2019 IEEE International Conference on Systems, Man and Cybernetics (SMC)
Bari, Italy, October 6-9, 2019

# Feature selection with genetic algorithm for protein function prediction

Bruno C. Santos, Marcos W. Rodrigues, Cristiano L. N. Pinto, Cristiane N. Nobre, Luis E. Zárate
Pontifical Catholic University of Minas Gerais
{brunocs90, marcoswanderrodrigues}@gmail.com, cristiano@emge.edu.br, {nobre, zarate}@pucminas.br

*Abstract*— **Knowing the function of proteins is essential in several areas such as bioinformatics, agriculture, and others. The processes to determine protein function that is realized in laboratories are costly and require a long time to be done. Therefore, it is necessary to provide efficient computational models that aim to find the function of a protein. There are currently several kinds of researches that deal with the prediction problem of protein function. However, each of them presents a different methodology, employing different classifiers as well. Based on this problem, we propose a methodology using a multi-objective genetic algorithm with the classifier $k$-NN to select the best characteristics and then apply several classifiers such as Artificial Neural Network, SVM, Random Forest, and $k$-NN, in order to compare their performance in the same methodology. Our methodology found the best performance to be the $k$-NN classifier, with an f-Measure value of $86.07\%$.**

## I. Introduction

Proteins are macromolecules formed by the sequence of several amino acids linked by chemical bonds, and play a vital role in biological systems [1]. Due to the importance of this role, the knowledge of the protein function is fundamental for the understanding of several biological mechanisms.

With advances in genome sequencing techniques, the number of protein sequences explored has been dramatically increased. This increase brings with it the need to develop computational methods to facilitate and automate the process of identifying protein functions.

A significant amount of information about proteins and their structure is continuously made available in the public repositories of data. However, are unknown which attributes, or physico-chemical characteristics are most relevant to distinguish the actual function of the protein. Therefore, there is a need to find a subset of these existing features that best represent the function of the protein so that they can be classified accurately.

The protein function prediction problem has been tackled by several authors, such as [2], [3] and [4]. These studies propose methodologies to address the protein function prediction problem, applying different approaches and classification algorithms, aiming to improve the performance of the classification models. The authors of these works utilized information from the STING_DB database [5], one of the largest repositories of physico-chemical, structural and biological characteristics of proteins. The STING_DB database has several features extracted from all the protein structural levels (primary, secondary, tertiary, and quaternary), which are frequently used in classification models for this type of problem.

The studies of Santos [6] and Brito [7] proposed different approaches to manipulate the set of characteristics of the database UniProtKB/Swiss-Prot[1]. This database contains a greater sample variability and protein characteristics that allow better results for the proposed problem. The authors developed their models with an SVM classifier, with satisfactory results.

The SVM is known for its performance, and also for a high computational cost during the learning phase. It has disadvantages when applied to large datasets because its memory consumption can reach quadratic scales, relative to the size of the dataset and cubic scales. In [7], the authors used Factor Analysis (FA) [8] to reduce the dimensionality of the dataset. With this statistical technique, they obtained better results than those found in the literature.

It is important to note that the problem of protein function prediction is complex and precise techniques to solve this problem. Based on the work [4], that made use of the multiobjective genetic algorithm to select the best subset of characteristics, using a $k$-NN classifier, we propose to use the database that was the target of the work of Brito [7] using another technique of selection of characteristics and using different classifiers.

Therefore, as the main objective of this study, we sought to find the best characteristics through a feature selection mechanism based on a multiobjective genetic algorithm, using the k-NN classifier. Besides, our study differs from previous research, as it proposes a methodology for the protein function prediction process, which consists of finding a smaller subset of characteristics, thus allowing a better understanding of the model found and its replication to other databases. We also observed that the classifiers of Random Forest and Artificial Neural Networks (ANN) had presented satisfactory results in the literature. Thus, it is also the objective of this study to compare the performance of different classifiers, namely SVM, ANN, $k$-NN and Random Forest, for the prediction problem of protein function.

This article is organized as follows: Section II presents the main studies for protein function prediction that are related to our proposal. Section III describes our methodology, bringing the dataset description, the preprocessing phase,

*The authors thank CNPq (National Council for Scientific and Technological Development, Brazil), CAPES (Coordination for the Improvement of Higher Education Personnel, Brazil) and FAPEMIG (Foundation for the Support to the Researches in Minas Gerais, Brazil), and the Pontifical Catholic University of Minas Gerais, Brazil, for the support received.

