Feature selection and comparison of classifiers for predicting protein class

Bruno C. Santos, Cora Silberschneider,
Marcos W. Rodrigues, Cristiano L. N. Pinto, Cristiane N. Nobre, Luis E. Zárate

Pontifical Catholic University of Minas Gerais, Brazil
brunocs90@gmail.com, cora.silberschneider@sga.pucminas.br,
marcoswanderrodrigues@gmail.com, cristiano@emge.edu.br, {nobre,zarate}@pucminas.br

Abstract. Knowing the function of proteins is essential for understanding several biological systems. The experiments in laboratory to determine protein class are costly and require a long time to be done. Therefore, it is necessary to provide efficient computational models to identify the class to which a protein belongs. Nowadays, a significant volume of information regarding proteins and their structure is continually being made available in public data repositories. For example, the STING_DB database has a lot of information extracted from all protein structural levels (primary, secondary, tertiary, and quaternary), which are frequently used in classification models for this type of problem. However, it is unknown which physical-chemical properties are the most relevant ones to contribute to the prediction of the class. Therefore, there is a need to identify the subset of more suitable properties. In this work, we propose an approach based on a multi-objective genetic algorithm with the classifier *k*-NN to select the best physical-chemical properties. Our strategy uses a multi-objective genetic algorithm to obtain a smaller subset of features that contribute significantly to the prediction problem. To improve the prediction's performance, we choose to perform a post enrichment process, then we compare the performance of our methodology with several classifiers: ANN, SVM, Random Forest, and *k*-NN. Our method achieved an average F-measure value of 70.22% with the Random Forest classifier. Finally, a comparative analysis, with statistical significance, shows the relevance of our approach in relation to other methodologies.

Categories and Subject Descriptors: H.2.8 [**Database Applications**]: Data Mining; I.2.6 [**Artificial Intelligence**]: Learning; J.3 [**Life and Medical Sciences**]: Biology and genetics

Keywords: Feature Selection, Classifiers, Multi-Objective Genetic Algorithm, Protein Prediction

1. INTRODUCTION

Proteins are macromolecules formed by chains of amino acids bonded by chemical connections, which perform a vital role in biological systems [Alberts et al., 2007]. There are many kinds of proteins according to its function, for example: transport, regulatory, defense, enzymes, structure, among other functions [Alberts et al., 2013]. Among these, the enzymes considered for this work are responsible for increasing the rates of chemical reactions and are classified in six classes: Hydrolases, Oxidoreductase, Transferases, Isomerases, Lyases e Ligases.

It is essential to highlight that a protein can perform several types of functions, according to how they are organized on a structural level. Some authors name these proteins as 'promiscuous proteins' [Tawfik and S., 2010]. The literature has studied these types of protein, but the existence of this option on the current classification can raise adjustment problems on the learning model. In this work, the enzymes are considered as proteins that perform only one function and belong to only one class.

Due to the importance of proteins in living beings, knowledge of protein class is fundamental for understanding several biological mechanisms. With the advances of genome sequencing techniques, the number of explored protein sequences has been greatly increased. Nowadays, a significant volume of information regarding proteins and their structure is continually being made available in public repositories. However, Nadzirin and Firdaus-Raih [2012] and Szalkai and Grolmusz [2018] highlight the difficulty to identify the function and the class of a protein. Facing this scenario, it is important the proposal of computacional methods to automatize and facilitate the process to identify the function and class of a protein. However, there still isn't a computacional approach capable to predict with accuracy the class of a big quantity of proteins. With that, the problem of prediction of a protein's class remains as a latent problem and a challenge for the molecular biology and bioinformatics.

The protein class prediction problem has been tackled by several authors, such as [Dobson and Doig, 2003], [Borro et al., 2006], [Santos et al., 2018a], [Santos et al., 2018b] and [Santos et al., 2018c]. These studies propose methodologies to address the prediction problem, applying different approaches and classification algorithms. The authors, cited previously, used information from the STING_DB database [Mancini et al., 2004], one of the largest repositories of physical-chemical, structural, and biological properties of proteins. The STING_DB database has several information extracted from all protein structural levels (primary, secondary, tertiary, and quaternary), which can be used in classification models for this type of problem. However, it is still unknown which physical-chemical properties are the most important ones to distinguish a protein class. Therefore, there is a need to identify the subset of properties that represent a protein and allow to better identify its class.

In [Santos et al., 2018a], [Santos et al., 2018b], and [Santos et al., 2018c] have proposed different approaches to use the physical-chemical properties from the STING_DB database. For example, in [Santos et al., 2018c], the authors considered ten (10) properties (of over 300 available) that were used in previous work [Dobson and Doig, 2003]. In [Santos et al., 2018b], the authors used a multi-objective architecture based on genetic algorithms (GA) to perform feature selection using a SVM classifier within the wrapper mechanism aiming to find a subset of relevant properties.

A limiting aspect of the chosen methodology in [Santos et al., 2018a] and [Santos et al., 2018b] is the use of the SVM classifier inside the wrapper architecture, that not only reaches quadradic scales in the memory usage but also cubic scales in the computational time to find a solution [Graf et al., 2004]. Besides, it is necessary the adjust of the hyperparameters Cost and Gama (of the kernel function) for each new subset of the training involved in the evolutive process of the wrapper architecture. To get around these difficulties, the authors fixed the Cost and Gama hyperparameters and also fixed the size of the population and the number of generations (of the genetical evolution) limiting the research space. The authors also suggest to use other types of classifiers, that dont request a costly parametric optimization, such as the classifier k-NN, considered in this paper.

The first objective of this work is to answer a question: what are the physical-chemical properties of the STING_DB database that can contribute to the protein class prediction? To answer this question, we consider a feature selection process using GA and k-NN to increase the search space and avoiding the setting of the SVM hyperparameters as in approaches early mentioned.

In [Santos et al., 2018c], the authors establish that it is always necessary the enrichment of dataset with other data sources. In order to improve the prediction performance, as second objective we are performing a posteriori enrichment process with new features (the amino acid frequency, alpha-carbon frequency, and statistical descriptors from the primary structure). After this procedure, we evaluated the performance of four classifiers: k-NN, SVM, Neural Networks and Random Forest. These classifiers were chosen because they were used with satisfactory performance in the context of protein class prediction.

It is important to highlight that our first objective is to identify a subset of physical-chemical properties, from STING_DB database, that can contribute to protein class prediction and discard other less relevant. In this work, for the first objective, we prefer not to make only one dataset, containing both data from physical-chemical properties as well as enrichment data. This is done to not mix the data effects from different sources, and to compare the results with works that used only the STING_DB features.

It is important to note that, Santos et al. [2018a] and Santos et al. [2018b] applied the Principal Component Analysis - PCA for the dimensionality reduction of the dataset (initially with 482 features). PCA can be a limiter to reproduce the same results and achieve the same performance in the classifiers when only the replicated with different training data set, in other words, with a different data source. Note that, the use of PCA implies the use of principal components which are orthogonal representations of a specific data set. Varying the data set, even slightly, can lead to significant changes in the principal components. Considering the usage the available model, a new entry, different from the representativeness of the original data, can lead to misclassification. Importantly, the protein classification problem is complex and we usually deal with scarce data and probably unrepresentative of the domain of the problem. In this work, we removed the PCA application stage to simplify the proposed methodology, but without compromising the results obtained.

This article is organized as follows: Section 2 presents the main works related to the proposed theme; Section 3 presents the methodology used in the development of the proposal; Section 4 discusses the results obtained; Finally, Section 5 presents the conclusions and suggestions for future improvements. Also, we available the Supplementary Material[1] presents the theoretical concepts, describing the protein and the execution of the genetic multi-objective algorithm.

2. RELATED WORKS

There are several approaches to generate computational models to predict protein class using information from their three-dimensional structure and/or chains of amino acid residues.

