MSc in Bioinformatics

MASTER THESIS

Development of a new methodology to predict and engineer thermostable proteins

September, 2020

Author: Ana Robles Martín

Project Tutor:

Víctor Guallar

Academic Tutor:

Laura Masgrau

Barcelona Supercomputing Center
Centro Nacional de Supercomputación

UAB
Universitat Autònoma de Barcelona

Signature page

This master thesis has been supervised by:

Project Tutor:

Víctor Guallar Tasies

(BSC)

Academic Tutor:

Laura Masgrau Fontanet

(UAB)

Author:

Ana Robles Martín

INDEX

ABSTRACT

Due to the advantage that resistant proteins provide, studies in the protein thermostability change has been of great interest because of its importance for industries, biotechnological and biomedical research. Multiple computational predictors have been created with the objective of identifying mutable residues that help improving the protein stability. However, the search of a highly accurate predictor has been unsuccessful. That is the reason why the idea of combining different predictors to create a consensus metapredictor comes up. Its main goal is to take advantage of all the other predictors' strengths and minimize their weaknesses. In this thesis, we have developed four metapredictors, from which RF-Classifier, and its combined model with RF-Regressor, has the best performance. Besides, we have conducted three retrospective studies and a prospective one with an alpha/beta hydrolase to filter and select the most stabilizing mutations.

1. INTRODUCTION

Protein stability is a very important characteristic of proteins that highly affects their function, activity and regulation. Sometimes, single point polymorphism appears and therefore protein decreases their thermostabily, so they produce new variants that cause diseases or reduced immunogenicity (local or global unfolding or aggregation) (Broom, Jacobi, Trainor, & Meiering, 2017). In other situations, organisms evolved to produce other variants that increase protein thermostability and allow them to live in extreme conditions (for example thermophilic bacterias), thanks to the new version of these proteins that can keep their functionality. Stability residues show high conservation between sequences. It is widely known the existence of both situations and, for this reason, studying protein stability is highly desirable for both biomedical and biotechnological applications. In particular, for an industrial purpose and in this master thesis, the main objective is increasing protein stability and aiding in finding new thermostable proteins (Chakravorty, Khan, & Patra, 2017; Musil et al., 2017). From an industrial point of view, these proteins can withstand harsh industrial environments and keep their activity, so there is a continuous interest in these type of studies.

There are several approaches to get thermostable proteins, although not all of them are feasible. For example, in vitro mutagenesis experiments, based on rational/semi rational or directed evolution approaches, are costly and time consuming. As a cheaper and faster alternative, computational methods, that have become more accurate and fast in the last years, can be helpful in estimating the effects of a mutation on a protein structure; we find an increasing number of predictions methods (Musil et al., 2017; Pucci, Bernaerts, Kwasigroch, & Rooman, 2018).

Protein stability is guided by numerous non-covalent interactions: hydrophobic, hydrogen bonding, electrostatic and Van der Waals interactions (Dill, 1990; Ponnuswamy & Michael Gromiha, 1994). From all them, the most important are hydrophobic interactions that are believed to be the driving force in folding and stability, although other cooperative long-range interactions between residues helps to overcome the local tendency to unfold (Gromiha & Selvaraj, 2004; Ponnuswamy, 1993).

1.1. ΔΔG explanation

To measure the change in protein stability upon a point mutation we use Delta Delta G (DDG or ΔΔG). ΔΔG is the change in Gibbs Free Energy between the folded and unfolded states in the wildtype protein and in the mutant one (Figure 1). This measurement is very useful to guess if a point mutation will increase protein stability or decrease in computational protein engineering. The units that are commonly used are Kcal per mol of substance.

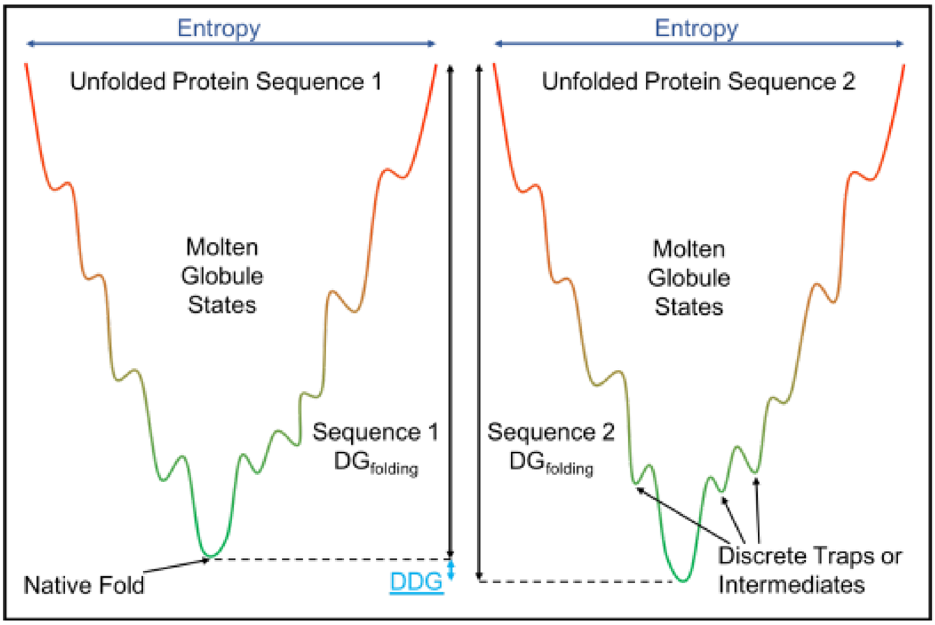

Entropy
Entropy
Unfolded Protein Sequence 1
Unfolded Protein Sequence 2
Molten Globule States
Molten Globule States
Sequence 1 $DG_{folding}$
Sequence 2 $DG_{folding}$
Native Fold
DDG
Discrete Traps or Intermediates

Figure 1. Schematic representation of the energy landscape for the wild-type protein and its mutant (left and right, respectively) where the x-axis represents the entropy and the y-axis the Gibbs energy. The minima represent the different conformational changes that explore the protein during folding. The difference between the global minima in the wildtype and in the mutant is the ΔΔG value. (Figure extracted from Cyrus Biotech | Molecular Modeling and Design. (n.d.), from https://cyrusbio.com/)

During protein folding, the internal energy decreases due to packing of hydrophobic residues, optimized polar groups orientation and adjustment of bond lengths and angles. Entropy of the protein decreases but the entropy of the system increases due to the solvent. Mutations affect the way that residues interact and stabilize the structure (Figure 2). There is a preference of hydrophobic residues to be buried as well as to avoid the appearance of cavities in the interface of the protein caused by mutations that reduce the size of the residue (Goldenzweig & Fleishman, 2018). It is important to consider the modification of Van der Waals interactions and hydrogen bonds. In general, flexible parts are more feasible to increase stability than rigid parts that tend to destabilize (Goldenzweig & Fleishman, 2018).

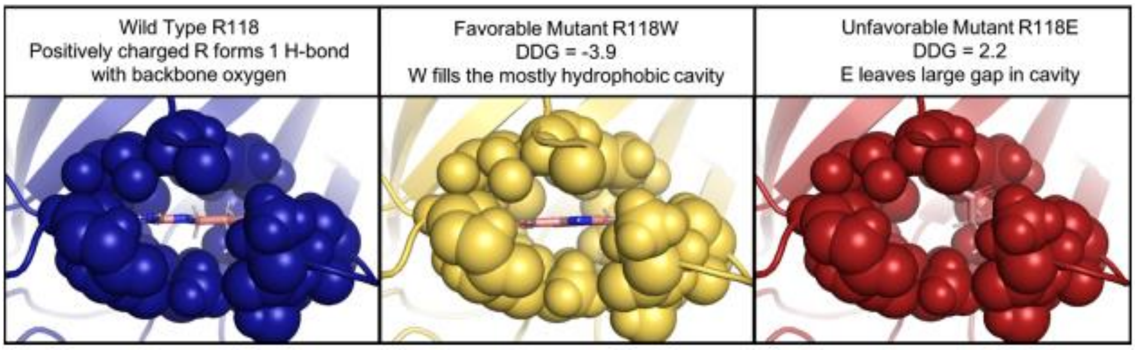

Wild Type R118
Positively charged R forms 1 H-bond
with backbone oxygen
Favorable Mutant R118W
DDG = -3.9
W fills the mostly hydrophobic cavity
Unfavorable Mutant R118E
DDG = 2.2
E leaves large gap in cavity

Figure 2. Example of protein interactions that affects the overall protein stability. (Figure extracted from Cyrus Biotech | Molecular Modeling and Design. (n.d.), from https://cyrusbio.com/)

There are different experimental techniques for protein stability measurement. Traditionally, the most used have been far-UV circular dichroism, differential scanning calorimetry and fluorescence spectroscopy using unfolding methods (Sanavia et al., 2020).

Other additional techniques have been developed: Differential Scanning Fluorimetry, SPROX (Stability of Proteins from Rates of Oxidation), high-throughput stability analysis using yeast surface two-hybrid system, Nuclear Magnetic Resonance, Pulse Proteolysis, Capillary Iso-electric Focusing with Whole-Column Imaging detection (CIEF-WCID) and 96-Well Microtitre Plates (Ó'Fágáin, 2017).

1.2. Types and classification of available computational protein stability predictive methods

There are lots of computational methods that predict the changes in protein stability caused by mutations as a measure of the difference in free energy of unfolding between wildtype and its mutant (Khan & Vihinen, 2010a). These methods are bioinformatics predictors that mainly combine computer science, physics, statistics, and mathematics (Farhoodi et al., 2017), being rather easy to use due to their simple input (protein structure, generally in PDB) format and output (a value or estimation for change in stability) (Broom et al., 2017).

We can classify different methods in four categories: physical potential approaches, statistical potential approaches, empirical potential approaches and machine learning methods. Newest and most robustness methods usually make combinations of them.

1.2.1. Physical Potential Approach

This type of approach, the earliest to be used, simulates the atomic force fields of the protein structure, typically by means of molecular mechanics in the form of molecular dynamics and/or monte carlo methods, to obtain relative free energies levels (Bash, Singh, Langridge, & Kollman, 1987; Prevost, Wodak, Tidor, & Karplus, 1991). The difference in unfolding free energy upon single mutations in the protein can be computed with the statistical mechanical relation (Pitera & Kollman, 2000). In particular, they use a thermodynamic cycle approach. The idea is to use processes that are easily studied theoretically (modelled) in replacement of the real physical process of interest, folding in this case, thus facilitating the computation of the former difference (Wong & McCammon, 1987).

Physical potential approaches are computationally very expensive, so they are used only on small sets of mutants (Guerois, Nielsen, & Serrano, 2002). One possible solution to the intense computation is the use of implicit terms for solvation energies and side-chains entropies, although it still needs a significant amount of time to get a reliable estimation (Guerois et al., 2002).

1.2.2. Statistical Potential Approach

Statistical potential approaches use potential functions derived from statistical analysis of protein features to make predictions. Protein features are extracted from experimentally databases of known proteins (Gilis & Rooman, 1997). Usually, they derive various types of potentials from databases and compute the changes in folding free energies. Then, they compared the results with the measures one in the real database. Statistical potentials are relatively simple, accurate and computationally efficient (Shen & Sali, 2006). It is important to select the correct potentials to estimate the stability changes caused by mutations, depending of the role that they play in the protein stability, so there are different approaches that can be used to characterize the contribution to folding stability (Gilis & Rooman, 1996; Zhou & Zhou, 2004). Probabilities are transformed into energy functions employing usually Boltzmann's law and it is justified by theory of conditional probabilities and linear and quadratic information theory (Cossio, Granata, Laio, Seno, & Trovato, 2012). A drawback is the difficulty to add improvements without introducing overlaps in the underlying energies (Guerois et al., 2002).

1.2.3. Empirical Potential Approach

Empirical potential approaches combine weighted physical, energy terms and structural descriptors to produce the energy function containing an optimised set of parameters. It is basically a combination of the elements of the two previous approaches (Schymkowitz et al., 2005).

They are computer algorithms whose energy terms have been weighted using empirical data. It is difficult to set the balance between the different energy terms that contribute to protein stability, for developing the protein force-field (Guerois et al., 2002). They combine a physical description of the interactions with features learned from experimental data. The resulting energy function uses a minimum of computational resources (Kellogg, Leaver-Fay, & Baker, 2011).

1.2.3. Machine Learning Approach

Machine learning approaches are trained with protein examples that have the wildtype protein and its mutant with the experimental measure of the change in unfolding free energy. They can be divided into sequence-based methods and structure-based methods depending on the input information.

Machine learning methods learn a function from large data sets, and a set of features established, so these methods map the input information to the energy change. It does not consider the physics underlying mutation stability, although they combine selected physical, statistical and empirical features (Fang, 2020). There are a wide range of machine learning algorithms to face the problem of predict thermostability

They often reduce the amount of computer resources needed when compared with other approaches. They usually require as features the sequence, solvent accessibility, pH, temperature, or structure information, to make accurate prediction, being generally applicable to any protein (Zhou, X., & Cheng, J., 2016). The main limitation is the size and selection of the training and testing datasets, with the caution of overtraining. Moreover, several performance reviews failed to reproduce the high accuracies reported by authors once they change the databases (Buß, Rudat, & Ochsenreither, 2018; Khan & Vihinen, 2010a), resulting in real applicability limitations.

1.3. Machine learning

Machine learning is a branch of artificial intelligence that studies computer algorithms that classify, group, and learn from data with the objective of improving themselves automatically through experience (Awad & Khanna, 2015). They build a model using a set of data points labelled by the corresponding output value. Once the model is trained, it can make predictions on a set of new data points.

There are traditionally three approaches: supervised learning, where the algorithm learns a rule to map inputs to outputs given a set of data. Unsupervised learning where the algorithm on its own search for a structure in the input information. Reinforcement learning, the algorithm learns from a dynamic environment when exact models are infeasible (Radford, Metz, & Chintala, 2016).

Machine learning finds generalizable predictive patterns. They have been used for many applications such as handwriting recognition, face detection, speaker identification, microarray expression data analysis… (Awad & Khanna, 2015). They have been also developed to help predict changes in protein stability upon point mutations and to infer critical residues.

Principal machine learning techniques of interest to this master thesis:

- Support vector machines (SVM). They were developed by Vladimir N. (1995) and can be used for solving classification and regression problems. SVM generates a regression function that maps implicitly input data using a kernel function (this kernel function can be linear or non linear – polynomial, sigmoid or radial). The drawback is that it can suffer from overfitting due to the use of kernel functions that can introduce them.

- Random forest (RF) was developed by Breiman (2001). It is capable for binary classification as well as regression problems. It is an ensemble learning method that utilizes the power of decision trees on multiple random sub-samples of the training set. Each decision tree infers a decision from the training data following a decision rule generated based on the value of a feature. It is resistant to overfitting problems and fast training processes.
- Artificial neural network (ANN), can be used both for classification and regression, requiring a large diversity of training sets. When an element of the neural network fails, it can continue thanks to its parallel nature that simulates biological neural networks (Jia, Yarlagadda, & Reed, 2015a).

1.4. State of art

Different types of approaches have been developed to increase the thermostability of proteins. One of the first experimental approaches was directed evolution which consists of subjecting a protein to random mutagenesis, experimentally testing and, finally, selecting the best mutants that have the properties we were looking for (Socha & Tokuriki, 2013). Then, the process is repeated once and again until the desired result is obtained. The drawbacks are that directed evolution is expensive and slow because the number of mutations that need to be tested increases exponentially.

As an alternative for experimental methods, other computational approaches quickly emerged, which were quite efficient and lowered costs.

Consensus design is a phylogeny-based stability design technique that has been widely used. The proteins that have been improved with this technique have achieved increments of up to 20 ºC (Lehmann, Pasamontes, Lassen, & Wyss, 2000). The main advantage is that neither the protein structure nor an energy model are required. During evolution, proteins accumulate a series of mutations that can be destabilizing, so when studying a family of homologous proteins, the most conserved aminoacids for each position will be the most stabilizing while maintaining the structure and function of the protein, as they have been conserved in evolution (Steipe, Schiller, Pluäckthun, & Steinbacher, 1994). As a drawback, it can lead to false positives since it does not consider the atomic details of the protein and may require experimental testing. Moreover, a large number of homologous sequences is needed to obtain unambiguous sequence alignments (Lehmann, Pasamontes, Lassen, & Wyss, 2000).

There are also some protocols, like FRESCO (Framework for Rapid Enzyme Stabilization by Computational libraries), which have a first computational design part that scans the entire protein structure to identify stabilizing disulphide bonds and point mutations. Then, it explores their effect by molecular dynamic simulations, and provides mutant libraries with variants that have a good chance to exhibit enhanced stability to produce highly robust enzymes in its second experimental validation part (Just, 2014; Muk et al., 2019). However, it needs a large amount of calculation power to carry out their molecular dynamics simulations since it increases along with the protein size and shape.

Other tools, like the webserver Fireprot (Musil et al., 2017), automatically designs thermostable multiple point mutant proteins combining a consensus of two bioinformatic predictors (FoldX and Rosetta). The iRDP webserver (Panigrahi, Sule, Ghanate, Ramasamy, & Suresh, 2015) is another existing platform that combines iCAPS, iStability and iMutant modules to an effective rational engineering of proteins.

In recent years, the development of metapredictors such as DUET (Pires, Ascher, & Blundell, 2014a), iStable 2.0 (Chen, Lin, Liao, Chang, & Chu, 2020), Dynamut (Rodrigues, Pires, & Ascher, 2018) and Threefoil (Broom et al., 2017) has increased in order to search for thermostable proteins. The metapredictors combine different individual predictors to which they give different importance for each type of mutation, so that the overall result is better than the individual results. The idea of using metapredictors that combine other tools has already been successfully used in other areas such as the covalent modification of proteins (Wan et al., 2008) or protein aggregation (Emily, Talvas, & Delamarche, 2013).

DUET combines mCSM (Pires, Ascher, & Blundell, 2014b) and SDM (Pandurangan, Ochoa-Montaño, Ascher, & Blundell, 2017). Dynamut studies both protein stability and dynamics, calculated with Bio3D (Grant, Rodrigues, ElSawy, McCammon, & Caves, 2006), ENCoM (Frappier, Chartier, & Najmanovich, 2015) and the DUET metapredictor. iStable is an online metapredictor that combines 10 individual predictors and its own machine learning algorithm. Threefoil combines 11 biopredictors by making a scale based on how well each one works.

However, very few of these predictive tools have been experimentally tested to improve protein stability (Deng et al., 2014; Floor et al., 2014; Heselpoth, Yin, Moult, & Nelson, 2015; Larsen et al., 2015; Song et al., 2013).

For this thesis, we will use the results of different multiple point mutation studies that achieved an increased thermostability of different enzymes: limonene epoxide hydrolase (LEH), a dimer of 149 aminoacids which increased +35 ºC the mean temperature of protein unfolding (Wijma et al., 2014), ω-transaminase (ω-TA), a dimer of 455 aminoacids which increased +23 ºC the mean temperature of protein unfolding (Meng et al., 2020) and short-chain dehydrogenase (ADHA), a tetramer of 246 aminoacids which increased +45 ºC the mean temperature of protein unfolding (Aalbers et al., 2020).

2. OBJECTIVE

The present master thesis consists in developing an integrated analysis tool for the prediction of stabilizing mutations as part of a more ambitious objective which is the development of a new methodology to get thermostable proteins. The following points are primary objectives:

1. Designing and developing a new metapredictor mixing different bioinformatics predictive tools.
2. Performance evaluation of the new metapredictor.
3. Application to three retrospective studies and a prospective one and an analysis of the comparative result.

3. MATERIALS AND METHODS

3.1. Single-mutation datasets

We built a dataset that contains single point mutations with its corresponding wildtype and the ΔΔG value experimentally determined as other studies have done before (Khan & Vihinen, 2010a). The signs for ΔΔG values are the opposite of those given in the Protherm database (Bava, Gromiha, Uedaira, Kitajima, & Sarai, 2004) because our consensus is that negative values are stabilizing ones. Moreover, we express the values in Kcal/mol.

We use a dataset from Varibench (Sasidharan Nair & Vihinen, 2013), which is a benchmark database suite that comprises several experimental validated subdatabases, screening and cleaning of redundant data and manually checking, that have been used previously for developing and testing other prediction tools. The data is derived from Protherm, a huge database that contains a collection of thermodynamic measures of protein stability and it is freely available at https://www.iitm.ac.in/bioinfo/ProTherm/. This dataset contains 1784 mutations from 80 proteins with experimentally determined ΔΔG values. There are 931 destabilizing mutations (ΔΔG < 0.5), 631 neutral mutations and 222 stabilizing mutations (Khan & Vihinen, 2010a; Kumar et al., 2006).

From the first dataset, we generate a subset where there are 79 different proteins and 1000 mutations (See Supplementary Table 2), where 448 mutations have a negative ΔΔG value (stabilizing), 44 mutations with a ΔΔG value equal to 0 and 508 mutations with a positive ΔΔG value (destabilizing). Then, we randomly divided our database in two parts, one for training (training data) and other for testing (testing data).

In testing data there are 59 different proteins and 300 mutations, where 145 mutations have a negative ΔΔG value (stabilizing), 11 mutations with a ΔΔG value equal to 0 and 144 mutations with a positive ΔΔG value (destabilizing).

Each mutation in the dataset has the PDB identification code, the mutation, wild-type and position and the reference to the study from which it was extracted.

3.2. Available published bioinformatic predictors used in this work

For ΔΔG predictions we used the following tools: MAESTRO, CUPSAT, AUTOMUTE-SVM and TR, FOLDX, INPS3D, MUPRO, I-MUTANT, EVOEF and IPTREESTAB which main characteristics are resume in Table 1.

3.2.1. MAESTRO (Multi AgEnt STability pRedictiOn, https://pbwww.che.sbg.ac.at/maestro/web/maestro/workflow)

MAESTRO (Laimer, Hiebl-Flach, Lengauer, & Lackner, 2016; Laimer, Hofer, Fritz, Wegenkittl, & Lackner, 2015) is a structure-based predictor of unfolding free energy change in proteins upon point mutations that requires protein structures as input. The structure can be either experimentally resolved or modelled. MAESTRO makes predictions using statistical scoring functions and protein properties, following a multi-agent machine learning strategy that combines three artificial neural networks, three support vector machines and a multiple linear regression to get a consensus value that is filtered to remove outliers. Finally, MAESTRO gives a consensus ΔΔG prediction and its corresponding confidence score. We specify the temperature and pH values when they are different from 25 ◦C and 7, respectively.

3.2.2. CUPSAT (Cologne University Protein Stability Analysis Tool, http://cupsat.tu-bs.de/)

CUPSAT (Parthiban, Gromiha, Abhinandan, & Schomburg, 2007; Parthiban, Gromiha, & Schomburg, 2006) is a structure-based predictor of the change in protein stability upon point mutations. It uses the specific environment of the mutation site, which is assessed with the aminoacid atom potentials and torsion angles potentials to build a prediction model that allows prediction of the difference in unfolding free energy between the wildtype and the mutant proteins. It uses statistical potentials without machine learning.

3.2.3. INPS3D (Impact of Non synonymous variations on Protein Stability, http://inpsmd.biocomp.unibo.it)

INPS 3D (Fariselli, Martelli, Savojardo, & Casadio, 2015; Savojardo, Fariselli, Martelli, & Casadio, 2016) is a structured-based predictor that uses nine sequence features (as the INPS sequence version), but now includes two new features derived from the protein 3D structure: the relative solvent accessibility (RSA) of the native residue and the local energy difference between wildtype and mutant protein structures. It uses a Support Vector Regression with a radial basis function kernel to predict the ΔΔG value. We will refer to it as INPS.

3.2.4. AUTOMUTE 2.0 (AUTOmated server for predicting functional consequences of amino acid MUTations in protEins, http://binf.gmu.edu/automute/)

AUTOMUTE (Masso & Vaisman, 2010) is a structure-based predictor that transforms aminoacids into coordinates in a 3D space. Then, it applies a tessellation and calculates a residue environment score for each tetrahedral in both the wildtype and the mutant, considering the spatial perturbation caused by the mutation in the 3D structure. It makes the vector difference of the profiles to calculate the environmental change. There are several classification and regression models available to predict the change in stability upon point mutations. We use two regression models available in the webserver, a Tree regression (REPTree) and Support Vector Machine regression (SVMreg) to obtain a predicted value of the change in free energy. We will refer them as AUTO-RT and AUTO-SVM respectively. We specify the temperature and pH values when they are different from 25 ◦C and 7, respectively.

3.2.5. iPTREE-STAB (interpretable decision tree based method for predicting protein stability, http://203.64.84.190:8080/IPTREEr/iptree.htm)

iPTREE-STAB or IPTREESTAB (L. T. Huang, Gromiha, & Ho, 2007) is a sequence-based predictor that uses the information of the three residues before and after the mutation site, experimental conditions, and a set of several rules from the knowledge of experimental conditions. With that information, it makes a classification and regression tree (CART) to get a prediction in stability change. There is also a predictive discrimination mode for classification into stabilizing and destabilizing. We specify the temperature and pH values when they are different from 25 ◦C and 7, respectively.

3.2.6. I-MUTANT 3.0.1 (http://gpcr2.biocomp.unibo.it/cgi/predictors/I-Mutant3.0/I-Mutant3.0.cgi)

I-MUTANT 3.0.1 (Capriotti, Fariselli, Calabrese, & Casadio, 2005; Capriotti, Fariselli, Rossi, & Casadio, 2008) is a structure-based or sequence-based support vector machine predictor that can classify the mutations into three classes: stabilizing, neutral or destabilizing. Moreover, it can be used as a regression estimator of the ΔΔG value upon mutation that works with Support Vector Machine. We use the protein structure version of the software to predict ΔΔG values. It is feature-based with machine learning. We specify the temperature and pH values when they are different from 25 ◦C and 7, respectively. It is a stand-alone executable.

3.2.7. MUpro (Prediction of Protein Stability Changes for Single Site Mutations from Sequences, http://mupro.proteomics.ics.uci.edu/)

We used the sequence-based predictor of MUPRO (Cheng, Randall, & Baldi, 2006) to predict the change of unfolding free energy upon point mutations. It uses a support vector machine method to predict the ΔΔG from the sequence and structure information around the mutation site. We specify the temperature and pH values when they are different from 25 ◦C and 7, respectively. It is a stand-alone feature-based machine learning executable.

3.2.8. FOLD-X (http://foldxsuite.crg.eu/)

FoldX (Schymkowitz et al., 2005) is a structure base predictor that uses an empirical potential approach. It requires to compute a previous step of reparation of the input data to minimize the energy and correct structure errors. Fold-X combines different energy terms to estimate the difference in free energy of the folded and unfolded protein. It is a stand-alone executable.

3.2.9. EVOEF (Energy Function for EvoDesign, https://zhanglab.ccmb.med.umich.edu/EvoEF/)

EVOEF (X. Huang, Pearce, & Zhang, 2020) is a structure-based predictor that employs a physics-base energy function. It is very similar to FoldX and it also requires the reparation of protein structure before computing the difference in stability between the mutant and the wildtype. It is a standalone executable.

Table 1. Resume of main characteristics of the selected predictors.

Predictor	Published (year)	Functionality	Time execution per mutation	Classification
MAESTRO	2016	Online/Stand-alone	< 15 s	Multi-agent machine learning system
CUPSAT	2006	Online	< 15 s	Statistics potential approach: specific atom potentials and torsion angle potentials.
INPS	2016	Online	< 20 s	Machine learning approach: support vector regression.
AUTOMUTE	2010	Online/Stand-alone	< 20 s	Machine learning approach: support vector machine and random forest.
iPTREESTAB	2007	Online	< 20 s	Machine learning approach: adaptive boosting algorithm, classification and regression tree.
I-mutant 3.0	2006	Online/Stand-alone	< 20 s	Machine learning approach: support vector machine based predictor.
MUPRO	2006	Online/Stand-alone	< 30 s	Machine learning approach: support vector machine based predictor.
Fold-X	2005	Stand-alone	< 30 s	Empirical potential approach: empirical force field calibrated with experimental ΔΔG values
EvoEF	2019	Online/Stand-alone	< 30 s	Empirical potential approach: empirical force field calibrated with experimental ΔΔG values

3.3. Creation of the dataset of descriptors to train machine learning

To make this process easier and increase the number of tests, we have developed a pipeline to automatically predict with the majority of the predictors the mutations from databases (ProTherm and VariBench) (Figure 3) and generate our own database of predictions and real values. We have developed scripts for each bioinformatic predictor and a general script that coordinates the request and download results, saving the data in the same format. In Supplementary Information 1, you can find a coding example with the coordination of three individual predictors.