[1] http://www.uniprot.org/uniprot/

978-1-7281-4568-6/19/$31.00 ©2019 IEEE

and the utilized methods. Section IV presents a summary of the utilized classifiers, as well as the architectures and parameter adjustments performed. In Section V, we offer our results and a discussion regarding the comparison of different classifiers. Finally, Section VI presents the conclusions and future studies suggestions.

## II. Related Works

Leijoto et al. [9] employed a standard Genetic Algorithm (GA) to select 11 physico-chemical characteristics from the STING_DB. The values of each attribute were normalized, and they applied a Discrete Cosine Transform (DCT) to handle the differences in the amino acid chain length. The authors considered the 75 first coefficients of the transform as the most significant. To validate their approach, they utilized an SVM classifier optimized with the Grid-Search parameter adjustment technique to choose the value for the Cost and $\gamma$ parameters of the classifier. The authors performed experiments adding the amino acid frequency to the DCT coefficients, increasing the classifier's average recall and precision to $68\%$ and $71\%$, respectively. They claimed that the GA was limited to process 50 generations and 10 individuals, due to the high computational processing demands.

In [4], a multi-objective GA methodology was proposed to select features from the STING_DB for a protein function prediction task. After the physico-chemical attributes had been selected, the dataset was enriched with other features, and an SVM classification model was built. Their methodology obtained an average precision of $77,3\%$ and f-Measure of $72,7\%$. However, the GA having utilized the SVM classifier, if they were to adjust its parameters at each execution, that would make the process very computationally expensive. Therefore, the authors chose not to perform parameter optimization during the GA evolution, which limited their classification performance.

In [6], the authors performed an investigation of different combinations of protein information using the four structures (primary, secondary, tertiary and quaternary), and an approach using the DTC (Discrete Transformation Cosine) so that the input vectors were the same size with Sting_DB [5] database. Their results obtained using the SVM classifier were 74.4% of f-measure. They concluded that only the Sting_DB was not sufficient for the prediction problem of protein function.

The work of Brito [7] used the UniProtKB/Swiss-Prot database and made a comparison of the results without the reduction of dimensionality and with a reduction, using the Factor Analysis. The results obtained without dimensionality reduction were better with f-Measure 79.77%. Its results were better than the methodology of Santos [6] on average of 5.35% for f-measure.

In this paper, we present a methodology that can be easily reproduced in addition to comparing the performance of several classification algorithms for the task of predicting protein function.

## III. Methodology

The methodology proposed in this article is presented in Figure 1. It involves several phases: the description of the dataset and its preprocessing, feature selection based on the GA and finally, the comparison of different classifiers followed by a validation of these results.

It is important to note that there are two main tasks in the methodology. The first task is to find the best subset of physico-chemical characteristics through a multi-objective GA. For this, we use the $k$-NN classifier because it requires fewer computational resources compared to $k$-NN, random forest, SVM, and ANN. We then used different classifiers to validate if the generated model is adequate for the prediction problem of protein function, so we compared the performance of classifiers.

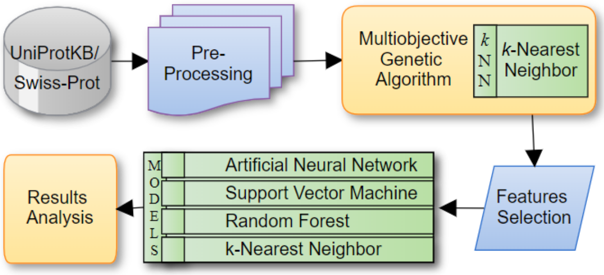


Fig. 1. The proposed methodology

### A. Dataset construction

The database used in this work, to validate the results obtained by the proposed methodology, was extracted from UniProtKB/Swiss-Prot, produced by the UniProt Consortium, Swiss Institute of Bioinformatics (SIB) and Protein Information Resource (PIR). UniProtKB/Swiss-Prot is a repository of protein sequence and function information that is annotated and manually revised [10]. The database has six different classes of enzymes. Table I shows the number of enzymes and amino acid chains.