The approaches based on chains of amino acid residues can be categorized into three groups, [Pandey et al., 2006]: 1) based on *sequence homology*, in which the known protein class is transferred to a protein with unknown class based on the similarity of the primary sequence; 2) based on the *subsequence*, which considers that many times, only a portion of the chain of amino acids is crucial for a protein to carry out its function [Liu and Califano, 2001]; and 3) based on *physical-chemical properties*, which consider that a protein can be represented by a set of features, which may include their structures [Dobson and Doig, 2004; Resende et al., 2012], physical-chemical properties [Borro et al., 2006; Leijoto et al., 2014] and amino acids composition [Kumar and Choudhary, 2012].

Borro et al. [2006] used the physical-chemical properties selection process of the STING_DB database 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. With the proposed methodology, average precision of about 53.9% was obtained, using Bayesian networks.

In [Leijoto et al., 2014], the authors employed a standard Genetic Algorithm (GA) to select 11 physical-chemical properties from the STING_DB database. The values of each attribute were normalized, and the authors applied a Direct Cosine Transform (DCT) to handle the differences in the length of chains of amino acid residues that compose protein structure. The authors considered the 75 first coefficients of the transform as the most significant. To validate their approach, they used a SVM classifier optimized with the Grid-Search hyperparameters adjustment technique, to choose the value for the Cost and γ hyperparameters of the classifier. The authors performed experiments adding the amino acid frequency to DCT coefficients, increasing the classifier's average recall and precision

[1] Supplementary Material Avaliable at: `https://drive.google.com/open?id=1ZI72cUoisb1fq1YtCGgJcbJKDFoBFKgq`

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 [Santos et al., 2018b], a multi-objective GA was proposed to select features from the STING_DB. After the physical-chemical properties had been selected, the dataset was enriched with other 202 features in order to obtain the prediction model based on SVM classifier. The adopted methodology achieved an average precision of 77.3% and F-Measure of 72.7%. It is important to highlight that the authors applied PCA to reduce the dimensionality of the dataset. But, as mentioned before, this procedure reduces the capacity of model generalization and compromises the reproduction of results.

Santos et al. [2018a] used a multi-objective GA for feature selection. The authors considered the k-NN classifier to raise the problem of hyperparameters adjustment in the SVM faced in [Santos et al., 2018b]. After performing the feature selection, they added attributes to enrich the dataset. The methodology obtained an average precision of 72.9% and F-Measure of 68.3%. The obtained values were slightly inferior to those in the literature. However, there was a considerable gain in processing times, with approximately 90% reduction. This gain was due to the use of the k-NN algorithm during the evolution process. One limitation of that work is that the authors fixed the number of neighbors $k = 1$ in the classifiers, during the evolution, which may have hindered the results.

Table I presents a comparison between the previous methodologies used as reference for the development of this work. Note that each work opted for some strategy to solve the proposed problem, such as: number of features, differed in population size, number of generation used, number of experiments, among others.

In addition, it is noted that all studies cited in this section, except the work of Borro et al. [2006], used GA and addressed the protein class prediction problem to propose a methodology that would solve this problem. Some of them limited the number of properties [Leijoto et al., 2014], while others employed methodologies that are not easy to reproduce [Santos et al., 2018b] and [Santos et al., 2018a], employing principal component analysis, which requires the eigenvalues previously used for its accurate reproduction in other data. In this article, we present a methodology that can be easily reproduced and compare the performance of several classification algorithms for a protein class prediction task.

Table I: Summary of Related Work

	Fernandes Leijoto et al. [2014]	**Santos et al. [2018b]**	**Santos et al. [2018a]**
DataSet	STING_DB	STING_DB + Enriched	STING_DB + Enriched
Features	338	334	334
Tuning Techniques	GA	GA	GA
Coding	Java (Weka)	Python (Deap/Sklearn)	Python (Deap/Sklearn)
Optimization	Mono-objective	Multi-objective	Multi-objective
Objective	Max(Precision)	Max(Precision), Min(features)	Max(Precision), Min(features)
Classifier GA	SVM	SVM	k-NN
Population	10	100, 300, 500	500
Generation	50	100, 300	200, 300
Crossover	65%	65%, 70%, 75%, 80%	70%, 80%, 90%
Mutation	1%	1%, 5%, 10%	1%
Number Experiment	-	600	150
Evaluation Measure GA	Precision	Precision	Precision
Feature Reduction	-	PCA	PCA
Classifier	SVM	SVM	SVM
Evaluation Procedure	10-CV	10-CV	10-CV
F-Measure Average	69.2%	72.7%	68.3%

3. METHODOLOGY

The methodology proposed in this article is shown in Figure 1. It involves several phases: the construction of the dataset, its preprocessing, the standardization of example (chains of amino acid residues) sizes, feature selection based on the GA and dataset enrichment. Finally, a comparison of different classifiers followed by the analysis of results is presented.

It is important to note that there are two main tasks in our methodology. The first task is to find the best subset of physical-chemical properties, of the STING_DB database, through a multi-objective GA. For that, we utilized the k-NN classifier which demands less computational effort when compared to Random Forests, SVMs, and ANNs. As mentioned, a protein class prediction problem is complex and requires as much relevant biological information as possible. Hence, the second task in the methodology is to enrich the dataset with other data sources (as suggested by Santos et al. [2018c]): the amino acid frequency, alpha-carbon frequency, and extracting statistical descriptors from the primary structure. Finally, comparing the performances of different classifiers.

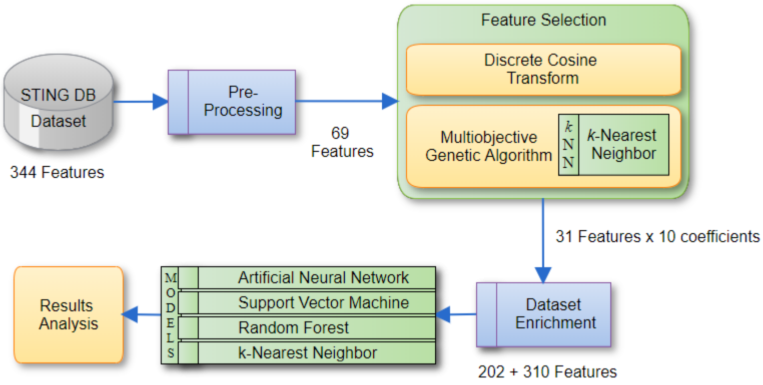


Fig. 1: The proposed methodology

3.1 Dataset construction

We used the STING_DB[2] to extract the physical-chemical properties from the set of six classes of enzyme investigated in this research: Oxidoreductases, Transferases, Hydrolases, Lyases, Isomerases, and Ligases. These are the same classes investigated by Dobson and Doig [2004], Borro et al. [2006], Leijoto et al. [2014], and Santos et al. [2018c].

The STING_DB is a data repository provided by the laboratory of computational biology, from the EMBRAPA institution, that has a set of software for data visualization and macromolecule analysis. The STING_DB has 344 physical-chemical properties for each amino acid of the chains that compose the enzyme. The number of enzymes and the number of chains used in our study are presented in Table II.

In [Santos et al., 2018c], the authors used this same database, which underwent a cleaning process, where the enzymes with a Summary PDB ASTRAL Check Index (SPACI) score[3] less than 0.3 were eliminated. The SPACI provides a numeric score that measures reliability and precision of a structure determined by crystallography in a PDB file. The SPACI score includes three components: the quality of the experimental data, how well the model fits the collected data, and the theoretical quality of the model considered. The enzymes were compared with the information contained in the *Protein Data Bank*[4] (PDB) [Berman et al., 2000], which made it possible to observe that some of these enzymes

[2] Available at: `https://www.cbi.cnptia.embrapa.br/SMS/index\_s.html`
[3] Available at: `https://scop.berkeley.edu/astral/spaci/ver=2.04`
[4] Available at: `http://www.rcsb.org/pdb/home/home.do`

were classified in a new class and therefore were reorganized. With this, we had a reduction in the number of enzymes used in this research, as shown in the last columns of Table II. Thus, in this work, to build the classification model, the 490 chains of amino acid residues were considered.

