Ensemble of Protein Stability Upon Point Mutation Predictors [⋆]

Eduardo Kenji Hasegawa de Freitas[1][0000−0002−1476−4779], Alex Dias Camargo[1][0000−0002−6377−9379], Maurício Balboni[1][0000−0001−7653−0962], Adriano V. Werhli[1][0000−0001−7107−5024], and Karina dos Santos Machado[1][0000−0002−8966−5708]

Computational Biology Laboratory - COMBI-LAB
Centro de Ciências Computacionais, Universidade Federal do Rio Grande - FURG.
Av. Itália, km 8. Rio Grande, RS, Brazil.
karina.machado@furg.br

Abstract. Computational determination of protein stability upon point mutations is very useful in a wide field of applications. The reliability of such computational predictions is essential. Unfortunately, existing computational tools frequently disagree in their results. In the present study, the usage of Ensemble Learning Algorithms to aggregate the results from different stability prediction tools is investigated. Techniques of Stacking, Bagging, and Boosting as well as different Machine Learning algorithms as combiner function are explored. All the investigation is carried out in real dataset ProTherm for which experimental results are known. The proposed methodology was validated considering two different experiments according to the training set. Results show that our proposed ensemble approach is appropriate to predict the effect of point mutations on protein stability showing more reliable results than the individual tools improving overall accuracy, precision, and/or recall.

Keywords: Ensemble learning · Bioinformatics · Protein Stability · Point Mutations.

1 Introduction

The latest research and development in biological experiments and techniques have created a deluge of experimental data. To deal with this quantity and variety of data the necessity of computational methods and tools is obvious. In some cases, the use of Data Science methodology is mandatory.

In all living species, biological information is stored in the DNA (deoxyribonucleic acid). DNA can be seen as the source code that contains all the instructions to create and sustain life. To transform the information contained in the DNA into functioning parts, DNA is transcript as RNA (ribonucleic acid) which is

[⋆] This study was supported by CAPES Edital Biologia Computacional (51/2013), CAPES Financial Code 001 and CNPq (439582/2018-0).

then translated into proteins. Proteins are the principal biological elements responsible for executing biological functions. Proteins can be thought of as the compiled result of the information stored in the DNA [23]. Moreover, proteins are the most abundant organic molecules in living systems and have the widest range of functions among macro-molecules. These functions can be structural, regulatory, contracting, or protective. Furthermore, they can serve as transport, storage, or membranes, as well as being enzymes or toxins. Each cell in a living system has thousands of proteins where each one has a function, making their structures very varied.

Proteins are the biological elements that execute functions and frequently they are modified (mutated) to achieve specific goals. These designed changes, mutations, are carried out in specific parts of the DNA using for example site-directed mutagenesis. These mutations can produce a strong impact on the structure and function of proteins [19, 17]. However, due to the vast number of possibilities, it is impractical to experimentally evaluate the impact of all possible mutations on protein structure and function [25].

A series of computational tools have been developed, with the common objective of predicting the effect of point mutations on the structure of a protein. In general, prediction tools are developed and used by professionals from different areas who, generally, do not access the same tools for their analysis, which implies discrepancies in results and difficulties in interpretation, as they will have different results for the same data entries. Among the many tools available we can cite: Dmutant [30], FoldX [12], I-Mutant2.0 [4], CUPSAT [17], Eris [28], I-Mutant3.0 [5], PoPMuSiC [7], SDM [25], mCSM [18], MUpro [6], MAESTRO [14], and DUET [19].

This computationally predicting the impact on proteins structure, and function, upon point mutations is of great scientific and commercial interest. One way of performing such predictions is to compute the $\Delta\Delta G$ (variation in the variation of the free energy). However, the existing methods and tools do not always obtain similar results for the same entries. This differences in results obtained from distinct tools might impact negatively in the experiments carried out by researchers. The resulting $\Delta\Delta G$ from the computational tools is usually discretized in Stabilizing and Destabilizing and this is an important information to define whether a mutation is viable or not.

In this context, where we have several answers for the same problem, Ensemble Learning techniques can be a good alternative. Ensemble learning aims to aggregate predictions from different classifier models in just one result. These combinations can be achieved in many different ways, e.g., average, major voting, etc. The resulting ensemble classifier usually obtains better results than the individual results of the classifiers used in the combination [27].

Therefore, the present work proposes the application of Ensemble Learning to aggregate the protein stability predictions of individual tools. Techniques of Stacking, Bagging, and Boosting as well as different Machine Learning algorithms as combiner function are explored. The ensemble models were trained and validated using data from ProTherm [2], a database composed by sequence and structural information obtained from experimental methods for wild-type and mutant proteins. The paper is organized as follows: section 2 presents bioinformatics and machine learning brief concepts related to this purpose; section 3 detailed the individual tools used on the ensemble; section 4 describes the proposed methodology; section 5 presents the results and discussion and finally section 6 concludes the paper and show some future work.