TABLE I

Amount of enzymes by class

| Type | Enzymes | Chain |
|---|---|---|
| Hydrolases | 5286 | 5888 |
| Isomerases | 834 | 898 |
| Lyases | 1309 | 1381 |
| Ligases | 920 | 1050 |
| Oxidoreductases | 2684 | 3003 |
| Transferases | 4602 | 5055 |
| **Total** | **15635** | **17275** |

This database went through a preprocessing process employed by [7] that divided it into four groups that together make up a total of 424 attributes available for feature selection.

(a) Primitive structural features of a protein: Information regarding protein structure without transformations. The

Protein Data Bank (PDB)[2] makes this information available. The number of information is 33.

(b) Descriptive features of the residues sequence: It presents information regarding the sequence sequences of residues, concerning the frequency of occurrence of the following classes of amino acids: Hydrophobic, Neutral, and Polar. The quantity of the group is 21.

(c) Amino acid chain features processed by tools form the scientific community: In this type, we have information about the tertiary structure of the protein, made available by CSM[3] and BioJava[4]. Besides, there is information from EMBOSS Pepstats on physico-chemical characteristics. Quantity of 204.

(d) Characteristics based on sequence alignment: In this group, we have details on making comparisons about the sequence alignment of residues. This information was extracted from Interpro[5]. The number of 166 characteristics.

All this information makes up a total of 424 characteristics [7]. Thus formed the database we need to carry out a next step of the process of data mining that is the preprocessing.

### B. Preprocessing

During the preprocessing phase, we first found and removed data redundancies. After that, we calculated the Pearson correlation between the quantitative variables in the dataset and found that several characteristics were strongly correlated. We chose to eliminate characteristics that presented a correlation above 0.9 to another characteristic in the dataset, which reduced the number of features from 424 to 275.

Following the transformation, we applied a normalization process to all characteristics to make their values vary in the interval [0,1]. This adjusting is necessary to avoid that some variables with a different scale influence the classifier, introducing bias. Equation 1 shows the Min-Max normalization function used.

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

Where: $X$ = represents the value to be normalized;
$max$ = is the largest value of the variable;
$min$ = represents the smallest value of the variable;
$X'$ = corresponds to the normalized value.

### C. Multi-Objective Genetic Algorithm

After the preprocessing phase, we applied the multi-objective GA algorithm Non-dominated Sorting Genetic Algorithm II (NSGA-II) [11], for feature selection. This algorithm implements concepts of dominance, and its choice was motivated by it being one of the main multi-objective algorithms in the literature.

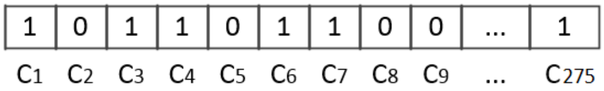
*1) Fitness:* In this study we considered two objectives for the algorithm: a) the model should have a small error percentage, increasing its reliability, and b) the model should have a small subset of features, so it is simplified. Thus, the GA algorithm was given the following directives:

(a) Maximize the average f-measure values of the $k$-NN classifier, where:

$$\text{f-Measure} = \frac{2 \times Precision \times Recall}{2 \times (Precision + Recall)} \tag{2}$$

(b) Minimize the number of selected features.

*2) Representation:* The representation we used for the chromosomes in the GA was a binary vector of 275 positions, each of them representing one of the characteristics. The positions in the vector can take the values 0 and 1, representing the absence or presence of that characteristic's features in the dataset. Figure 2 illustrates the representation of an individual in the GA.



| 1 | 0 | 1 | 1 | 0 | 1 | 1 | 0 | 0 | ... | 1 |
|---|---|---|---|---|---|---|---|---|---|---|
| $C_1$ | $C_2$ | $C_3$ | $C_4$ | $C_5$ | $C_6$ | $C_7$ | $C_8$ | $C_9$ | ... | $C_{275}$ |

Fig. 2. Representation of the individual classes

*3) Feature Selection:* Based on preliminary experiments the best parameters for GA were found: Population = 100, Generations = 300, Crossover ratio = 0.8 and Mutation ratio = 0.01. Using these parameters, we performed our experiments with 5 different seeds, using the $k$-NN classifier and performing the adjustment of the nearest neighbor number parameter.