Table II: Amount of enzymes per class

	Proteins used by Dobson and Doig		Proteins after cleaning process	
Type	Protein	Chain	Protein	Chain
Hydrolases	160	312	122	162
Isomerases	51	89	35	56
Lyases	60	131	43	61
Ligases	20	22	15	16
Oxidoreductases	79	124	52	78
Transferases	128	162	82	117
Total	**498**	**840**	**349**	**490**

3.2 Dataset preprocessing

During the preprocessing stage, we calculated Pearson's correlation between quantitative variables in the dataset to find redundant data. We found that several physicochemical properties were strongly correlated. We reduced properties that presented a correlation above 0.9 to another property in the dataset, which reduced the number of features from 344 to 69. Figure 2 illustrates the data preprocessing phase of our methodology.

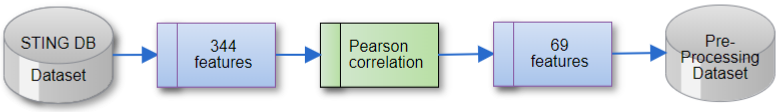


Fig. 2: Features reduction by Pearson's correlation method

The matrix with the properties $(c_1, \ldots, c_{69})$ of each amino acid, obtained from the STING_DB repository, are presented by Equation 1.

$$C = \begin{bmatrix} c_{1,1,1,1} & \cdots & c_{1,1,1,L} \\ \vdots & \vdots & \vdots \\ c_{1,1,A_{1,1},1} & \cdots & c_{1,1,A_{1,1},L} \\ \vdots & \vdots & \vdots \\ c_{p,s_p,1,1} & \cdots & c_{p,s_p,1,L} \\ \vdots & \vdots & \vdots \\ c_{p,s_p,A_{p,s_p},1} & \cdots & c_{p,s_p,A_{p,s_p},L} \end{bmatrix}_{\sum_{i=1}^{p}\sum_{j=1}^{s_i} A_{i,j} \mathrm{x} L} \tag{1}$$

$C = c_{i,j,k,l}$, where
$i = \{1, \ldots, P\}$, P corresponds to number of proteins;
$j = \{1, \ldots, s_i\}$, s_i corresponds to number of chains of amino acid residues that compouse a protein i;
$k = \{1, \ldots, A_{i,j}\}$, $A_{i,j}$ corresponds to number of amino acids from protein i in the chain j;
$l = \{1, \ldots, L\}$, L corresponds to number of physical-chemical properties from STING_DB, where $L = 69$.

Where, $c_{i,j,k,l}$ should be read as: physical-chemical property l of amino acid k, present in the chain of amino acid j in protein i.

These 69 properties are composed of physical-chemical information obtained from the attractions that occur from the several types of bonds between amino acids.

3.3 Standardizing example size with the Discrete Cosine Transform

To enable the use of a classification algorithm, the size of all input vectors (examples in the dataset) must be the same. However, due to the difference in the number of amino acids in each chain of proteins (each protein is formed with a different number of amino acids residues), the dataset had examples of different sizes. To solve this problem we employed the Discrete Cosine Transform (DCT) [Ahmed et al., 1974] to each property of the $c_{i,j,k,l}, \ldots, c_{i,j,A_{i,j},l}$ set (see Equation 1). The DCT, Equation 2, was chosen because it preserves in its first values the most significant coefficients, with decreasing information carried in each subsequent coefficient from the first.

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

where $\alpha_k = \frac{1}{\sqrt{N}}$, $\forall\ k = 0$ or $\alpha_k = \sqrt{\frac{2}{N}}$, $\forall\ k = 1, \ldots, N$, and N is the number of amino acids in each chain $A_{i,j}$.

We adopted the methodology used by Santos et al. [2018a], where the number of used DCT coefficients is $k = 10$. Figure 3 shows the DTC process applied for each protein. The final dataset has 490 chains of amino acid (examples), each of those described by 69 physical-chemical properties that are represented by 10 coefficients each, totalizing 690 features in the dataset is given in Equation 3.

$$T = [T^k_{ij1}], \cdots, [T^k_{ij69}] \tag{3}$$

where $i = \{1, \ldots, P\}$, $j = \{1, \ldots, s_i\}$ and $k = \{1, \ldots, 10\}$ (coefficients DCT).

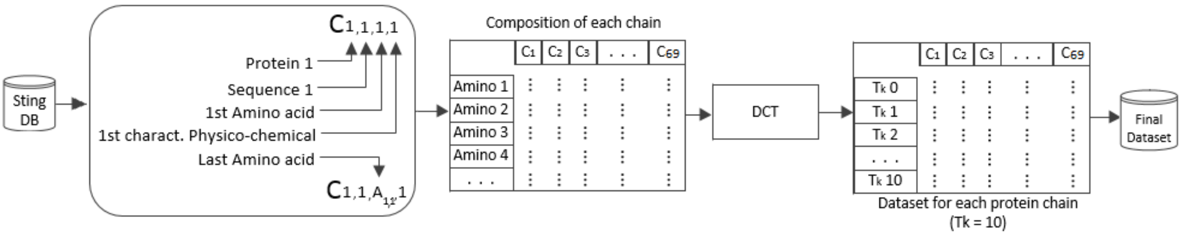


Fig. 3: Discrete Cosine Transform process.

3.4 Multi-Objective Genetic Algorithm for Feature Selection

After the preprossessing phase, we applied the multi-objective GA algorithm Non-dominated Sorting Genetic Algorithm II (NSGA-II) [Deb et al., 2002] 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. 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 guidelines:

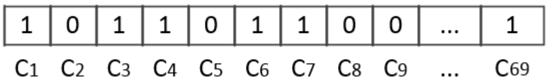
(1) Maximize the average F-Measure values of the k-NN classifier, where

$$\overline{F - Measure} = \frac{\sum_{i=1}^{m} \sum_{j=1}^{n} \text{F-Measure}_{ij}}{mn} \tag{4}$$

where m is the number of classes of enzymes ($m = 6$) and n is the number of cross-validation folds ($n = 10$);

(2) Minimize the number of selected features (physical-chemical properties from the STING_DB).

The F-Measure was adopted because it is the harmonic medium of precision and recall. It also has a unique number that indicates the overall quality of the model. The representation used for the chromosomes in the GA was a binary vector of 69 positions, each of them representing one of the physical-chemical properties. The positions in the vector can take the values 0 and 1, representing the absence or presence of that property's features in the dataset. Figure 4 illustrates the representation of an individual in the GA. Each of the selected properties is represented by a set of 10 DTC coefficients, obtained from the DCT transformation.



1	0	1	1	0	1	1	0	0	...	1
C1	C2	C3	C4	C5	C6	C7	C8	C9	...	C69

Fig. 4: Representation of the individual classes

3.4.1 *Feature Selection.* In [Santos et al., 2018a], the best GA parameters found were: Population = 500, Generations = 200, Crossover ratio = 0.7 and Mutation ratio = 0.01. Considering these experimental values as reference, we performed a new set of experiments to traverse search spaces that had not been traversed by [Santos et al., 2018a]. We define the population size and the number of generations to 500 to increase the domain's representativeness, besides avoiding a premature convergence for local solutions. These new experiments were carried out with 5 different seeds for each combination of the chosen parameters. This was done in order to evaluate the results of the feature's selection process. There were 120 performed experiments in total within the indicated ranges, using the classifier k-NN, according to Table III. Notice that the population size of the number of neighbors k was varied by a range of 1 to 10 in our experiments to find the value that would produce the best-ranking performance for the selected subset of features. The experiments were performed according to the stipulated parameter ranges and the number of random seeds, which are presented in Table III.

Table III: Parameters of the experiments

Initialization of population	:	Random	Population size	:	200, 500
Representation	:	Binary	Number of generations	:	200, 500
Crossover	:	two points	Crossover Selection	:	Tournament = 2
Crossover (Pc)	:	70%, 75%, 80%	New generation	:	Not dominated
Mutation	:	One point	Stopping Criterion	:	Number of generations
Mutation (Pm)	:	1%, 5%	Number of Random Seeds	:	5

3.4.2 *Search for the best experiment.* To search the best solution we performed 120 experiments, combining the parameters of Table III. Figure 5 illustrates the decision making process.