2 Background

2.1 Point mutations and their effects on protein structures

DNA is formed by a sequence of molecules, or base pairs. Some parts of DNA are transcript as RNA which in turn is translated as a protein. In translation, each three bases of RNA is translated as one amino acid. The order of amino acids determines the 3D structure and consequently the function of proteins.

A point mutation is when a single base pair is exchanged [29, 1]. They can be: **i) silent:** when the resulting amino acid is the same, no changes in the resulting protein; **ii) missense:** when the resulting amino acid is different from the original hence the protein is different and **iii) nonsense:** the resulting mutations indicate a stop signal, again the result is a modified protein. In this study, the focus is on point mutations that results in an exchange of amino acids and thus in a modified protein.

2.2 Gibbs free energy (G)

Gibbs free energy is stabilized when the protein is in constant equilibrium, both in pressure and temperature. Therefore, the protein in its native state (or wild type, WT) has a certain amount of energy G_{WT}. When the protein folding process occurs, the Gibbs free energy undergoes a variation, providing a variation of energy ΔG_{WT}. In the same way that, if we analyze the same protein in a mutated state, we have, in its equilibrium, an energy G_{mutant} and, when this mutated protein folds, we can calculate the $\Delta G_{\mathrm{mutant}}$.

Then, with the folding of the native and mutant proteins, one can calculate the $\Delta\Delta G$, which is calculated using the Equation 1 [21] [10]. The $\Delta\Delta G$ is a metric for predicting how a single-point mutation affects protein stability. With this, the present work uses the $\Delta\Delta G$ as the main attribute to be evaluated, where the unit kcal/mol is used for the quantitative values.

$$\Delta\Delta G = \Delta G_{\mathrm{WT}} - \Delta G_{\mathrm{mutant}} \tag{1}$$

2.3 Supervised machine learning

Machine learning (ML) investigates how computers can learn based on data and has proven to be of great practical value in a variety of application domains[13]. Supervised learning is the ML task of learning a model (or function) that maps an input to an output based on example input-output pairs. Thus, the objective of these tasks is to predict the value of a particular attribute (target) based on the values of other attributes (explanatory or independent variables) [22].

Since the data considered in this work has a target value, the $\Delta\Delta G$, we are applying supervised ML. This target is discretized between *Stabilizing* or *Destabilizing* (section 2.2). In this way, we are applying as meta-classifiers in ensemble learning tasks the classification algorithms Decision Trees (DT), Multilayer perceptron (MLP), Naive Bayes (NB) and Support Vector Machines (SVM).

2.4 Ensemble Learning

Ensemble learning is a popular method that combines multiple algorithms to make a decision usually in supervised ML [8]. Ensembles have been applied in several research areas as economy, logistic, medicine, education and so on. In Bioinformatics, it has been applied in different contexts as reviewed by Yang [26]. In Eickhotl & Cheng [9], the authors proposed to apply boosting ensembles of deep networks for a sequence based prediction of protein disorder. Mendonza *et al.* [15] discuss the use of ensemble-based solutions for inference of gene regulatory networks and prediction of microRNAs targets. Saha *et al.* [20] applied an Ensemble Learning method on majority voting on datasets of protein-protein interactions data.

The main premise of this method is that by combining multiple models, the errors of a single predictor can be compensated by others [8]. Moreover, the objective of ensemble methods is not only to improve the predictions but also to enhance their capability of generalization. An ensemble method for classification consists in a set of k models, $M_1, M_2, \ldots, M_k$, used to build an improved predictive model. A dataset D is used to obtain k, $D_1, D_2, \ldots, D_k$ datasets in which D_i ($1 \leq i \leq k$-1) is used to build the classification model M_i [13]. There are several methodologies for forming an ensemble considering the k induced models: modifying the $D_1, D_2, \ldots, D_k$ datasets (sub-sampling, bagging, randomly sampling, etc), modifying the learning task, exploiting the predictive algorithms characteristics and so on. Thus, given a new instance to classify, each one of the induced models contributes to compose the final decision of the ensemble. Combining the predictive models can reduce the risk to choose a bad result since the result of the combination usually is better than the best induced model. The definition of the best combination of classifiers is a combinatorial problem and there are different implementations of ensemble methods. In this paper we propose to generate stacking [16] , bagging [3] and boosting [11] ensemble models combining different classification algorithms.

Stacking [16] is a heterogeneous ensemble method where the models are obtained considering different algorithms and one unique dataset. Then the answers of the predictive models are combined in a new dataset [31] and a meta-classifier is introduced. Bagging and Boosting are homogeneous ensemble methods. In Bagging [3], the individual models are built parallel, are different from each other and all the instances have the same weight. In Boosting [11] individual models are built sequentially where the outputs of the first model is used on the next and so on. So, during training, the algorithm defines weights for the instances and the ensemble models are incrementally built.