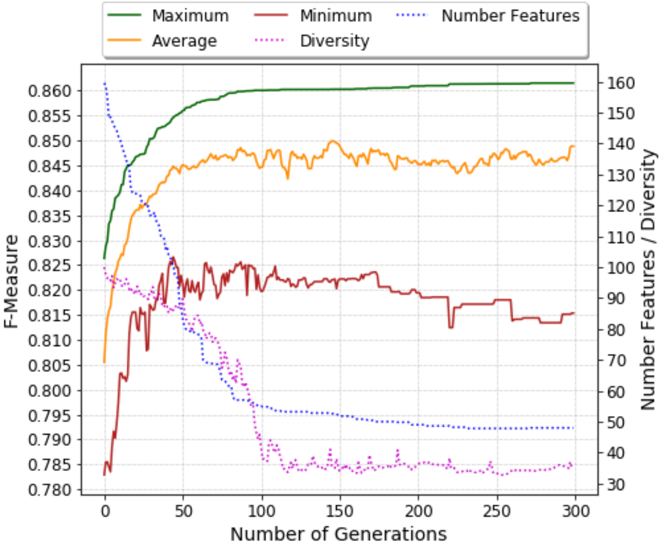


Fig. 3. Average of execution of 5 experiments

The number of neighbors $k$ was varied in a range from 1 to 8 in our experiments, to find the value that would yield the best classification performance for the selected feature subset. We observe in Figure 3 the diversity metric, which

[2] http://www.rcsb.org/pdb
[3] http://homepages.dcc.ufmg.br/dpires/csm
[4] https://github.com/biojava/biojava-tutorial/tree/master/structure
[5] http://www.ebi.ac.uk/interpro/

indicates the number of individuals that are different from their parents, which shows that the GA converged next to the 200 generations. Besides, f-Measure got its best values of around 225 generations. The best mean values were obtained with average f-measure of 86.07% using 49 attributes in the dataset.

The GA algorithm had as its output a set of 49 physico-chemical characteristics as the best set to classify the six enzyme classes used in this study. These number of characteristics' groups are presented in Table II.

One of the objectives of our study is to compare four different classification algorithm to find the one with the best f-Measure result for the protein function prediction task. The classification algorithms utilized in the experiments are described in the following sections. We employed a 10-fold cross-validation process to ensure confidence in the accuracy of the results. The cross-validation process is a standard way to adjust classification bias in the classification model.

## IV. Classifiers and parameters

In this Section, we present the classifiers that were used to validate the proposed methodology, as well as the adjustments of its parameters for a better predictive accuracy that will be described next.

### *A. k-Nearest Neighbour*

The $k$-Nearest Neighbour ($k$-NN) is another supervised learning algorithm. It can be applied to both regression and classification problems but is more commonly used for the latter. The $k$-NN is non-parametric because it makes no assumptions on the data distribution, and the structure of the model is generated from the data itself.

The $k$ value represents the number of neighbors used to consider when classifying an instance. A high value of $k$ can result in a less precise result, but a value that is too low makes the model more sensitive to outliers. For example, if $k = 1$, the instances in the test dataset will receive the same class label as the closest instance from the training dataset. For our dataset and problem (in the classifier comparison phase), we found that the best course was to implement our $k$-NN algorithm with $k = 2$, as proposed by Aha [12].

The $k$-NN predicts a class label by finding the most similar instances in the training set. The most popular distance metrics are Euclidean, Manhattan, Hamming, and Minkowski distances. We utilized the Manhattan distance due to its simplicity, avoiding quadratic calculations.

### *B. Support Vector Machine*

*Support Vector Machine* (SVM) is a supervised learning technique used for the classification task. The algorithm tries to find the hyperplane that maximizes the separation margin between the instances of different classes in a dataset.

For our study, we used the Radial Basis Function kernel, because it is a simple and frequently utilized function for SVM classifiers. The kernel function is responsible for mapping the data and find the hyperplane separating the classes. Also, this kernel function is used to improve the accuracy of the classifier, along with the following parameters:

- **Regularization ($C$):** This is used to reduce the chance of incorrect classifications in the training set. High values of $C$ mean a smaller margin for the generated hyper-plan, maximizing accuracy. Typically, the SVM optimization tries to find a more significant margin for the hyper-plan even if that means more training errors.
- **Gamma ($\gamma$):** This parameter defines the influence the training instances have over others they are distant too. A small value of $\gamma$ means that the distant points in the hyper-plan will also be considered, whereas larger values will make the algorithm consider only the closest instances.