When one predictor fails to predict due to exceptions or errors, we put a 0 for that prediction as it cannot estimate any change in stability.

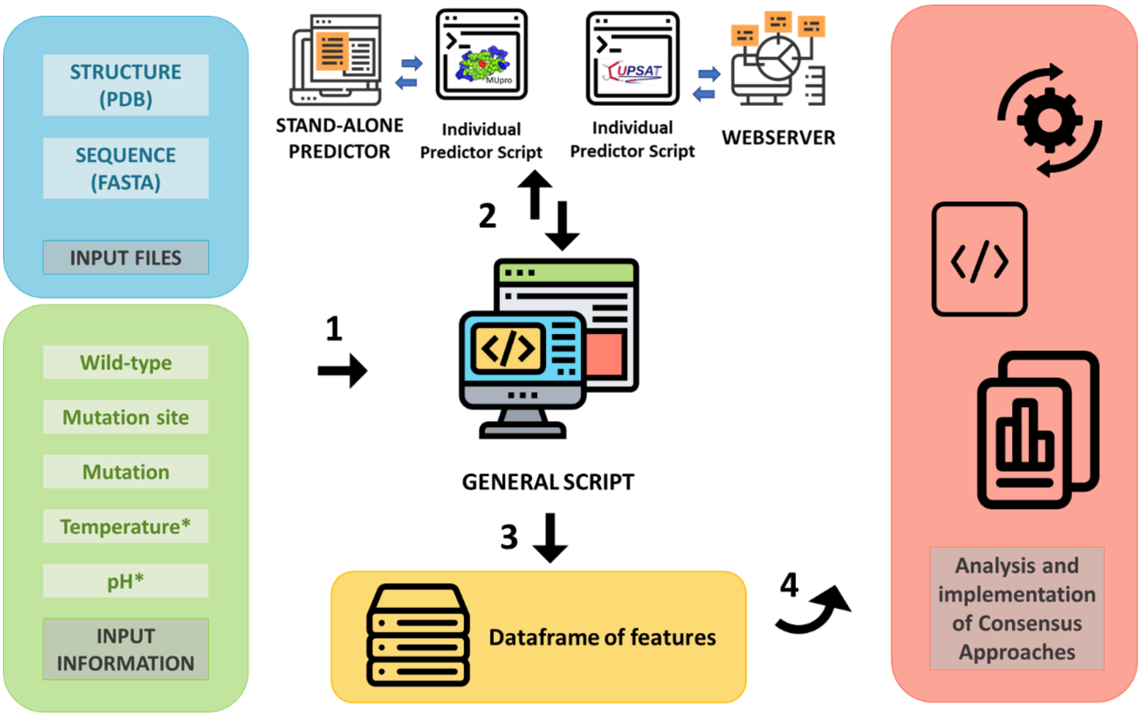


Figure 3. Scheme of the general process followed to extract and analyse data from ProTherm and VariBench databases and estimate the accuracy of the thermostability predictors. The pipeline is divided in four parts. 1) The input information required is the wild-type residue, the position and the mutation. Moreover, some predictors require the temperature and pH conditions, when it is not provided the value, we assume 25 ºC and pH 7. 2) A general/master script coordinates the request and download of different predicted values with 10 bioinformatic predictors previously selected. 3) Generation of a library with experimental mutations and predicted values. 4) Use of the new library to generate a new metapredictor that combines all previous information. *This input data is not needed for all predictors.

3.4. Machine learning algorithms to combine the methods

The input data to train the machine learning algorithm consist of a comma-separate value (csv) file where the first column is a protein-mutation identifier, followed by the ΔΔG real value extracted from the database and obtained experimentally. Then, we have set up a column called “STABILITY” that will take value 1 if ΔΔG is negative and therefore stabilizing, and value 0 otherwise. The rest of columns will be filled with the ΔΔG values predicted by each predictor for each mutation indicated in the identifier column. We create a vector of features with the prediction values of the different bioinformatic predictors.

If the energy change ΔΔG is negative, the mutation increases stability and is classified as a positive example. Otherwise, we will consider it as a negative example.

3.4.1. Random Forest Classifier

We are going to use it to classify whether an isolated mutation in a protein is stabilizing or not from the ΔΔG data provided by various available predictors. Each Random Forest Decision Tree takes as input data the results ΔΔG of a random subset of predictors and as output it will have a value of 1 (stabilizer) or value 0 (otherwise). You can also get a quantity between 0 and 1 that indicates the proportion of trees that return output 1 (stabilizer). By default, we consider a mutation as stabilizing if the proportion is bigger than 50 %.

We selected the parameters for the Random Forest that are common to RF-Classifier and RF Regressor by default ('bootstrap': True, 'min_samples_leaf': 1, 'n_estimators': 100, 'min_samples_split': 2, 'max_features': 'auto', 'max_depth': None, 'max_leaf_nodes': None)

- Bootstrap: True - means that all data is divided in random groups to train different decision trees.
- Min_samples_leaf: 1 – the tree is going to subdivide until each division gives 1.
- N_estimators: 100 – this is the number of trees in the forest. It was set to 100 after the optimization of the hyperparameter. Less trees reduced the accuracy, while increasing the number did not change the performance when we tested it.
- Min_samples_split: 2 – data is divided until there is less than this amount of data.
- Max_features: ‘auto’ – how many predictors will catch in each tree. By default, they select the root square of the number of predictors.

- Max_depth: None – this is the maximum number of divisions in each tree.
- Max_leaf_node: None – we can set the number of leaves that will have each tree.

3.4.2. Random Forest Regressor

We are going to use a random forest regressor to estimate the ΔΔG value of a mutation from the values provided by different available predictors. We keep the parameters by default.

parameters = {'bootstrap': True, 'min_samples_leaf': 1,'n_estimators': 100, 'min_samples_split': 2,'max_features': 'auto', 'max_depth': None, 'max_leaf_nodes': None, 'random_state':1}

3.4.3. 10-Fold Cross-Validation

Cross-validation procedure is a method that is used to optimize the use of available data for training and testing. The goal is to estimate the accuracy of our trained machine learning models, dividing the training and testing 10 times.

3.5. Statistical analysis of predictor's performances

3.5.1. Analysis of the classification performance

To analyse the classification performance of the different thermostability predictors, we transform all ΔΔG estimated and real values into a binary classification: stabilizing or destabilizing where stabilizing values are negatives ones and destabilizing are the positives.

To evaluate the result and analyse the predictive ability of the different predictors we develop a contingency table or confusion matrix (Stephen, 1997). In the contingency table we count the number of true positives (TP), which are the number of times in which a predictor estimates correctly that a mutation is stabilizing, true negatives (TN), are the number of times a predictor estimates correctly that a mutation is not stabilizing. We also count the number of false positives or type I error (FP), the number of times that a predictor estimates that a mutation is stabilizing while it is not, and the number of false negatives or type II error (FN), the number of times that a predictor estimates that a mutation is not stabilizing while it is.

From the handling of these terms, a series of statistics have been developed that provide information on the different methods and help to assess which is the best: accuracy (Acc), sensitivity (Sn), specificity (Sp), Precision (Pr) and Matthews Correlation Coefficient (MCC).

The accuracy corresponds to the proportion of data that have been correctly estimated by the predictors as an indicator of the overall performance:

$$Accuracy\ (Acc) = \frac{TP+TN}{TP+FP+TN+FN}$$

Precision is the number of hits in estimating positives out of the total number of positives, both false positives and true positives (Equation 2).

$$Precision\ (\mathrm{Pr}) = \frac{TP}{TP+FP}$$

Although accuracy seems to intuitively carry a lot of weight, having a low number of false positives results in higher precision (Trevethan, 2017). Other types of accuracy that are used are specificity and sensitivity. The first focuses on those correctly estimated negative values of the total existing negatives (ability to exclude true negatives). The second is the same as the previous one but for the positives (ability to include true positives).

$$Specificity\ (Sp)\ = \frac{TN}{TN+FP}$$

$$Sensitivity\ (Sn) = \frac{TP}{TP+FN}$$

Matthews Correlation Coefficient is a coefficient that measures the quality of binary classifications in the field of bioinformatics and machine learning. Their range of values goes from 1, for perfect predictions, to - 1 for opposite predictions while 0 corresponds with random predictions.

$$Matthews\ Correlation\ Coefficient\ (MCC)\ = \frac{TP\ x\ TN - FP\ x\ FN}{\sqrt{(TP+FP)(TP+FN)(TN+FP)(TN+FN)}}$$

With this data, we can define a ROC (Receiver Operating Characteristic) space, a type of plot used to illustrate the different predictor’s performances as binary classifiers (Metz, 1978). We represent the sensitivity in the y-axis and 1-specificity in the x-axis, where the perfect classifier would be in the coordinates (0,1) corresponding with a tool with no false positives and no false negatives.

3.5.2. Analysis of the regression performance

To evaluate the performance of the ability to predict an estimated value for ΔΔG, we consider the Pearson Correlation Coefficient (PCC), Mean Square Error (MSE) and Mean Absolute Error (MAE).

Pearson Correlation Coefficient (Pcc or ρ) is a measure of the linear dependence of two variables whose value ranges between 1 and -1 and is independent of the scale. In this work, we study the relationship between the estimated value (X) and the real value (Y). For a perfect predictor, the estimated and the real value should be equalled, so there would be a perfect positive correlation ρ = 1. In the case of perfect negative correlations, it would be ρ = -1. Those values that are close to 0 indicate that the data is not correlated.

$$\rho = \frac{cov(X,Y)}{sdXsdY}$$

The Mean Square Error (MSE) is a statistic that calculates the average error made when calculating the difference between the actual (Y) and the estimated value (X). It is an estimator of the bias of the measures that considers the variance.

$$MSE = \frac{\sum_{i=1}^{n}(X_i - Y_i)^2}{n}$$

The Mean Absolute Error (MAE) is a statistical measure used to estimate the precision of a method. It would be the equivalent of the distance that separates each of the data from the exact value it should have.

$$MAE = \frac{\sum_{i=1}^{n}|X_i - Y_i|}{n}$$

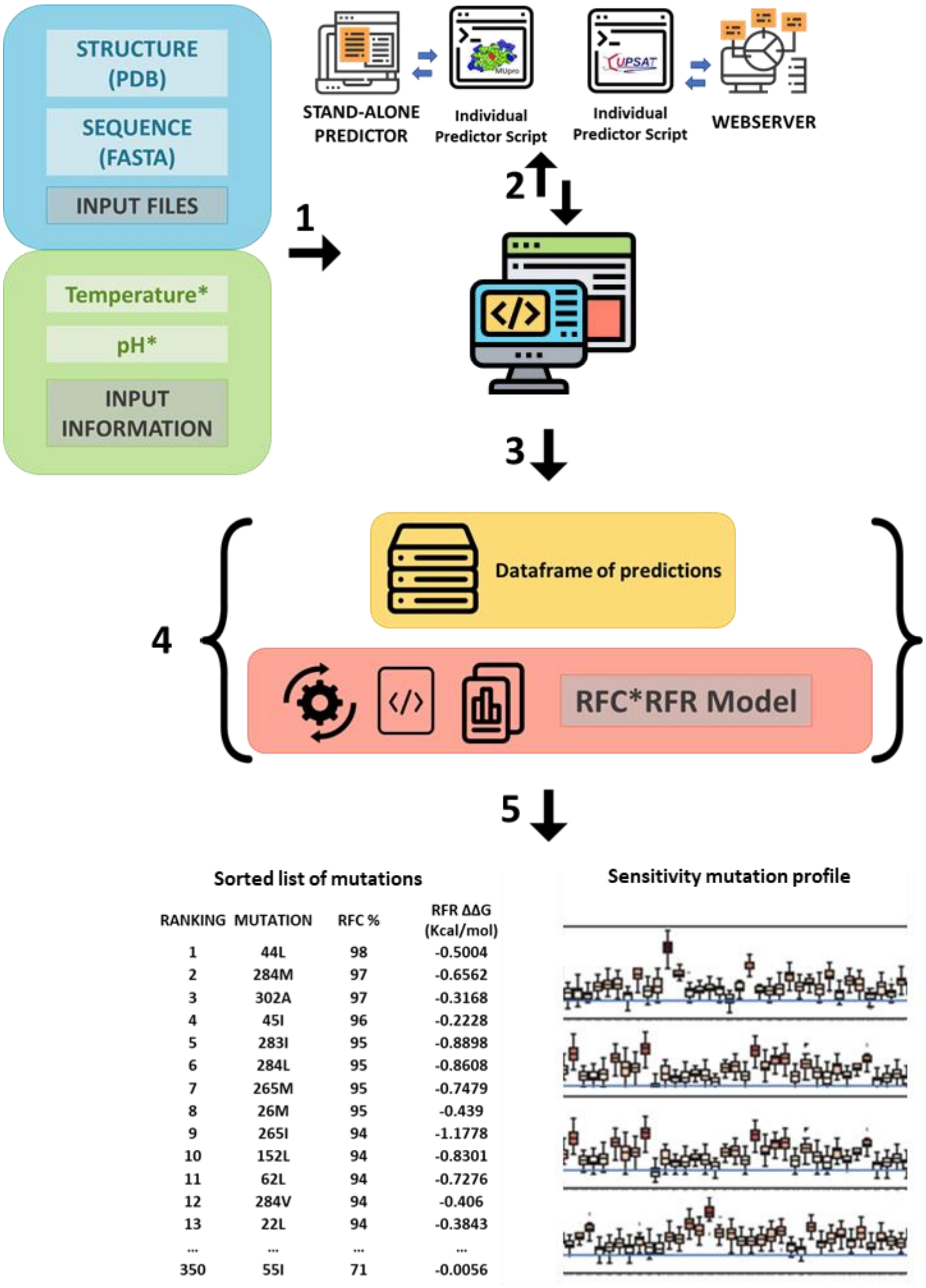
3.6. Description of the Consensus Metapredictor

Our metapredictor coordinates the massive download of the thermostability changes predicted by each of the 10 bioinformatic predictors through a series of scripts (in the Supplementary Data there is an example of MUPRO script). Once all the data is downloaded, it is integrated into a single python dictionary. Then, from that dictionary we generated a dataframe with all the predictions and we saved it in a csv file.

The required input by the combined consensus Random Forest Classifier and Regressor model is the structure of the protein or its PDB code, the protein amino acid sequence, the temperature, and the pH in which we want to work and calculate the changes in protein stability.

In resume, the process consists in a pipeline that combines web server functionalities and stand-alone tools. We combine all the predictions in a consensus so the result of the metapredictor will be more accurate than the result of all predictors separately (Figure 4).

The output consists in a list of the most stabilizing mutations obtained, sorted from highest to lowest stabilizing ΔΔG value. In addition, we also generate a thermostability sensitivity profile in the form of boxplots for a quick visual inspection.



RANKING	MUTATION	RFC %	RFR ΔΔG (Kcal/mol)
1	44L	98	-0.5004
2	284M	97	-0.6562
3	302A	97	-0.3168
4	45I	96	-0.2228
5	283I	95	-0.8898
6	284L	95	-0.8608
7	265M	95	-0.7479
8	26M	95	-0.439
9	265I	94	-1.1778
10	152L	94	-0.8301
11	62L	94	-0.7276
12	284V	94	-0.406
13	22L	94	-0.3843
...	...	...	...
350	55I	71	-0.0056

Figure 4. Scheme of the general process followed to obtain the sensitive thermostability mutation profile and the topmost stabilizing single mutations for a protein.

The consensus metapredictor model will be used for three retrospective thermostability improved proteins, limonene epoxide hydrolase (LEH), ω-transaminase (ω-TA) and short-chain dehydrogenase (ADHA). In addition, a prospective study will be carried out on an alpha/beta hydrolase enzyme (MGS-MilE3) (Stogios, 2016). The aim of the prospective is carried out a posterior experimental validation and thus verify the effectiveness of the new method. Table 2 summarizes the main characteristics of these four proteins.

Table 2. Resume of retrospective and prospective proteins information.

Study	Protein	PDB ID	SEQ	ΔTm (°C)	Total Predicted	Total Stabilizing	Hit ratio %
Retrospective	LEH	1NWW	149 aa	+ 35	268	47	17.5
	ω-TA	6G4B	455 aa	+ 23	204	31	15.2
	ADHA	6TQ5	246 aa	+ 45	177	21	11.9
Prospective	MGS-MilE3	5JD5	321 aa	-	-	-	

3.7. Python libraries

Most of the code generated has been written using the Python 3 language for the development of this thesis. The following free python libraries have been widely used:

- Math is a python library that facilitates the use of mathematical utilities.
- Matplotlib and Seaborn Python are comprehensive libraries for creating visualizations. All the graphs produced in this work have been created using any of them.
- Numpy is a python library used to work with matrices, arrays, and calculations.
- Pandas Python is a very useful library for handling large amounts of data and statistical analysis. It has been widely used for the creation of dictionaries and dataframes and the exportation of the results in csv files format.
- Pickle is a python library that allows you to manipulate data structures and save them in pkl format for later use.
- Scikit-learn is a Python library widely used for predictive data analysis. It is an open source, built on SciPy, Matplotlib and Numpy.

- Selenium Python: provides a webdriver protocol to control web browsers and automate web scraping and the control and managements of the request in the online webservers, yet it simulates clicking on buttons, filling forms or downloading results files.

4. RESULTS AND DISCUSSION

4.1. Generation of input data features for training and testing

4.1.1. Real experimental values data

A complete database has been generated with real values and the estimations of ten bioinformatic predictors that belong to empirical, statistical, machine learning and combined approaches.

Other studies that use several predictive tools (Broom et al., 2017; Khan & Vihinen, 2010b) have also generated their own databases to test and analyse the performance of the different predictors. The results from recalculation of experimental values with the different predictors are usually quite different from what the authors reported. It has been shown that the non-symmetry in the distribution of ΔΔG values and the imbalances in the proportion of stabilizing and destabilizing mutations have a great effect on machine learning based predictions (Sanavia et al., 2020). Many of the published and available datasets are usually biased to destabilizing (Montanucci, Capriotti, Frank, Ben-Tal, & Fariselli, 2019).

Some studies (Khan & Vihinen, 2010b; Montanucci, Capriotti, Frank, Ben-Tal, & Fariselli, 2019; Sasidharan Nair & Vihinen, 2013) (DDGun, Performance, Varibench…) have tried to face this problem selecting subsets of the database and controlling the proportions of both types of mutant, in a way of fixing the disbalance and to avoid a prediction bias toward destabilizing variations (Montanucci et al., 2019).

From all above, we have selected those mutations that have been reported to be manually checked in other studies or that have been experimentally reproduced (Sasidharan Nair & Vihinen, 2013). Moreover, we have tried to keep a continuous and normal distribution of all values, to create a balanced dataset (Figure 5).

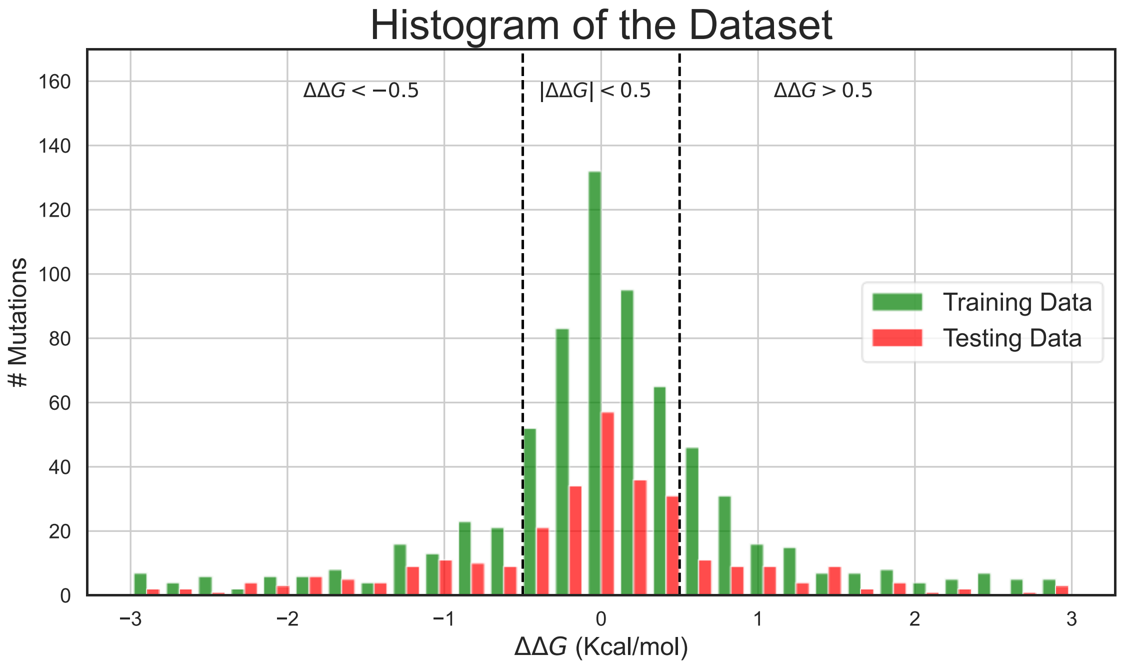


Figure 5. Histogram of the distribution of training and testing data in green and red, respectively. We can differentiate three regions depending on the value of ΔΔG, stabilizing, neutral and destabilizing.

It is very difficult to assure that datasets do not include variations used in the bioinformatic predictors training step, because the number of mutations in the database is limited, and all predictors try to use as much mutants as possible, so most of the predictors have been trained on subsets of the ProTherm database (Jia, Yarlagadda, & Reed, 2015b).

In our study, we have distributed all the stabilizing mutants and we have selected destabilizing values in the same proportion. If we observe the distribution of the values of ΔΔG in kcal/mol of all the mutants of our database, we can observe that they are distributed symmetrically (Figure 6).

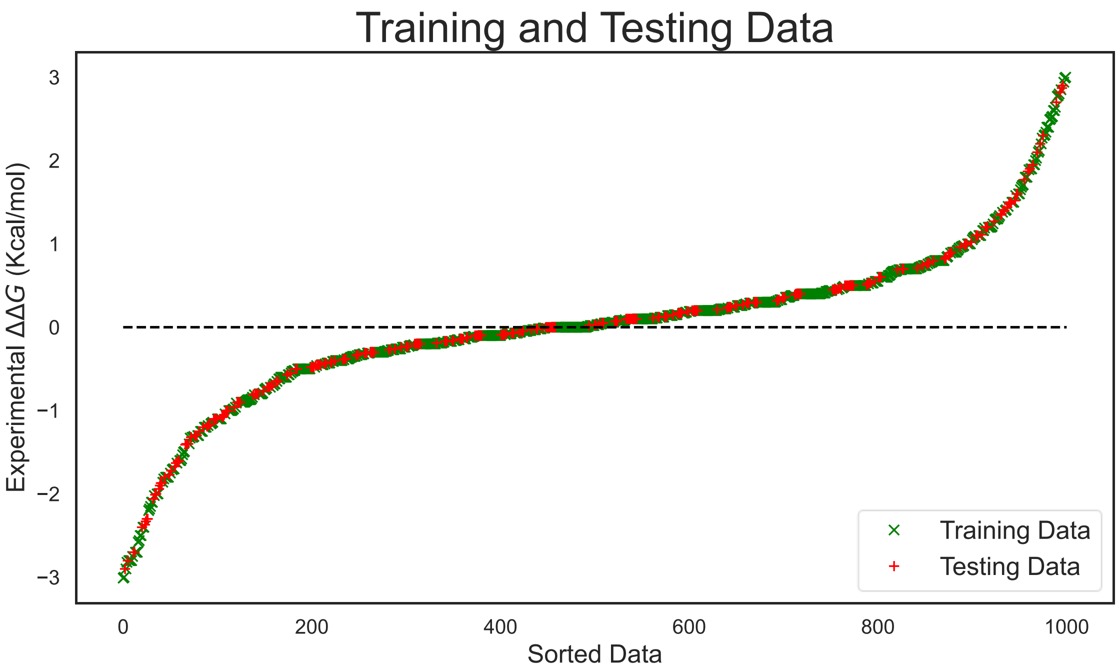


Figure 6. Distribution of empirical ΔΔG values for training and testing data. In x-axis all mutants are sorted from minimal ΔΔG value to maximum ΔΔG value and y-axis contains the empirical values in kcal/mol extracted from Protherm database and Varibench database. Training data is color green (x) and testing data in color red (+).

We need to take care when considering what we call "real data". The results from experimental data that we consider as the real values are subjected to common errors derived to the technical procedure and measure errors. Besides, authors sometimes approximate the values when there are redundancies in the databases (Kumar et al., 2006), due many times to the fact that we can find the same mutation with different values or in different conditions, so some authors average the result. Other authors, like Capriotti (2008), divides all ΔΔG values in three categories, clearly stabilizing (ΔΔG < -0.5 with our consensus sign), neutral (|ΔΔG|<0.5) and destabilizing (ΔΔG > 0.5). With this criterion we can be more confident to say that one mutation is stabilizing or destabilizing and we discard all those data that are slightly destabilizing when we need to make a decision.

Reliable reference data is an important issue for all computational simulation tools, as we cannot forget that they are models and approximations that try to explain reality. The more accurate and precise of the experimental data the more reliable will be the result. The existence of experimental databases with thermodynamic information becomes essential. They should work to increment the number, reliability and truthfulness of the data (Potapov, Cohen, & Schreiber, 2009).

4.1.2. Predictive values data

We have developed a pipeline to automatize the request and download of predictions from the different webservers to generate as output a dataframe that contains the real value, from the database (Khan & Vihinen, 2010a) and all the estimations keeping a consensus format the sign for the ΔΔG value and the units in kcal/mol. Each predictor uses its own criteria, so we need to unify them to develop a formal consensus. The most logical sense will be considered negative values for ΔΔG as stabilizing, agreeing with thermodynamics estimations.

A practical, usefulness and easy handling criteria was used to the selection of the different predictive tools with preferable short execution times and stand-alone functionalities. Some predictors such as Duet (Pires et al., 2014a), DynaMut (Rodrigues et al., 2018), mCSM (Pires et al., 2014b) and SDM (Pandurangan et al., 2017) (Supplemental Table 1) take too long to run each computation, so they were discarded as they were unfeasible to calculate all possible mutations for a given protein, which is one of the objectives of this thesis.

Other problems affecting the predictors were: i) internet crashes, ii) websites temporarily unavailable, iii) impractical request and download formats (some predictors requested an email to send the results individually or no more than one request could be sent).

Thus, the possibility to obtain from a same script several predictions for a same mutant in a relatively fast way is advantageous. And, moreover, it is the first step to an analysis to combine all that information and increment in a synergistically way the result.

Although mostly freely available published predictors report high reliable and accuracy ratio, actually, only few of them have been laboratory tested. Comparative studies have usually shown underperformances (Broom et al., 2017). Moreover, it should be noted that there are few databases that have not been used to train predictors, so we make redundancy mistakes.

4.2. Creation of a thermostability consensus predictor

In this thesis, we try to generate a consensus solution between several bioinformatic predictor tools that improve the sensitivity to estimate the stability of a mutation in a significant way, in comparison with the results of the different biopredictors taken into account separately. Our idea is to use the predictive power of all predictors combined as a first step of a methodology that will increase proteins' thermostability with an interest mainly focused on the industry, leaving aside the biomedical study of destabilizing mutations (Figure 7).

In order to achieve our goal, we selected: MAESTRO, CUPSAT, AUTOMUTE-SVM and TR, FOLDX, INPS, MUPRO, I-MUTANT, EVOEF and IPTREESTAB.

Our first consensus approach was "MEAN-PRED", a mean consensus which consists in the average of the outputs of all the predictors. Those values that differed from the average by three times the value of the standard deviation were excluded. These outliers have a probability of occurrence of less than a 3 %.