3 Individual tools for predicting the effects of point mutations in protein stability

There are many tools for predicting the effects of point mutations in protein stability. In this work we have chosen the tools based on the following characteristics: they produce a numerical prediction of $\Delta\Delta G$; their response time should be short; they consider protein structure as input; they permit automatic submission and license of use for academic purposes is available. In doing so, the selected tools are: CUPSAT [17], SDM [25], mCSM [18], DUET [19], MAESTRO [14] and PoPMuSic [7]. Besides, these tools have similar input forms and therefore similar outputs, which allow us to better manipulate the data.

3.1 CUPSAT

CUPSAT (Cologne University Protein Stability Analysis Tool) [17] is a web tool that analyzes and predicts protein stability changes for point mutations. It uses the specific structural environment of potential atoms and potential torsion angles, in the unwinding, to predict the $\Delta\Delta$G. Their results consist of informing the structural characteristics about the mutation of this location, such as: solvent accessibility, secondary structure and torsion angles.

It also looks at the ability of mutated amino acids to adapt to twist angles. It uses several validation tests (split-sample, jack-knife and k-fold) to guarantee the reliability, accuracy and transferability of the prediction methods, which guarantee an accuracy of more than 80% on their tests [17].

3.2 SDM

Site Directed Mutator (SDM) [25] web tool is a potential energy statistical function with the aim of predicting the effect that single nucleotide polymorphisms will have on protein stability. Using amino acid substitution frequencies in specific environments within protein families, the tool calculates a stability score that is analogous to the variation in free energy between the native and mutant state of the protein.

3.3 mCSM

Mutation Cutoff Scanning Matrix (mCSM), is a predictor that uses structural graph-based signatures of proteins to study the impact of mutations [18]. mCSM signatures were derived from the graph-based concept of Cutoff Scanning Matrix (CSM), originally proposed to represent network topology by distance patterns in the study of biological systems. In mCSM each mutation is represented by a pharmacophore vector that is used to train and test machine learning prediction methods, both in regression and predictive classification.

On their proposed algorithm, first mCSM signature is calculated using a set of mutations, wild-type structure, the atomic categories (or pharmacophore) to be considered, and a cutoff range and step. For each mutation, based on the cutoff, the residue environment is calculated, followed by the determination of a distance matrix of the pairwise distances between all pairs of atoms of the residue environment. Then, the distance matrix is scanned using the cutoff range and step generating a cumulative distribution of the distances by atomic category. Finally, the pharmacophoric changes between wild-type and mutant residue are appended to the signature. This signatures are used for training the predictive models.

3.4 DUET

The DUET [19] tool is an integrated computational approach available online, which aims to predict the impact of mutations in protein stability. Given a single point mutation, DUET combine/consensus the predictions of two other tools SDM and mCSM in a non-linear way, similar to the ensemble learning method, to make predictions and optimize results using Support Vector Machines (SVMs) with radial basis function kernel.

As a filtering step, residue relative solvent accessibility (RSA) is used to optimize the standard SDM predictions using a regression model tree (M5P algorithm) before combining it with mCSM. Then, the mCSM an optimized SDM predictions, together with secundary structure from SDM and pharmacophore vector from mCSM are used as input for SVM generating the combined DUET output. All their training was carried out by data sets with little redundancy and validated with random data sets.

3.5 MAESTRO

In order to predict changes in stability on point mutations of proteins, MAESTRO [14] is based on the structure of the protein and, although it has a predictive similar to other methods, it differs because of the following points: it implements a multi-agent system for learning; provides in-depth mutation scanning at various points where mutation types can be comprehensively controlled; and provides a specific mode for stabilizing disulfide bond predictions.

MAESTRO is based on two different machine learning approaches: artificial neural networks (ANN) and SVM (Gaussian Kernel), and also on multiple linear regression (MLR). In order to improve generalization, a set of ANNs is utilized rather than a single one.

3.6 PoPMuSic

PopMuSic [7] is a web server for predicting thermodynamic changes in point mutations of proteins, which uses a linear combination of statistical potentials whose coefficients depend on the solvent accessibility of the mutated residue.

It is a fast tool that allows you to make predictions for all possible mutations of a protein, average size, in less than a minute. It has the ability to detect which and how optimal the amino acids are in each protein, so that mutation experiments can be carried out, also demonstrating structural weaknesses by quantifying how much these sites are optimized for protein function, rather than stability.

4 Proposed methodology

Our proposed methodology is presented in Figure 1. We are applying Stacking and Bagging/Boosting ensemble learning on the results of six protein stability changes upon point mutations predictors: CUPSAT [17], SDM [25], mCSM [18], DUET [19], MAESTRO [14] and PopMuSic [7].