In order to estimate these parameters, we adopted the Grid-Search strategy [13]. The utilized values were Cost $= 32.0$ and $\gamma = 3.174802$.

### *C. Random Forest*

The Random Forest (RF) algorithm [14] is an ensemble learning algorithm, that combines several models called weak predictors, in order to create a reliable final prediction.

Our implementation uses the same algorithm proposed by Breiman [15], which needs the following parameter adjustments:

- **Max_depth**: Indicate the maximum depth of each decision tree in the forest.
- **N_estimators**: Is the number of decision tree that will be created. This number should not be too small, to guarantee the variability that makes the ensemble learning reliable.
- **Min_samples**: The minimum number of instances to be used in a node partition.
- **Max_features**: The number of randomly selected features to be used on each decision tree.
- **Criterion**: Quality function used in the node split function. The supported criteria are the Gini Index, a measure of impurity, and Entropy, a measure of information gain.

After executing the Grid-Search, we obtained the following parameter values: **Max_depth**: None, **N_estimators**: 100, **Min_samples**: 3, **Max_features**: 10, **Criterion**: entropy.

### *D. Artificial Neural Network*

An Artificial Neural Network (ANN) is organized in interconnected artificial neuron layers: the input layer, the hidden layers, and the output layer. It is possible to utilize a large number of neurons, expanding the number of hidden layers, magnifying the complexity of the ANN.

Based on different experiments, we find the following model for the neural network with a multi-layer perceptron architecture, for the prediction problem of protein function.

- **Number of layers and neurons:** The first is the input layer with 49 neurons referring to each feature in the

TABLE II

CHARACTERISTICS SELECTED BY THE GENETIC ALGORITHM

| Group | Amount | Features |
|---|---|---|
| A | 11 | volume - quantidadeCobre - quantidadeFerro - quantidadeFAD - quantidadeNAD - quantidadeFolhas - quantidadeOutrasEstruturas - quantidadeHelices - areaSuperficieAcessivelTotal - % area residuo G - % area residuo R. |
| B | 9 | areaSuperficieDimensaoFractalTotal - % D - % G - % K - % L - % M - % P - % S - % W |
| C | 21 | Desc. Local - % H->N, Desc. Local. % P->N - Desc. Local. % N->P - Desc. Local. % Neutro, Desc. Local. % Polar - 100.0_Hidrofobico - 100.0_Neutro - 25.0_Hidrofobico - 25.0_Neutro - 50.0_Hidrofobico - 50.0_Neutro - 50.0_Polar - 75.0_Hidrofobico - 75.0_Polar - Emboss1 - Emboss3 - Emboss4 - Emboss10 - Emboss19 - Emboss25 - Emboss31 |
| D | 8 | Active_site_IPR008266 - Domain_IPR011009 - Domain_IPR014721 - Domain_IPR014727 - Domain_IPR014748 - Domain_IPR014882 - Domain_IPR029064 - Family_IPR004441 |

dataset. Following, there are 3 hidden layers in the network, with their neuron numbers being defined by the number of inputs in the dataset n, as 2n + 1 [16], totaling 99 neurons for each. The output layer has only 6 neurons, that represent each of the target enzyme classes.

- **Activation function:** The activation function in the hidden layers was the Hyperbolic Tangent (Tanh) function. The output layer represents the canonical responses corresponding to the classes, and each neuron is activated proportionally through the softmax activation function.
- **Learning method:** We employed the back-propagation strategy with the Adam optimization function in the training of our ANN.

## V. RESULTS AND DISCUSSION

In the next phase of our methodology, we compared the classification performance of the $k$-NN, RF, SVM, and ANN classifiers, for predicting one of six enzyme classes in our dataset. We applied a 10-fold cross-validation process, and analyzed the f-Measure metric.