The execution of the genetic algorithm with the variation of the parameters of the experiments resulted in a set of candidate solutions. Each of these candidate solutions contains a set of 10 individuals. To choose the one that best fits a protein problem, we apply the 10-fold cross-validation to each group of candidate solutions and we determine the average F-measure. The set that got the best average was the set chosen.

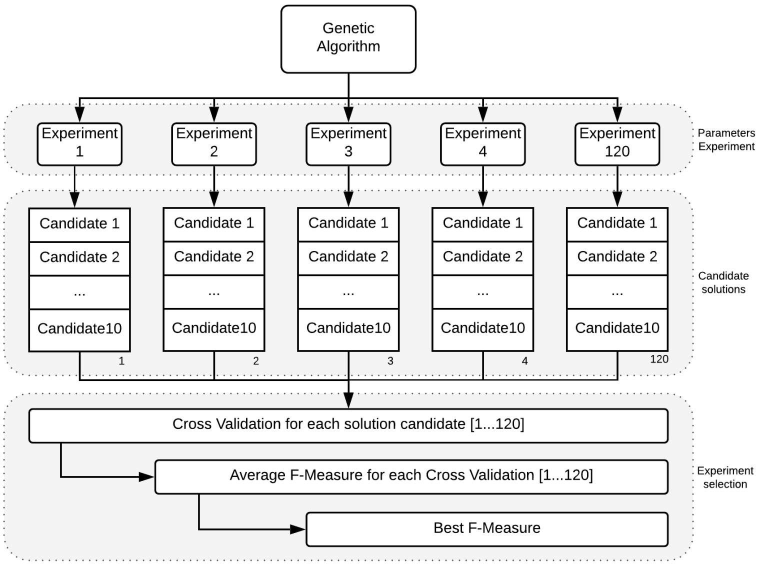


Fig. 5: Decision making process used in AG.

After this process, we have a group of 10 possible candidates. Thus, we chose the one that obtained the best result for objective 1 (F-measure) and in case of a tie we chose the one with the lowest number of attributes (objective 2).

The best parameters found during the execution of multi-objective GA were: Population = 500, Generations = 200, Crossover = 70% and Mutation = 1%. Figure S2[5] presents an analysis of GA behavior during the process of genetic evolution. Thus, the GA algorithm had as its output a set of 31 properties, presented in Table IV, as the best subset to classify the six enzyme classes. These properties describing proteins' structure, stability, function and interaction with other macromolecules and, in general, can be grouped into: 1) *Conservation* are calculated from changes in proteins, which is, how much their sequences have evolved over time; 2) *Interatomic contacts* are calculated from the contact between atoms present in each residue of a protein; 3) *Physical-chemical* are obtained from attractions exerted by the many types of connections between amino acids; 4) *Geometric and structural* are calculated from the three-dimensional structure of a protein and 5) *Relevants Sites* are calculated from cavities in the surface of a protein to which ligands bind themselves. A detailed explanation of the meaning of each of these properties can be found in the Supplementary Material in Table S2[5].

From the 31 properties selected by our mechanism based on GA, 14 are common to those ones selected in [Santos et al., 2018a]. These common properties are in bold in Table IV. It is important to notice that these numbers represent 54% of convergence. This can indicate that a great local solution was reached. Despite that convergence, the average F-Measure values for these six classes of protein (Hydrolases, Isomerases, Lyases, Ligases, Oxidoreductases, Transferases) were respectively 65.8%, 67.5%, 57.3%, 38.5%, 59.3% and 61.5%. The obtained results were low, which takes us to the hypothesis that only the features from the base STING_DB are not enough for the resolution of a enzymes's class prediction problem. With the aim to improve the precision of the classifier, we have done the process of enrichment with new features that will be described in the next section.

It is important to highlight that after finishing this step (reaching the first objective of this paper) is to be able to count with a subclass of physical-chemical properties that can be representative for

[5] Supplementary Material Avaliable at: `https://drive.google.com/open?id=1ZI72cUoisb1fq1YtCGgJcbJKDFoBFKgq`

Table IV: Properties selected by the GA.

3DEntropyCAsw(3,3)	**3DEntropyINT(6)**	**3DEntropyINT(9)**
DiffReliability()	**ACCC()**	ACCR()
DistanceC()	**DistanceN()**	CloCA()
PHI()	**DensityCA(3)**	EnergyDensityCAsw(3,3)
EnergyDensityIFR(1)	EnergyDensityIFR(9)	**EnergyDensityLHAsw(3,3)**
InternalContactsEnergy(t,t)	IFRDensityCA(3)	HydroKD()
NumberOfHydrophobicPLC()	PLC-Max	**NumberofIFRContacts(3)**
NumberofIFRContacts(4)	NumberofIFRContacts(7)	**NumberofIFRContacts(11)**
NumberofIFRContacts(14)	NumberofINTContacts(8)	**NumberofINTContacts(3)**
NumberofINTContacts(10)	**NumberofINTContacts(13)**	NumberofINTContacts(14)
EPabsolute()		

the improvement of computational models that deal with the enzyme's class prediction problem.

3.5 Dataset Enrichment

In [Santos et al., 2018c], the authors demonstrated that the physical-chemical properties in the STING_DB are not enough to reliably identify the six major classes of enzyme. The authors suggested to use other information, such as amino acid frequency, α-carbon frequency and statistical data of primary structure properties. Therefore, we used additional features in order to improve the classification's performance. These features were added at the end of the GA execution, as described below:

(1) Amino Acid Frequency: for each of the considered chains of amino acid, we counted the frequency of each of the 20 amino acids. This set of data is defined in Equation 5.

$$F = \begin{bmatrix} f_{1,1,1} & \cdots & f_{1,1,N} \\ \vdots & \ddots & \vdots \\ f_{1,S_1,1} & \cdots & f_{1,S_1,N} \\ \vdots & \ddots & \vdots \\ f_{p,1,1} & \cdots & f_{p,1,N} \\ \vdots & \ddots & \vdots \\ f_{p,S_p,1} & \cdots & f_{p,S_p,N} \end{bmatrix}_{\sum_{i=1}^{p} S_i \times N} \tag{5}$$

Thereby, $F = f_{i,j,q}$ where $i = \{1, \dots, P\}$, $j = \{1, \dots, S_i\}$ and $n = \{1, \dots, N\}$, N is the number of amino acids ($N = 20$), P is the number of proteins, S_i is the number of chain of the protein i.

(2) α-carbon Frequency: Distribution pattern of the Euclidean distance between the α-carbons of the residues along the chain of amino acid (151 additional features) [Pires et al., 2011], given by Equation 6.

$$D = \begin{bmatrix} d_{1,1,1} & \cdots & d_{1,1,Q} \\ \vdots & \ddots & \vdots \\ d_{1,S_1,1} & \cdots & d_{1,S_1,Q} \\ \vdots & \ddots & \vdots \\ d_{p,1,1} & \cdots & d_{p,1,Q} \\ \vdots & \ddots & \vdots \\ d_{p,S_p,1} & \cdots & d_{p,S_p,Q} \end{bmatrix}_{\sum_{i=1}^{p} S_i \times Q} \tag{6}$$

Thereby, $D = d_{i,j,q}$ where $i = \{1, \ldots, P\}$, $j = \{1, \ldots, S_i\}$ and $q = \{1, \ldots, q\}$, being $Q = 151$ and P is the number of proteins, S_i is the number of chain of the protein i and Q is the number of distance between the carbons.