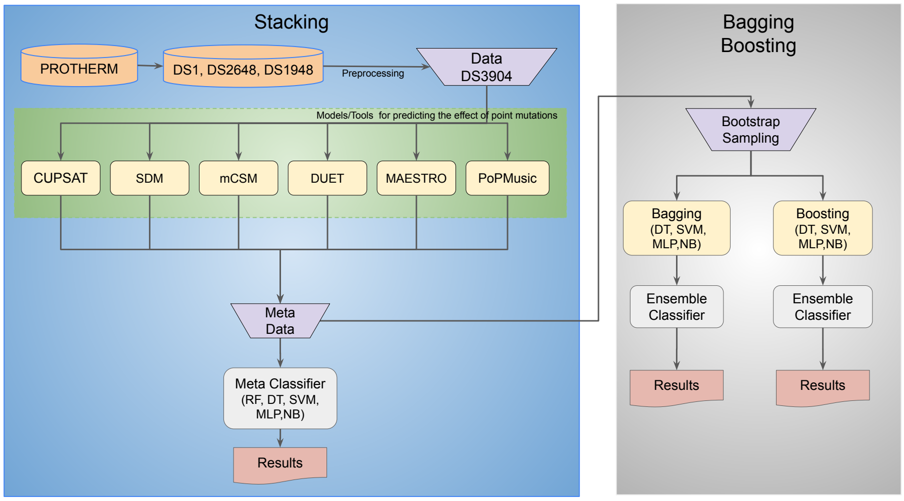


Fig. 1. Proposed methodology of ensemble learning applied on protein stability changes upon point mutations predictors.

4.1 Input data

The first step on our proposed methodology is to prepare the input dataset. In this paper we are considering as input data sets of proteins from which we can obtain the experimental value of Gibbs free energy ($\Delta\Delta$G) between wild-type and mutants (our target attribute on ensemble learning). This thermodynamic experimental data was obtained from ProTherm [2]. ProTherm is a database containing several thermodynamic parameters along with sequence and structural information, experimental methods and conditions parameters for wild-type and mutant proteins.

The non-disjoint datasets DS1, DS2648 and DS1948, Table 1 lines 1-3, are taken from ProTherm [2] and considered in the individual tools for training their models. DS1 and DS2648 were used by PoPMuSic [7], SDM [25] and MAESTRO [14], while DS1948 was used by I-Mutant [4] tool. These three datasets DS1, DS2648 and DS1948 have different formats for presenting their data, in addition to containing several duplicates. Thus, we unified and preprocess all these data totaling 7244 instances (4th line on Table 1). After preprocessing all these data, we remove duplicates of the same protein experimental results resulting in our final input dataset DS3904 (highlighted on Table 1).

Table 1. Input datasets: number of instances, number of unique proteins, number of stabilizing mutations and number of destabilizing mutations.

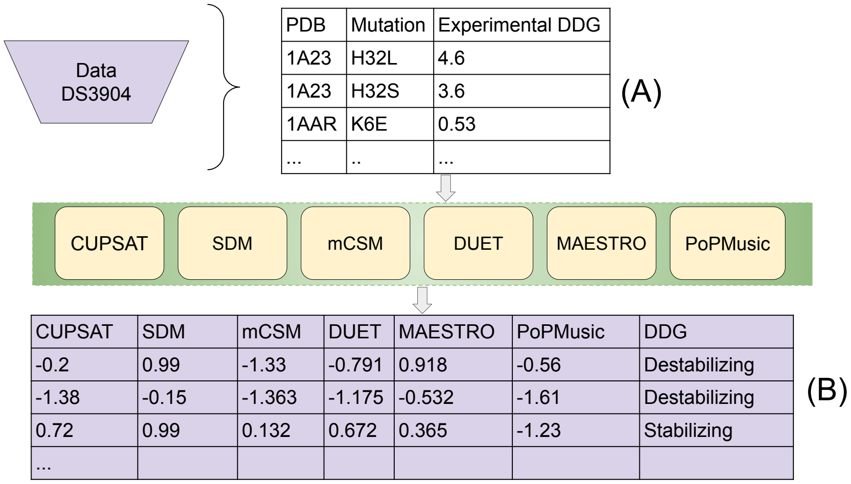
Dataset	Total of instances	Proteins	Stabilizing	Destabilizing
DS1	2648	99	2.046	602
DS2648	2648	100	568	2080
DS1948	1948	58	562	1386
DS1 ∪ DS2648 ∪ DS1948	7244	298	3738	4068
DS3904	**3904**	**151**	**951**	**2953**

4.2 Meta data

After having DS3904 properly prepared, the following step is submitting this list of protein wild-type/mutations in each of the tools. In Figure 2 (A) we describe a part of DS3904 where we have the PDB ID, the mutation in the format Original residue type - position - mutation and the experimental $\Delta\Delta$G. For example, the first line, PDB ID 1A23 has as original residue in position 32 a Histidine (H) that when mutated to a Leucine (L) has an experimental $\Delta\Delta$G of 4.6.