MEAN-PRED has been previously tested with other predictors (Khan & Vihinen, 2010b). The main disadvantage is that all predictors have the same weight when averaging the values, so if one predictor is less precise, the accuracy of the others decreases. In some studies, they average the value of each predictor depending on the characteristics of the mutation, based on a previous study in which they determine which predictor is the most reliable in each case (Broom et al., 2017). Although the result was promising, it was necessary to test it more times.

Another possibility was to create a classifier to separate stabilizing mutations from destabilizing ones depending on the majority consensus sign. This approach was called "SIGN-PRED". If half or more of the predictors estimate that the mutation is stabilizing, we trust the consensus and decide that it is so. In case of a draw, we assume that the mutation is stabilizing to avoid losing predictive power (type II error).

Finally, we are facing a decision problem for each position trying to guess if a given mutation is stabilizing or destabilizing. As we have mentioned before in the introduction, there are robust and important algorithms in machine learning that are specialized in learning how to decide given a set of features. Within the machine learning algorithms, Random Forest stands out for its robustness. The approach using Random Forest is much more efficient and performs better when making decisions since this type of algorithm learns how to distribute the accuracy of the different predictors to increase the overall performance.

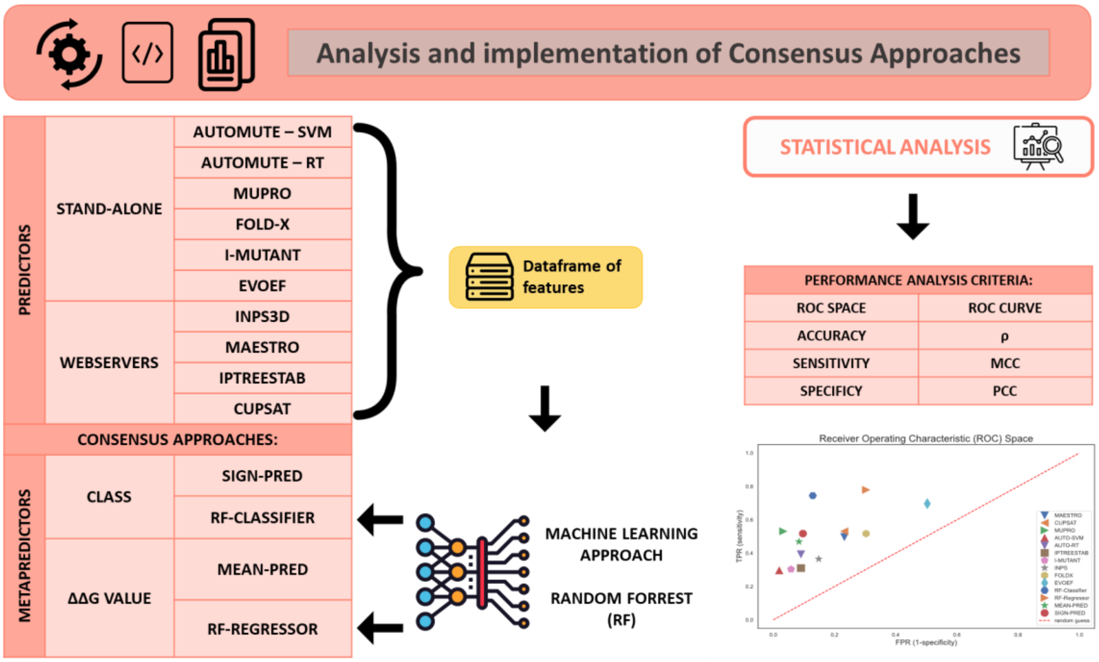


Figure 7. Scheme of the analysis and implementation of Consensus Approaches. We can differentiate the already developed predictors (divided in stand-alone or webservers) and the metapredictors generated in this work that combines the outputs of the previous ones. To select the best method, we performed a statistical analysis of the different predictors' characteristics.

We used both versions available from Random Forest ensemble, Classifier and Regressor, and we have generated two models, RF-Classifier and RF-Regressor, which have stood out for their good results. On the one hand, the former can tell apart the stabilizing mutations from the destabilizing ones thanks to their percentage associated with the probability of the estimation. Cappriotti (2004) reports that in many cases, the classification of the mutation is more relevant than its value. On the other hand, the latter can approximate the real value of ΔΔG.

All different predictors that participated in the training are important. After defining the Random Forest Classifier and training with the learning data previously selected, we can see the mean proportional importance of each predictor in the predictions made by the RF-Classifier. Among the different predictors that work as features for the model, the most important is MUPRO which participates with a relevance of 26 % followed by I-MUTANT 12.9 % and the least important is AUTOMUTE-SVM with a relevance of 5.5 % (Figure 8A).

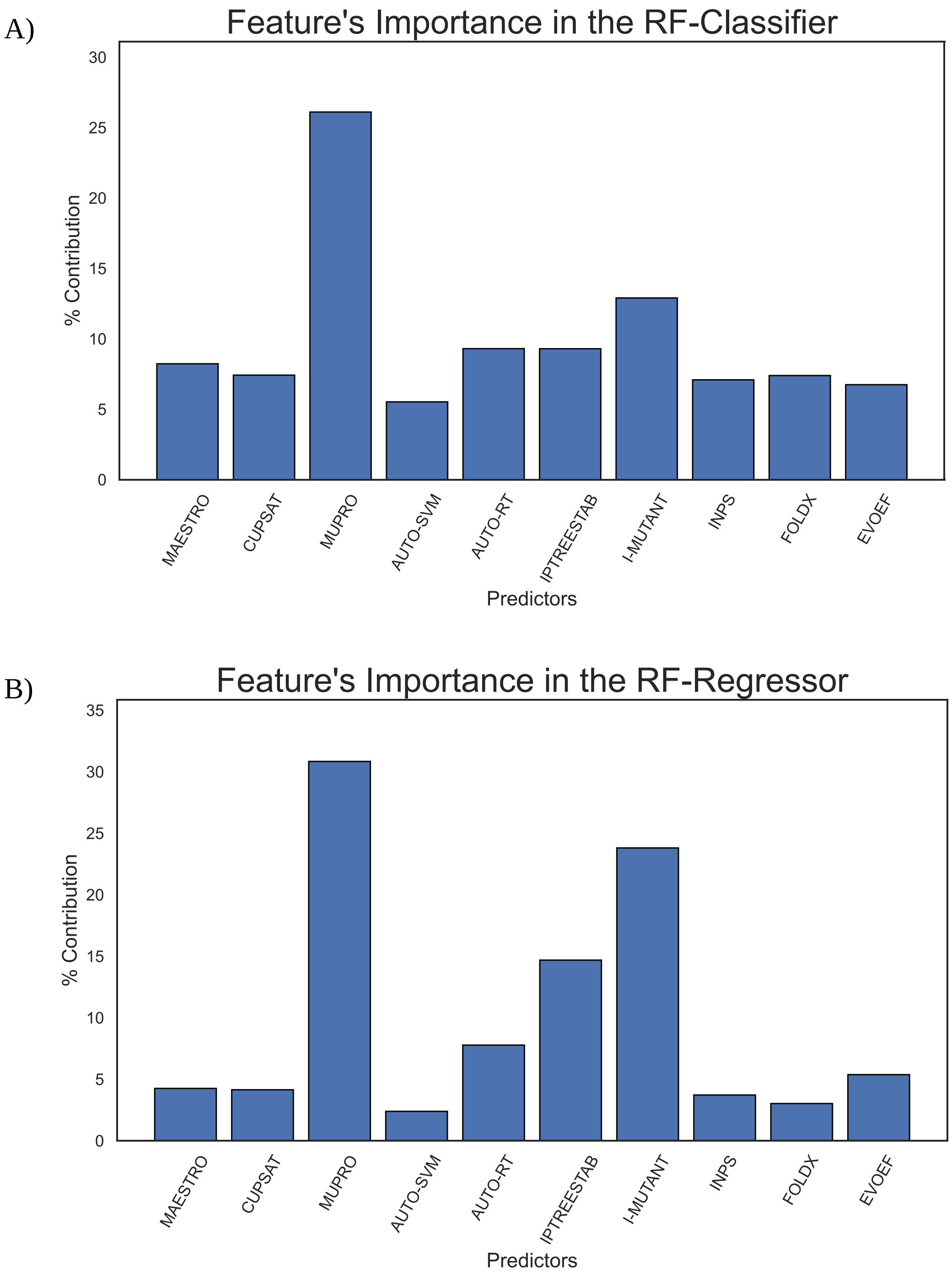


Figure 8. Feature's importance in RF-Classifier (A) and RF-Regressor (B). The importance of each predictor is highlighted by a barplot.

In RF-Regressor, when we trained the algorithm, the predictors that acquire more relevance are: MUPRO with a relevance of 30.8 % followed by I-MUTANT with 23 % and IPTREESTAB with 14.7 % (Figure 8B). The least representative is again AUTOMUTE-SVM with a 3 %.

Although there may be significant differences in the apparent contributions made by each predictor when these are compared, Random Forest learns from all the features achieving a synergistic effect that takes advantage of all the predictors.

4.3. Analysis and evaluation of the performances

One advantage of Random Forest Predictor is its robustness and the low risk of overfitting. After applying a 5-fold cross-validation and repeating 10 times this validation for RF-Classifier, we obtain a mean accuracy of 0.77 (+/- 0.07). The accuracy can vary in an interval equal to (0.70, 0.84) with a high probability (in a normal distribution that probability is 95%), regardless of the division of the dataset into test and training. Similarly, in RF-Regressor the mean square error is 0.43 Kcal/mol and can vary in an interval equal to (0.22, 0.64) with high probability.

4.3.1 Accuracy of the regressions

In a predicted value vs experimental value axis graph, the accuracy of the predictions can be seen to the extent that the resulting points for each mutation in the test group are distributed along the diagonal line of the first quadrant (Figure 9A). We compare this line with the linear regression line of the resulting scatter plot, also showing its Pearson correlation coefficient and the least squares error associated with each predictor (Figure 9B).

The best predictors are RF-Regressor with a correlation coefficient of 0.792, followed by MEAN-PRED, I-MUTANT and MUPRO, whose Pcc are 0.595, 0.585 and 0.55 respectively.

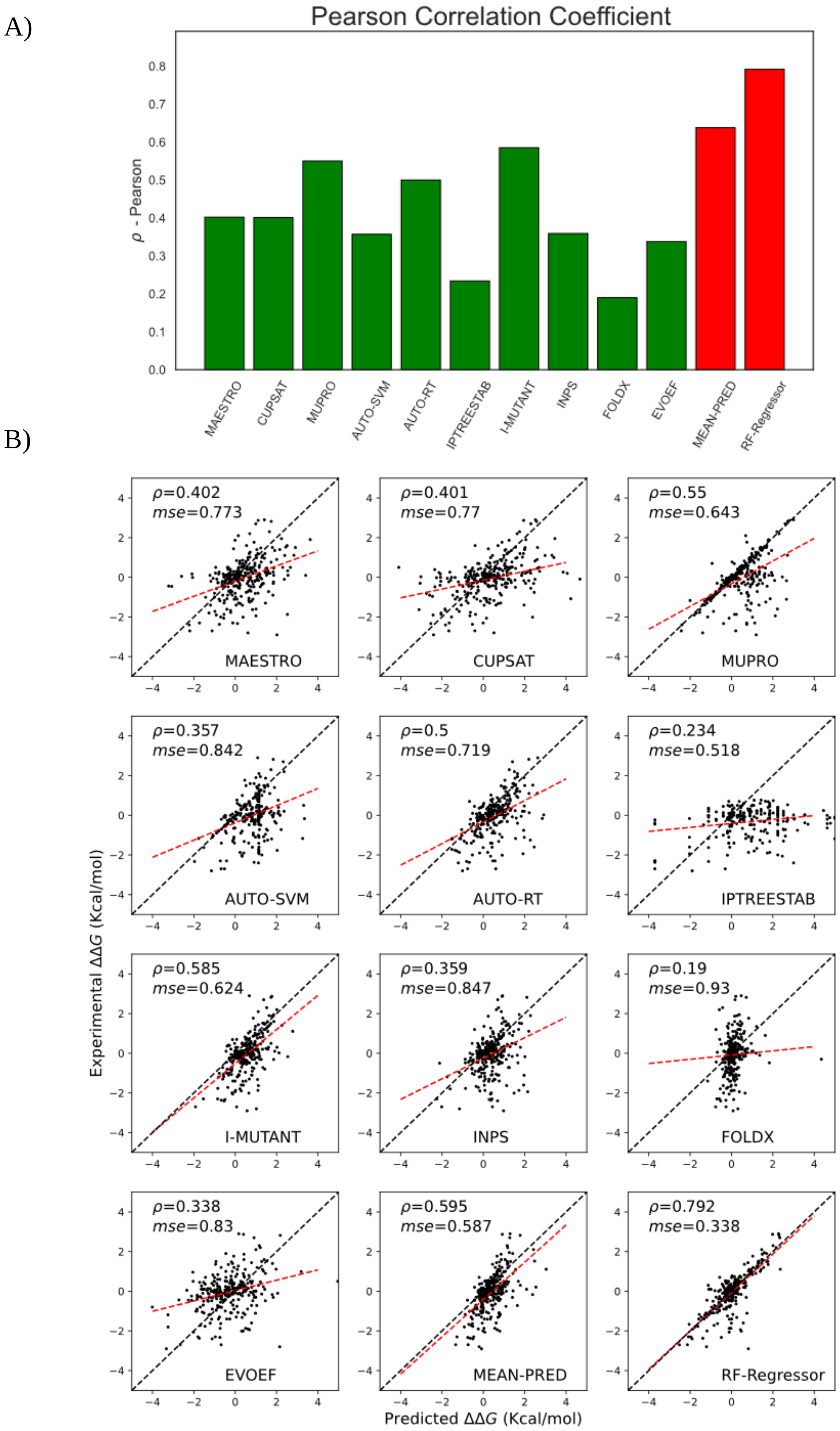


Figure 9. A) Representation of the Pearson Correlation Coefficient of each predictor. B) Relationship between predicted and experimental changes in stability for the different predictors. The dotted black line indicates a perfect correlation, whereas the red dotted line indicates the correlation of the predicted and experimental values.

We also calculate the mean absolute error committed by each predictor as another measure of the accuracy. It is evident that the metapredictor RF-Regressor is the most accurate following this criterion since it has the lowest mean absolute error (Figure 10A). Moreover, if we observe the distribution of the error values, best predictors are MUPRO and RF-Regressors, as their boxplot and mean are proximate to 0, which is no error at all. Narrow boxes indicate that the different measures have approximate absolute errors. The worst is IPTREESTAB, whose median of error values is the furthest from 0 and has the biggest dispersion of absolute errors. MUPRO has a very low median, but its dispersion is bigger than RF-Regressor (Figure 10B).

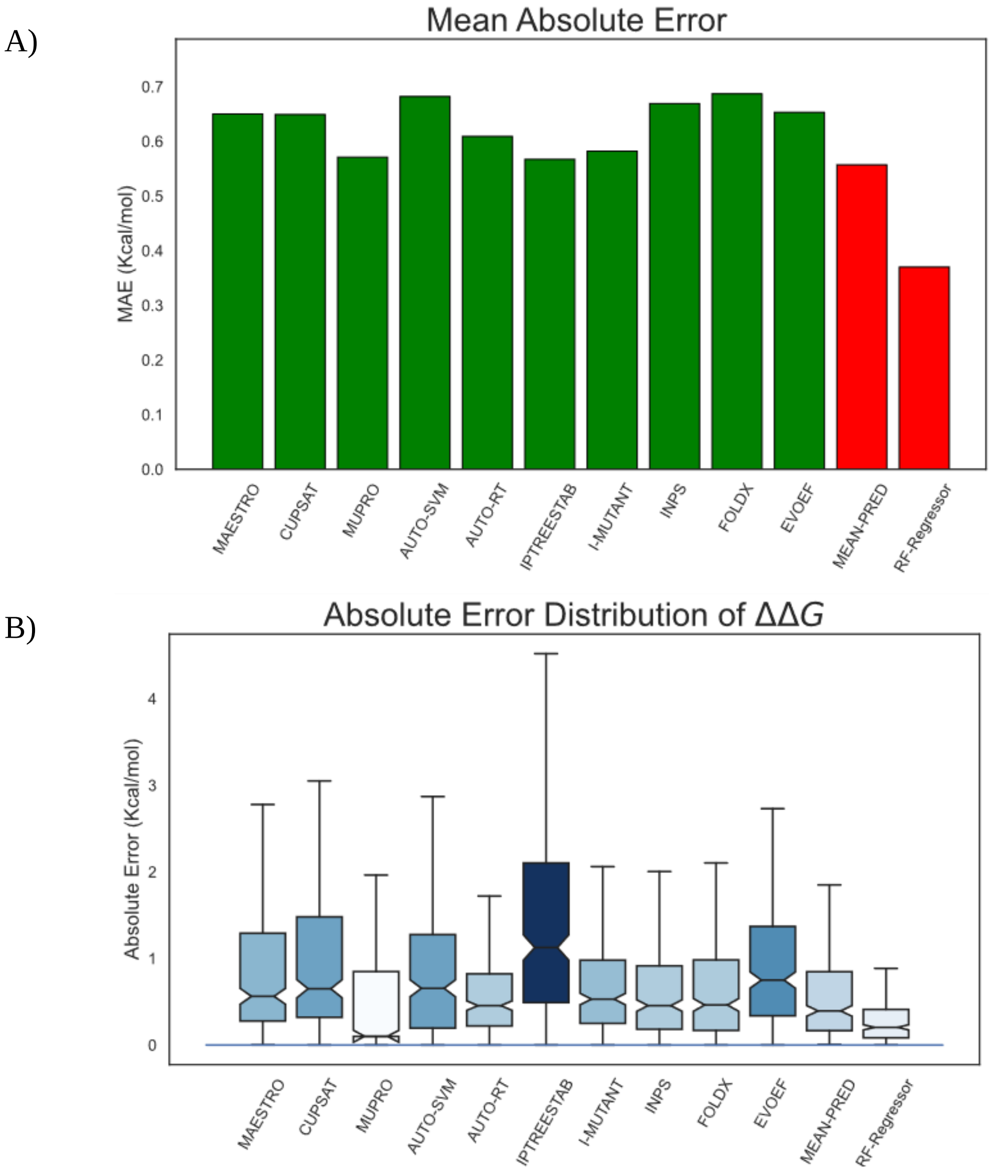


Figure 10. A) Representation of the mean absolute error of each predictor. B) Distribution of the absolute error of measuring ΔΔG. The median absolute error corresponds with the notched region. The coloured region of each box includes the 50 % of the data, the rest are inside the whiskers. Outliers have been omitted in this plot.

4.3.2. Accuracy of the classifications

We calculate the sensitivity, specificity, precision, accuracy and MCC of all predictors to be able to compare with the same conditions. For further information about these statistics see materials and methods.

We use ROC analysis considering the sensitivity and specificity of each predictor as a classifier of the stability of the mutations, representing them as points in the ROC space (Figure 11). Predictors are better classifiers insofar as they are represented as points closest to the point (0,1), which represents an ideal infallible classifier (see materials and methods). A quick visualization of the best predictor can be done with ROC space.

The results show that our method, using the three encoding schemes, performs better than all other methods using most evaluation measures.

The best predictor is RF-Classifier followed by RF-Regressor. The worst is EvoEf with the worst specificity, precision, accuracy and MCC. Random Forest increase the mean values of all the other predictors, improving strengths and decreasing weaknesses.

RF-Classifier is the closest predictor to position (0,1) that corresponds to a perfect classifier (0.28 u), whereas the furthest is AUTO-SVM (0.70 u). However, the worst predictor would be EvoEF due to its distance from both axes, so it is less specific than any other.

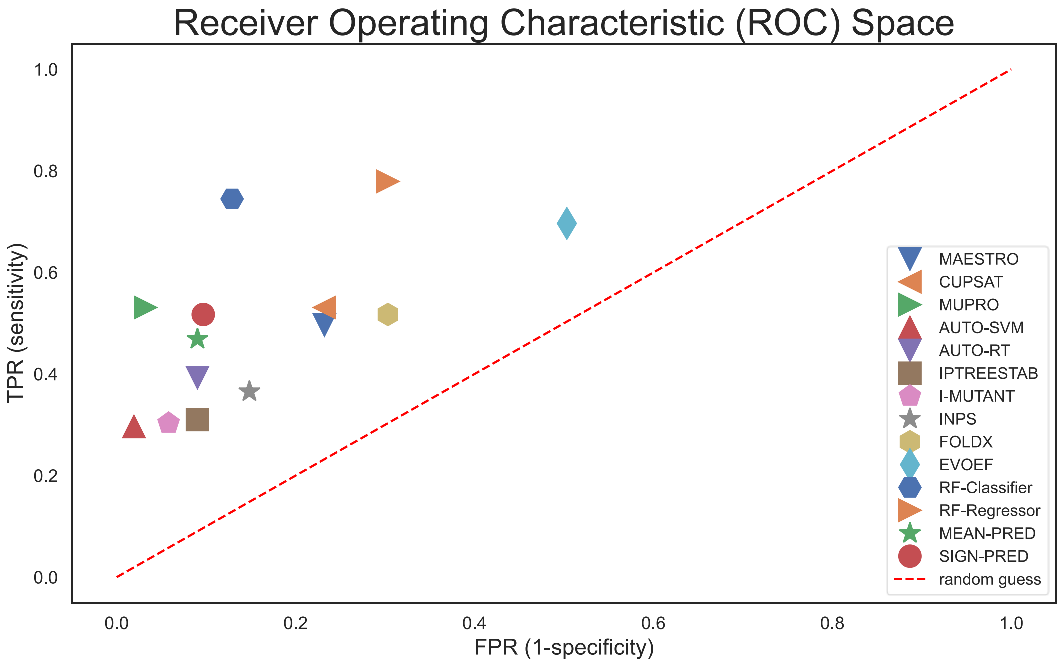


Figure 11. Comparison of the overall performance in Receiver Operating Characteristic (ROC) Space. Each predictor is represented by a different symbol. The red dotted line diagonal represents random guess performances.

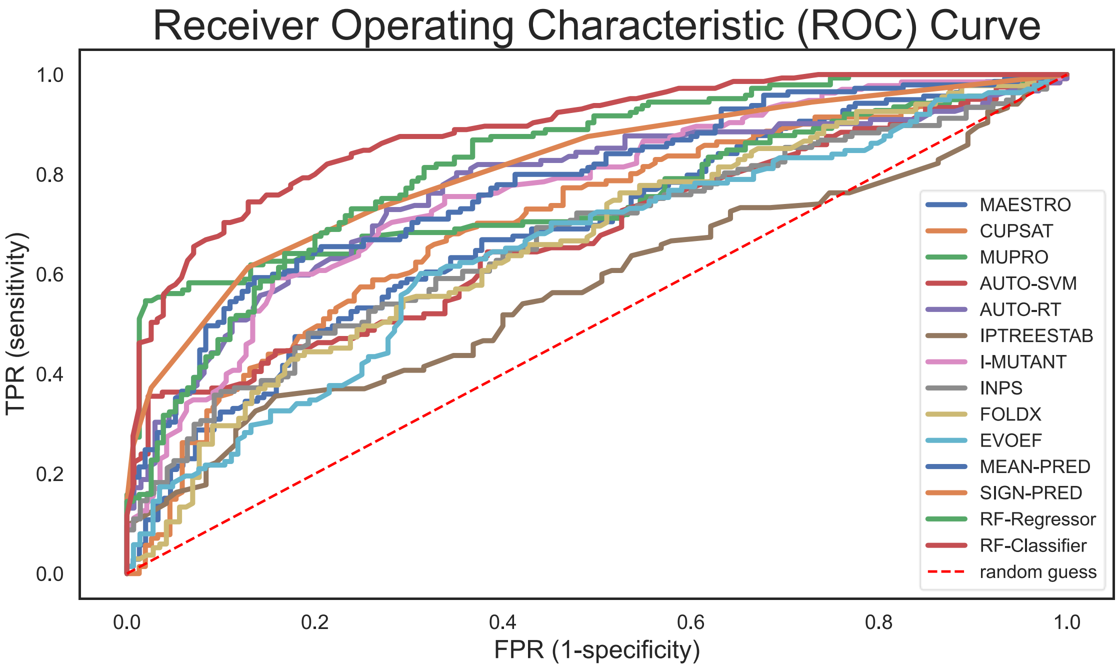


Figure 12. Comparison of the overall performance in Receiver Operating Characteristic (ROC) Space. Each predictor is represented by a different color line. The red dotted line diagonal represents random guess performances.

Another indicator widely used to compare classification methods is the AUC (Area Under the ROC Curve). The classification of a mutation between stabilizing or destabilizing is based on a threshold ΔΔG value which has been set to 0 by default. Each ROC curve is generated varying the threshold from the minimum ΔΔG value to the maximum and calculating the corresponding associated dot in the ROC space.

The AUC is the mean sensitivity value for all possible specificity values and allows the comparison between predictors (Hanley & McNeil, 1982). A perfect predictor will have an area of 1, the larger the area, the better the predictor. The four metapredictors have the highest AUC values (Figure 12, Table 3) and the best predictor is RF-Classifier (0.887), followed by RF-Regressor (0.821).

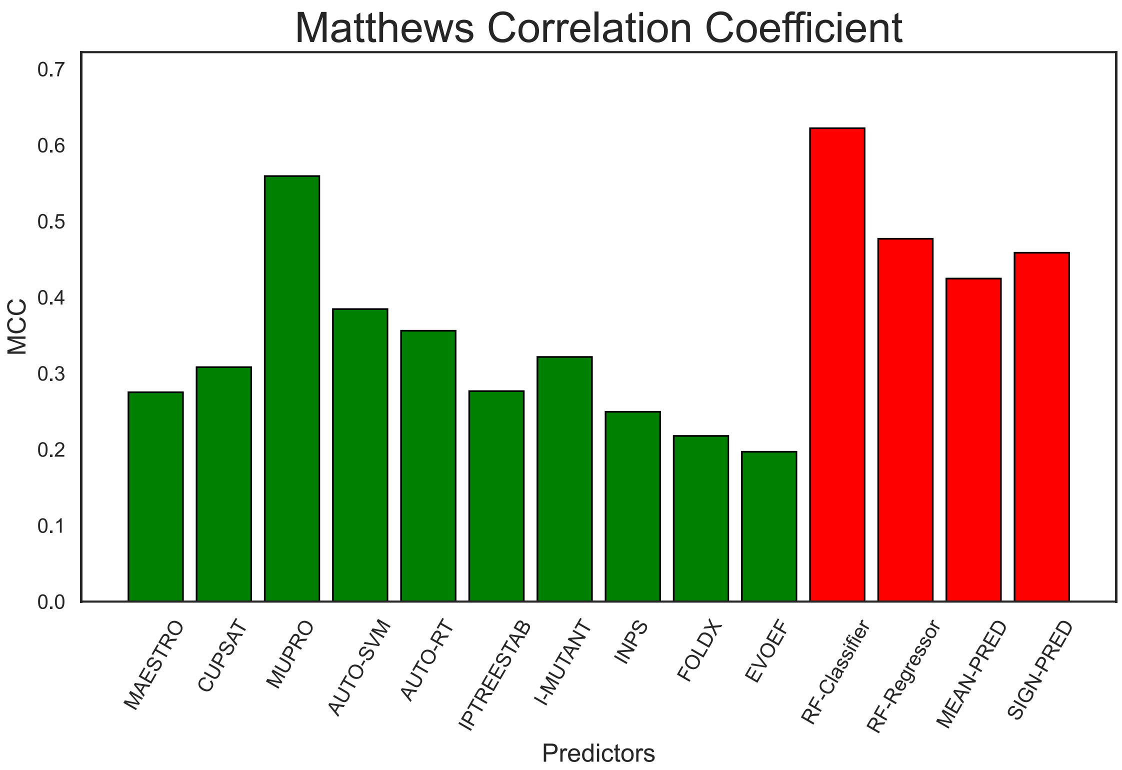


Figure 13. Representation of the Matthews Correlation Coefficient of each predictor.