(3) Statistical data from the primary protein structure: Statistical descriptors of the chains of amino acid residues (31 additional features).

$$E = \begin{bmatrix} e_{1,1,1} & \cdots & e_{1,1,M} \\ \vdots & \ddots & \vdots \\ e_{1,S_1,1} & \cdots & e_{1,S_1,M} \\ \vdots & \ddots & \vdots \\ e_{p,1,1} & \cdots & e_{p,1,M} \\ \vdots & \ddots & \vdots \\ e_{p,S_p,1} & \cdots & e_{p,S_p,M} \end{bmatrix}_{\sum_{i=1}^{p} S_i \times M} \tag{7}$$

Where, $E = e_{i,j,m}$ with $i = \{1, \ldots, P\}$, $j = \{1, \ldots, S_i\}$ and $m = \{1, \ldots, M\}$, being $M = 31$ and P is the number of proteins, S_i is the number of chain of the protein i and M is the number of EMBOSS Pepstats attributes.

Thus, we had a total of 202 additional features to enrich the dataset and improve classification performance. In this way, summed to the features selected by the GA, the dataset had a total of 512 features[6]. Equation 8 defines the dataset containing the best physical-chemical properties and data of the different sources of enrichment:

$$Q = \left[[T^k_{ij1}], \cdots, [T^k_{ij31}], f_{ijn}, d_{ijq}, e_{ijm}\right]_{\sum_{i=1}^{p} S_i \text{x} 512} \tag{8}$$

where $k = \{1, \ldots, 10\}$ (coefficients DTC).

3.6 Dataset preparation and classifier parameter adjustments

One of the objectives of our study is to compare four different classification algorithm to find the one with the best $F - Measure$ metric for a protein class prediction task. The normalization process, class balancing, and the classification algorithms used in the experiments are described in the following sections.

3.6.1 *Normalization of data.* As the properties selected by the GA are sourced from the STING_DB dataset, and the added features from the enrichment task come from different sources, there is a variation on the scale of the variables, creating the need to normalize the data. This normalization is required for several machine learning estimators. For this task, we employed the Standard Scaler method, Equation 9, available on the Sciki-learn framework [Pedregosa et al., 2011].

$$\text{Standard Scaler} = \frac{x_i - mean(x)}{stdev(x)} \tag{9}$$

3.6.2 *Cross-validation and class balancing.* We employed a 10-fold cross-validation process to ensure confidence in the accuracy of the results and we applied a class balancing process to avoid the model bias.

[6] As the GA had found 31 properties and 10 values represented each of them, we had 310 features in the dataset so far.

Preliminary experiments showed low precision value in the classifiers. One of the factors to explain this is the small number of examples in the dataset, as is the case with classes Ligases Isomerases and Lyases. For example, the Ligase class has only 16 examples. For that reason, the classifiers have a low performance to predict the minority classes. To address this issue, we applied class balancing with the ADASYN strategy (ADAptive SYnthetic), an oversampling approach for unbalanced learning [He et al., 2008]. This technique is an improved version of SMOTE strategy [Chawla et al., 2002]. After creating examples linearly correlated with the parent, it adds small random values to the examples, increasing by little the variance. With this strategy, all classes were grown to become equivalent to the number of examples of the majority class.

It is essential to mention that the class balancing task was made during the cross-validation, which is, each training fold had the balancing applied to it separately, to avoid tendentious or biased learning. After balancing all classes were left with 162 instances, equivalent to the majority class. The number of features correspond to 512, as indicated by Equation 8.

It is also important to mention that the process of cross-validation was applied during the features selection process (GA with k-NN), and after the enrichment, during the training of the classifiers. However, the classes' balancing process was applied on the last step, compared to the classifiers performance.

3.6.3 *Artificial Neural Network.* An Artificial Neural Network is organized in interconnected artificial neuron layers: the input layer, the hidden layers, and the output layer. It is possible to use a large number of neurons, expanding the amount of the hidden layer, magnifying the complexity of the ANN. In our experiments, we used a four layer Multilayer Perceptron (MLP) [da Silva et al., 2016], [Haykin, 2001] for which the Backpropagation Learning Rule was used. A detailed description of the network structure used is described below.

- **Number of layers and neurons:** The first is the *input layer* with 512 neurons referring to each feature in the dataset. Then, there are 2 hidden layers in the network, with their neuron numbers being defined by the number of inputs in the dataset n, as $2n + 1$, (according to the Kolmogorov's Theorem [Li and Vitányi, 1990]), totalizing 1025 neurons for each. The output layer has only 6 neurons, individually activated, that represents each of the targeted enzyme classes. The Kolmogorov's theorem was taken into consideration as a heuristic criterion. According to [Kovacs, 2002], the theorem could be reformulated to the field of ANN as follows: Given an arbitrary continuous function (f), there is always for f an exact implementation with a three-layer neural network, being the entry layer an N-dimensional vector, the hidden layer containing $(2N + 1)$ neurons, and the output layer containing M neurons representing the m components of the output vector. In [Kůrková, 1992], based on Kolmogorov's theorem, the author gives a proof of the universal approximation capabilities of neural networks with four hidden layers.
- **Activation function:** The activation function in the hidden layers was the Rectified Linear Unit (ReLu) function [Hahnloser et al., 2000]. The output layer represents the canonical responses corresponding to the classes, and each neuron is activated individually through the Softmax activation function.
- **Stop criterion**: We employed the EarlyStopping [Chollet et al., 2015] as the stop criterion for the ANN, considering the smallest value of the error function. This method was used because it enabled us to establish several training epochs and suspend the training when the ANN stops improving.
- **Learning method:** We employed the back-propagation strategy with the Adam optimization function during the training of our ANN.

3.6.4 *Support Vector Machine (SVM).* It is a supervised learning technique used for the task of classification. The algorithm tries to find the hyperplane that maximizes the separation margin between the examples of different classes in a dataset. Usually, SVM uses kernel methods to transform a set of data from dimension n to another m, with $m > n$, to increase the separability of classes. In other words, the kernel function is responsible for mapping the data and making it easier to find the hyperplane separating the classes. The 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 hyperplan, maximizing accuracy. Usually, the SVM optimization tries to find a more significant margin for the hyperplan even if that means more training errors.

—**Gamma (γ):** This parameter defines the influence the training examples have over others they are distant to. A small value of γ means that the distant points in the hyperplan will also be considered, whereas larger values will make the algorithm consider only the closest examples.

For our study, we used the Radial Basis Function (RBF) kernel, since, in preliminary experiments, we obtained better results with this chosen type. It is a simple and often used function for SVM classifiers. To estimate these parameters, we adopted the Grid Search strategy [Hearst, 1998], which consists of the process of performing the hyperparameter adjustment to determine the optimal values for a given model. The values used were Cost $= 8.0$ and $\gamma = 0.00195313$. Other techniques, as Random Search, can be applied to this purpose [Bergstra and Bengio, 2012].

3.6.5 *Random Forest.* The Random Forest algorithm [Ho, 1995] 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 [2001], which needs the following parameter adjustments:

—**Max_depth**: Indicates the maximum depth of each decision tree in the forest.

—**N_estimators**: Is the number of decision trees that will be created. This should not be too small, to guarantee the variability that makes the ensemble learning reliable.

—**Min_examples**: The minimum number of examples 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_ examples*: 3, *Max_features*: 10, *Criterion*: entropy.

3.6.6 *k-Nearest Neighbor.* The k-Nearest Neighbor (k-NN) is another supervised learning algorithm; it can be used for classification or regression problems. 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 dataset itself.

The k value represents the number of neighbors used to consider when classifying an example. A high number of k can result in a less precise result, but a value that is too low makes the model more sensible to outliers. For example, if $k = 1$, the examples in the test dataset will receive the same class label as the closest example 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 $k = 1$, as proposed by Aha et al. [1991].

The k-NN predicts a class label by finding the most similar examples 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.

4. RESULTS AND DISCUSSION

For predicting the enzyme's classes, we compared the performance of the k-NN, RF, SVM, and ANN classifiers. Due to the small number of training examples (mainly in the minority class) we opted to perform tests varying the number of folds, as discussed in [Kohavi, 1995]. According to the author most of the estimates are almost unbias between 10 and 20 folds. Through preliminary experiments, the number of folds that showed the best result was $k = 16$ folds. Table S3[7] presents the results applying the proposed methodology without balancing of dataset. We considered the arithmetic and weighted[8] averages of the F-measure metric to evaluate the quality of the classifiers. Values in parentheses correspond to the standard deviation. As there is a significant unbalance between the classes (see Table II), we consider the weighted average, as it considers the number of examples in each class. However, as the related works evaluated the arithmetic average, we added these values for comparison purposes.