In order to obtain the predicted $\Delta\Delta$G of each tool to generate our meta data we would need to manually submit each entry of DS3904 to the tools. To perform this task in batch we use a browser plugin called iMacros. iMacros is a scripting language that allows users to simulate online activities for filling out forms, uploading and downloading images and files, or even importing and exporting information from databases, CSV files, XML files, among others. PopMusic is different of the other tools since it allows the user to inform the PDB code for each of the 151 proteins of DS3904 (and their specific mutation positions) in a list.

The results were properly stored in a local database generating our meta data (Figure 2 (B)) where we have as predictive attributes the predicted $\Delta\Delta$G by the tools and as target attribut the experimental $\Delta\Delta$G discretized as *Destabilizing* ($\Delta\Delta G < 0$) or *Stabilizing* ($\Delta\Delta G \geq 0$).



PDB	Mutation	Experimental DDG
1A23	H32L	4.6
1A23	H32S	3.6
1AAR	K6E	0.53
...	..	...

CUPSAT	SDM	mCSM	DUET	MAESTRO	PoPMusic	DDG
-0.2	0.99	-1.33	-0.791	0.918	-0.56	Destabilizing
-1.38	-0.15	-1.363	-1.175	-0.532	-1.61	Destabilizing
0.72	0.99	0.132	0.672	0.365	-1.23	Stabilizing
...						

Fig. 2. Part of input data. (A) DS3904 (B) Meta data.

4.3 Ensemble learning - Stacking

Considering as input the meta data generated by heterogeneous predictive tools in the previous step we performed stacking ensemble learning considering as meta classifiers: Random Forest (RF), Decision Trees (DT), Support Vector Machines (SVM), Multilayer Perceptron - Radial basis function (MLP) and Naive Bayes (NB). The parameters of these algorithms were set as default values: DT (maxDepth=unlimited; numFeatures=unlimited; numTrees=100), SVM (degree of the kernel=3; kernel type=radial; eps=0.001), MLP (autoBuild=true; decay=false; hiddenLayers='a'; learningrate=0.3; momentum=0.2; validationThreshold=20) and NB (useKernelEstimator=false and useSupervisedDiscretization=false). We evaluate all the generated ensemble models using accuracy, precision, recall and F-measure.

4.4 Ensemble learning - Bagging/Boosting

In order to evaluate the impact of ensemble learning not only with stacking configuration we decided to also apply bagging and boosting. We have performed different configurations for this ensemble experiments: Bagging and Boosting. The parameters of these algorithms were set as default values: Bagging (bagSizePercent=100; calcOutofBag=false; Classifiers=SVM, DT, MLP and NV - same parameters used in Stacking; numIterations=10), Boosting (numIterations=10; Classifiers=SVM, DT, MLP and NV - same parameters used in Stacking; useResampling=false; weightThreshold=100). The stacking, bagging and boosting ensemble learning algorithms were performed using WEKA 3.8 [24].

5 Results and Discussion

To evaluate our proposed methodology of ensemble learning for this problem we performed two experiments: *Experiment* 1 considers a balanced training dataset from where 500 random instances where selected from DS3904 for each target value *Stabilizing* and *Destabilizing*; in *Experiment* 2 we divided DS3904 in 70% (Training) and 30% (Testing), and we have not balanced training/testing datasets. Table 2 describes the datasets for *Experiments* 1 and 2.

Table 2. Training and testing datasets for *Experiments* 1 and 2.

Experiment	**Training**			**Testing**		
	Stabilizing	Destabilizing	Total	Stabilizing	Destabilizing	Total
1	500	500	1000	451	2453	2904
2	666	2066	2732	285	887	1172

5.1 Experiment 1: Balanced training dataset

DS3904 is unbalanced according to the target attribute. Although it is common in real datasets can cause some problems in generating predictive models as being highly specialized (specially to detect the rare class) or susceptible to presence of noise [22]. Thus, we decided to train the stacking and bagging/boosting ensemble models in *Experiment* 1 with 500 random instances of each class, totaling 1000 instances for training. The remaining 2904 instances are part of the test set. In order to compare the results of the generated ensemble models with the individual Tools we calculated the metrics accuracy, precision, recall and F-measure for the test set for all.

First, we evaluate the individual tools considering the test set with 2904 instances (see Table 2)and comparing the predicted $\Delta\Delta$G with the experimental value from ProTherm (Table 3). We can observe that all tools obtained good accuracy values but with low values for precision. It is important to mention that our test set probably was used by the tools to train their models.

Table 3. Evaluation of individual tools in Experiment 1.