Finally, we consider the Matthews Correlation Coefficient (MCC). The higher MCC the better. The lowest values for FoldX and EvoEF indicate that the predicted stability and the true states are weakly correlated. The best predictor is RF-Classifier, followed by MUPRO and the rest of consensus approaches (Figure 13).

Table 3. Resume comparison table of predictors and meta-predictors. The best result for each metric is highlight in green whereas the worst is coloured red.

Predictor	Sensitivity	Specificity	Precision	Accuracy	MCC	AUC
MAESTRO	0.497	0.768	0.667	0.637	0.275	0.688
CUPSAT	0.531	0.768	0.681	0.653	0.308	0.707
MUPRO	0.531	0.968	0.939	0.757	0.559	0.760
AUTO-SVM	0.297	0.981	0.935	0.650	0.384	0.675
AUTO-RT	0.393	0.910	0.803	0.660	0.356	0.769
IPTREESTAB	0.310	0.910	0.763	0.620	0.277	0.574
I-MUTANT	0.303	0.942	0.830	0.633	0.322	0.766
INPS	0.366	0.852	0.697	0.617	0.249	0.668
FOLDX	0.517	0.697	0.615	0.610	0.218	0.662
EVOEF	0.697	0.497	0.564	0.593	0.197	0.646
RF-Classifier	0.745	0.871	0.844	0.810	0.622	0.887
RF-Regressor	0.779	0.697	0.706	0.737	0.477	0.821
MEAN-PRED	0.469	0.916	0.840	0.700	0.433	0.782
SIGN-PRED	0.517	0.903	0.833	0.717	0.459	0.812

4.4. Selection of most stabilizing mutations

The final objective is selecting the mutations more stabilizing given a protein sequence, so we are going to do a test to see with each predictor which percentage of the most stabilizing mutations will be selected. In this way, if we must select a fixed number of mutations as the most stabilizing, we will take the first ones in this ranking.

Since there is not available databases with the estimated ΔΔG values for all possible mutations in a protein, we alternatively propose to compare how well each predictor would order mutations from its lowest ΔΔG to the highest, even if they come from different proteins.

If we sort the list of 300 testing data mutations from the lowest ΔΔG to the highest real ΔΔG, we compare which predictor would have better performance sorting that list so its 100 top mutations were the largest possible proportion of the true top 100 values.

By pure chance, if we choose m mutations of all of them by random in an sorted list of n elements, approximately a quantity given by m*m/n (that is, a proportion of m/n of them) will result from the first m. Therefore, a predictor makes a good selection of mutations if an amount above that proportion results in the ranking of the first m. We will call that amount above “improved selection”. For instance, if we select the top 100 mutations from 300, we need to exceed a minimum of 30 by random guess and it is an evidence that the predictor is sorting this particular ranking well. This can be done for each predictor and compared with the ranking provided by our RF-Regressor and MEAN-PRED predictors.

On the other hand, we can study the way to combine the information on classification and the estimation of ΔΔG values to improve our selection ranking. Two strategies are proposed: the first, weighting the ΔΔG value estimated by RF-Regressor multiplying it by the value returned by RF-Classifier, which represents the proportion of trees in the Random Forest that determine the stabilizing character of a mutation. In this way, mutations with a high proportion of stabilizing predictor trees will gain positions in the ranking of the best mutations compared to other mutations with a low proportion or consensus on their stabilizing character. We will call this way of establishing the ranking "RFC * RFR". A second strategy consists of forming ordered pairs of values given by (RF-Classifier, RF-Regressor) and ordering the ranking by such ordered pairs. In this way we give priority to the proportion of stabilizers and, in the event of a tie, it is ordered by the lowest estimated value of ΔΔG. We will name this way of establishing the ranking as "(RFC, RFR)".

Once again, we will use the test data to compare the different rankings of the most stabilizing mutations (lower ΔΔG). To compare them with each other, we will take the ranking of the 100 best mutations according to experimental data and we will see what percentage of them fall within the ranking of each predictor used. We will finally choose the predictor that provides the best percentage (Figure 14A).

When doing so, the combination of both RF algorithms turned out to be the best solution taking advantage of the performances of the two metapredictors. Our metapredictor is better than all of them, and the basic fundamental is that we are combining tools that are quite good separately, but in combination, the percentages of accuracy are multiplied and we take the best of each one (Figure 14B).

It can be observed with the test data that the predictor that manages to introduce the majority of mutations in the top group through its particular ranking depends on the top group. It is likely that if we prioritize the guarantee that a mutation is stabilizing against the higher or lower value of its ΔΔG, the best predictor turns out to be the (RFC, RFR), while if our priority is to achieve very low values of the ΔΔG, then will be better predictors RFR or RFC * RFR.

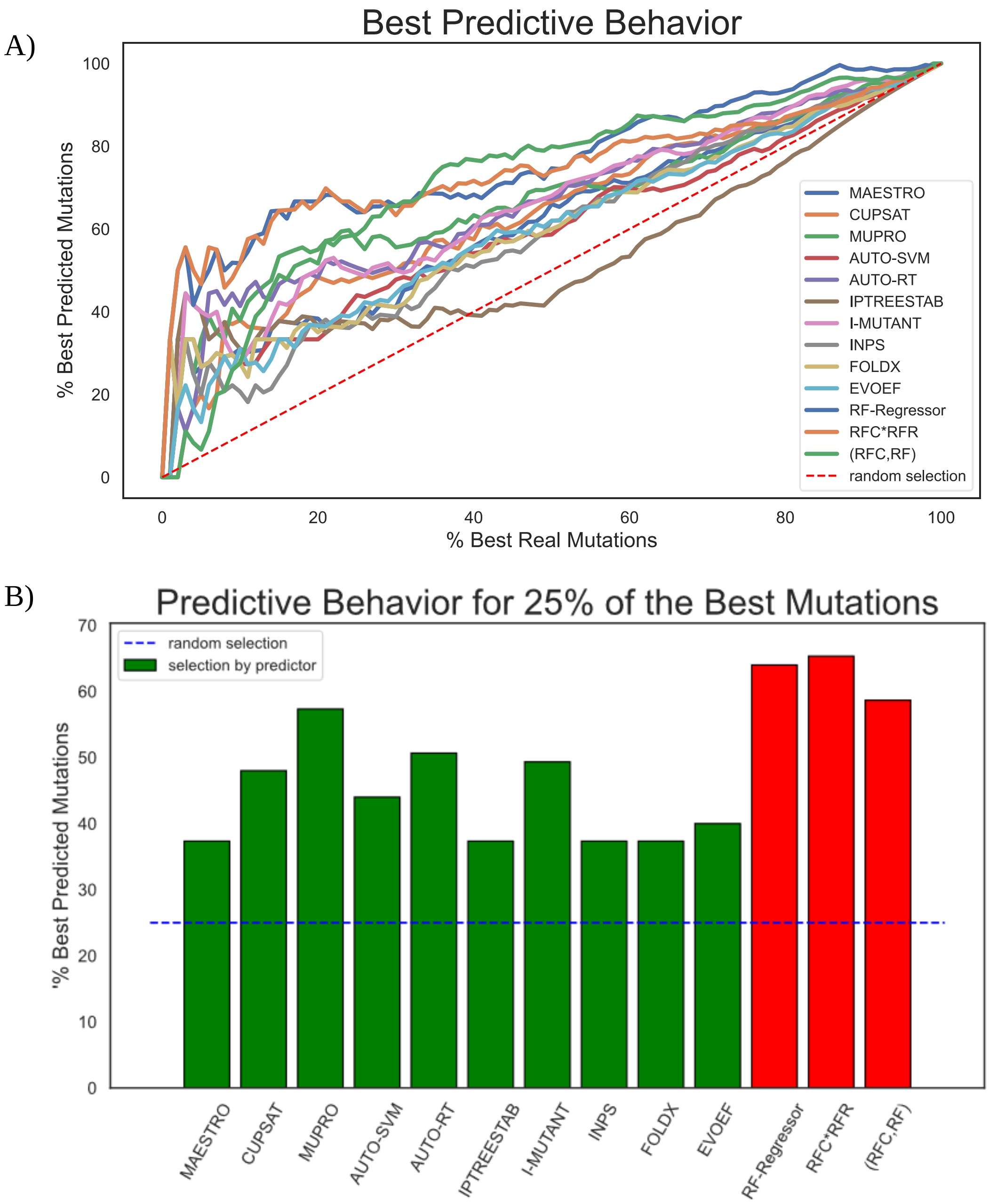


Figure 14. A) Representation of the best predictive behavior. Each predictor is represented by a different color. The red dotted line diagonal represents random guess performances. On the x-axis we show the size of the top group (in percentage of the size of the test group) and on the y-axis we show the percentage of mutations that have really entered the true top group. B) Representation of the predictive behavior for 25 % of the best mutations. Blue dotted line represent random selection, individual tools are colored green and metapredictors in red.

4.5. Studies with real-case data

The final objective of the metapredictor is to locate those mutations that allow obtaining thermostable proteins of biotechnological interest from their structure. In order to test the procedure that would be followed with a prospective study, we have searched in the recent publications and other studies that have been carried out following the FRESCO protocol which first combines two biopredictors (Rosetta and FoldX) to select the most stabilizing mutations. After the process of selection, they applied a molecular filter, visual inspection and finally experimental verification where they recorded the change in mean temperature of denaturalization obtained for each mutant. The three retrospective studies that were selected are: an alcohol dehydrogenase, a transaminase, and a hydrolase (Table 2). From these studies, a list of mutations as well as the ΔΔGs that the authors predicted with the FRESCO protocol and experimentally tested values were extracted. This data can be used to check whether with our approach, we would have still selected those mutations that experimentally have been shown to be stabilizers.

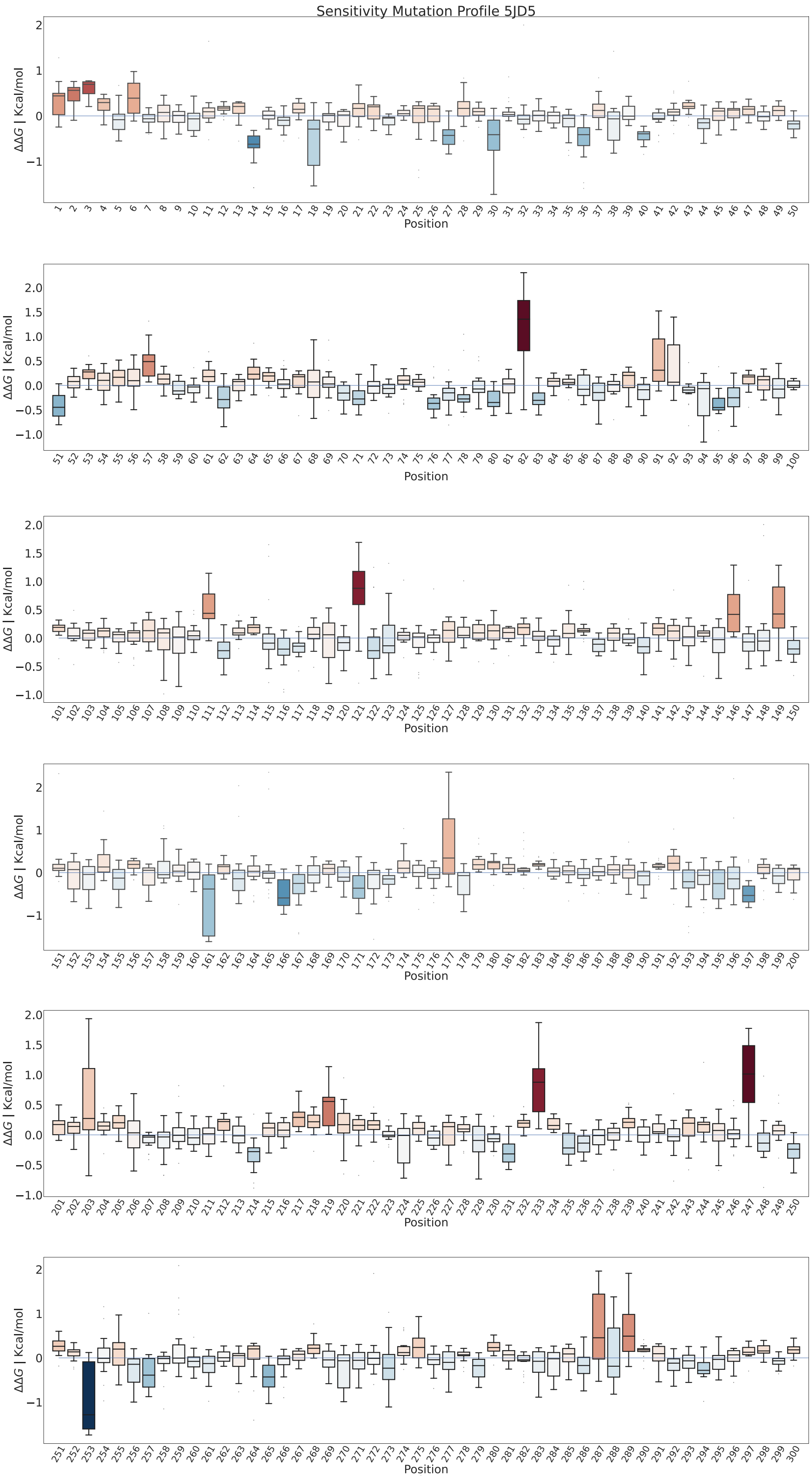
All the ΔΔG values estimated by the 10 predictors were calculated, as described in section of Materials and Methods, and the RFR * RFC model was applied, which values can be seen for a quick inspection in the sensitivity profile of mutations for the prospective (Figure 15) and retrospectives (Supplementary Figure 1). Besides, we provide the first 150 mutation most stabilizing in Supplementary Data 1. Our metapredictor uses the predictions made by FoldX as one of the descriptors, so we will also take this into account when estimating the results to compare the predictive power of both methods.

For 6TQ5, in the FRESCO protocol (FoldX + Rosetta) there were 21 stabilizing mutations from a total of 177 mutations (see Table 2), although 11 of those were initially ruled out because they inactivated the enzyme. From those 21, FoldX without Rosetta predicts 17 are stabilizers, while our metapredictor RFR * RFC predicts only 7 are stabilizers.

Then, 1NWW reported 47 stabilizing mutations from 268 mutations that were predicted with FRESCO (see Table 2), and FoldX without Rosetta predicts 32 of them are stabilizers, while RFC * RFR predicts 18. Finally, 6G4B published 31 stabilizing mutations from 204 that were tested with FRESCO (see Table 2), and FoldX only predicts 15 of them are stabilizers, while RFC * RFR predicts 18.

From the mutations selected by FRESCO, which are the result from following other steps such as visually filtering as well as experimental verification, it seems that our metapredictor does not consider most of them to be stabilizers. Since our metapredictor seems to be more accurate, we cannot jump to conclusions because it may detect other highly stabilizing point mutations that may have gone unnoticed by FRESCO. Therefore, it will be necessary to wait in order to carry out a prospective study with the 5JD5 protein and its experimental validation to study the percentage of stabilizing mutations successfully detected.

Figure 15. Sensitive Mutation Profile of the protein MGS-MilE3 (5JD5). Representation of all ΔΔG values calculated with the RFC * RFR model for each position. The colour gradually varies from dark red for positions with a more destabilizing estimated mean value, with a positive ΔΔG value, to navy blue for positions with a more stabilizing estimated mean value. A horizontal blue line indicates the neutral values that do not modify the protein stability. The most interesting mutations are negative outliers and blue-coloured positions.



5. CONCLUSION AND FUTURE PROSPECTS

The design of thermostable enzymes is of great interest for industries and biomedical research. As we have seen, there are many predictors whose purpose is to screen all possible mutations that a protein may undergo in order to reduce production time and experimental costs. The idea of using metapredictors, which combine the estimates of different individual tools to produce a more reliable result, has been spreading for some years. We have developed two metapredictors under the Machine Learning paradigm: Random Forest Classifier and Random Forest Regressor metapredictors. Each of them combines a total of 10 already published protein stability predictors. Moreover, we proposed two ways of organizing them to increase their predictive power: sorting them by the result of multiplying their outputs or by the vector that contains both outputs.

After the study of the performance of all the predictors throughout the previously mentioned tests, the following conclusions can be obtained:

- In all cases, the RF-Classifier has behaved with greater accuracy than the rest.
- There are predictors that contribute very little to the accuracy of our RF-Classifier. The computational cost would have to be weighed against the loss of accuracy. Candidate predictors to be suppressed from RF are those that are very far from the optimum accuracy (point (0,1) in the ROC space) or those for which the FPR is very large since they provide too many false positives, which can negatively affect the choice of the ranking of the most stabilizing mutations (rightmost points in the ROC space). Following this criteria, we could delete the "EVOEF" predictor.
- From now on, we will stop using the "MEAN-PRED" and "SIGN-PRED" based strategies, which have been clearly outperformed in accuracy by RF-Regressor and RF-Classifier, respectively.

When making a ranking of the most stabilizers, we will use the "RFC * RFR" predictor if the percentage of the most stabilizing of the total size is less than ~30% and if it is not, we will use the "(RFC, RFR)". Although we cannot draw decisive conclusions from retrospective studies, we are going to send an experimental evaluation of the result of the predictions in order to verify the real efficacy of our metapredictor.

After experimental validation, the following steps will include cluster and accumulation of the effective stabilizing mutations to produce multiple point stabilizing mutations. To cluster mutations we could use a clustering algorithm like k-means algorithm, which is very powerful in data mining problems (Jain, 2010). Our goal is decreasing the risk of combining mutations with antagonistic effects and independence of specific structure. Finally, we should experimentally validate the result of the prospective study with 5JD5.

6. BIBLIOGRAPHY

Aalbers, F. S., Fürst, M. J. L. J., Rovida, S., Trajkovic, M., Rubén Gómez Castellanos, J., Bartsch, S., … Fraaije, M. W. (2020). Approaching boiling point stability of an alcohol dehydrogenase through computationally-guided enzyme engineering. ELife, 9. https://doi.org/10.7554/eLife.54639

Awad, M., & Khanna, R. (2015). Week 3 - ML. Efficient Learning Machines, 1–18. https://doi.org/10.1007/978-1-4302-5990-9_1

Bash, P. A., Singh, U. C., Langridge, R., & Kollman, P. A. (1987). Free energy calculations by computer simulation. Science, 236(4801), 564–568. https://doi.org/10.1126/science.3576184

Bava, K. A., Gromiha, M. M., Uedaira, H., Kitajima, K., & Sarai, A. (2004). ProTherm, version 4.0: Thermodynamic database for proteins and mutants. Nucleic Acids Research, 32(DATABASE ISS.), D120. https://doi.org/10.1093/nar/gkh082

Breiman, L. (2001). Random forests. Machine Learning, 45(1), 5–32. https://doi.org/10.1023/A:1010933404324

Broom, A., Jacobi, Z., Trainor, K., & Meiering, E. M. (2017). Computational tools help improve protein stability but with a solubility tradeoff. Journal of Biological Chemistry, 292(35), 14349–14361. https://doi.org/10.1074/jbc.M117.784165

Buß, O., Rudat, J., & Ochsenreither, K. (2018). FoldX as Protein Engineering Tool: Better Than Random Based Approaches? Computational and Structural Biotechnology Journal, 16, 25–33. https://doi.org/10.1016/j.csbj.2018.01.002

Capriotti, E., Fariselli, P., Calabrese, R., & Casadio, R. (2005). Predicting protein stability changes from sequences using support vector machines. Bioinformatics, 21(SUPPL. 2). https://doi.org/10.1093/bioinformatics/bti1109

Capriotti, E., Fariselli, P., Rossi, I., & Casadio, R. (2008). A three-state prediction of single point mutations on protein stability changes. BMC Bioinformatics, 9(SUPPL. 2), S6. https://doi.org/10.1186/1471-2105-9-S2-S6

Chakravorty, D., Khan, M. F., & Patra, S. (2017). Multifactorial level of extremostability of proteins: can they be exploited for protein engineering? Extremophiles, 21(3), 419–444. https://doi.org/10.1007/s00792-016-0908-9

Chen, C. W., Lin, M. H., Liao, C. C., Chang, H. P., & Chu, Y. W. (2020). iStable 2.0: Predicting protein thermal stability changes by integrating various characteristic modules. Computational and Structural Biotechnology Journal, 18, 622–630. https://doi.org/10.1016/j.csbj.2020.02.021

Cheng, J., Randall, A., & Baldi, P. (2006). Prediction of protein stability changes for single-site mutations using support vector machines. Proteins: Structure, Function and Genetics, 62(4), 1125–1132. https://doi.org/10.1002/prot.20810

Cortes, C., & Vapnik, V. (1995). Support-Vector Networks. Machine Learning, 20(3), 273–297. https://doi.org/10.1023/A:1022627411411

Cossio, P., Granata, D., Laio, A., Seno, F., & Trovato, A. (2012). A simple and efficient statistical potential for scoring ensembles of protein structures. Scientific Reports, 2(1), 1–8. https://doi.org/10.1038/srep00351

Deng, Z., Yang, H., Li, J., Shin, H. D., Du, G., Liu, L., & Chen, J. (2014). Structure-based engineering of alkaline α-amylase from alkaliphilic Alkalimonas amylolytica for improved thermostability. Applied Microbiology and Biotechnology, 98(9), 3997–4007. https://doi.org/10.1007/s00253-013-5375-y

Dill, K. A. (1990). Dominant Forces in Protein Folding. Biochemistry, 29(31), 7133–7155. https://doi.org/10.1021/bi00483a001

Emily, M., Talvas, A., & Delamarche, C. (2013). MetAmyl: A METa-Predictor for AMYLoid Proteins. PLoS ONE, 8(11), e79722. https://doi.org/10.1371/journal.pone.0079722

Fang, J. (2020). A critical review of five machine learning-based algorithms for predicting protein stability changes upon mutation. Briefings in Bioinformatics, 21(4), 1285–1292. https://doi.org/10.1093/bib/bbz071

Farhoodi, R., Haspel, N., Shelbourne, M., Hutchinson, B., Hsieh, R., & Jagodzinski, F. (2017). Predicting the effect of point mutations on protein structural stability. ACM-BCB 2017 - Proceedings of the 8th ACM International Conference on Bioinformatics, Computational Biology, and Health Informatics, 17, 247–252. https://doi.org/10.1145/3107411.3107492

Fariselli, P., Martelli, P. L., Savojardo, C., & Casadio, R. (2015). INPS: Predicting the impact of non-synonymous variations on protein stability from sequence. Bioinformatics, 31(17), 2816–2821. https://doi.org/10.1093/bioinformatics/btv291

Floor, R. J., Wijma, H. J., Colpa, D. I., Ramos-Silva, A., Jekel, P. A., Szymański, W., … Janssen, D. B. (2014). Computational library design for increasing haloalkane dehalogenase stability. ChemBioChem, 15(11), 1660–1672. https://doi.org/10.1002/cbic.201402128

Frappier, V., Chartier, M., & Najmanovich, R. J. (2015). ENCoM server: Exploring protein conformational space and the effect of mutations on protein function and stability. Nucleic Acids Research, 43(W1), W395–W400. https://doi.org/10.1093/nar/gkv343

Gilis, D., & Rooman, M. (1996). Stability changes upon mutation of solvent-accessible residues in proteins evaluated by database-derived potentials. Journal of Molecular Biology, 257(5), 1112–1126. https://doi.org/10.1006/jmbi.1996.0226

Gilis, D., & Rooman, M. (1997). Predicting protein stability changes upon mutation using database-derived potentials: Solvent accessibility determines the importance of local versus non-local interactions along the sequence. Journal of Molecular Biology, 272(2), 276–290. https://doi.org/10.1006/jmbi.1997.1237

Grant, B. J., Rodrigues, A. P. C., ElSawy, K. M., McCammon, J. A., & Caves, L. S. D. (2006). Bio3d: An R package for the comparative analysis of protein structures. Bioinformatics, 22(21), 2695–2696. https://doi.org/10.1093/bioinformatics/btl461

Gromiha, M. M., & Selvaraj, S. (2004, October). Inter-residue interactions in protein folding and stability. Progress in Biophysics and Molecular Biology, Vol. 86, pp. 235–277. https://doi.org/10.1016/j.pbiomolbio.2003.09.003

Goldenzweig, A., & Fleishman, S. J. (2018, June 20). Principles of Protein Stability and Their Application in Computational Design. Annual Review of Biochemistry, Vol. 87, pp. 105–129. https://doi.org/10.1146/annurev-biochem-062917-012102

Guerois, R., Nielsen, J. E., & Serrano, L. (2002). 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. https://doi.org/10.1016/S0022-2836(02)00442-4

Hanley, J. A., & McNeil, B. J. (1982). The meaning and use of the area under a receiver operating characteristic (ROC) curve. Radiology, 143(1), 29–36. https://doi.org/10.1148/radiology.143.1.7063747

Heselpoth, R. D., Yin, Y., Moult, J., & Nelson, D. C. (2015). Increasing the stability of the bacteriophage endolysin PlyC using rationale-based FoldX computational modeling. Protein Engineering, Design and Selection, 28(4), 85–92. https://doi.org/10.1093/protein/gzv004

Huang, L. T., Gromiha, M. M., & Ho, S. Y. (2007). iPTREE-STAB: Interpretable decision tree based method for predicting protein stability changes upon mutations. Bioinformatics, 23(10), 1292–1293. https://doi.org/10.1093/bioinformatics/btm100

Huang, X., Pearce, R., & Zhang, Y. (2020). EvoEF2: Accurate and fast energy function for computational protein design. Bioinformatics, 36(4), 1135–1142. https://doi.org/10.1093/bioinformatics/btz740

Jia, L., Yarlagadda, R., & Reed, C. C. (2015a). Structure based thermostability prediction models for protein single point mutations with machine learning tools. PLoS ONE, 10(9), 1–19. https://doi.org/10.1371/journal.pone.0138022

Jia, L., Yarlagadda, R., & Reed, C. C. (2015b). Structure Based Thermostability Prediction Models for Protein Single Point Mutations with Machine Learning Tools. PLOS ONE, 10(9), e0138022. https://doi.org/10.1371/journal.pone.0138022

Kellogg, E. H., Leaver-Fay, A., & Baker, D. (2011). Role of conformational sampling in computing mutation-induced changes in protein structure and stability. Proteins: Structure, Function and Bioinformatics, 79(3), 830–838. https://doi.org/10.1002/prot.22921

Khan, S., & Vihinen, M. (2010a). Performance of protein stability predictors. Human Mutation, 31(6), 675–684. https://doi.org/10.1002/humu.21242

Khan, S., & Vihinen, M. (2010b). Performance of protein stability predictors. Human Mutation, 31(6), 675–684. https://doi.org/10.1002/humu.21242

Kumar, M. D. S., Bava, K. A., Gromiha, M. M., Prabakaran, P., Kitajima, K., Uedaira, H., & Sarai, A. (2006). ProTherm and ProNIT: thermodynamic databases for proteins and protein-nucleic acid interactions. Nucleic Acids Research, 34(Database issue). https://doi.org/10.1093/nar/gkj103

Laimer, J., Hiebl-Flach, J., Lengauer, D., & Lackner, P. (2016). MAESTROweb: A web server for structure-based protein stability prediction. Bioinformatics, 32(9), 1414–1416. https://doi.org/10.1093/bioinformatics/btv769

Laimer, J., Hofer, H., Fritz, M., Wegenkittl, S., & Lackner, P. (2015). MAESTRO - multi agent stability prediction upon point mutations. BMC Bioinformatics, 16(1). https://doi.org/10.1186/s12859-015-0548-6

Larsen, D. M., Nyffenegger, C., Swiniarska, M. M., Thygesen, A., Strube, M. L., Meyer, A. S., & Mikkelsen, J. D. (2015). Thermostability enhancement of an endo-1,4-β-galactanase from Talaromyces stipitatus by site-directed mutagenesis. Applied Microbiology and Biotechnology, 99(10), 4245–4253. https://doi.org/10.1007/s00253-014-6244-z

Lehmann, M., Pasamontes, L., Lassen, S. F., & Wyss, M. (2000, December 29). The consensus concept for thermostability engineering of proteins. Biochimica et Biophysica Acta - Protein Structure and Molecular Enzymology, Vol. 1543, pp. 408–415. https://doi.org/10.1016/S0167-4838(00)00238-7