The results without balancing are shown in Table S3[7]. The RF classifier got the best results, with weighted average F-Measure of 75.63%. The results for ANN, SVM and k-NN classifiers had very similar results: 70.84%, 71.51%, and 67.94%, respectively. It is important to highlight that, for the Lyase class (one of three minority classes), the performance in all of the classifiers was low (with the best F-measure of 53.33% for RF classifier), which reduced the global mean. This indicates the need for new researches to find other features that can improve the prediction performance of this class, without harming the prediction of the others.

On the other hand, the performance of the other minority classes, such as Isomerase (N = 56) and Ligase (N = 16), showed promise, with F-measure of 66.15% and 71.71%, respectively. This may suggest that the properties selected by the GA, along with the enrichment of the dataset, contributed positively to the classification of these two minority classes, in addition to the majority classes. Note that for the two major classes, Transferase (N = 117) and Hydrolase (N = 162), the performance was above 72%, reaching 87.46% for the Hydrolase class.

As mentioned before, there are significant differences in the number of examples for the six classes in our dataset. For example, the Ligases, Isomerase and Lyase classes have, only 16, 56 and 61 examples, respectively, and the Hydrolases class has 162 (see Table II). The unbalancing can impair the performance of classifiers. Thus, we also perform experiments using the balancing technique with the ADASYN method, which increases the number of examples of the minority classes.

The results obtained with the class balancing strategy do not show improvements. Again, the RF classifier got the best results, with 75.47% weighted average F-Measure. The ANN, SVM and k-NN classifiers had very similar results for the experiment without class balancing: 70.82%, 71.10%, and 67.78%, respectively.

In order to confirm the small relevance of the balancing process we performed a statistical test comparing the results with and without balancing. For this test we considered the result of the RF classifier, which presented the best average among the other classifiers. For the comparison, two-tailed hypothesis tests for two examples were considered. A $t - test$, (for $n < 30$) with a significance level of α=5% was applied for each one of the 6 protein classes. The examples considered for each test correspond to the results of the 16 folds obtained in our experiments. Before applying the $t - test$, the $F - test$ ($Fcritical$=2.403) was applied in order to evaluate if the variances of both examples, for each one of the 6 protein classes, are equal or different. After the tests, it was confirmed that the samples, for each protein classes, had equivalent variances. After this confirmation, the $t - test$ ($tcritical$=2.131) was applied to compare the sample average. The null hypothesis $Ho : \mu A = \mu B$ was accped in Hydrolase, Lyases and Oxidoreductases classes. It was rejected in the Isomerase class, with

[7] Supplementary Material Avaliable at: `https://drive.google.com/open?id=1ZI72cUoisb1fq1YtCGgJcbJKDFoBFKgq`

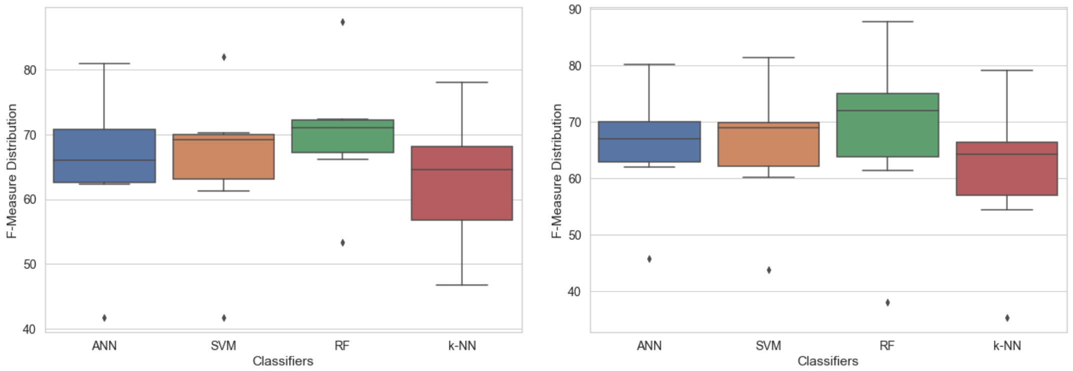
[8] The calculated average considers the number of examples in each class.

better results for the experiment without balancing, and it was rejected in Ligases and Transferases classes, favorable to the experiment with balancing.

In general, we observe that even when applying the class balancing strategy, there was no improvement in the predictive precision of the classifiers. Instead, in the three classes with the fewest examples, the quality of the model worsened for two of them, Isomerase and Lyase. In the Lyase class, for example, the balance worsened by 15.3 percentage points. One of the factors that can explain these results is a low variability in examples of minority classes, which may have resulted in artificial examples generated by the balancing strategy that is not representative enough for the problem. As a result, the performance of the classifiers cannot be improved.

Figure 6 shows the distribution of F-Measure for both strategies. We observe in Figure 6a that the maximum and minimum values of k-NN have a larger range. Data dispersion for the k-NN classifier showed greater variability. Also, the median value of the Random Forest shows its superiority over other classifiers, since we can see that its lower limit was practically the upper limit of the k-NN classifier.

Figure 6b has a different behavior because the variability of the Random Forest classifier was higher and there was a slight reduction in the ANN classifier, a fact that can be observed with the difference between the first and third quartile. The ANN symmetry draws attention in this figure, showing the midline between the first and third quartiles. There are also discrepancies for all classifiers, which leads us to conclude that these values are out of range that can be detected. Finally, the median value of the Random Forest classifier again shows its better performance compared to the other classifiers.



(a) Without Balancing

(b) With Balancing

Fig. 6: F-Measure Distribution

After the general analysis of both strategies (with and without balancing), we made a comparison with the work of [Borro et al., 2006], [Leijoto et al., 2014], [Santos et al., 2018a], [Santos et al., 2018b] and [Santos et al., 2018c]. All of these approaches considered the STING_DB database, see Figure 7. It is noticed that the proposed methodology loses, with a larger difference, in classifying the lyase class. However, we observe that the features selection approach using genetic algorithms is very promising. The five approaches that used GA achieved similar results in most classes, and generally much better than work that did not use GA, for example [Borro et al., 2006], except for the work of [Santos et al., 2018c] that had very good result for the Lyase class, which eventually raised the overall average.

It is also observed that our methodology (considering the RF classifier) has some gain in relation to hydrolase class. The results for the Isomerase, Lyase, and Oxidoreductase classes were very close and down concerning the Ligase class. However, it is important to highlight that there was an improvement over [Santos et al., 2018a], where there was a gain in the 5 protein classes (Hydrolase, Isomerase, Ligase, Oxidoreductase and Transferase), and a decline over the class (Lyase).

Looking at the features used in the work of [Santos et al., 2018c], we see that 8 out of 10 features used in the work belong to the categories physical-chemical (electrostatic potential@CA, electrostatic potential@LHA, electrostatic potential@surface, electrostatic potential: atom average) and geometrics (distance from center of gravity, cross presence order, cross link order, Hydrophobicity Isolation). Although this work has selected features of these two categories, they are slightly different (see Supplementary Material). Thus, a suggestion for improving these results may be to add these additional features to the model.