Individual tools	**Accuracy**	**Precision**	**Recall**	**F-measure**
CUPSAT	0.75	0.31	0.54	0.39
SDM	0.65	0.23	0.60	0.33
mCSM	0.81	0.35	0.31	0.34
DUET	0.78	0.34	0.58	0.43
MAESTRO	0.70	0.24	0.53	0.34
PoPMusic	0.78	0.29	0.35	0.32

Both Stacking (Table 4), and Bagging/Boosting ensemble models (Table 5), obtained good values of accuracy and precision. From these results, we can highlight SVM , Multilayer Perceptron and Random Forest close to mCSM Tool. From the Experiment 1 results we can notice that better precision measures were obtained if we compare with individual tools.

Table 4. Evaluation of Stacking Ensemble in Experiment 1.

Stacking ensemble: meta classifiers	**Accuracy**	**Precision**	**Recall**	**F-measure**
Naive Bayes (NB)	0.63	0.84	0.26	0.40
Multilayer Perceptron (MLP)	0.74	0.76	0.32	0.46
SVM (radial)	0.75	0.76	0.34	0.47
Decision Trees (DT)	0.76	0.70	0.35	0.47
Random Forest (RF)	0.80	0.70	0.40	0.51

Table 5. Evaluation of Bagging/Boosting in Experiment 1.

Bagging/Boosting Ensemble	**Accuracy**	**Precision**	**Recall**	**F-measure**
Bagging (SVM)	0.75	0.76	0.35	0.48
Bagging (DT)	0.75	0.76	0.34	0.47
Bagging (MLP)	0.74	0.75	0.33	0.46
Bagging (NB)	0.63	0.84	0.26	0.40
Boosting (SVM)	0.70	0.79	0.30	0.44
Boosting (DT)	0.75	0.71	0.33	0.45
Boosting (MLP)	0.74	0.76	0.33	0.46
Boosting (NB)	0.69	0.80	0.30	0.43

The predictions about point mutation effects on protein structures are going to be used in the decision of generating experimentally some proposed protein mutations, which is expensive in terms of time and costs. Considering *Stabilizing* as a positive class, *false positives* are mutations predicted as *stabilizing* but they are *destabilizing.* This error can make the laboratory spends money and time on a not viable protein. On the other side, *false negatives* are mutations predicted as *destabilizing* but they are *stabilizing.* These errors can discard some important mutation that impacts on the protein function of interest, for example. Both errors can impact experimental laboratory activities according to the project. So, we expect to achieve good values for *precision* and *recall* since it is important to guarantee that the *stabilizing* or *destabilizing* predictions are correct.

5.2 Experiment 2: Unbalanced training set

In Experiment 2 we are training the ensemble models with unbalanced dataset randomly dividing the DS3904 into 70% of for training and the remaining 30% of the instances for testing.

Table 6 presents the results of the individual tools. It is observed that although they present good accuracy, the values for precision and recall varies. Some tools (mCSM and PoPMuSic) have better Recall values, while SDM and MAESTRO have better precision (around 0.6).

Table 6. Evaluation of individual tools in Experiment 2.

Individual Tools	**Accuracy**	**Precision**	**Recall**	**F-measure**
CUPSAT	0.72	0.43	0.51	0.47
SDM	0.66	0.38	0.60	0.47
mCSM	0.78	0.57	0.36	0.45
DUET	0.77	0.52	0.59	0.56
MAESTRO	0.68	0.40	0.60	0.48
PoPMusic	0.75	0.49	0.42	0.45

Tables 7 and 8 present the Ensemble learning models evaluation for Experiment 2. We can observe better accuracy, recall and precision values. Boosting(SVM) model has the best accuracy (0.84) and good recall (0.77). We can see similar results for Stacking learning (MLP, SVM, DT and RF), Bagging (SVM, DT and MLP) and also Boosting (MLP).

Table 7. Evaluation of Stacking Ensemble in Experiment 2.

Stacking ensemble: meta classifiers	**Accuracy**	**Precision**	**Recall**	**F-measure**
NaiveBayes (NB)	0.73	0.78	0.46	0.58
Multilayer Perceptron (MLP)	0.83	0.43	0.77	0.56
SVM (radial)	0.84	0.47	0.79	0.59
Decision Trees (DT)	0.82	0.35	0.77	0.49
Random Forest(RF)	0.84	0.40	0.85	0.55

Table 8. Evaluation of Bagging/Boosting in Experiment 2.

Bagging/Boosting Ensemble	**Accuracy**	**Precision**	**Recall**	**F-measure**
Bagging (SVM)	0.84	0.45	0.80	0.57
Bagging (DT)	0.84	0.52	0.74	0.61
Bagging (MLP)	0.84	0.56	0.73	0.63
Bagging (NB)	0.73	0.78	0.46	0.58
Boosting (SVM)	0.85	0.52	0.77	0.62
Boosting (DT)	0.82	0.60	0.63	0.62
Boosting (MLP)	0.83	0.44	0.78	0.56
Boosting (NB)	0.80	0.67	0.58	0.62

These are good models for avoiding to incorrectly predict as destabilizing a stabilizing mutation (*falses negatives*). On the other side, Stacking learning, Bagging and Boosting with Naive Bayes (NB) also have good accuracy but presented better precision values. It means that ensemble models with NB can be good options for avoiding to predict destabilizing mutations as stabilizing (*falses positives*).