Masso, M., & Vaisman, I. I. (2010). AUTO-MUTE: Web-based tools for predicting stability changes in proteins due to single amino acid replacements. Protein Engineering, Design and Selection, 23(8), 683–687. https://doi.org/10.1093/protein/gzq042

Meng, Q., Capra, N., Palacio, C. M., Lanfranchi, E., Otzen, M., Van Schie, L. Z., … Janssen, D. B. (2020). Robust ω-Transaminases by Computational Stabilization of the Subunit Interface. ACS Catalysis, 10(5), 2915–2928. https://doi.org/10.1021/acscatal.9b05223

Metz, C. E. (1978). Basic principles of ROC analysis. Seminars in Nuclear Medicine, 8(4), 283–298. https://doi.org/10.1016/S0001-2998(78)80014-2

Montanucci, L., Capriotti, E., Frank, Y., Ben-Tal, N., & Fariselli, P. (2019). ΔΔGun: An untrained method for the prediction of protein stability changes upon single and multiple point variations. BMC Bioinformatics, 20(S14), 335. https://doi.org/10.1186/s12859-019-2923-1

Montanucci, L., Capriotti, E., Frank, Y., Ben-Tal, N., & Fariselli, P. (2019). DDGun: An untrained method for the prediction of protein stability changes upon single and multiple point variations. BMC Bioinformatics, 20(S14), 335. https://doi.org/10.1186/s12859-019-2923-1

Musil, M., Stourac, J., Bendl, J., Brezovsky, J., Prokop, Z., Zendulka, J., … Damborsky, J. (2017). FireProt: web server for automated design of thermostable proteins. Nucleic Acids Research, 45(W1), W393–W399. https://doi.org/10.1093/nar/gkx285

Ó'Fágáin, C. (2017). Protein stability: Enhancement and measurement. In Methods in Molecular Biology (Vol. 1485, pp. 101–129). https://doi.org/10.1007/978-1-4939-6412-3_7

Pandurangan, A. P., Ochoa-Montaño, B., Ascher, D. B., & Blundell, T. L. (2017). SDM: A server for predicting effects of mutations on protein stability. Nucleic Acids Research, 45(W1), W229–W235. https://doi.org/10.1093/nar/gkx439

Panigrahi, P., Sule, M., Ghanate, A., Ramasamy, S., & Suresh, C. G. (2015). Engineering proteins for thermostability with iRDP web server. PLoS ONE, 10(10). https://doi.org/10.1371/journal.pone.0139486

Parthiban, V., Gromiha, M. M., & Schomburg, D. (2006). CUPSAT: Prediction of protein stability upon point mutations. Nucleic Acids Research, 34(WEB. SERV. ISS.), W239–W242. https://doi.org/10.1093/nar/gkl190

Parthiban, V., Gromiha, M. M., Abhinandan, M., & Schomburg, D. (2007). Computational modeling of protein mutant stability: Analysis and optimization of statistical potentials and structural features reveal insights into prediction model development. BMC Structural Biology, 7, 54. https://doi.org/10.1186/1472-6807-7-54

Pires, D. E. V., Ascher, D. B., & Blundell, T. L. (2014a). DUET: A server for predicting effects of mutations on protein stability using an integrated computational approach. Nucleic Acids Research, 42(W1), W314. https://doi.org/10.1093/nar/gku411

Pires, D. E. V., Ascher, D. B., & Blundell, T. L. (2014b). MCSM: Predicting the effects of mutations in proteins using graph-based signatures. Bioinformatics, 30(3), 335–342. https://doi.org/10.1093/bioinformatics/btt691

Pitera, J. W., & Kollman, P. A. (2000). Exhaustive mutagenesis in silico: Multicoordinate free energy calculations on proteins and peptides. Proteins: Structure, Function and Genetics, 41(3), 385–397. https://doi.org/10.1002/1097-0134(20001115)41:3<385::AID-PROT100>3.0.CO;2-R

Ponnuswamy, P. K. (1993). Hydrophobic characteristics of folded proteins. Progress in Biophysics and Molecular Biology, Vol. 59, pp. 57–103. https://doi.org/10.1016/0079-6107(93)90007-7

Ponnuswamy, P. K., & Michael Gromiha, M. (1994). On the conformational stability of folded proteins. Journal of Theoretical Biology, 166(1), 63–74. https://doi.org/10.1006/jtbi.1994.1005

Potapov, V., Cohen, M., & Schreiber, G. (2009). Assessing computational methods for predicting protein stability upon mutation: Good on average but not in the details. Protein Engineering, Design and Selection, 22(9), 553–560. https://doi.org/10.1093/protein/gzp030

Prevost, M., Wodak, S. J., Tidor, B., & Karplus, M. (1991). Contribution of the hydrophobic effect to protein stability: Analysis based on simulations of the Ile-96 → Ala mutation in barnase. Proceedings of the National Academy of Sciences of the United States of America, 88(23), 10880–10884. https://doi.org/10.1073/pnas.88.23.10880

Pucci, F., Bernaerts, K. V., Kwasigroch, J. M., & Rooman, M. (2018). Quantification of biases in predictions of protein stability changes upon mutations. Bioinformatics, 34(21), 3659–3665. https://doi.org/10.1093/bioinformatics/bty348

Radford, A., Metz, L., & Chintala, S. (2016). DCGAN original paper 2016. 1–16. Retrieved from https://arxiv.org/pdf/1511.06434.pdf

Rodrigues, C. H. M., Pires, D. E. V., & Ascher, D. B. (2018). DynaMut: Predicting the impact of mutations on protein conformation, flexibility and stability. Nucleic Acids Research, 46(W1), W350–W355. https://doi.org/10.1093/nar/gky300

Sanavia, T., Birolo, G., Montanucci, L., Turina, P., Capriotti, E., & Fariselli, P. (2020, January 1). Limitations and challenges in protein stability prediction upon genome variations: towards future applications in precision medicine. Computational and Structural Biotechnology Journal, Vol. 18, pp. 1968–1979. https://doi.org/10.1016/j.csbj.2020.07.011

Sanavia, T., Birolo, G., Montanucci, L., Turina, P., Capriotti, E., & Fariselli, P. (2020, January 1). Limitations and challenges in protein stability prediction upon genome variations: towards future applications in precision medicine. Computational and Structural Biotechnology Journal, Vol. 18, pp. 1968–1979. https://doi.org/10.1016/j.csbj.2020.07.011

Sasidharan Nair, P., & Vihinen, M. (2013). VariBench: A Benchmark Database for Variations. Human Mutation, 34(1), 42–49. https://doi.org/10.1002/humu.22204

Savojardo, C., Fariselli, P., Martelli, P. L., & Casadio, R. (2016). INPS-MD: A web server to predict stability of protein variants from sequence and structure. Bioinformatics, 32(16), 2542–2544. https://doi.org/10.1093/bioinformatics/btw192

Schymkowitz, J., Borg, J., Stricher, F., Nys, R., Rousseau, F., & Serrano, L. (2005). The FoldX web server: An online force field. Nucleic Acids Research, 33(SUPPL. 2). https://doi.org/10.1093/nar/gki387

Shen, M., & Sali, A. (2006). Statistical potential for assessment and prediction of protein structures. Protein Science, 15(11), 2507–2524. https://doi.org/10.1110/ps.062416606

Socha, R. D., & Tokuriki, N. (2013). Modulating protein stability - directed evolution strategies for improved protein function. FEBS Journal, 280(22), 5582–5595. https://doi.org/10.1111/febs.12354

Song, X., Wang, Y., Shu, Z., Hong, J., Li, T., & Yao, L. (2013). Engineering a More Thermostable Blue Light Photo Receptor Bacillus subtilis YtvA LOV Domain by a Computer Aided Rational Design Method. PLoS Computational Biology, 9(7), e1003129. https://doi.org/10.1371/journal.pcbi.1003129

Steipe, B., Schiller, B., Pluäckthun, A., & Steinbacher, S. (1994, July 14). Sequence statistics reliably predict stabilizing mutations in a protein domain. Journal of Molecular Biology, Vol. 240, pp. 188–192. https://doi.org/10.1006/jmbi.1994.1434

Stogios, P. J., Xu, X., Cui, H., Martinez-Martinez, M., Chernikova, T. N., Golyshin, P. N., … Savchenko, A. (2016). Crystal structure of MGS-MilE3, an alpha/beta hydrolase enzyme from the metagenome of pyrene-phenanthrene enrichment culture with sediment sample of Milazzo Harbor, Italy. https://doi.org/10.2210/pdb5jd5/pdb

Trevethan, R. (2017). Sensitivity, Specificity, and Predictive Values: Foundations, Pliabilities, and Pitfalls in Research and Practice. Frontiers in Public Health, 5, 307. https://doi.org/10.3389/fpubh.2017.00307

Wan, J., Kang, S., Tang, C., Yan, J., Ren, Y., Liu, J., … Li, T. (2008). Meta-prediction of phosphorylation sites with weighted voting and restricted grid search parameter selection. Nucleic Acids Research, 36(4), e22. https://doi.org/10.1093/nar/gkm848

Wijma, H. J., Floor, R. J., Jekel, P. A., Baker, D., Marrink, S. J., & Janssen, D. B. (2014). Computationally designed libraries for rapid enzyme stabilization. Protein Engineering, Design and Selection, 27(2), 49–58. https://doi.org/10.1093/protein/gzt061

Wong, C. F., & McCammon, J. A. (1987). Thermodynamics of Enzyme Folding and Activity: Theory and Experiment. https://doi.org/10.1007/978-3-642-71705-5_12

Zhou, H., & Zhou, Y. (2004). Quantifying the Effect of Burial of Amino Acid Residues on Protein Stability. Proteins: Structure, Function and Genetics, 54(2), 315–322. https://doi.org/10.1002/prot.10584

Zhou, X., & Cheng, J. (2016). DNpro: A Deep Learning Network Approach to Predicting Protein Stability Changes Induced by Single-Site Mutations. Retrieved from https://pdfs.semanticscholar.org/69fa/0b70fc0ca2a6826f70a6fb306bec84d97201.pdf

7. SUPPLEMENTARY DATA

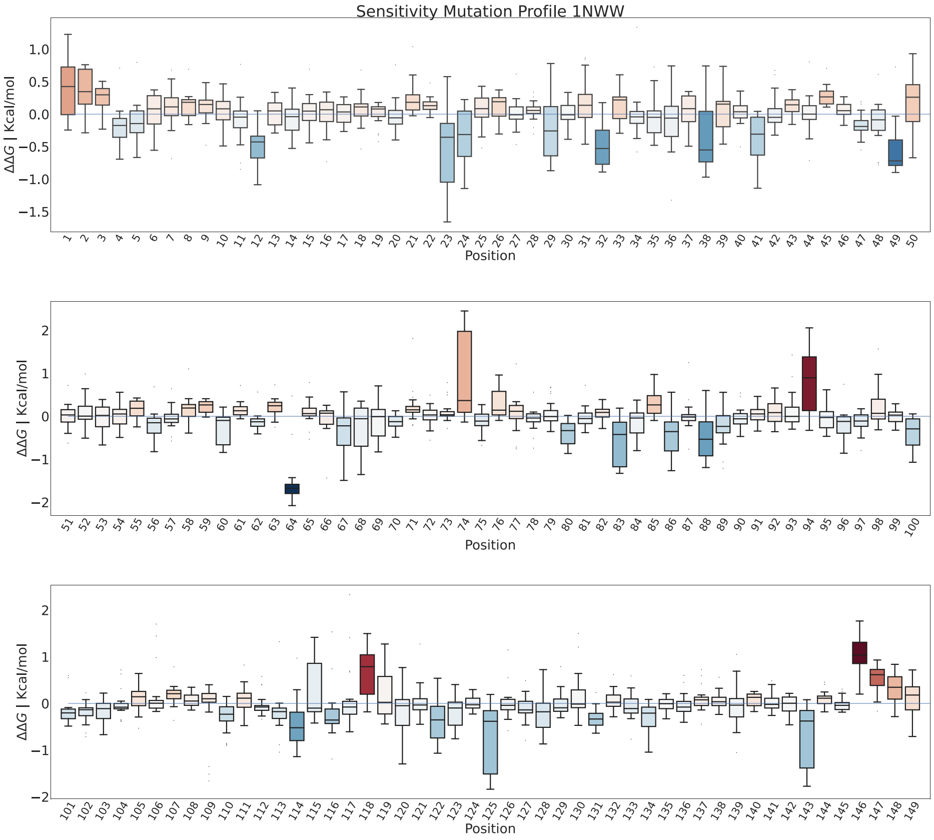
Supplementary table 1. Already published protein stability predictors.

Predictor	Description	Published (year)
AUTOMUTE	Automated server for predicting functional consequences of amino acid mutations in proteins. http://binf.gmu.edu/automute/AUTO-MUTE_Stability_ΔΔG.html	2010
COREX	Web browser-based predictor that calculates regional stability variations within protein structures http://best.utmb.edu/BEST/	2005
CUPSAT	Predicts changes in protein stability upon point mutations. http://cupsat.tu-bs.de/	2006
DUET	A web server for an integrated computational approach to study missense mutations in proteins. http://biosig.unimelb.edu.au/duet/stability	2014
DynaMut	Predicting the impact of mutations on protein conformation, flexibility and stability. http://biosig.unimelb.edu.au/dynamut/prediction	2018
EASE-MM	Sequence-Based Prediction of Mutation-Induced Stability Changes with Feature-Based Multiple Models. http://sparks-lab.org/server/ease	2016
ERIS	Predict impact of mutation, stability. https://dokhlab.med.psu.edu/eris/login.php	2007
EvoEF2	Accurate and fast energy function for computational protein design (de novo sequence design on a given fixed protein backbone (standalone) https://zhanglab.ccmb.med.umich.edu/EvoEF	2020
FoldX 5	Provide quantitative estimation of the importance of the interactions contributing to the stability of proteins and protein complexes. http://foldxsuite.crg.eu/	2005
I-MUTANT 3.0	Predicting stability changes upon mutation from the protein sequence or structure. http://gpcr2.biocomp.unibo.it/cgi/predictors/I-Mutant3.0/I-Mutant3.0.cgi	2005
INPS-MD	The prediction of protein stability changes upon single point variation from protein sequence and/or structure. http://inps.biocomp.unibo.it	2016
IPTREESTAB	Interpretable decision tree based method for predicting protein stability changes upon mutations. http://203.64.84.190:8080/IPTREEr/iptree.htm	2007
iSTABLE	Predicting protein stability changes. Stability change, input PDB or sequence, stability. http://ncblab.nchu.edu.tw/iStable2	2020
MAESTRO	A web server for structure based protein stability prediction. https://pbwww.che.sbg.ac.at/maestro/web	2016

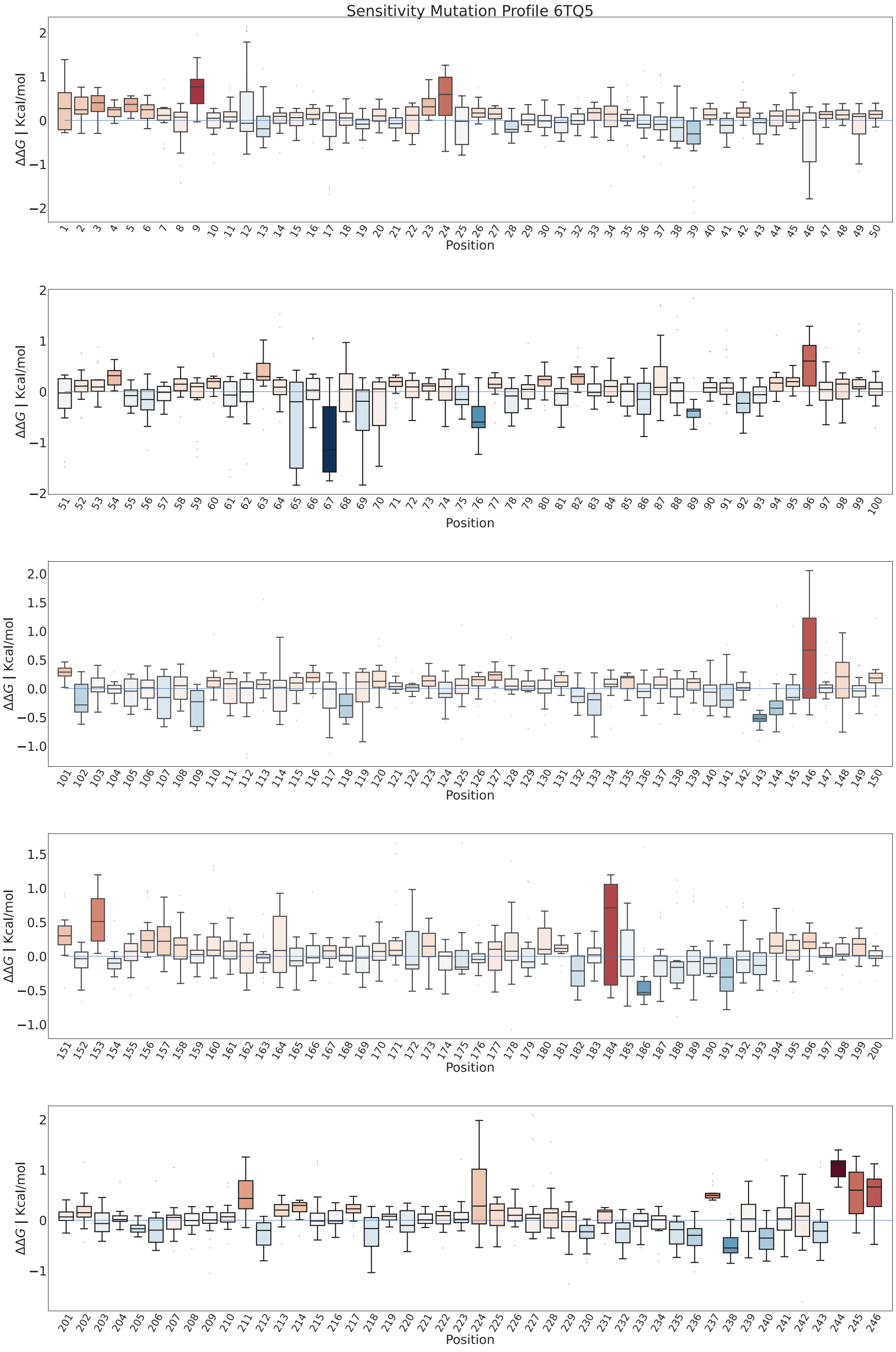
Predictor	Description	Published (year)
mCSM	Effect of mutations on stability, PPI, protein-DNA http://biosig.unimelb.edu.au/mcsm/stability	2014
MUpro	Prediction of Protein Stability Changes for Single-Site Mutations from Sequences. http://mupro.proteomics.ics.uci.edu/	2005
Mutation tool - NeEMO	A method using residue interaction networks to improve prediction of protein stability upon mutation. http://protein.bio.unipd.it/neemo/	2014
POPMUSIC	An algorithm for predicting protein mutant stability changes. https://soft.dezyme.com/query/create/pop	2009
pPerturb	Predicting Long-Distance Energetic Couplings and Mutation-Induced Stability Changes in Proteins via Perturbations. https://pbl.biotech.iitm.ac.in/pPerturb/	2020
PPSC	Prediction of Protein Stability Changes http://structure.bmc.lu.se/PPSC/	2012
PremPS	Predicting the Effects of Mutations on Protein Stability. https://lilab.jysw.suda.edu.cn/research/PremPS/	2020
Pro-Maya	Protein Mutant stability Analyzer http://bental.tau.ac.il/ProMaya	2011
ProTSPoM	Estimating the Effect of Single Point Mutations on Protein Thermodynamic Stability. https://cosmos.iitkgp.ac.in/ProTSPoM/	2020
pStab	Prediction of Stable Mutants. http://pbl.biotech.iitm.ac.in/pStab	2018
SDM2	Site Directed Mutator for predicting stability changes upon mutation. http://marid.bioc.cam.ac.uk/sdm2/	2017
SRide	Identification of Stabilizing Residues in proteins, stability changes. http://sride.enzim.hu	2005
StaRProtein	A Web Server for Prediction of the Stability of Repeat Proteins	2015
STRUM	A method for predicting the fold stability change (delta-delta-G) of protein molecules upon single-point nssnp mutationsv. https://zhanglab.ccmb.med.umich.edu/STRUM/	2016
PROVEAN	Stability prediction for a protein sequence. http://provean.jcvi.org/seq_submit.php	2012
ELASPIC	Predict stability effects of mutations on protein folding and interactions. http://elaspic.kimlab.org/	2016

Supplementary Figure 1. Sensitive Mutation Profile of the retrospective proteins: A) LEH (1NWW), ADHA (6TQ5) and C) ω-TA (6G4B). Representation of all ΔΔG values calculated with the RFC * RFR model for each position. The colour gradually varies from dark red for positions with a more destabilizing estimated mean value, with a positive ΔΔG value, to navy blue for positions with a more stabilizing estimated mean value. A horizontal blue line indicates the neutral values that do not modify the protein stability. The most interesting mutations are negative outliers and blue-coloured positions. Each window shows 50 residues of the total protein sequence.

A)

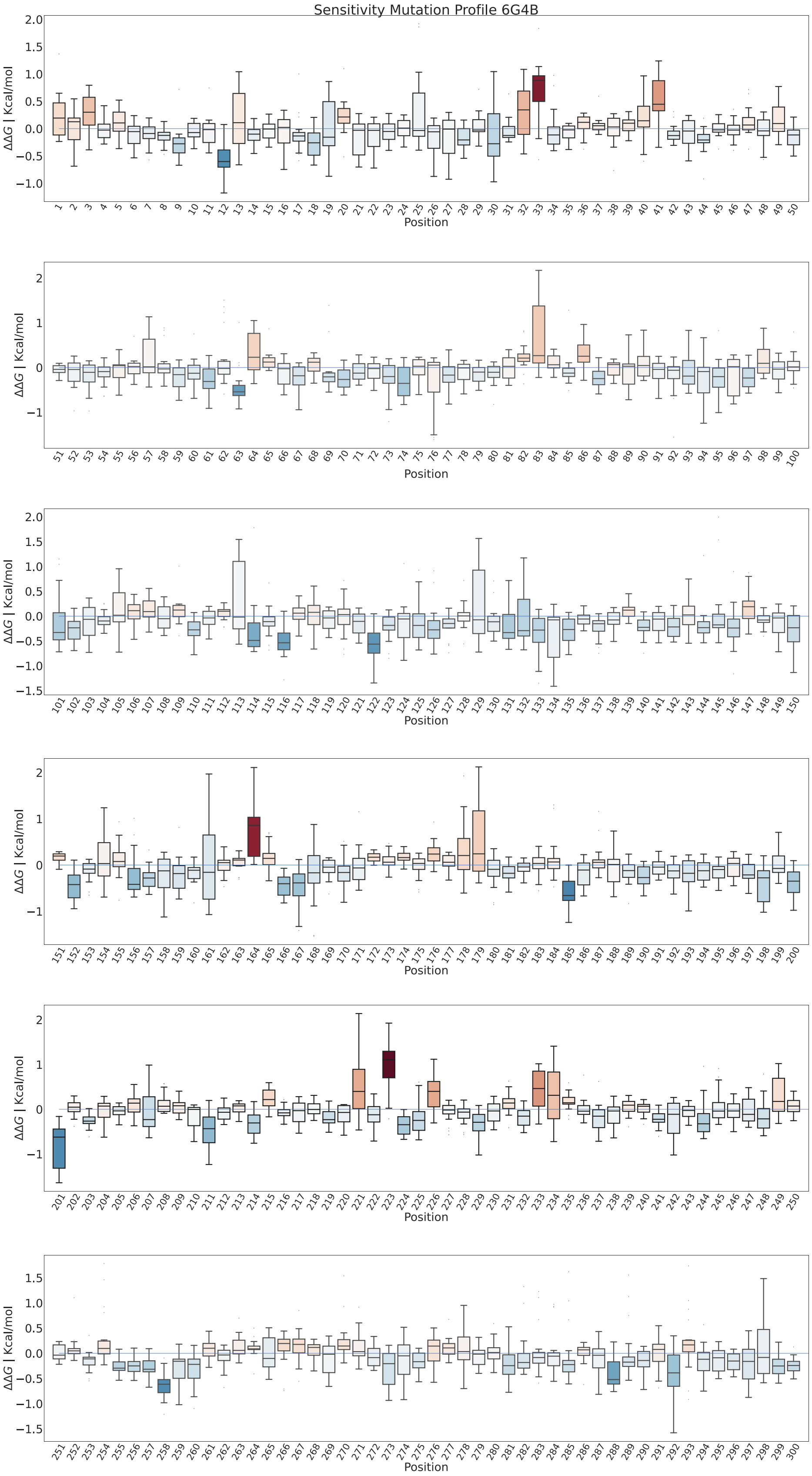


B)

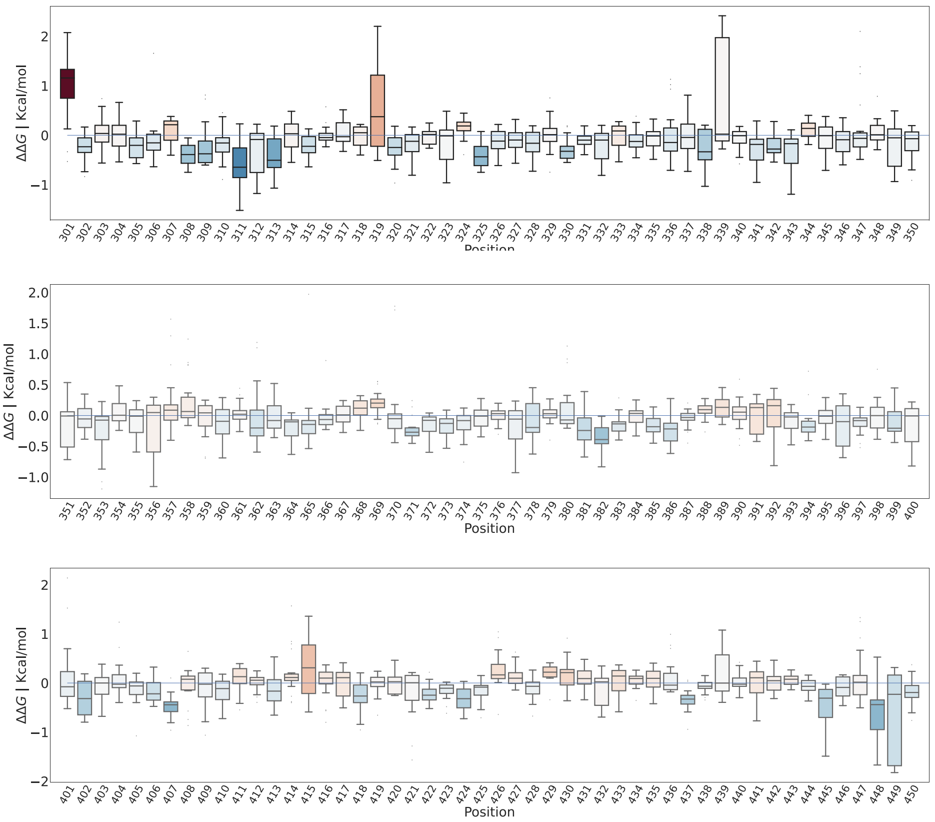


D

C)


Sensitivity Mutation Profile 6G4B
ΔΔG | Kcal/mol
Position
2.0 1.5 1.0 0.5 0.0 −0.5 −1.0
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50
2 1 0 −1
51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100
2.0 1.5 1.0 0.5 0.0 −0.5 −1.0 −1.5
101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128 129 130 131 132 133 134 135 136 137 138 139 140 141 142 143 144 145 146 147 148 149 150
2 1 0 −1
151 152 153 154 155 156 157 158 159 160 161 162 163 164 165 166 167 168 169 170 171 172 173 174 175 176 177 178 179 180 181 182 183 184 185 186 187 188 189 190 191 192 193 194 195 196 197 198 199 200
2 1 0 −1
201 202 203 204 205 206 207 208 209 210 211 212 213 214 215 216 217 218 219 220 221 222 223 224 225 226 227 228 229 230 231 232 233 234 235 236 237 238 239 240 241 242 243 244 245 246 247 248 249 250
1.5 1.0 0.5 0.0 −0.5 −1.0 −1.5
251 252 253 254 255 256 257 258 259 260 261 262 263 264 265 266 267 268 269 270 271 272 273 274 275 276 277 278 279 280 281 282 283 284 285 286 287 288 289 290 291 292 293 294 295 296 297 298 299 300

E


ΔΔG | Kcal/mol
2
1
0
−1
301 302 303 304 305 306 307 308 309 310 311 312 313 314 315 316 317 318 319 320 321 322 323 324 325 326 327 328 329 330 331 332 333 334 335 336 337 338 339 340 341 342 343 344 345 346 347 348 349 350
Position
ΔΔG | Kcal/mol
2.0
1.5
1.0
0.5
0.0
−0.5
−1.0
351 352 353 354 355 356 357 358 359 360 361 362 363 364 365 366 367 368 369 370 371 372 373 374 375 376 377 378 379 380 381 382 383 384 385 386 387 388 389 390 391 392 393 394 395 396 397 398 399 400
Position
ΔΔG | Kcal/mol
2
1
0
−1
−2
401 402 403 404 405 406 407 408 409 410 411 412 413 414 415 416 417 418 419 420 421 422 423 424 425 426 427 428 429 430 431 432 433 434 435 436 437 438 439 440 441 442 443 444 445 446 447 448 449 450
Position

F

Supplementary data 1: part of the final output sorted list of the different proteins calculated with the model RFR*RFC.