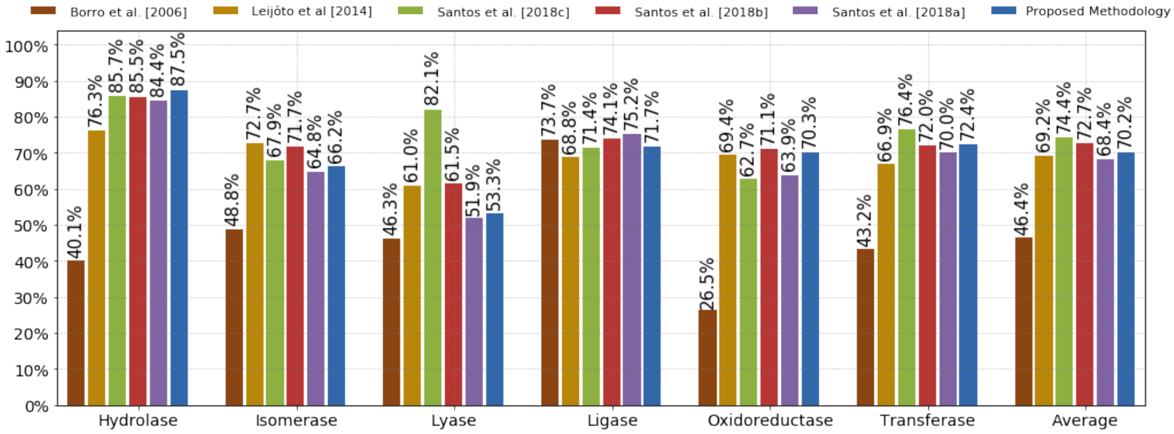


Fig. 7: Comparison between related works and proposed methodology

Table V shows a summary of the approaches used in each of the approaches listed in Figure 7. In order to increase the research space, our methodology considered the k-NN classifier within the wrapper mechanism. In Santos et al. [2018b], the authors considered the SVM classifier, more robust, however it demanding more computational effort. Importantly, the use of SVM within the wrapper mechanism improves the feature selection process. This happens due to the fact that the Genetic algorithm (GA) is designed to search the optimal solution via weeding out the worse gene strings based on a fitness function. This function is better adjusted when more robust classifiers are used, such as the SVM. However, those have more parameters to be adjusted, which can result in an excessive computational time. Thus, unlike the proposal presented in [Santos et al., 2018b] that used SVM as a wrapper mechanism, we observe that the proposal to use k-NN is promising, as it obtained very similar results, 70.2% against 72.7% considering the average, with a much less computational time, saving about 90% of the time.

Table V: Comparative of adopted methodologies

	Feature Select with Genetic Evolution			**Dataset After Enrichment**				
Works	GA	SVM	k-NN	PCA	SVM	ANN	RF	k-NN
Borro et al [2006]				✓				
Leijôto et al [2014]	✓	✓						
Santos et al. [2018c]				✓	✓			
Santos et al. [2018b]	✓	✓		✓	✓			
Santos et al. [2018a]	✓		✓	✓	✓			
Proposed Methodology	✓		✓		✓	✓	✓	✓

Starting from available data, two statistical analysis were applied in order to compare the proposed approach of this paper with the presented results in Santos et al. [2018a] and Santos et al. [2018b]. For both comparisons, two-tailed hypothesis tests for two examples were considered. A $t-test$, (for $n < 30$) with a significance level of α=5% was applied for each one of the 6 protein classes, for both comparisons. The examples considered for each test correspond to the results of the 16 folds obtained in our experiments.

Before applying the $t-test$, the $F-test$ ($Fcritical$=2.403) was applied to evaluate if the variances of both examples, for each one of the 6 protein classes, considering the approach presented in this paper, first with [Santos et al., 2018a] and after with Santos et al. [2018b], are equal or different. After the tests, it was confirmed that the samples, for each protein classes, had equivalent variances, in both comparisons. After this confirmation, the $t-test$ ($tcritical$=2.131) was applied on them to compare the sample means. The null hypothesis $Ho : \mu A = \mu B$ was rejected in all classes, except on the Lyases class, where the null hypothesis was accepted. The same results were observed in both comparisons.

Based on these results we can conclude that the proposed approach is superior to the approach presented in Santos et al. [2018a] for presenting a superior mean for all protein classes. In relation to Santos et al. [2018b], the means are very close, with a better result for the Hydrolase and Transferase classes. We ratify that the Lyases class presents the same low performance, independent from the considered strategy. In general, the results show the importance of increasing the number of examples of the minority classes (in our study, Ligase, Isomerase and Lyase), as well as adding other physical chemical features that help to better separate classes.

Furthermore, in order to simplify the proposed method, our methodology removed the step of dimensional reduction based on PCA, considered for the other approaches. After these changes, the results were superior to previous work by Santos et al. [2018a], showing to be so efficient in the process of protein class prediction, as shown in Figure 7.

5. CONCLUSIONS

In this study, we presented an approach for the protein class prediction problem. The proposed methodology was composed of two stages. In the first stage, the aim was to identify a subset of physical-chemical properties from the STING_DB database, which can be used in classification models for this type of problem. For this, we applied a multi-objective Genetic Algorithm based on the k-NN classifier. The second stage involved enriching that dataset and comparing the classification model with four different algorithms. We compared the performances of Artificial Neural Networks, Support Vector Machines, Random Forests, and k-Nearest Neighbors.

Towards the experiments in the feature selection stage, the k parameter from the k-NN classifier was adjusted to find the best classification results. However, even with that adjustment, the results were unsatisfactory as expected, once we had denoted in previous works that the physical-chemical properties from the STING_DB database are not sufficient to reliably predict a protein class. Importantly, one of the main results of this work is the identification of a subset of properties that could be considered for the development of new protein class prediction models, without having to consider the complete database.

A critical aspect to mention is the class unbalance issue. When we applied the class balancing strategy, matching the number of examples in all classes, there was no significant improvement in the results. One of the factors that can explain these results is a low variability in examples of minority classes, which may have resulted in artificial examples generated by the balancing strategy that is not representative enough for the problem. This fact shows the importance of finding a representative amount of protein examples for the minority classes, or the proposal of methods for evaluating the effectiveness of balancing strategies. As a future paper we consider the possibility to apply the Cost-

Sensitive Learning, which is a kind of learning that takes the misclassification costs, among other types of costs. The goal of this type of learning is to minimize the total cost.

On the other hand, our proposal was able to improve a previous work [Santos et al., 2018b], where the authors apply Principal Component Analysis (PCA) to reduce the dimensionality of the dataset (from 482 to 127 features, in their dataset). Their proposal reached an overall average F-Measure of 72.7%, but their metodology is difficult to reproduce due to its dependency on the PCA. It is noteworthy that the weakness of using PCA is the feature became uninterpretable, and it is necessary to define the optimal number of the principal component to get the optimum result [Nasreen, 2014], [Wicaksono and Afif, 2018]. Our proposal disregards the PCA step, utilizing all 512 features in the dataset. The RF classifier had an average F-Measure of 70.2%, a superior performance to [Santos et al., 2018a]'s work and very similar to [Santos et al., 2018b], but a reduction of approximately 90% of computational time.

As future studies, we suggest joining the subset of selected physical-chemical properties of the STING_DB and enrichment datasets, adding new information such as protein Graph Models. Finally, we also suggest new approaches for the selection of biological properties to be added for the dataset, such as Particle Swarm Optimization (PSO) and Ant Colony Optimization. Concerning the techniques of class balancing, we suggest proposing new techniques and algorithms, more efficient, to generate a balanced dataset from a set with few examples of minority classes.

Acknowledgments

This study was financed in part by the Coordination for the Improvement of Higher Education Personnel (CAPES) - Finance Code 001. The authors also thank the National Council for Scientific and Technological Development of Brazil (CNPq - Brazilian National Council for Scientific and Technological Development) and the Foundation for Research Support of the Minas Gerais State (FAPEMIG). The work was developed at the Pontifical Catholic University of Minas Gerais, PUC Minas in the Applied Computational Intelligence laboratory – LICAP.

REFERENCES

W. Aha, D. Kibler, and M. K. Albert. Instance-based learning algorithms. *Machine Learning*, 6: 37–66, 01 1991. doi: 10.1023/A:1022689900470.

N. Ahmed, T. Natarajan, and K. R. Rao. Discrete cosine transform. *Computers, IEEE Transactions on*, C-23:90–93, 1974.

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

B. Alberts, D. Bray, K. Hopkin, A. Johnson, J. Lewis, M. Raff, K. Roberts, and P. Walter. *Essential Cell Biology*. CRC Press, 2013. ISBN 9781317806271. URL `https://books.google.com.br/books?id=Cg4WAgAAQBAJ`.