6 Conclusion

The knowledge about the impact on proteins stability upon point mutations is an important scientific and commercial research topic. One way is to computationally predict the variation of the free energy $\Delta\Delta G$ and its consequently impact on protein stability. There are different methods and tools for this purpose that do not always obtain similar results for the same entries.

Besides, these predictions are used to define point mutations on protein structures that are experimentally generated, which is expensive in terms of time and costs. Errors on this predictions can have important impacts. For example, consider a point mutation as *destabilizing* but when it is *stabilizing* can discard this candidate. At the same time that a wrong prediction of *stabilizing* when the mutation destabilizes the protein structure can made the laboratory spends money and time on a not viable protein. Thus, in this paper we propose to aggregate the results from different stability prediction tools using Stacking, Bagging/Boosting Ensemble Learning techniques achieving to obtain models with not only good accuracy but also good precision and/or recall.

In our proposed methodology we are considering six individual $\Delta\Delta G$ predictors: CUPSAT [17], SDM [25], mCSM [18], DUET [19], MAESTRO [14] and PoPMuSic [7]. We prepared an input dataset from ProTherm [2] with 3,904 instances about wild-type and mutation proteins. After submitting this list to the tools we obtained our meta-data composed by their $\Delta\Delta G$ predictions. Having this meta-data, we applied Stacking ensemble with meta-classifiers (RF, DT, SVM, MLP and NB), Bagging (SVM, DT, MLP and NV) and Boosting .

For both ensemble techniques we validated considering two different experiments according to the training set. In Experiment 1 (balanced training set) results we can observe that even all individual tools obtained good accuracy values, they presented low precision. On the other side, both Stacking and Bagging/Boosting ensemble models obtained also good accuracy but with better precision.

In Experiment 2 (unbalanced training set) results we can observe better accuracy, recall and precision values for ensemble models. Boosting(SVM) obtained the best accuracy (0.84) and good recall (0.77). We can see similar results for Stacking learning (MLP, SVM, DT and RF), Bagging (SVM, DT and MLP) and also Boosting (MLP). These are good models for avoiding to incorrectly predict as destabilizing a stabilizing mutation. On the other side, Stacking learning, Bagging and Boosting with Naive Bayes (NB) also have good accuracy but presented better precision values. It means that ensemble models with NB can be good options for avoiding to predict destabilizing mutations as stabilizing.

Thus, according with the results, our proposed ensemble approach is appropriate to predict the effect of point mutations on protein stability showing more reliable results than the individual tools. As future work, we can combine the individual tools using genetic algorithms to determine the weight of each $\Delta\Delta$G predicted and also include more tools in the ensemble. In addition, other balancing strategies can be performed on the training datasets.

References

1. Auclair, J., Busine, M.P., Navarro, C., Ruano, E., Montmain, G., Desseigne, F., Saurin, J.C., Lasset, C., Bonadona, V., Giraud, S., et al.: Systematic mrna analysis for the effect ofmlh1 andmsh2 missense and silent mutations on aberrant splicing. Human Mutation **27**(2), 145–154 (2006). https://doi.org/10.1002/humu.20280
2. Bava, K.A., Gromiha, M.M., Uedaira, H., Kitajima, K., Sarai, A.: ProTherm, version 4.0: thermodynamic database for proteins and mutants. Nucleic Acids Research **32**(1), 120–121 (01 2004). https://doi.org/10.1093/nar/gkh082, https://doi.org/10.1093/nar/gkh082
3. Breiman, L.: Bagging predictors. Machine learning **24**(2), 123–140 (1996)
4. Capriotti, E., Fariselli, P., Casadio, R.: A neural-network-based method for predicting protein stability changes upon single point mutations. Bioinformatics **20**(Suppl 1), i63–i68 (2004). https://doi.org/10.1093/bioinformatics/bth928
5. Capriotti, E., Fariselli, P., Rossi, I., Casadio, R.: A three-state prediction of single point mutations on protein stability changes. BMC Bioinformatics **9**(Suppl 2) (2008). https://doi.org/10.1186/1471-2105-9-s2-s6
6. Cheng, J., Randall, A., Baldi, P.: Prediction of protein stability changes for single-site mutations using support vector machines. Proteins: Structure, Function, and Bioinformatics **62**(4), 1125–1132 (2005). https://doi.org/10.1002/prot.20810
7. Dehouck, Y., Kwasigroch, J.M., Gilis, D., Rooman, M.: Popmusic 2.1: a web server for the estimation of protein stability changes upon mutation and sequence optimality. BMC Bioinformatics **12**(1) (2011). https://doi.org/10.1186/1471-2105-12-151
8. Dong, X., Yu, Z., Cao, W., Shi, Y., Ma, Q.: A survey on ensemble learning. Frontiers of Computer Science **14**(2), 241–258 (2020)
9. Eickholt, J., Cheng, J.: Dndisorder: predicting protein disorder using boosting and deep networks. BMC bioinformatics **14**(1), 1 (2013)
10. Fersht, A.R.: Protein folding and stability: the pathway of folding of barnase. FEBS Letters Volume 325, Issues 1–2, 28 June 1993 pp. 5–16 (1993)
11. Freund, Y., Schapire, R.E., et al.: Experiments with a new boosting algorithm. In: icml. vol. 96, pp. 148–156 (1996)
12. Guerois, R., Nielsen, J.E., Serrano, L.: Predicting changes in the stability of proteins and protein complexes: A study of more than 1000 mutations. Journal of Molecular Biology **320**(2), 369–387 (2002). https://doi.org/10.1016/s0022-2836(02)00442-4
13. Han, J., Pei, J., Kamber, M.: Data mining – concepts and techniques. San Francisco: Morgan and Kaufmann (2006)
14. Laimer, J., Hofer, H., Fritz, M., Wegenkittl, S., Lackner, P.: Maestro - multi agent stability prediction upon point mutations. BMC Bioinformatics **16**(1) (2015). https://doi.org/10.1186/s12859-015-0548-6