A) List of the 150 most stabilizing mutations for ADHA (6TQ5) from a total of 4674 possible mutations:

1) 39I, -2.11 Kcal/mol, 78. %; 2) 65L, -1.70 Kcal/mol, 91. %; 3) 65I, -1.68 Kcal/mol, 89. %; 4) 61I, -1.68 Kcal/mol, 83. %; 5) 17L, -1.53 Kcal/mol, 88. %; 6) 39V, -1.84 Kcal/mol, 73. %; 7) 51L, -1.48 Kcal/mol, 85. %; 8) 61L, -1.52 Kcal/mol, 81. %; 9) 62I, -1.42 Kcal/mol, 85. %; 10) 8V, -1.43 Kcal/mol, 84. %; 11) 65V, -1.71 Kcal/mol, 70. %; 12) 67W, -1.75 Kcal/mol, 67. %; 13) 242I, -1.61 Kcal/mol, 72. %; 14) 46F, -1.78 Kcal/mol, 65. %; 15) 34I, -1.49 Kcal/mol, 77. %; 16) 229F, -1.26 Kcal/mol, 90. %; 17) 46I, -1.61 Kcal/mol, 70. %; 18) 46M, -1.67 Kcal/mol, 67. %; 19) 67F, -1.69 Kcal/mol, 66. %; 20) 69M, -1.66 Kcal/mol, 67. %; 21) 67I, -1.70 Kcal/mol, 65. %; 22) 67A, -1.57 Kcal/mol, 70. %; 23) 17I, -1.69 Kcal/mol, 64. %; 24) 65F, -1.84 Kcal/mol, 57. %; 25) 69L, -1.77 Kcal/mol, 60. %; 26) 39L, -1.51 Kcal/mol, 70. %; 27) 76M, -1.23 Kcal/mol, 86. %; 28) 46L, -1.64 Kcal/mol, 64. %; 29) 67R, -1.61 Kcal/mol, 65. %; 30) 67E, -1.49 Kcal/mol, 68. %; 31) 67M, -1.50 Kcal/mol, 66. %; 32) 17V, -1.58 Kcal/mol, 63. %; 33) 46V, -1.46 Kcal/mol, 68. %; 34) 56V, -1.15 Kcal/mol, 86. %; 35) 67V, -1.61 Kcal/mol, 60. %; 36) 65Y, -1.46 Kcal/mol, 66. %; 37) 76L, -1.20 Kcal/mol, 79. %; 38) 59I, -1.13 Kcal/mol, 83. %; 39) 69F, -1.83 Kcal/mol, 51. %; 40) 65M, -1.62 Kcal/mol, 56. %; 41) 8I, -1.03 Kcal/mol, 88. %; 42) 67Y, -1.46 Kcal/mol, 62. %; 43) 67L, -1.52 Kcal/mol, 59. %; 44) 76R, -1.15 Kcal/mol, 77. %; 45) 59L, -0.98 Kcal/mol, 88. %; 46) 65W, -1.41 Kcal/mol, 61. %; 47) 49V, -1.16 Kcal/mol, 74. %; 48) 69I, -1.58 Kcal/mol, 53. %; 49) 112M, -1.13 Kcal/mol, 71. %; 50) 112I, -1.19 Kcal/mol, 67. %; 51) 117I, -1.12 Kcal/mol, 70. %; 52) 51F, -1.38 Kcal/mol, 56. %; 53) 10I, -0.96 Kcal/mol, 81. %; 54) 70W, -1.46 Kcal/mol, 53. %; 55) 69V, -1.58 Kcal/mol, 47. %; 56) 218L, -1.03 Kcal/mol, 70. %; 57) 49I, -0.98 Kcal/mol, 73. %; 58) 133I, -0.84 Kcal/mol, 84. %; 59) 49L, -0.85 Kcal/mol, 82. %; 60) 70Y, -1.28 Kcal/mol, 53. %; 61) 49F, -0.99 Kcal/mol, 68. %; 62) 8L, -0.74 Kcal/mol, 90. %; 63) 119I, -0.92 Kcal/mol, 72. %; 64) 59V, -1.28 Kcal/mol, 50. %; 65) 10V, -0.76 Kcal/mol, 82. %; 66) 130M, -0.62 Kcal/mol, 98. %; 67) 56L, -0.68 Kcal/mol, 89. %; 68) 62L, -0.63 Kcal/mol, 94. %; 69) 182L, -0.64 Kcal/mol, 92. %; 70) 81I, -0.70 Kcal/mol, 84. %; 71) 212R, -0.80 Kcal/mol, 73. %; 72) 185I, -0.72 Kcal/mol, 80. %; 73) 137L, -0.61 Kcal/mol, 94. %; 74) 109W, -0.69 Kcal/mol, 83. %; 75) 81L, -0.65 Kcal/mol, 87. %; 76) 100F, -0.71 Kcal/mol, 79. %; 77) 182I, -0.64 Kcal/mol, 87. %; 78) 76Q, -0.70 Kcal/mol, 79. %; 79) 232M, -0.76 Kcal/mol, 73. %; 80) 98I, -0.61 Kcal/mol, 90. %; 81) 117L, -0.76 Kcal/mol, 72. %; 82) 239I, -0.74 Kcal/mol, 74. %; 83) 98L, -0.57 Kcal/mol, 95. %; 84) 86I, -0.88 Kcal/mol, 62. %; 85) 56I, -0.62 Kcal/mol, 87. %; 86) 41L, -0.61 Kcal/mol, 89. %; 87) 97M, -0.64 Kcal/mol, 84. %; 88) 76A, -0.72 Kcal/mol, 75. %; 89) 67C, -0.72 Kcal/mol, 75. %; 90) 119L, -0.78 Kcal/mol, 69. %; 91) 243W, -0.79 Kcal/mol, 67. %; 92) 109L, -0.73 Kcal/mol, 73. %; 93) 62M, -0.60 Kcal/mol, 86. %; 94) 236Q, -1.02 Kcal/mol, 51. %; 95) 230L, -0.76 Kcal/mol, 68. %; 96) 67T, -0.82 Kcal/mol, 63. %; 97) 239V, -0.71 Kcal/mol, 72. %; 98) 229L, -0.61 Kcal/mol, 84. %; 99) 109F, -0.69 Kcal/mol, 74. %; 100) 39F, -0.69 Kcal/mol, 74. %; 101) 56F, -0.61 Kcal/mol, 83. %; 102) 17M, -0.66 Kcal/mol, 77. %; 103) 182M, -0.54 Kcal/mol, 93. %; 104) 109E, -0.65 Kcal/mol, 78. %; 105) 76N, -0.69 Kcal/mol, 73. %; 106) 232W, -0.74 Kcal/mol, 68. %; 107) 109M, -0.67 Kcal/mol, 74. %; 108) 70L, -0.60 Kcal/mol, 82. %; 109) 70F, -0.82 Kcal/mol, 60. %; 110) 229V, -0.67 Kcal/mol, 73. %; 111) 86L, -0.77 Kcal/mol, 64. %; 112) 36V, -0.82 Kcal/mol, 59. %; 113) 117F, -0.85 Kcal/mol, 56. %; 114) 56M, -0.51 Kcal/mol, 92. %; 115) 92I, -0.81 Kcal/mol, 57. %; 116) 109K, -0.71 Kcal/mol, 66. %; 117) 230C, -0.84 Kcal/mol, 55. %; 118) 195I, -0.52 Kcal/mol, 88. %; 119) 76Y, -0.62 Kcal/mol, 73. %; 120) 70I, -0.66 Kcal/mol, 68. %; 121) 25I, -0.79 Kcal/mol, 56. %; 122) 212M, -0.61 Kcal/mol, 73. %; 123) 235L, -0.59 Kcal/mol, 75. %; 124) 37T, -1.01 Kcal/mol, 44. %; 125) 63I, -0.76 Kcal/mol, 57. %; 126) 134R, -0.68 Kcal/mol, 64. %; 127) 39M, -0.60 Kcal/mol, 73. %; 128) 58L, -0.48 Kcal/mol, 90. %; 129) 235I, -0.57 Kcal/mol, 75. %; 130) 133V, -0.53 Kcal/mol, 80. %; 131) 154L, -0.47 Kcal/mol, 90. %; 132) 12M, -0.76 Kcal/mol, 56. %; 133) 241L, -0.72 Kcal/mol, 59. %; 134) 126I, -0.45 Kcal/mol, 93. %; 135) 130F, -0.46 Kcal/mol, 92. %; 136) 70M, -0.67 Kcal/mol, 63. %; 137) 25A, -0.74 Kcal/mol, 56. %; 138) 209G, -1.05 Kcal/mol, 40. %; 139) 136L, -0.46 Kcal/mol, 90. %; 140) 72P, -0.56 Kcal/mol, 74. %; 141) 76F, -0.65 Kcal/mol, 64. %; 142) 242L, -0.59 Kcal/mol, 70. %; 143) 152I, -0.49 Kcal/mol, 84. %; 144) 36I, -0.81 Kcal/mol, 51. %; 145) 76H, -0.55 Kcal/mol, 75. %; 146) 222I, -0.55 Kcal/mol, 75. %; 147) 70V, -0.64 Kcal/mol, 64. %; 148) 70R, -0.77 Kcal/mol, 53. %; 149) 76V, -0.57 Kcal/mol, 71. %; 150) 232F, -0.60 Kcal/mol, 68. %;

B) List of the 150 most stabilizing mutations for MGS-MilE3 (5JD5) from a total of 6099 possible mutations:

1) 307I, -1.57 Kcal/mol, 84. %; 2) 18F, -1.47 Kcal/mol, 89. %; 3) 18Y, -1.38 Kcal/mol, 94. %; 4) 307L, -1.63 Kcal/mol, 79. %; 5) 307V, -1.78 Kcal/mol, 71. %; 6) 318L, -1.67 Kcal/mol, 75. %; 7) 18W, -1.53 Kcal/mol, 81. %; 8) 25L, -1.35 Kcal/mol, 90. %; 9) 167L, -1.40 Kcal/mol, 83. %; 10) 14F, -1.57 Kcal/mol, 74. %; 11) 318I, -1.48 Kcal/mol, 78. %; 12) 18V, -1.35 Kcal/mol, 85. %; 13) 307M, -1.60 Kcal/mol, 70. %; 14) 14M, -1.58 Kcal/mol, 69. %; 15) 30I, -1.67 Kcal/mol, 65. %; 16) 193I, -1.40 Kcal/mol, 77. %; 17) 253L, -1.72 Kcal/mol, 62. %; 18) 307F, -1.75 Kcal/mol, 61. %; 19) 18M, -1.13 Kcal/mol, 93. %; 20) 36F, -1.59 Kcal/mol, 66. %; 21) 161W, -1.61 Kcal/mol, 63. %; 22) 167I, -1.43 Kcal/mol, 70. %; 23) 253W, -1.70 Kcal/mol, 59. %; 24) 25I, -1.19 Kcal/mol, 84. %; 25) 193L, -1.26 Kcal/mol, 79. %; 26) 313L, -1.82 Kcal/mol, 54. %; 27) 172I, -1.56 Kcal/mol, 63. %; 28) 265L, -1.03 Kcal/mol, 93. %; 29) 36L, -1.46 Kcal/mol, 64. %; 30) 30F, -1.72 Kcal/mol, 54. %; 31) 253F, -1.74 Kcal/mol, 53. %; 32) 256L, -0.99 Kcal/mol, 91. %; 33) 253M, -1.72 Kcal/mol, 52. %; 34) 18L, -1.03 Kcal/mol, 85. %; 35) 265I, -0.96 Kcal/mol, 91. %; 36) 253I, -1.64 Kcal/mol, 53. %; 37) 253V, -1.57 Kcal/mol, 54. %; 38) 161L, -1.53 Kcal/mol, 55. %; 39) 30V, -1.22 Kcal/mol, 68. %; 40) 264I, -1.40 Kcal/mol, 57. %; 41) 18I, -0.97 Kcal/mol, 83. %; 42) 161M, -1.45 Kcal/mol, 55. %; 43) 161F, -1.51 Kcal/mol, 53. %; 44) 161Y, -1.53 Kcal/mol, 52. %; 45) 313I, -1.59 Kcal/mol, 50. %; 46) 273R, -1.10 Kcal/mol, 71. %; 47) 62I, -0.84 Kcal/mol, 92. %; 48) 178L, -0.91 Kcal/mol, 84. %; 49) 14R, -1.03 Kcal/mol, 73. %; 50) 253R, -1.50 Kcal/mol, 50. %; 51) 30L, -1.13 Kcal/mol, 66. %; 52) 94L, -1.03 Kcal/mol, 72. %; 53) 161V, -1.55 Kcal/mol, 47. %; 54) 283M, -0.83 Kcal/mol, 88. %; 55) 171M, -0.96 Kcal/mol, 76. %; 56) 161I, -1.43 Kcal/mol, 51. %; 57) 36H, -0.90 Kcal/mol, 80. %; 58) 94Y, -1.15 Kcal/mol, 62. %; 59) 256I, -0.77 Kcal/mol, 91. %; 60) 166I, -0.97 Kcal/mol, 72. %; 61) 256M, -0.77 Kcal/mol, 89. %; 62) 265M, -0.73 Kcal/mol, 94. %; 63) 265A, -0.78 Kcal/mol, 88. %; 64) 62L, -0.73 Kcal/mol, 93. %; 65) 178I, -0.79 Kcal/mol, 86. %; 66) 284M, -0.71 Kcal/mol, 96. %; 67) 265V, -0.75 Kcal/mol, 90. %; 68) 318V, -0.86 Kcal/mol, 78. %; 69) 283F, -0.88 Kcal/mol, 76. %; 70) 302M, -0.74 Kcal/mol, 90. %; 71) 302I, -0.72 Kcal/mol, 92. %; 72) 253T, -1.30 Kcal/mol, 51. %; 73) 253Y, -1.27 Kcal/mol, 52. %; 74) 256V, -0.76 Kcal/mol, 86. %; 75) 253S, -1.28 Kcal/mol, 51. %; 76) 178M, -0.71 Kcal/mol, 91. %; 77) 284L, -0.70 Kcal/mol, 93. %; 78) 62M, -0.69 Kcal/mol, 94. %; 79) 62F, -0.71 Kcal/mol, 89. %; 80) 313F, -1.28

Kcal/mol, 49. %; 81) 253A, -1.30 Kcal/mol, 48. %; 82) 171W, -0.89 Kcal/mol, 70. %; 83) 214I, -0.88 Kcal/mol, 69. %; 84) 94I, -0.99 Kcal/mol, 61. %; 85) 302L, -0.74 Kcal/mol, 82. %; 86) 166M, -0.87 Kcal/mol, 69. %; 87) 171Y, -0.86 Kcal/mol, 69. %; 88) 27V, -0.83 Kcal/mol, 71. %; 89) 30M, -0.92 Kcal/mol, 64. %; 90) 121C, -0.79 Kcal/mol, 74. %; 91) 194L, -0.81 Kcal/mol, 72. %; 92) 178Y, -0.64 Kcal/mol, 91. %; 93) 94F, -1.08 Kcal/mol, 54. %; 94) 108F, -0.98 Kcal/mol, 59. %; 95) 145L, -0.71 Kcal/mol, 81. %; 96) 214F, -0.80 Kcal/mol, 72. %; 97) 318P, -0.87 Kcal/mol, 66. %; 98) 40W, -0.84 Kcal/mol, 68. %; 99) 270Y, -0.98 Kcal/mol, 57. %; 100) 194I, -0.93 Kcal/mol, 61. %; 101) 256C, -0.71 Kcal/mol, 80. %; 102) 208M, -0.67 Kcal/mol, 85. %; 103) 257V, -0.87 Kcal/mol, 65. %; 104) 208L, -0.67 Kcal/mol, 84. %; 105) 116V, -0.90 Kcal/mol, 62. %; 106) 44M, -0.60 Kcal/mol, 93. %; 107) 35Y, -0.86 Kcal/mol, 64. %; 108) 155P, -0.81 Kcal/mol, 68. %; 109) 51L, -0.80 Kcal/mol, 69. %; 110) 270F, -0.90 Kcal/mol, 61. %; 111) 77L, -0.56 Kcal/mol, 98. %; 112) 302V, -0.60 Kcal/mol, 90. %; 113) 195L, -0.84 Kcal/mol, 65. %; 114) 286L, -0.69 Kcal/mol, 78. %; 115) 253E, -0.86 Kcal/mol, 63. %; 116) 166T, -0.89 Kcal/mol, 61. %; 117) 27M, -0.75 Kcal/mol, 72. %; 118) 270L, -0.76 Kcal/mol, 71. %; 119) 164I, -0.68 Kcal/mol, 79. %; 120) 109L, -0.61 Kcal/mol, 88. %; 121) 123I, -0.63 Kcal/mol, 84. %; 122) 318W, -0.63 Kcal/mol, 84. %; 123) 167M, -0.75 Kcal/mol, 70. %; 124) 283I, -0.58 Kcal/mol, 91. %; 125) 243L, -0.58 Kcal/mol, 91. %; 126) 318M, -0.63 Kcal/mol, 83. %; 127) 108M, -0.74 Kcal/mol, 70. %; 128) 27I, -0.73 Kcal/mol, 71. %; 129) 14T, -0.71 Kcal/mol, 73. %; 130) 265F, -0.57 Kcal/mol, 90. %; 131) 145V, -0.66 Kcal/mol, 78. %; 132) 173L, -0.57 Kcal/mol, 90. %; 133) 14P, -0.66 Kcal/mol, 78. %; 134) 51W, -0.70 Kcal/mol, 73. %; 135) 161A, -0.80 Kcal/mol, 64. %; 136) 27F, -0.67 Kcal/mol, 76. %; 137) 159P, -0.74 Kcal/mol, 69. %; 138) 90L, -0.57 Kcal/mol, 89. %; 139) 193V, -0.80 Kcal/mol, 64. %; 140) 120I, -0.58 Kcal/mol, 88. %; 141) 27E, -0.75 Kcal/mol, 67. %; 142) 264L, -0.74 Kcal/mol, 68. %; 143) 167K, -0.68 Kcal/mol, 74. %; 144) 190L, -0.59 Kcal/mol, 85. %; 145) 292L, -0.64 Kcal/mol, 79. %; 146) 266C, -0.89 Kcal/mol, 56. %; 147) 77W, -0.80 Kcal/mol, 62. %; 148) 125L, -0.70 Kcal/mol, 71. %; 149) 40R, -0.67 Kcal/mol, 74. %; 150) 26M, -0.54 Kcal/mol, 91. %;

C) List of the 150 most stabilizing mutations for LEH (1NWW) from a total of 2831 possible mutations.

1) 125F, -1.77 Kcal/mol, 80. %; 2) 125I, -1.83 Kcal/mol, 77. %; 3) 125L, -1.81 Kcal/mol, 77. %; 4) 23W, -1.57 Kcal/mol, 87. %; 5) 23V, -1.58 Kcal/mol, 82. %; 6) 23I, -1.65 Kcal/mol, 77. %; 7) 23F, -1.47 Kcal/mol, 84. %; 8) 64W, -2.07 Kcal/mol, 59. %; 9) 64R, -1.79 Kcal/mol, 68. %; 10) 125V, -1.63 Kcal/mol, 71. %; 11) 23L, -1.45 Kcal/mol, 78. %; 12) 125M, -1.54 Kcal/mol, 73. %; 13) 109V, -1.66 Kcal/mol, 66. %; 14) 143I, -1.48 Kcal/mol, 73. %; 15) 64C, -1.64 Kcal/mol, 65. %; 16) 64V, -1.81 Kcal/mol, 57. %; 17) 125Y, -1.50 Kcal/mol, 70. %; 18) 143F, -1.77 Kcal/mol, 59. %; 19) 83L, -1.27 Kcal/mol, 82. %; 20) 36L, -1.32 Kcal/mol, 79. %; 21) 143W, -1.49 Kcal/mol, 69. %; 22) 64Y, -1.83 Kcal/mol, 56. %; 23) 68V, -1.35 Kcal/mol, 76. %; 24) 64L, -1.88 Kcal/mol, 54. %; 25) 143V, -1.48 Kcal/mol, 68. %; 26) 68M, -1.20 Kcal/mol, 83. %; 27) 64Q, -1.72 Kcal/mol, 56. %; 28) 64H, -1.71 Kcal/mol, 56. %; 29) 67L, -1.49 Kcal/mol, 65. %; 30) 83F, -1.32 Kcal/mol, 73. %; 31) 86I, -1.20 Kcal/mol, 80. %; 32) 64A, -1.79 Kcal/mol, 53. %; 33) 143L, -1.56 Kcal/mol, 61. %; 34) 83I, -1.20 Kcal/mol, 79. %; 35) 64K, -1.69 Kcal/mol, 56. %; 36) 143M, -1.34 Kcal/mol, 70. %; 37) 116M, -1.19 Kcal/mol, 77. %; 38) 64F, -1.79 Kcal/mol, 51. %; 39) 83M, -1.30 Kcal/mol, 70. %; 40) 66I, -1.43 Kcal/mol, 63. %; 41) 24M, -1.14 Kcal/mol, 77. %; 42) 68F, -1.22 Kcal/mol, 71. %; 43) 12L, -1.08 Kcal/mol, 80. %; 44) 64M, -1.65 Kcal/mol, 52. %; 45) 86L, -1.26 Kcal/mol, 67. %; 46) 64P, -1.48 Kcal/mol, 56. %; 47) 64T, -1.58 Kcal/mol, 53. %; 48) 24L, -1.06 Kcal/mol, 78. %; 49) 120I, -1.29 Kcal/mol, 64. %; 50) 64S, -1.56 Kcal/mol, 53. %; 51) 114F, -1.08 Kcal/mol, 76. %; 52) 88W, -1.10 Kcal/mol, 74. %; 53) 110L, -0.85 Kcal/mol, 95. %; 54) 109L, -1.35 Kcal/mol, 60. %; 55) 122R, -1.06 Kcal/mol, 76. %; 56) 114Y, -1.13 Kcal/mol, 71. %; 57) 83W, -1.16 Kcal/mol, 69. %; 58) 69L, -0.82 Kcal/mol, 97. %; 59) 83Y, -1.22 Kcal/mol, 65. %; 60) 68L, -0.94 Kcal/mol, 84. %; 61) 68I, -0.91 Kcal/mol, 86. %; 62) 110F, -0.88 Kcal/mol, 89. %; 63) 64E, -1.62 Kcal/mol, 48. %; 64) 120L, -1.27 Kcal/mol, 61. %; 65) 64I, -1.61 Kcal/mol, 48. %; 66) 110V, -0.90 Kcal/mol, 86. %; 67) 64N, -1.42 Kcal/mol, 54. %; 68) 114W, -1.11 Kcal/mol, 69. %; 69) 41L, -1.13 Kcal/mol, 67. %; 70) 89V, -1.21 Kcal/mol, 63. %; 71) 88P, -1.19 Kcal/mol, 63. %; 72) 128I, -0.86 Kcal/mol, 86. %; 73) 88V, -1.04 Kcal/mol, 70. %; 74) 109F, -1.51 Kcal/mol, 48. %; 75) 88L, -1.03 Kcal/mol, 70. %; 76) 88F, -0.91 Kcal/mol, 79. %; 77) 100L, -1.06 Kcal/mol, 67. %; 78) 84I, -0.79 Kcal/mol, 88. %; 79) 88Y, -0.93 Kcal/mol, 75. %; 80) 114M, -1.03 Kcal/mol, 67. %; 81) 84L, -0.77 Kcal/mol, 89. %; 82) 67F, -0.93 Kcal/mol, 73. %; 83) 60I, -0.81 Kcal/mol, 83. %; 84) 134L, -0.92 Kcal/mol, 73. %; 85) 41F, -1.02 Kcal/mol, 66. %; 86) 41V, -1.08 Kcal/mol, 62. %; 87) 23M, -0.82 Kcal/mol, 81. %; 88) 23Y, -0.90 Kcal/mol, 73. %; 89) 149I, -0.70 Kcal/mol, 93. %; 90) 149L, -0.70 Kcal/mol, 93. %; 91) 83V, -0.89 Kcal/mol, 73. %; 92) 49P, -0.89 Kcal/mol, 73. %; 93) 86M, -0.86 Kcal/mol, 76. %; 94) 120V, -0.98 Kcal/mol, 66. %; 95) 11M, -0.85 Kcal/mol, 76. %; 96) 24W, -0.83 Kcal/mol, 77. %; 97) 69M, -0.72 Kcal/mol, 89. %; 98) 60L, -0.77 Kcal/mol, 81. %; 99) 89I, -1.06 Kcal/mol, 59. %; 100) 60C, -0.84 Kcal/mol, 74. %; 101) 114V, -0.75 Kcal/mol, 82. %; 102) 12M, -0.86 Kcal/mol, 72. %; 103) 60M, -0.77 Kcal/mol, 80. %; 104) 4L, -0.69 Kcal/mol, 89. %; 105) 127F, -0.79 Kcal/mol, 77. %; 106) 139T, -1.05 Kcal/mol, 57. %; 107) 128L, -0.79 Kcal/mol, 76. %; 108) 24F, -0.82 Kcal/mol, 73. %; 109) 88I, -0.85 Kcal/mol, 70. %; 110) 110M, -0.63 Kcal/mol, 94. %; 111) 12F, -0.85 Kcal/mol, 69. %; 112) 88M, -0.75 Kcal/mol, 78. %; 113) 60F, -0.72 Kcal/mol, 81. %; 114) 123F, -0.75 Kcal/mol, 77. %; 115) 133W, -0.76 Kcal/mol, 75. %; 116) 122Y, -0.87 Kcal/mol, 65. %; 117) 134V, -0.89 Kcal/mol, 64. %; 118) 48M, -0.77 Kcal/mol, 73. %; 119) 24V, -0.81 Kcal/mol, 69. %; 120) 67I, -0.96 Kcal/mol, 57. %; 121) 12I, -0.80 Kcal/mol, 69. %; 122) 67M, -0.94 Kcal/mol, 57. %; 123) 80W, -0.86 Kcal/mol, 63. %; 124) 49R, -0.83 Kcal/mol, 65. %; 125) 32R, -0.84 Kcal/mol, 64. %; 126) 32L, -0.89 Kcal/mol, 60. %; 127) 49W, -0.84 Kcal/mol, 63. %; 128) 29I, -0.87 Kcal/mol, 61. %; 129) 80C, -0.78 Kcal/mol, 67. %; 130) 16Y, -0.73 Kcal/mol, 72. %; 131) 7L, -0.68 Kcal/mol, 77. %; 132) 86C, -0.93 Kcal/mol, 56. %; 133) 38L, -0.96 Kcal/mol, 54. %; 134) 38W, -0.95 Kcal/mol, 55. %; 135) 48L, -0.74 Kcal/mol, 70. %; 136) 97V, -0.79 Kcal/mol, 65. %; 137) 110I, -0.55 Kcal/mol, 94. %; 138) 49K, -0.71 Kcal/mol, 72. %; 139) 126L, -0.58 Kcal/mol, 88. %; 140) 49F, -0.83 Kcal/mol, 61. %; 141) 48F, -0.75 Kcal/mol, 67. %; 142) 12R, -0.65 Kcal/mol, 77. %; 143) 143R, -0.79 Kcal/mol, 63. %; 144) 29M, -0.83 Kcal/mol, 60. %; 145) 4I, -0.54 Kcal/mol, 91. %; 146) 149M, -0.54 Kcal/mol, 92. %; 147) 14I, -0.52 Kcal/mol, 94. %; 148) 122F, -0.79 Kcal/mol, 62. %; 149) 60Y, -0.64 Kcal/mol, 77. %; 150) 122W, -0.77 Kcal/mol, 63. %;