J. Bergstra and Y. Bengio. Random search for hyper-parameter optimization. *J. Mach. Learn. Res.*, 13(null):281–305, Feb. 2012. ISSN 1532-4435.

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*, 28:235–242, 2000.

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*, 1:193–202, 2006.

L. Breiman. Random forests. *Machine Learning*, 45(1):5–32, Oct 2001. ISSN 1573-0565. doi: 10.1023/A:1010933404324. URL `https://doi.org/10.1023/A:1010933404324`.

N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer. Smote: Synthetic minority oversampling technique. *J. Artif. Int. Res.*, 16(1):321–357, June 2002. ISSN 1076-9757. URL http://dl.acm.org/citation.cfm?id=1622407.1622416.

F. Chollet et al. Keras. https://keras.io, 2015.

I. N. da Silva, D. H. Spatti, R. A. Flauzino, L. H. B. Liboni, and S. F. dos Reis Alves. *Artificial Neural Networks: A Practical Course.* Springer Publishing Company, Incorporated, 1st edition, 2016. ISBN 3319431617.

K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan. A fast and elitist multiobjective genetic algorithm: Nsga-ii. *Trans. Evol. Comp*, 6(2):182–197, Apr. 2002. ISSN 1089-778X. doi: 10.1109/4235.996017. URL http://dx.doi.org/10.1109/4235.996017.

P. D. Dobson and A. J. Doig. Distinguishing enzyme structures from non-enzymes without alignments. *Molecular Biology*, 330:771–783, 2003.

P. D. Dobson and A. J. Doig. Predicting enzyme class from protein structure without alignments. *Molecular Biology*, 345:187–199, 2004.

H. P. Graf, E. Cosatto, L. Bottou, I. Dourdanovic, and V. Vapnik. Parallel support vector machines: The cascade svm. In *Advances in neural information processing systems*, pages 521–528, 2004.

R. H. R. Hahnloser, R. Sarpeshkar, M. A. Mahowald, R. J. Douglas, and H. S. Seung. Digital selection and analogue amplification coexist in a cortex-inspired silicon circuit. *Nature*, 405:947–951, 2000.

S. Haykin. *Redes Neurais - 2ed.* Bookman, 2001. ISBN 9788573077186. URL https://books.google.com.br/books?id=lBp0X5qfyjUC.

H. He, Y. Bai, E. A. Garcia, and S. Li. Adasyn: Adaptive synthetic sampling approach for imbalanced learning. In *2008 IEEE International Joint Conference on Neural Networks (IEEE World Congress on Computational Intelligence)*, pages 1322–1328, June 2008. doi: 10.1109/IJCNN.2008.4633969.

M. A. Hearst. Support vector machines. *IEEE Intelligent Systems*, 13(4):18–28, July 1998. ISSN 1541-1672. doi: 10.1109/5254.708428. URL http://dx.doi.org/10.1109/5254.708428.

T. K. Ho. Random decision forests. In *Proceedings of the Third International Conference on Document Analysis and Recognition (Volume 1) - Volume 1*, ICDAR '95, pages 278–, Washington, DC, USA, 1995. IEEE Computer Society. ISBN 0-8186-7128-9. URL http://dl.acm.org/citation.cfm?id=844379.844681.

R. Kohavi. A study of cross-validation and bootstrap for accuracy estimation and model selection. pages 1137–1143. Morgan Kaufmann, 1995.

Z. Kovacs. *Redes Neurais Artificiais - Fundamentos e Aplicações.* Saraiva, 2002.

C. Kumar and A. Choudhary. A top-down approach to classify enzyme functional classes and subclasses using random forest. *EURASIP Journal on Bioinformatics and Systems Biology*, 2012(1):1, Feb 2012. ISSN 1687-4153. doi: 10.1186/1687-4153-2012-1. URL https://doi.org/10.1186/1687-4153-2012-1.

V. Kůrková. Kolmogorov's theorem and multilayer neural networks. *Neural Networks*, 5(3):501 – 506, 1992. ISSN 0893-6080. doi: https://doi.org/10.1016/0893-6080(92)90012-8. URL http://www.sciencedirect.com/science/article/pii/0893608092900128.

L. F. Leijoto, T. Assis De Oliveira Rodrigues, L. Zarate, 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*, pages 168–174. Computer Society Digital Library, Nov 2014. doi: 10.1109/BIBE.2014.42.

M. Li and P. M. Vitányi. Chapter 4 - kolmogorov complexity and its applications. In J. V. LEEUWEN, editor, *Algorithms and Complexity*, Handbook of Theoretical Computer Science, pages 187 – 254. Elsevier, Amsterdam, 1990. ISBN 978-0-444-88071-0. doi: https://doi.org/10.1016/B978-0-444-88071-0.50009-6.

A. H. Liu and A. Califano. Functional classification of proteins by pattern discovery and top-down clustering of primary sequences. *IBM Systems Journal*, 40(2):379–393, 2001. doi: 10.1147/sj.402.0379.

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*, 20: 2145–2147, 2004.

N. Nadzirin and M. Firdaus-Raih. Proteins of unknown function in the protein data bank (pdb): an inventory of true uncharacterized proteins and computational tools for their analysis. *International journal of molecular sciences*, 13(10):12761–12772, Oct 2012. ISSN 1422-0067. doi: 10.3390/ijms131012761. URL `https://www.ncbi.nlm.nih.gov/pubmed/23202924`. 23202924[pmid].

S. Nasreen. A survey of feature selection and feature extraction techniques in machine learning,sai,2014. 08 2014.

G. Pandey, V. Kumar, and M. Steinbach. Computational approaches for protein function prediction: A survey. *Twin Cities: Department of Computer Science and Engineering, University of Minnesota*, 01 2006.

F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

D. E. Pires, R. C. de Melo-Minardi, M. A. dos Santos, C. H. da Silveira, M. M. Santoro, and W. Meira. Cutoff scanning matrix (csm): structural classification and function prediction by protein inter-residue distance patterns. *BMC Genomics*, 12(4):S12, 2011. ISSN 1471-2164. doi: 10.1186/1471-2164-12-S4-S12.

W. K. Resende, R. A. Nascimento, C. R. Xavier, I. F. Lopes, and C. N. Nobre. The use of support vector machine and genetic algorithms to predict protein function. In *2012 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, pages 1773–1778, Oct 2012. doi: 10.1109/ICSMC.2012.6377994.

B. C. Santos, S. C., M. W. Rodrigues, C. N. Nobre, and L. E. Zárate. Seleção de características utilizando algoritmo genético multiobjetivo e k-nn para predição de função de proteína. In *6th Symposium on Knowledge Discovery, Mining and Learning*, pages 36–43, São Paulo, Brazil, 2018a. Bracis 2018. `https://bracis2018.mybluemix.net/files/anais-kdmile-2018.pdf`.

B. C. 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)*, pages 1–6, July 2018b. doi: 10.1109/CEC.2018.8477981.

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*, 2018c. (To be published `http://http://www.inderscience.com/info/ingeneral/forthcoming.php?jcode=ijbra`).

B. Szalkai and V. Grolmusz. Near perfect protein multi-label classification with deep neural networks. *Methods*, 132:50 – 56, 2018. ISSN 1046-2023. doi: https://doi.org/10.1016/j.ymeth.2017.06.034. URL `http://www.sciencedirect.com/science/article/pii/S104620231730035X`. Comparison and Visualization Methods for High-Dimensional Biological Data.

O. K. Tawfik and D. S. Enzyme promiscuity: A mechanistic and evolutionary perspective. *Annual Review of Biochemistry*, 79(1):471–505, 2010. doi: 10.1146/annurev-biochem-030409-143718. URL `https://doi.org/10.1146/annurev-biochem-030409-143718`. PMID: 20235827.

A. Wicaksono and A. Afif. Hyper parameter optimization using genetic algorithm on machine learning methods for online news popularity prediction. *International Journal of Advanced Computer Science and Applications*, 9, 01 2018. doi: 10.14569/IJACSA.2018.091238.