15. Mendoza, M.R., Bazzan, A.L.C.: The wisdom of crowds in bioinformatics: What can we learn (and gain) from ensemble predictions? In: Proceedings of the Twenty-Seventh AAAI Conference on Artificial Intelligence. pp. 1678–1679 (2013)
16. Noçairi, H., Gomes, C., Thomas, M., Saporta, G.: Improving stacking methodology for combining classifiers; applications to cosmetic industry. Electronic Journal of Applied Statistical Analysis **9**(2), 340–361 (2016)
17. Parthiban, V., Gromiha, M.M., Schomburg, D.: Cupsat: prediction of protein stability upon point mutations. Nucleic Acids Research **34**(Web Server) (Jan 2006). https://doi.org/10.1093/nar/gkl190
18. Pires, D.E.V., Ascher, D.B., Blundell, T.L.: mcsm: predicting the effects of mutations in proteins using graph-based signatures. Bioinformatics **30**(3), 335–342 (2013). https://doi.org/10.1093/bioinformatics/btt691
19. Pires, D.E., Ascher, D.B., Blundell, T.L.: Duet: a server for predicting effects of mutations on protein stability using an integrated computational approach. Nucleic acids research **42**(W1), W314–W319 (2014)
20. Saha, I., Zubek, J., Klingström, T., Forsberg, S., Wikander, J., Kierczak, M., Maulik, U., Plewczynski, D.: Ensemble learning prediction of protein–protein interactions using proteins functional annotations. Molecular BioSystems **10**(4), 820–830 (2014)
21. Sugita, Y., Kitao, A.: Dependence of protein stability on the structure of the denatured state: free energy calculations of i56v mutation in human lysozyme. Biophysical journal **75**(5), 2178–2187 (1998)
22. Tan, P.N., Steinbach, M., Kumar, V.: Introduction to data mining. Pearson Education (2016)
23. Verli, H.: Bioinformática: da biologia à flexibilidade molecular. Sociedade Brasileira de Bioquímica e Biologia Molecular (2014)
24. Witten, I.H., Frank, E., Hall, M.A.: Data Mining Practical Machine Learning Tools and Techniques Third Edition. Morgan Kaufmann (2016)
25. Worth, C.L., Preissner, R., Blundell, T.L.: Sdm—a server for predicting effects of mutations on protein stability and malfunction. Nucleic acids research **39**(suppl_2), W215–W222 (2011)
26. Yang, P., Hwa Yang, Y., B Zhou, B., Y Zomaya, A.: A review of ensemble methods in bioinformatics. Current Bioinformatics **5**(4), 296–308 (2010)
27. Yang, Y.: Temporal Data Mining via Unsupervised Ensemble Learning. Elsevier (2016)
28. Yin, S., Ding, F., Dokholyan, N.V.: Eris: an automated estimator of protein stability. Nature Methods **4**(6), 466–467 (2007). https://doi.org/10.1038/nmeth0607-466
29. Zhang, Z., Miteva, M.A., Wang, L., Alexov, E.: Analyzing effects of naturally occurring missense mutations. Computational and Mathematical Methods in Medicine **2012**, 1–15 (2012). https://doi.org/10.1155/2012/805827
30. Zhou, H., Zhou, Y.: Distance-scaled, finite ideal-gas reference state improves structure-derived potentials of mean force for structure selection and stability prediction. Protein Science **11**(11), 2714–2726 (2009). https://doi.org/10.1110/ps.0217002
31. Zhou, Z.H.: Ensemble methods: foundations and algorithms. Chapman and Hall/CRC (2019)