D) List of the 150 most stabilizing mutations for ω-TA (6G4B) from a total of 8645 possible mutations:

1) 449F, -1.70 Kcal/mol, 87. %; 2) 449L, -1.68 Kcal/mol, 84. %; 3) 449M, -1.63 Kcal/mol, 86. %; 4) 449I, -1.68 Kcal/mol, 83. %; 5) 421L, -1.56 Kcal/mol, 87. %; 6) 76L, -1.50 Kcal/mol, 89. %; 7) 201I, -1.62 Kcal/mol, 80. %; 8) 201L, -1.64 Kcal/mol, 77. %; 9) 292L,

-1.57 Kcal/mol, 80. %; 10) 448W, -1.57 Kcal/mol, 80. %; 11) 201F, -1.62 Kcal/mol, 76. %; 12) 76I, -1.61 Kcal/mol, 74. %; 13) 201M, -1.55 Kcal/mol, 76. %; 14) 311W, -1.51 Kcal/mol, 76. %; 15) 92I, -1.55 Kcal/mol, 72. %; 16) 421I, -1.28 Kcal/mol, 85. %; 17) 452F, -1.65 Kcal/mol, 64. %; 18) 167I, -1.41 Kcal/mol, 74. %; 19) 168I, -1.52 Kcal/mol, 68. %; 20) 448V, -1.67 Kcal/mol, 62. %; 21) 185Y, -1.24 Kcal/mol, 83. %; 22) 167M, -1.32 Kcal/mol, 77. %; 23) 94I, -1.24 Kcal/mol, 82. %; 24) 449W, -1.77 Kcal/mol, 56. %; 25) 133I, -1.35 Kcal/mol, 72. %; 26) 445L, -1.49 Kcal/mol, 65. %; 27) 312F, -1.15 Kcal/mol, 83. %; 28) 134F, -1.41 Kcal/mol, 68. %; 29) 201V, -1.44 Kcal/mol, 66. %; 30) 343W, -1.19 Kcal/mol, 80. %; 31) 168V, -1.53 Kcal/mol, 62. %; 32) 312L, -1.17 Kcal/mol, 81. %; 33) 292M, -1.21 Kcal/mol, 78. %; 34) 409L, -1.06 Kcal/mol, 89. %; 35) 353L, -1.19 Kcal/mol, 79. %; 36) 167L, -1.30 Kcal/mol, 72. %; 37) 464L, -1.07 Kcal/mol, 87. %; 38) 312I, -1.11 Kcal/mol, 83. %; 39) 313L, -1.06 Kcal/mol, 86. %; 40) 167V, -1.24 Kcal/mol, 74. %; 41) 133M, -1.11 Kcal/mol, 82. %; 42) 274L, -0.91 Kcal/mol, 99. %; 43) 94L, -1.11 Kcal/mol, 81. %; 44) 292V, -1.45 Kcal/mol, 62. %; 45) 94M, -1.20 Kcal/mol, 74. %; 46) 449Y, -1.82 Kcal/mol, 49. %; 47) 201W, -1.18 Kcal/mol, 75. %; 48) 201Y, -1.03 Kcal/mol, 84. %; 49) 122I, -1.10 Kcal/mol, 78. %; 50) 353I, -1.07 Kcal/mol, 79. %; 51) 122K, -1.34 Kcal/mol, 63. %; 52) 258Q, -1.21 Kcal/mol, 70. %; 53) 350I, -0.91 Kcal/mol, 93. %; 54) 116I, -1.28 Kcal/mol, 66. %; 55) 76V, -1.56 Kcal/mol, 54. %; 56) 292I, -1.27 Kcal/mol, 66. %; 57) 146V, -1.16 Kcal/mol, 72. %; 58) 449V, -1.76 Kcal/mol, 47. %; 59) 185I, -0.89 Kcal/mol, 92. %; 60) 73I, -1.20 Kcal/mol, 68. %; 61) 27L, -0.92 Kcal/mol, 88. %; 62) 152M, -0.94 Kcal/mol, 86. %; 63) 94V, -0.99 Kcal/mol, 81. %; 64) 448M, -1.12 Kcal/mol, 72. %; 65) 26L, -0.86 Kcal/mol, 93. %; 66) 158F, -1.12 Kcal/mol, 71. %; 67) 311E, -1.10 Kcal/mol, 72. %; 68) 134L, -1.06 Kcal/mol, 74. %; 69) 242F, -1.02 Kcal/mol, 77. %; 70) 356M, -1.03 Kcal/mol, 75. %; 71) 26I, -0.87 Kcal/mol, 88. %; 72) 185L, -0.80 Kcal/mol, 95. %; 73) 448F, -1.12 Kcal/mol, 68. %; 74) 185M, -0.80 Kcal/mol, 94. %; 75) 356F, -0.85 Kcal/mol, 88. %; 76) 201R, -0.97 Kcal/mol, 77. %; 77) 12A, -1.17 Kcal/mol, 63. %; 78) 356V, -1.15 Kcal/mol, 64. %; 79) 449R, -1.50 Kcal/mol, 49. %; 80) 418Y, -0.95 Kcal/mol, 77. %; 81) 180I, -0.85 Kcal/mol, 86. %; 82) 407M, -0.96 Kcal/mol, 76. %; 83) 407L, -0.80 Kcal/mol, 90. %; 84) 67M, -0.94 Kcal/mol, 77. %; 85) 168L, -0.88 Kcal/mol, 82. %; 86) 166I, -0.81 Kcal/mol, 88. %; 87) 445I, -1.01 Kcal/mol, 71. %; 88) 377I, -0.91 Kcal/mol, 78. %; 89) 185F, -0.77 Kcal/mol, 92. %; 90) 258E, -0.98 Kcal/mol, 72. %; 91) 400F, -0.82 Kcal/mol, 86. %; 92) 377L, -0.90 Kcal/mol, 78. %; 93) 166V, -0.77 Kcal/mol, 90. %; 94) 134I, -1.30 Kcal/mol, 53. %; 95) 150C, -1.13 Kcal/mol, 61. %; 96) 198M, -0.89 Kcal/mol, 77. %; 97) 408I, -0.86 Kcal/mol, 79. %; 98) 185D, -0.75 Kcal/mol, 90. %; 99) 59V, -0.73 Kcal/mol, 93. %; 100) 152I, -0.80 Kcal/mol, 85. %; 101) 266L, -0.73 Kcal/mol, 92. %; 102) 312M, -0.97 Kcal/mol, 70. %; 103) 210I, -0.72 Kcal/mol, 94. %; 104) 313I, -1.04 Kcal/mol, 65. %; 105) 52I, -0.96 Kcal/mol, 70. %; 106) 200L, -0.97 Kcal/mol, 69. %; 107) 185S, -0.71 Kcal/mol, 94. %; 108) 185E, -0.71 Kcal/mol, 94. %; 109) 350L, -0.69 Kcal/mol, 96. %; 110) 30C, -0.97 Kcal/mol, 69. %; 111) 293W, -0.91 Kcal/mol, 73. %; 112) 464I, -0.78 Kcal/mol, 85. %; 113) 74M, -0.82 Kcal/mol, 81. %; 114) 311I, -1.06 Kcal/mol, 62. %; 115) 448R, -0.91 Kcal/mol, 72. %; 116) 400M, -0.72 Kcal/mol, 90. %; 117) 259L, -1.02 Kcal/mol, 64. %; 118) 356I, -1.00 Kcal/mol, 65. %; 119) 160M, -0.81 Kcal/mol, 79. %; 120) 325L, -0.69 Kcal/mol, 92. %; 121) 150W, -1.03 Kcal/mol, 62. %; 122) 259I, -1.01 Kcal/mol, 63. %; 123) 332F, -0.81 Kcal/mol, 79. %; 124) 343L, -0.76 Kcal/mol, 83. %; 125) 89I, -0.72 Kcal/mol, 88. %; 126) 180L, -0.81 Kcal/mol, 77. %; 127) 198F, -0.89 Kcal/mol, 70. %; 128) 198W, -1.02 Kcal/mol, 61. %; 129) 402M, -0.79 Kcal/mol, 78. %; 130) 308L, -0.74 Kcal/mol, 83. %; 131) 89M, -0.69 Kcal/mol, 89. %; 132) 337L, -0.73 Kcal/mol, 85. %; 133) 16V, -0.74 Kcal/mol, 83. %; 134) 185T, -0.65 Kcal/mol, 94. %; 135) 311F, -0.93 Kcal/mol, 66. %; 136) 74F, -0.82 Kcal/mol, 74. %; 137) 312V, -1.15 Kcal/mol, 53. %; 138) 448L, -0.98 Kcal/mol, 62. %; 139) 210L, -0.67 Kcal/mol, 91. %; 140) 396M, -0.65 Kcal/mol, 93. %; 141) 441M, -0.77 Kcal/mol, 79. %; 142) 273M, -0.93 Kcal/mol, 65. %; 143) 332M, -0.77 Kcal/mol, 78. %; 144) 76F, -0.84 Kcal/mol, 72. %; 145) 402I, -0.77 Kcal/mol, 78. %; 146) 95A, -1.00 Kcal/mol, 60. %; 147) 198P, -0.77 Kcal/mol, 78. %; 148) 437M, -0.94 Kcal/mol, 64. %; 149) 158M, -0.77 Kcal/mol, 78. %; 150) 152L, -0.70 Kcal/mol, 85. %;

Supplementary Table 2: Dataset extracted from Khan's dataset (2010a) and Kumar's dataset (2006). For each mutation we specify the PDB ID, the mutation (wild-type residue, position and new mutant residue) and the ΔΔG in Kcal/mol. For further information you can review the original datasets.

PDB ID	Mut	ΔΔG	PDB ID	Mut	ΔΔG	PDB ID	Mut	ΔΔG	PDB ID	Mut	ΔΔG	PDB ID	Mut	ΔΔG
1A23	H32Y	-0.5	1LRP	G48S	-0.68	1YCC	N52M	-2.58	1FLV	G87L	0.21	1CSP	E12K	0.19
1AAR	F45W	-0.32	1LRP	K4Q	-0.43	1YCC	N52Q	0.08	1FLV	G87V	-0.12	1CSP	E19K	0.5
1AAR	H68Q	-0.55	1LRP	Q33Y	-1.32	1YCC	N52S	0.05	1FLV	V18I	-0.08	1CSP	E21K	0.07
1AAR	K29N	1.48	1LRP	Q44Y	0.02	1YCC	N52T	0.53	1FLV	V18L	-0.04	1CSP	E21Q	0.24
1AAR	K6E	-0.53	1LRP	Y88C	-2.4	1YCC	N52V	-1.67	1FTG	A101V	-0.16	1CSP	E3K	-2.75
1AAR	K6Q	-0.26	1LZ1	A32L	0.1	1YCC	P76G	0.78	1FTG	A84G	0.47	1CSP	E3L	-1.01
1AAR	R42E	-1.63	1LZ1	A32S	0.33	1YCC	P76V	1.07	1FTG	D126K	-0.03	1CSP	E3Q	-1.1
1AAR	R72Q	0.33	1LZ1	A47P	-0.1	1YCC	T69E	-1.3	1FTG	D150K	-0.01	1CSP	E3R	-1.72
1AJ3	W22Y	0.23	1LZ1	A92S	-0.81	1YCC	T96A	-0.8	1FTG	D43A	0.03	1CSP	E42K	0
1AKK	L94V	1.2	1LZ1	A96M	-0.02	2ABD	D21A	-0.42	1FTG	D65K	-0.09	1CSP	E42Q	0.12
1AM7	H31N	-1.8	1LZ1	A9S	0.02	2ABD	E67A	0.36	1FTG	D75K	0.01	1CSP	E43G	-2.82
1ANK	R88G	0.2	1LZ1	D102N	-0.07	2ABD	F5A	2.52	1FTG	E107A	-0.15	1CSP	E43K	-0.14
1AQH	Q58C	-1.87	1LZ1	D120N	-0.05	2ABD	K32A	1.02	1FTG	E20K	-0.14	1CSP	E43Q	-0.02
1ARR	N29A	-1.32	1LZ1	D18N	0.53	2ABD	K32E	1.68	1FTG	E40K	0.06	1CSP	E43S	-0.29
1ARR	P8L	-2.4	1LZ1	D49N	0	2ABD	K52M	-0.18	1FTG	E61K	-0.22	1CSP	E50K	-0.02
1AYF	C95S	-0.96	1LZ1	D67N	0.24	2ABD	K54M	-0.27	1FTG	G68A	-0.38	1CSP	E53K	0.05
1AYF	D73E	-0.45	1LZ1	D91P	0.4	2ABD	T35A	1.09	1FTG	I156V	0.69	1CSP	E53Q	-0.12
1AYF	T51S	-0.05	1LZ1	E35L	0.53	2ABD	Y31N	1.52	1FTG	I21A	0.02	1CSP	E66K	-2.18
1AYF	Y79F	0.01	1LZ1	E7Q	-0.1	2ABD	Y73F	-0.27	1FTG	I21G	-0.17	1CSP	E66L	-1.77
1AYF	Y79L	0.36	1LZ1	G127A	0.55	2AFG	C117S	0.26	1FTG	I22A	0.67	1CSP	F17A	-0.55
1AYF	Y79S	0.33	1LZ1	G129A	-0.14	2AFG	F108Y	-0.33	1FTG	I51V	0.41	1CSP	F27A	0.12
1AYF	Y79W	0.21	1LZ1	G19A	1.77	2AFG	H102Y	-0.1	1FTG	I52V	0.18	1CSP	F38A	-0.31
1B26	E231A	0.17	1LZ1	G37A	0.29	2AFG	H21Y	-0.38	1FTG	L143A	0.11	1CSP	K13E	0.29
1B26	K193A	-0.23	1LZ1	G37Q	0.26	2AFG	H93G	-1.08	1FTG	L50A	-0.02	1CSP	K13Q	0.07

PDB ID	Mut	ΔΔG	PDB ID	Mut	ΔΔG	PDB ID	Mut	ΔΔG	PDB ID	Mut	ΔΔG	PDB ID	Mut	ΔΔG
1B26	R190A	0.15	1LZ1	G48A	-0.45	2AFG	L44F	-0.31	1FTG	L6A	1.11	1CSP	K39E	0.05
1B5M	D58R	-0.14	1LZ1	G68A	0.12	2AFG	L44P	0.14	1FTG	N97A	0.22	1CSP	K39Q	0
1BCX	S100C	-3	1LZ1	G72A	0.36	2AFG	L73V	0.89	1FTG	Q111G	-0.22	1CSP	K65E	0.79
1BCX	V98C	-1.3	1LZ1	H78A	0.14	2AFG	V109I	0.08	1FTG	Q99A	-0.43	1CSP	K65I	-1.53
1BNI	A32C	1	1LZ1	H78G	0.12	2CHF	D12A	-2.5	1FTG	S110A	0.18	1CSP	K65Q	0.12
1BNI	A32F	0.7	1LZ1	I106A	0.93	2CHF	D13A	-2.7	1FTG	S71A	0.11	1CSP	M1R	-1.75
1BNI	A32H	0.8	1LZ1	I106V	0.72	2CI2	D42A	0.96	1FTG	T122S	-0.11	1CSP	N10D	-0.26
1BNI	A32K	0.2	1LZ1	I23V	0.36	2CI2	E26A	0.47	1FTG	V117A	0.95	1CSP	N55D	-0.38
1BNI	A32L	0.3	1LZ1	I56L	0.1	2CI2	E26Q	0.62	1FTG	V18I	-0.19	1CSP	N55K	-0.1
1BNI	A32M	0.3	1LZ1	I59L	0	2CI2	E33A	-0.64	1FTG	V31A	0.46	1CSP	N55S	0.17
1BNI	A32N	0.7	1LZ1	I89V	0.35	2CI2	E33D	0.52	1FTG	V83I	-0.05	1CSP	R56Q	-0.17
1BNI	A32Q	0.5	1LZ1	N118A	-0.19	2CI2	E33Q	0.29	1G6N	S129A	0.26	1CSP	S48E	0
1BNI	A32S	0.4	1LZ1	N118G	-0.05	2CI2	E34Q	0.47	1G6N	S129P	-0.14	1CSP	S48K	-0.48
1BNI	A32T	0.8	1LZ1	P103G	0.1	2CI2	E45A	0.32	1HFY	A30I	-0.3	1CSP	S48R	-1.58
1BNI	A32V	0.9	1LZ1	P71G	1.6	2CI2	E60A	0.68	1HFY	A30T	-0.06	1CYO	F35H	2.82
1BNI	D12A	-0.28	1LZ1	Q58A	-0.91	2CI2	I48V	0.92	1HFY	T29I	-0.12	1CYO	V45H	1.34
1BNI	D22M	0.27	1LZ1	Q58G	-1.87	2CI2	I49V	-0.08	1HFY	T29V	-1.86	1CYO	V61K	2.34
1BNI	D44E	-0.1	1LZ1	R21A	-1.32	2CI2	I56A	0.03	1HFY	T33I	-0.18	1DYJ	D27N	-1.4
1BNI	D54A	3	1LZ1	R21G	-1.15	2CI2	I76V	-0.09	1HFZ	H107Y	0.19	1EL1	A93S	0.26
1BNI	D8A	-0.1	1LZ1	R50A	-0.43	2CI2	K21A	0.55	1HFZ	H32Y	-0.07	1EL1	H21G	0.48
1BNI	E29Q	0	1LZ1	R50G	-0.26	2CI2	K21M	0.67	1HFZ	L110E	0.19	1EL1	I56L	0.24
1BNI	G34H	2.6	1LZ1	V100A	0.26	2CI2	K30A	-0.42	1HFZ	L110R	-0.43	1FC1	K392A	0.4
1BNI	G65S	-0.5	1LZ1	V100F	1.65	2CI2	K36A	0.49	1HFZ	Q54A	0.41	1FC1	L351A	1.3
1BNI	H18G	-0.31	1LZ1	V100T	0.29	2CI2	K37A	-0.21	1HFZ	Y103P	0.22	1FC1	L398A	0.1
1BNI	H18K	1.19	1LZ1	V110A	-0.07	2CI2	K37G	0.97	1HME	G35H	0.24	1FLV	A101V	0.69
1BNI	I25V	1.1	1LZ1	V110D	-0.17	2CI2	K43A	-0.26	1HME	I34H	0	1FLV	G87A	0.09
1BNI	I4A	1.4	1LZ1	V110F	0.05	2CI2	K72N	0	1HME	N47H	-0.27	1RTB	A4S	0.22
1BNI	I51V	1.8	1LZ1	V110G	-0.48	2CI2	L27A	2.64	1HUE	A56S	0.13	1RTB	A5S	-0.08
1BNI	I55V	0.3	1LZ1	V110I	-0.86	2CI2	L51I	0.26	1HUE	M69I	0	1RTB	D121A	0.72
1BNI	I76V	0.8	1LZ1	V110L	-0.07	2CI2	L51V	0.5	1HUE	S31T	-0.41	1RTB	D121N	0.76
1BNI	I88L	0.3	1LZ1	V110M	-0.53	2CI2	N75D	1.21	1HUE	V42I	0.82	1RTB	H119A	-0.2
1BNI	I96V	0.9	1LZ1	V110N	-0.07	2CI2	Q41A	0.02	1IGV	D19N	-0.52	1RTB	I107A	2.85
1BNI	K108R	-0.9	1LZ1	V110P	-0.5	2CI2	Q41G	0.6	1IGV	E17Q	-0.26	1RTB	I107V	0.08
1BNI	K19R	-0.2	1LZ1	V110R	-0.89	2CI2	Q47M	-0.32	1IGV	E26Q	-0.09	1RTB	I81V	0.43
1BNI	K27G	0.4	1LZ1	V110Y	0.14	2CI2	S31G	0.8	1IOB	K97R	0.5	1RTB	S123A	-0.46
1BNI	K66A	-0.2	1LZ1	V125A	1.32	2CI2	T22A	0.85	1IOB	K97V	-0.8	1RTB	V108I	0.44
1BNI	L89V	0.3	1LZ1	V130A	0.84	2CI2	T22V	0.32	1IOB	T9G	2.6	1RTB	V108L	0.7
1BNI	N58A	2.7	1LZ1	V2D	1.44	2CI2	T55A	-0.23	1IRO	I33L	-0.76	1RTB	V116A	0.67
1BNI	N58D	-0.5	1LZ1	V2I	-1.1	2CI2	T55S	0.02	1IRO	V24I	-0.36	1RTB	V118G	2.78
1BNI	Q104A	0.2	1LZ1	V2M	0.31	2CI2	T55V	0.76	1K9Q	D26T	-0.35	1RTB	V54I	1.95
1BNI	Q15A	0.2	1LZ1	V2R	0.38	2CI2	T58A	0.69	1K9Q	L22Y	-0.17	1RTB	V63A	2.03
1BNI	Q15I	-1	1LZ1	V2S	1.41	2CI2	T58D	-0.04	1KFW	G253P	0.65	1RTP	A21P	-0.45
1BNI	Q31A	-0.1	1LZ1	V2Y	0.36	2CI2	V38A	0.46	1KFW	G405Q	-0.62	1RTP	H26P	-1.25
1BNI	Q31G	0.98	1LZ1	V74D	0.43	2CI2	V53A	0.63	1KFW	G92P	-0.5	1RTP	K80S	0.29
1BNI	Q31S	0.2	1LZ1	V74F	0.29	2CI2	V70A	1.95	1KFW	N197K	-0.81	1RX4	E139Q	1.36
1BNI	R110A	0.4	1LZ1	V74G	0.22	2CI2	V82A	1.45	1L63	F104M	0.4	1RX4	L28R	-0.5
1BNI	R69M	2.12	1LZ1	V74I	-0.45	2HPR	S46D	-0.7	1L63	I50M	0.4	1RX4	V75A	0.2
1BNI	S28A	-0.41	1LZ1	V74L	-0.19	2LZM	A129V	0.7	1L63	I78A	1.2	1RX4	V75C	0.2
1BNI	S28E	-0.4	1LZ1	V74M	-0.65	2LZM	A130S	1	1L63	L66M	1	1RX4	V88A	-0.2
1BNI	S28G	0.45	1LZ1	V74N	0.33	2LZM	A134S	0.1	1L63	M106A	1.9	1SHG	A11V	-0.6
1BNI	S85A	0.12	1LZ1	V74R	0.07	2LZM	A146T	1.5	1LRP	A15G	0.55	1SHG	F52Y	0.44
1BNI	S91A	1.16	1LZ1	V74S	0.38	2LZM	A41D	-0.29	1LRP	A20G	-0.87	1SHG	K18F	-2.33
1BNI	T100G	2.8	1LZ1	V74Y	0.24	2LZM	A41S	0.6	1LRP	A37G	-0.62	1SHG	K18Y	0.05
1BNI	T16A	0.27	1LZ1	V93A	0.74	2LZM	A41V	-0.3	1LRP	A49G	-1.25	1SHG	K59F	-1.71
1BNI	T16R	-0.2	1LZ1	V93T	0.67	2LZM	A73S	0.4	1LRP	A63G	-1.49	1SHG	K59Y	-0.84
1BNI	T26D	0.08	1LZ1	V99T	0.5	2LZM	A82P	0.07	1LRP	A66G	-0.07	1SSO	I29V	0.4
1BNI	T26E	0.05	1LZ1	Y124F	0.36	2LZM	A82S	0.3	1LRP	A66T	2.99	1STN	D77G	2.2
1BNI	T26G	1.5	1LZ1	Y20F	0.5	2LZM	A93P	-0.03	1LRP	G46A	-0.66	1STN	F61A	2.4
1BNI	T26N	1.29	1LZ1	Y38A	2.49	2LZM	A93S	0.2	1LRP	G48A	-0.87	1STN	F76V	0
1BNI	T6D	-0.11	1LZ1	Y45F	-0.07	2LZM	A93T	-0.06	1LRP	G48N	-0.79	1STN	G88V	-0.5
1BNI	T6N	1.27	1LZ1	Y63F	0.24	2LZM	C54T	-0.3	1TUP	R249S	1.95	1STN	H46Y	0
1BNI	T6Q	1.87	1MGR	Y33F	-0.5	2LZM	C54V	0.7	1VQB	A86T	0.7	1STN	I18M	0.2
1BNI	T6S	0.22	1MGR	Y55F	2.1	2LZM	D127A	-0.24	1VQB	A86V	-0.5	1STN	K116G	-1
1BNI	T79V	-0.3	1MGR	Y89F	0	2LZM	D20A	0.3	1VQB	C33A	0.5	1STN	N118D	2.5
1BNI	V36A	1.3	1PGA	D47A	-0.36	2LZM	D20N	-1.3	1VQB	C33V	0.2	1STN	S141A	0.12
1BNI	W94L	-0.33	1PGA	I6A	0	2LZM	D20S	-0.7	1VQB	E30F	-1.17	1STN	V66L	-0.3
1BNI	W94Y	0.99	1PGA	I6E	-0.16	2LZM	D20T	-0.9	1VQB	E30M	-0.6	1STN	W140F	0.6
1BNI	Y103F	0	1PGA	I6F	-2.1	2LZM	D47A	0.28	1VQB	E40T	0.4	1STN	W140H	0.4
1BNI	Y13F	0.41	1PGA	I6K	-1.6	2LZM	D92N	0.1	1VQB	F73W	-0.8	1SUP	G169A	-0.3
1BNI	Y17A	2	1PGA	I6L	-2.8	2LZM	E11A	-1.1	1VQB	H64C	-0.5	1SUP	M50F	-0.48
1BNI	Y17F	0.3	1PGA	I6N	-1.94	2LZM	E11F	-1.7	1VQB	I47L	0.6	1SUP	N218S	-1.07
1BNI	Y24F	0	1PGA	I6R	-0.16	2LZM	E11H	-0.1	1VQB	I6V	0.04	1SUP	N76D	-0.45
1BPI	D3A	-0.2	1PGA	I6T	-1.11	2LZM	E11M	-1.6	1VQB	I78V	1.3	1SUP	Q206C	-1.25
1BPI	D50A	0.4	1PGA	I6V	-2.5	2LZM	E128A	-0.16	1VQB	K24V	-0.8	1SUP	Y217K	-0.72
1BPI	E49A	0.2	1PGA	K10P	0.2	2LZM	E128K	1.16	1VQB	K69M	-0.1	1TEN	E887A	-1.61
1BPI	G28A	1	1PGA	K50A	0.45	2LZM	E22K	-0.57	1VQB	L28V	-1.1	1TUP	C242S	2.94
1BPI	G56A	0.2	1PGA	T16I	-1.82	2LZM	E45A	-0.01	1VQB	L32W	-2.8	2LZM	V111A	1.1
1BPI	G57A	0.2	1PGA	T16L	-2.06	2LZM	F153I	0.2	1VQB	L65P	1.5	2LZM	V111I	0.69
1BPI	K15A	0.4	1PGA	T16V	-2.15	2LZM	F153L	-0.2	1VQB	M77C	0	2LZM	V131A	-0.15
1BPI	K26A	0	1PGA	T49A	0.86	2LZM	F153M	0.6	1VQB	M77F	0.2	2LZM	V131D	-0.08