The results obtained are shown in Table III. The $k$-NN classifier got the best results, with 86.07% average f-Measure, and the SVM, RF, and ANN had very similar results for the experiment with 85.17%, 85.80%, and 80.67% respectively. We noticed a slight gain of the k-NN classifier over the other classifiers around 0.9%, 0.2% and 5.4% on the SVM, Random Forest, and ANN. One of the reasons for this gain is that the genetic algorithm has its fitness based on the $k$-NN classifier, which consequently produces a set of data for this classifier. However, this fact also shows that although the k-NN classifier was used to accelerate the genetic evolution process, the chosen dataset was useful for the other classifiers, a fact proven in the metrics found for each of these classifiers.

After the results found with the feature selection of the genetic algorithm, we performed a comparison with the work that [7] that used this same dataset. Figure 4 presents the results of the previous work and the proposed methodology, considering, thus, the best set of values that were found using the $k$-NN classifier.

We can note that the average results found were higher at 6.3%. Note that the proposed methodology was superior in five classes of protein (Hydrolase, Isomerase, Lyase, Oxidoreductase, Transferase) and lower in one (Ligase). There was also a significant increase in the Isomerase and Lyase classes with an average gain of 12.42% and 17.36% respectively. Also, using the genetic algorithm to select features we have a greater simplification of the generated model, which makes possible the understanding of the results found and the reproduction of the methodology proposed for future works.

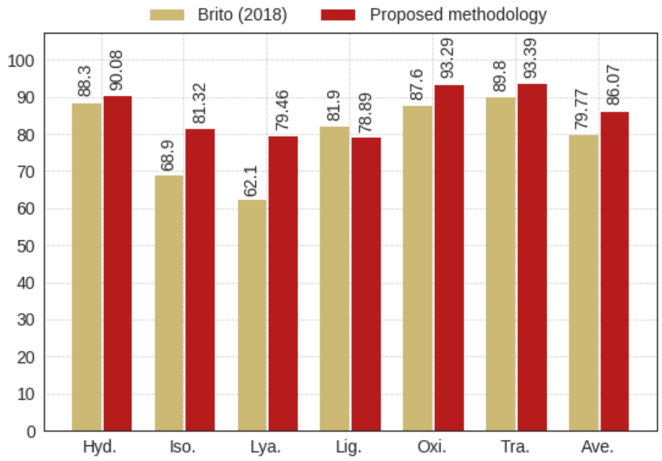


Fig. 4. Comparison of f-measure the two methodologies

It is worthwhile to note that the best model for each of the classifier was adequately adjusted for the protein function prediction problem. This care demonstrates the consistency of our methodology for this application and highlights the importance of parameter adjustments for classifiers.

## VI. CONCLUSIONS

In this study, we presented an approach for the protein function prediction problem. The proposed methodology was composed of two phases. In the first phase, we applied a multi-objective Genetic Algorithm based on the $k$-NN classifier, to find the best subset of physico-chemical characteristics to use in the dataset for the prediction problem. The second phase involved constructing the classification models with four different algorithms. Finally, we compared the performances of these algorithms, namely Artificial Neural Networks, Support Vector Machines, Random Forests, and $k$-Nearest Neighbours, for the protein function prediction task.

Regarding the experiments in the feature selection phase, the $k$ parameter of the $k$-NN classifier was adjusted to find the best classification results. Thus, the results were

TABLE III

Comparison of f-Measure for both strategies

| Classifiers/<br>Class Protein | f-Measure | | | | | | |
|---|---|---|---|---|---|---|---|
| | Hydrolase | Isomerase | Lyase | Ligase | Oxidoreductase | Transferase | **Average** |
| *Neural Network* | 86.92 (± 0.010) | 76.94 (± 0.039) | 64.98 (± 0.028) | 73.41 (± 0.029) | 90.76 (± 0.014) | 90.96 (± 0.008) | **80.67 (± 0.099)** |
| *SVM* | 89.37 (± 0.008) | 79.59 (± 0.028) | 78.51 (± 0.031) | 78.05 (± 0.028) | 92.66 (± 0.011) | 92.83 (± 0.010) | **85.17 (± 0.069)** |
| *Random Forest* | 87.75 (± 0.008) | 77.15 (± 0.034) | 82.10 (± 0.027) | 80.60 (± 0.025) | 92.30 (± 0.011) | 94.87 (± 0.009) | **85.80 (± 0.067)** |
| *k-NN* | 90.08 (± 0.007) | 81.32 (± 0.020) | 79.46 (± 0.022) | 78.89 (± 0.026) | 93.29 (± 0.078) | 93.39 (± 0.008) | **86.07 (± 0.065)** |

satisfactory, which leads us to conclude that the information used in the Uniss/Prot database is sufficient to predict protein function. Contrary to what was demonstrated in Santos [6], which concluded that only the physicochemical characteristics of the database STING_DB were not sufficient for the prediction problem of protein function.

