[14] A. Lehninger, D. L. Nelson, and M. M. Cox, *Lehninger Principles of Biochemistry*. Macmillan, 2008.

[15] C. Cortes and V. Vapnik, "Support-vector networks," *Machine Learning*, vol. 20, no. 3, pp. 273–297, September 1995.

[16] V. N. Vapnik, "An overview of statistical learning theory," *Transactions on Neural Networks*, vol. 10, pp. 988–999, 1999.

[17] C. wei Hsu, C. chung Chang, and C. jen Lin, "A practical guide to support vector classification," National Taiwan University, Taiwan, Tech. Rep., 2010.

[18] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I. H. Witten, "The weka data mining software: An update," *ACM SIGKDD Explorations Newsletter*, vol. 11, pp. 10–18, 2009.

[19] J. G. de Carvalho Pereira, "Caracterização dos aminoácidos da interface proteína-proteína com maior contribuição na energia de ligação e sua predição a partir dos dados estruturais," Master's thesis, Universidade Estadual de Campinas, 2012.

[20] C.-L. Huang and C.-J. Wang, "A ga-based feature selection and parameters optimization for support vector machines," *Expert System with Application*, vol. 31, pp. 231–240, 2006.

[21] C. L. Huang and J.-F. Dun, "A distributed pso–svm hybrid system with feature selection and parameter optimization," *Applied Soft Computing*, vol. 8, p. 1381–1391, 2008.

[22] J. Holland, "Adaptation in natural and artificial systems." *The University of Michigan Press, Ann Arbor.*, 1975.

[23] P. D. Dobson and A. J. Doig, "Predicting enzyme class from protein structure without alignments," *Molecular Biology*, vol. 345, p. 187–199, 2004.

[24] H. M. Berman, J. Westbrook, Z. Feng, G. Gililand, T. N. Bhat, H. Weissig, I. N. Shindyalov, and P. E. Bourne, "The protein data bank," *Nucleic Acids Research*, vol. 28, pp. 235–242, 2000.

[25] G. Neshich, W. Rocchia, A. L. Mancini, M. E. B. Yamagishi, P. R. Kuser, R. Fileto, C. Baudet, I. P. Pinto, A. J. Montagner, J. F. Palandrani, J. N. Krauchenco, R. C. Torres, S. Souza, R. C. Togawa, and R. H. Higa, "Javaprotein dossier: a novel web-based data visualization tool for comprehensive analysis of protein structure," *Nucleic Acids Research*, vol. 32, pp. W595–W601, 2004.

[26] N. Ahmed, T. Natarajan, and K. R. Rao, "Discrete cosine transform," *Computers, IEEE Transactions on*, vol. C-23, pp. 90–93, 1974.

[27] D. E. Goldberg, *Genetic Algorithms in Search, Optimization and Machine Learning*. Addison-Wesley Longman Publishing Co., Inc., 1989.

[28] I. M. Oliver, D. J. Smith, and J. R. C. Holland, "A study of permutation crossover operators on the traveling salesman problem," in *Proceedings of the Second International Conference on Genetic Algorithms on Genetic Algorithms and Their Application*. Hillsdale, NJ, USA: L. Erlbaum Associates Inc., 1987, pp. 224–230.