In addition, our proposal was able to improve a previous work [7], where the authors proposed to perform the prediction of protein function with the reduction of dimensionality using factor analysis and without it. Our methodology has found superior results which shows that it is better adjusted for the problem. With this, we can select a smaller set of data, which provides a better clarity in understanding the problem.

For future studies, we suggest the choice of new attributes to refine the problem of protein function prediction using the genetic algorithm. Finally, we also suggest new approaches to the selection of biological characteristics to be added to the dataset.

## References

[1] B. Alberts, A. Johnson, J. Lewis, M. Raff, K. Roberts, and P. Walter, *Molecular Biology of the Cell*, 5th ed. Garland Science, Nov. 2007. [Online]. Available: http://www.amazon.com/exec/obidos/redirect?tag=citeulike07-20\&path=ASIN/0815341059

[2] 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.

[3] P. D. Dobson and A. J. Doig, "Distinguishing enzyme structures from non-enzymes without alignments," *Molecular Biology*, vol. 330, pp. 771–783, 2003.

[4] B. C. Dos Santos, C. N. Nobre, and L. E. Zárate, "Multi-objective genetic algorithm for feature selection in a protein function prediction context," in *2018 IEEE Congress on Evolutionary Computation (CEC)*, July 2018, pp. 1–6.

[5] 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.

[6] G. O. Santos, C. N. Nobre, and L. E. Zárate, "Biological characteristics evaluation to predict enzyme classes with support vector," *International Journal of Bioinformatics Research and Applications*, 2018, (To be published http://http://www.inderscience.com/info/ingeneral/forthcoming.php?jcode=ijbra).

[7] L. H. Brito, A. L. C. V. Lara, L. E. Zárate, and C. N. Nobre, "Improving the quality of enzime prediction by using feature selection and dimensionaly reduction," in *2019 International Joint Conference on Neural Networks (IJCNN)*, 2019, (To be published).

[8] R. L. Peterson, C.-F. Chien, and R. Xing, "Factor analysis in data mining," in *Encyclopedia of Data Warehousing and Mining*, 2005.

[9] L. Fernandes Leijoto, T. Assis De Oliveira Rodrigues, L. Zárate, and C. Nobre, "A genetic algorithm for the selection of features used in the prediction of protein function," in *Bioinformatics and Bioengineering (BIBE), 2014 IEEE International Conference on*. Computer Society Digital Library, Nov 2014, pp. 168–174.

[10] U. Consortium and M. Magrane, "UniProt Knowledgebase: a hub of integrated protein data," *Database*, vol. 2011, 03 2011. [Online]. Available: https://doi.org/10.1093/database/bar009

[11] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: Nsga-ii," *Trans. Evol. Comp*, vol. 6, no. 2, pp. 182–197, Apr. 2002. [Online]. Available: http://dx.doi.org/10.1109/4235.996017

[12] W. Aha, D. Kibler, and M. K. Albert, "Instance-based learning algorithms," *Machine Learning*, vol. 6, pp. 37–66, 01 1991.

[13] C. wei Hsu, C. chung Chang, and C. jen Lin, "A practical guide to support vector classification," 2010.

[14] T. K. Ho, "Random decision forests," in *Proceedings of the Third International Conference on Document Analysis and Recognition (Volume 1) - Volume 1*, ser. ICDAR '95. Washington, DC, USA: IEEE Computer Society, 1995, pp. 278–. [Online]. Available: http://dl.acm.org/citation.cfm?id=844379.844681

[15] L. Breiman, "Random forests," *Machine Learning*, vol. 45, no. 1, pp. 5–32, Oct 2001. [Online]. Available: https://doi.org/10.1023/A:1010933404324

[16] M. Li and P. M. Vitányi, "Chapter 4 - kolmogorov complexity and its applications," in *Algorithms and Complexity*, ser. Handbook of Theoretical Computer Science, J. V. LEEUWEN, Ed. Amsterdam: Elsevier, 1990, pp. 187 – 254.