PDB ID	Mut	ΔΔG	PDB ID	Mut	ΔΔG	PDB ID	Mut	ΔΔG	PDB ID	Mut	ΔΔG	PDB ID	Mut	ΔΔG
1BPI	K41A	0.4	1PGA	T53A	-0.08	2LZM	G113A	-0.1	1VQB	M77I	-1.6	2LZM	V131E	-0.2
1BPI	K46A	-0.1	1PGA	T53E	-0.23	2LZM	G113E	-0.3	1VQB	M77L	1.2	2LZM	V131I	-0.16
1BPI	L29A	0	1PGA	T53H	-0.37	2LZM	G30A	-0.1	1VQB	M77T	0.8	2LZM	V131L	-0.09
1BPI	P8A	0.3	1PGA	T53I	-1.25	2LZM	G77A	-0.4	1VQB	M77V	-1.2	2LZM	V131M	-0.12
1BPI	P9A	0.8	1PGA	T53K	-0.35	2LZM	I100M	1.6	1VQB	T48V	0	2LZM	V131S	0.05
1BPI	R17A	0.3	1PGA	T53L	-0.45	2LZM	I100V	0.4	1VQB	T62C	0.7	2LZM	V149I	0
1BPI	R1A	0.5	1PGA	T53M	-0.9	2LZM	I3A	0.7	1VQB	T62V	-1.3	2LZM	V71A	1.5
1BPI	R39A	0	1PGA	T53N	-0.52	2LZM	I3F	0.53	1VQB	V19C	0.3	2LZM	V87I	0.3
1BPI	R42A	0.5	1PGA	T53Q	-0.38	2LZM	I3L	-0.4	1VQB	V19T	0.6	2LZM	V94A	1.8
1BPI	R53A	0.1	1PGA	T53R	-0.4	2LZM	I3M	0.3	1VQB	V35I	0.6	2LZM	W138Y	1.71
1BPI	T11A	0	1PGA	T53S	-0.87	2LZM	I3V	0.4	1VQB	V35M	1	2RN2	A125T	0
1BPI	T32A	0.1	1PGA	T53V	-2.37	2LZM	K124G	0	1VQB	V43T	1.6	2RN2	A24V	-0.73
1BPI	T54A	0.1	1PGA	T53W	-1.04	2LZM	K147E	-0.1	1VQB	V45C	0.1	2RN2	A52C	-0.8
1BPI	V34A	1.2	1PGA	T53Y	-1.63	2LZM	K16E	-0.1	1VQB	Y41A	0.4	2RN2	A52D	1.9
1BPI	Y35A	1.1	1PGA	V21P	0.1	2LZM	K60H	0.1	1VQB	Y41F	0.6	2RN2	A52F	-0.03
1BTA	C82A	-0.48	1PGA	V54A	0.1	2LZM	K60P	0	1W4E	A139G	2.3	2RN2	A52I	-1.19
1BVC	A130K	0.9	1POH	K49E	-2	2LZM	K83H	0.4	1W4E	A168G	0.3	2RN2	A52M	-0.5
1BVC	A130L	-0.1	1POH	S46D	-1.4	2LZM	L118M	0.7	1W4E	D145N	0.9	2RN2	A52T	0.8
1BVC	D44A	0.15	1POH	S46N	-0.3	2LZM	L121M	0.8	1W4E	E141A	0.7	2RN2	D10A	-2.4
1BVC	D60A	0.1	1REX	T43V	-0.96	2LZM	L133F	0.2	1W4E	E141Q	0.4	2RN2	D10E	-1.05
1BVC	E109A	-0.17	1RGG	D17K	1.1	2LZM	L133M	0.4	1W4E	I130A	0.7	2RN2	D10N	0.7
1BVC	E109G	0.89	1RGG	D25H	-0.9	2LZM	L99F	-0.03	1W4E	I130V	0.2	2RN2	D134E	-0.72
1BVC	E18A	0.5	1RGG	D79E	0.3	2LZM	L99I	1.4	1W4E	I163V	0.5	2RN2	D134L	-1.33
1BVC	E4A	0.55	1RGG	D79H	-1.8	2LZM	L99M	0.4	1W4E	L159G	1.9	2RN2	D134Q	-0.48
1BVC	G129A	-1.1	1RGG	D79I	-2.8	2LZM	M102L	0.74	1W4E	L167V	0.2	2RN2	D134S	-0.26
1BVC	G23A	-0.74	1RGG	D79K	-2.3	2LZM	M106I	-0.2	1W4E	V129A	0.7	2RN2	D134T	-0.12
1BVC	G65A	0.11	1RGG	D79L	-2.7	2LZM	M106L	-0.5	1WQ5	C81A	0.69	2RN2	D134V	-0.31
1BVC	H116A	-0.2	1RGG	D79N	-1.18	2LZM	M120A	0.2	1WQ5	C81G	1.58	2RN2	D70A	0.1
1BVC	H36Q	0.1	1RGG	D79R	-2.7	2LZM	M120L	-0.5	1WQ5	E49C	-0.01	2RN2	D70E	-0.1
1BVC	I111M	1.1	1RGG	D79W	-2.3	2LZM	M120Y	0.1	1WQ5	E49D	0.03	2RN2	D94R	-1.1
1BVC	I142L	-0.6	1RGG	D79Y	-2.9	2LZM	M6I	1.38	1WQ5	E49G	-0.08	2RN2	E119V	0.06
1BVC	I142M	-0.9	1RGG	E41K	0.7	2LZM	N116A	-0.17	1WQ5	E49H	-0.33	2RN2	E135K	0.22
1BVC	I28V	0	1RGG	E74K	-0.1	2LZM	N116D	0.1	1WQ5	E49I	-0.46	2RN2	E48A	0.1
1BVC	K140A	-0.35	1RGG	H85Q	0	2LZM	N132F	-1.3	1WQ5	E49K	-0.2	2RN2	E48D	-0.2
1BVC	K56A	0.2	1RGG	N39A	2.2	2LZM	N132I	-1.2	1WQ5	E49L	-0.44	2RN2	E48Q	0.1
1BVC	K77A	0.02	1RGG	N39S	2.3	2LZM	N132M	-1.5	1WQ5	E49M	0.05	2RN2	G23A	-0.5
1BVC	L11A	0.4	1RGG	Q38A	-0.4	2LZM	N144D	0.1	1WQ5	E49N	-0.27	2RN2	H62D	0.17
1BVC	L135M	0.8	1RGG	Q94K	-0.2	2LZM	N144E	-0.2	1WQ5	E49P	0	2RN2	H62R	0.03
1BVC	L149A	1.6	1RGG	R65A	1	2LZM	N144H	-0.3	1WQ5	E49Q	0.23	2RN2	K117R	-0.03
1BVC	L29A	2.4	1RGG	T16V	-0.3	2LZM	N163D	0.21	1WQ5	E49S	-0.09	2RN2	K91R	0
1BVC	L29M	-0.1	1RGG	T18V	1.4	2LZM	N40A	-0.2	1WQ5	E49T	0.26	2RN2	K95A	-0.1
1BVC	L69A	1.2	1RGG	T5V	0	2LZM	N40D	-0.44	1WQ5	E49V	-0.14	2RN2	K95G	-1.8
1BVC	L69I	0	1RGG	T67V	0	2LZM	N53A	0.8	1WQ5	E49W	0.97	2RN2	K95N	-0.88
1BVC	L69M	0	1RGG	T82V	1.7	2LZM	N55G	0.5	1WQ5	E49Y	0.17	2RN2	Q113P	0.2
1BVC	L69V	0.1	1RGG	V43T	0.5	2LZM	N68A	0.05	1WQ5	F139W	0.07	2RN2	Q76L	-0.24
1BVC	P88A	-0.6	1RGG	Y30F	-0.4	2LZM	Q105A	0.6	1WQ5	F258W	0.16	2RN2	Q80L	0.03
1BVC	Q8A	-0.98	1RGG	Y49F	0.2	2LZM	Q105E	0.5	1WQ5	G211D	-0.2	3SSI	M73K	-0.05
1BVC	Q8G	0.5	1RGG	Y80F	1.5	2LZM	Q122A	0.24	1WQ5	G211E	-0.3	3SSI	M73L	0.13
1BVC	S117A	0.3	1RGG	Y81F	1.2	2LZM	Q123E	-0.1	1WQ5	G211R	-0.1	3SSI	V13I	0.84
1BVC	T67A	0.3	1RGG	Y86F	0.3	2LZM	Q69P	2.9	1WQ5	G211V	-0.2	451C	Q37R	-0.5
1BVC	V114A	-0.31	1RN1	A21C	0.74	2LZM	R119A	0.17	1WQ5	G211W	0.5	451C	V13M	-0.4
1BVC	V68T	0.2	1RN1	A21D	-0.33	2LZM	R119E	0	1WQ5	P132G	0.78	4LYZ	A31I	-1.4
1BVC	W14F	0.5	1RN1	A21E	-0.05	2LZM	R119M	-0.1	1WQ5	P57A	0.04	4LYZ	A31L	-1.8
1BVC	W7F	-0.1	1RN1	A21H	0.17	2LZM	R154E	-0.2	1WQ5	P62A	0.52	4LYZ	A31V	-1.2
1C2R	K32E	-0.2	1RN1	A21I	0.44	2LZM	R80K	0.17	1WQ5	Y175C	0.1	4LYZ	D101E	0
1C9O	E12K	0.1	1RN1	A21K	0.51	2LZM	S117A	-1.16	1YCC	C102A	-2.9	4LYZ	D101F	-0.72
1C9O	E21A	0.29	1RN1	A21L	0.13	2LZM	S117F	-1.2	1YCC	C102S	-2.8	4LYZ	D101G	-0.45
1C9O	E21K	-0.17	1RN1	A21M	0.15	2LZM	S117I	-1.7	1YCC	F82Y	0.7	4LYZ	D101K	-0.19
1C9O	E36K	0.19	1RN1	A21N	-0.34	2LZM	S117V	-2	1YCC	H33P	0.8	4LYZ	D101N	-0.04
1C9O	E50K	-0.26	1RN1	A21Q	0.26	2LZM	S38A	0.77	1YCC	K73I	0.4	4LYZ	D101Q	0.08
1C9O	G23Q	0.1	1RN1	A21R	0.41	2LZM	S38D	0.1	1YCC	K73V	-0.1	4LYZ	D101R	-0.27
1C9O	H29E	0.29	1RN1	A21S	0.4	2LZM	S38N	0	1YCC	N52G	-0.9	4LYZ	D101S	-0.87
1C9O	L66E	1.24	1RN1	A21V	0.66	2LZM	S44A	-0.34	1YCC	N52I	-2.02	4LYZ	F34Y	-0.19
1C9O	N11S	-0.22	1RN1	A21W	0.3	2LZM	S44C	0.11	1YCC	N52L	-2.56	4LYZ	F3Y	0.45
1C9O	N55K	-0.1	1RN1	A21Y	0.39	2LZM	S44D	0.11	2RN2	R41C	-0.12	4LYZ	G102A	-0.02
1C9O	Q2L	-0.29	1RN1	D49H	-1.1	2LZM	S44E	0	2RN2	S68G	2.4	4LYZ	G102R	-0.38
1C9O	Q53E	0	1RN1	E58A	0.77	2LZM	S44F	-0.06	2RN2	S68L	0.5	4LYZ	G102V	0.04
1C9O	R3A	1.05	1RN1	H92A	0.62	2LZM	S44G	0.53	2RN2	S68V	-0.6	4LYZ	G49N	0.96
1C9O	R3K	0.14	1RN1	N36A	0	2LZM	S44H	-0.04	2RN2	V74I	-0.6	4LYZ	G71A	0.38
1C9O	R3L	0.22	1RN1	N81A	2.87	2LZM	S44I	-0.31	2RN2	V74L	-1	4LYZ	I55A	2.77
1C9O	R56E	-0.24	1RN1	N9A	0.71	2LZM	S44K	-0.2	2TRX	L79C	-2.7	4LYZ	I55L	0.45
1C9O	S24D	-0.19	1RN1	S17A	-0.57	2LZM	S44L	-0.39	2TRX	T77C	-2.1	4LYZ	I55M	2.27
1C9O	T31S	-0.17	1RN1	Y42F	-1.14	2LZM	S44N	0.14	2WSY	A18G	0.3	4LYZ	I55V	0
1C9O	V64T	0.19	1RN1	Y42W	0.14	2LZM	S44Q	-0.27	2WSY	A18V	-0.2	4LYZ	M12F	-0.28
1C9O	Y15F	0.05	1RN1	Y56F	0.71	2LZM	S44R	-0.24	2WSY	I232V	0.5	4LYZ	N103D	-0.24
1CAH	C206S	-0.1	1RN1	Y57F	0.44	2LZM	S44T	-0.01	2WSY	L209V	0.2	4LYZ	N77H	0.38
1CAH	I256C	0.3	1ROP	D30A	-0.3	2LZM	S44V	-0.1	2WSY	Y175Q	0.9	4LYZ	Q121H	0.45
1CAH	L60C	0.1	1ROP	D30C	-0.8	2LZM	S44W	-0.05	2ZTA	H19R	-0.22	4LYZ	R114H	-0.68
1CAH	S56C	0.5	1ROP	D30E	-1	2LZM	S44Y	-0.19	3BLS	Y150F	0.7	4LYZ	R21Q	0.15
1CAH	W123C	0	1ROP	D30G	-2	2LZM	T109D	0.3	3HHR	E74D	0.55	4LYZ	R73K	-0.23

PDB ID	Mut	ΔΔG	PDB ID	Mut	ΔΔG	PDB ID	Mut	ΔΔG	PDB ID	Mut	ΔΔG	PDB ID	Mut	ΔΔG
1CAH	W16F	-0.3	1ROP	D30H	-0.9	2LZM	T109N	0	3HHR	S71T	-0.33	4LYZ	S91A	0.15
1CAH	W5F	-0.1	1ROP	D30I	0.8	2LZM	T115E	0	3MBP	D55N	0	4LYZ	S91D	2.31
1CDC	A40G	-0.72	1ROP	D30K	-0.9	2LZM	T151S	-0.39	3MBP	P133A	0.3	4LYZ	S91T	-0.99
1CDC	I18V	-1.17	1ROP	D30L	0.1	2LZM	T152C	0.5	3MBP	P159A	1.8	4LYZ	S91V	0.08
1CDC	I57V	-0.43	1ROP	D30M	-0.6	2LZM	T152I	0.4	3MBP	P159S	2.1	4LYZ	T40S	0.27
1CDC	L16V	-2.01	1ROP	D30N	-0.8	2LZM	T152V	-0.2	3MBP	P48A	-0.5	5CRO	A36S	-0.4
1CDC	L50V	0.37	1ROP	D30Q	-1.8	2LZM	T157A	0.5	3MBP	P48S	-0.3	5CRO	Q16L	-2.8
1CDC	L89V	-0.14	1ROP	D30R	-0.8	2LZM	T157D	1.1	3MBP	T345I	-0.7	5CRO	Y26C	-2.2
1CDC	L95V	-0.75	1ROP	D30S	-1	2LZM	T157I	1.2	3SSI	M103I	1.89	5CRO	Y26D	-2.7
1CDC	V78A	-3.01	1ROP	D30V	0.4	2LZM	T157R	-0.3	3SSI	M103L	-0.15	5CRO	Y26F	-0.4
1CSP	A46E	-0.02	1ROP	D30W	0.4	2LZM	T26S	-0.57	3SSI	M103V	1.35	5CRO	Y26H	-1.9
1CSP	A46K	-1.41	1ROP	D30Y	-0.2	2LZM	T34A	0.2	3SSI	M73A	-0.13	5CRO	Y26L	-1.1
1CSP	D24K	0.36	1ROP	L41V	2.53	2LZM	T59D	0.9	3SSI	M73D	-1.31	5CRO	Y26Q	-1.4
1CSP	D24N	0.45	1RRO	P21A	0.74	2LZM	T59N	0.6	3SSI	M73E	-0.35	5CRO	Y26V	-0.9
1CSP	D25K	1.08	1RRO	P26A	0.74	2LZM	T59S	0.2	3SSI	M73G	0.06	5CRO	Y26W	0.1
1CSP	D25Q	0.41	1RTB	A109G	0.43	2LZM	V103I	0.5	3SSI	M73I	0.73	8TIM	K193A	-1

Supplementary Information 1: Part of the general script that was used for the coordination and download of the different mutations using different predictors (only shown for MAESTRO (webserver), IPTREESTAB (webserver) and MUPRO (stand-alone).

A) Coding example of the general script:

```
#!/user/bin/python

##########################################
#                                        #
#         General Script Database        #
#                                        #
##########################################

##
import re
import sys
import os
from csv import reader
from Bio.PDB import *
import numpy as np
import pandas as pd

######################################################
# 1. Definition of the necessary functions           #
######################################################

### Input file is the csv with the data from ProTherm

input_file = open(str(sys.argv[1]),"rt")
lines = []
# read csv file as a list of lists with all the mutations in the database
with open(sys.argv[1],'rt') as input_file:
    # pass the file object to reader() to get the reader object
    csv_reader = reader(input_file,delimiter="\t")
    # Pass reader object to list() to get a list of lists
    lines = list(csv_reader)

lines=lines[1::] # Eliminate the header

print("\n")
print("We select chain A by default, in case we want to diferenciate chains, we should
change that argument.")
print("\n")

# function to extract the input information from the database
def extract_real_data(i):
        print("Extraction of data:")
        pdb_pos = i[0]
        pdb = pdb_pos.split(",")[0]
        mutation = i[2]
        chain = i[1]
        newres = mutation[-2]
        wt = mutation[0]
        position = re.findall("\d+",mutation)[0]
        ddg_real = i[3]
        ph = i[4]
        temp = i[5]
        print("PDB: "+str(pdb)+"\n"+"Chain: "+str(chain)+"\n"+"Real DDG value (kcal/mol):
"+str(ddg_real)+"\n"+"Temperature: "+str(temp)+" ยบC\n"+"pH: "+str(ph)+"\n")
        return [pdb,chain,wt,position,newres,ddg_real,ph,temp]

### Download and move to directory pdb and fasta files

# Assign paths
fasta_files_path="./fasta_files"
pdb_files_path="./pdb_files"
script_path = "./scripts"


# function to download and save PDB and FASTA files for each different protein
# creation of a index dictionary to avoid redundant downloads
dic_pdb_seq = {}

def get_pdb_fasta(pdb):
    pdb_files = os.listdir(pdb_files_path)
    if pdb+".pdb" not in pdb_files:
        ## PDB"
        os.system("wget -O - https://files.rcsb.org/download/"+str(pdb)+".pdb >
"+str(pdb)+".pdb")
        os.system("mv "+pdb+".pdb "+pdb_files_path)
        ## fasta
        os.system("wget -O - https://files.rcsb.org/download/"+pdb+".pdb 2>/dev/null
| python3 -c \"import sys; from Bio import SeqIO; SeqIO.convert(sys.stdin, 'pdb-atom',
sys.stdout, 'fasta')\" > "+pdb+".fasta")
        os.system("mv "+str(pdb)+".fasta "+fasta_files_path)
        pdb_files.append(pdb)
        with
open("/home/anarobmr/Documentos/all_programs_database/fasta_files/"+pdb+".fasta",'rt')
as input_file:
            lines = [line.strip() for line in input_file]
        chains = ""
        for line in lines:
            if line[0]!=">":
                chains+=line
        sequence = list(chains)
        dic_pdb_seq[pdb] = sequence
        return pdb_files
    else:
        return pdb_files

#### IPTREESTAB

## This function is going to generate the data necessary to the webserver request. From
a FASTA file it extracts the sequence and builds a diccionary with the pdb code as key.

# Path to the individual iptreestab request script
iptreestab_script =
"/home/anarobmr/Documentos/all_programs_database/scripts/iptreestab_script.py"

translation = {'CYS': 'C', 'ASP': 'D', 'SER': 'S', 'GLN': 'Q', 'LYS': 'K',
     'ILE': 'I', 'PRO': 'P', 'THR': 'T', 'PHE': 'F', 'ASN': 'N',
     'GLY': 'G', 'HIS': 'H', 'LEU': 'L', 'ARG': 'R', 'TRP': 'W',
     'ALA': 'A', 'VAL':'V', 'GLU': 'E', 'TYR': 'Y', 'MET': 'M'}

def iptreestab_single_request(iptreestab_script,data,structure):
    pdb = data[0]
    chain = data[1]
    wt = data[2]
    pos = data[3]
    mut_to = data[4]
    ph = data[6]
    temp = data[7]
    ### Bio.PDB from Biopython; we extract the residue from the fasta file
    res_list = structure.get_residues()
    # We try to extract the three aminoacids in three code letter before and after the
mutation.
```

```
    # Sometimes, FASTA file contains positions does not agree with real
    # positions. For that, we manage PDB files and bio.PDB where we can call specific
positions.
    try:
        wt = translation[(structure[0][str(chain)][int(pos)]).get_resname()]
        b3 = translation[(structure[0][str(chain)][int(pos)-3]).get_resname()]
        b2 = translation[(structure[0][str(chain)][int(pos)-2]).get_resname()]
        b1 = translation[(structure[0][str(chain)][int(pos)-1]).get_resname()]
        a1 = translation[(structure[0][str(chain)][int(pos)+1]).get_resname()]
        a2 = translation[(structure[0][str(chain)][int(pos)+2]).get_resname()]
        a3 = translation[(structure[0][str(chain)][int(pos)+3]).get_resname()]

        # Execution of the individual script for IPTREESTAB using OS library.
        iptreestab_dic = os.system("python3 "+iptreestab_script+" "+pdb+" "+chain+"
"+wt+" "+str(pos)+" "+mut_to+" "+ph+" "+temp+" "+b3+" "+b2+" "+b1+" "+a1+" "+a2+" "+a3)
        return True
    except:
        # In case of error, we return False and continue with the script
        return False

# This function extract useful information once it is request and downloaded.
def extract_iptreestab_data(dic_line):
    # IPTREESTAB individual script has created the output file
"result_iptreestab.output"
    input_file_iptreestab =
open("/home/anarobmr/Documentos/all_programs_database/result_iptreestab.output","rt") #
open file
    input_file_iptreestab.readline() # read file
    ddg_values_iptreestab = (-1)*float((input_file_iptreestab.readline()).split()[5]) #
Extract predicted DDG value
    dic_line['IPTREESTAB'] = float(ddg_values_iptreestab) # Add to the dictionary for
this mutation the corresponding value for this predictor
    os.system("rm
/home/anarobmr/Documentos/all_programs_database/result_iptreestab.output") # Remove the
file
    return dic_line    # Return the dictionary

mupro_script = "/home/anarobmr/Documentos/all_programs_database/scripts/mupro_script.py"
def mupro_single_request(mupro_script,data,structure):
    pdb = data[0]
    chain = data[1]
    wt = data[2]
    pos = data[3]
    res_list=Selection.unfold_entities(structure, "R")
    pos_initial = (res_list[0]).id[1]
    pos = (int(pos) - int(pos_initial)) + 1
    mut_to = data[4]
    ph = data[6]
    temp = data[7]
    mupro_dic = os.system("python3 "+mupro_script+" "+pdb+" "+chain+" "+wt+"
"+str(pos)+" "+mut_to+" "+ph+" "+temp)
    return mupro_dic

def extract_mupro_data(dic_line):
    input_file_mupro =
open("/home/anarobmr/Documentos/all_programs_database/muproresultadofinal.txt","rt")
    input_file_mupro.readline()
    ddg_values_mupro = (-1)*float((input_file_mupro.readline()).split()[0])
    dic_line['MUPRO'] = float(ddg_values_mupro)
    os.system("rm
/home/anarobmr/Documentos/all_programs_database/muproresultadofinal.txt")

    return dic_line


### MAESTRO
maestro_script
="/home/anarobmr/Documentos/all_programs_database/scripts/maestro_script.py"

def maestro_single(maestro_script,data):
    pdb = data[0]
    chain = data[1]
    wt = data[2]
    pos = data[3]
    mut_to = data[4]
    ph = data[6]
    temp = data[7]
    maestro_dic = os.system("python3 "+maestro_script+" "+pdb+" "+chain+" "+wt+" "+pos+"
"+mut_to+" "+ph+" "+temp)
    return maestro_dic

def extract_maestro_data(dic_line):
    input_file_maestro =
open("/home/anarobmr/Documentos/all_programs_database/maestro.output","rt")
    input_file_maestro.readline()
    ddg_values_maestro = float((input_file_maestro.readline()).split()[5])
    dic_line['MAESTRO'] = float(ddg_values_maestro)
    os.system("rm /home/anarobmr/Documentos/all_programs_database/maestro.output")

    return dic_line

######################################################
# 2. MAIN PROCESS INTEGRATION                        #
######################################################

# Creation of two empty list that will be the future dataframe indexes
all_dics = []
all_mutations = []

counter = 0
os.system("touch
/home/anarobmr/Documentos/all_programs_database/result"+str(counter)+".csv")
for line in lines:
    # For each mutation in the dataset, we create an empty dictionary which will be an
empty descriptor template
    dic_line = {'real':0,'I-
Mutant':0,'MAESTRO':0,'INPS':0,'CUPSAT':0,'MUPRO':0,'FoldX':0,'AUTO-SVM':0,'AUTO-
RT':0,'IPTREESTAB':0,'EVOEF':0}
    data = extract_real_data(line) # Data for the first line of the csv_reader

    pdb = data[0]
    mutation = pdb+"."+data[2]+data[3]+data[4] # The name of each row is
PDB.WT+POSITION+MUTATION, (for example: 8TIM.S27A)

    all_mutations.append(mutation) # To create the index of the Dataframe
    dic_line['real'] = float(data[5])

    get_pdb_fasta(pdb) # DOWNLOAD FASTAS AND PDB AND SAVE THEM IN FOLDERS. IT ALSO
UPDATE A LIST WITH ALL THE PDBs
    p = PDBParser()
    structure=p.get_structure("X",
"/home/anarobmr/Documentos/all_programs_database/pdb_files/"+pdb+".pdb")

    result = iptreestab_single_request(iptreestab_script,data,structure)
    print("---->> IPTREESTAB request done. \n \n")
    if result == False:
        print("Iptreestab cannot calculate the result.")
    else:
        dic_line = extract_iptreestab_data(dic_line)
        print("---->> IPTREESTAB data added to Dataframe. \n \n")

    mupro_single_request(mupro_script, data,structure)
    print("---->> MUPRO request done. \n \n")
    dic_line = extract_mupro_data(dic_line)
    print("---->> MUPRO data added to Dataframe. \n \n")
    maestro_single(maestro_script,data)
    print("---->> MAESTRO request done. \n \n")
    dic_line = extract_maestro_data(dic_line)
    print("---->> MAESTRO data added to Dataframe. \n \n")


    counter+=1
    print("\n###########################################################")
    print("\n"+str(counter)+" of "+str(len(lines))+" have been calculated.\n")
    print("#############################################################")

    all_dics.append(dic_line) # To later create the content of the Dataframe

    print(all_dics)
    # In each mutation we update the final output result and delete the previous one.
    # This is a save way of keep the information in case of something interrupt the
overall process
    df = pd.DataFrame(all_dics, index = all_mutations)
    df.to_csv('resultado'+str(counter)+'.csv',sep=';')
    os.system("rm 
/home/anarobmr/Documentos/all_programs_database/resultado"+str(counter-1)+".csv")

    ## The end of the main function will be a list with all the values ddg of the 
different predictors
```

B) Coding example of the IPTREESTAB individual script:

```
#!/user/bin/python

##########################################
#                                        #
#       IPTREESTAB Script Database       #
#                                        #
##########################################

from selenium import webdriver
from selenium.webdriver.common.keys import Keys
from selenium.webdriver.support.ui import Select # Select from a desplegable menu
from selenium.webdriver.support.ui import WebDriverWait
from selenium.webdriver.common.action_chains import ActionChains
from selenium.webdriver.chrome.options import Options # hide browser
from bs4 import BeautifulSoup
from selenium.common.exceptions import NoSuchElementException # Avoid error 
NoSuchElementException when the link is not found

import time
import os
import re
import sys

window_size = "1920,1080"

def internet_browser(pdb,mut_from,pos,mut_to,temp_value,ph_value,b3,b2,b1,a1,a2,a3): 
#The arguments varies for each predictor internet browser

    # Initialize a Firefox webdriver
    chrome_options = Options()
    chrome_options.add_argument("--headless")
    chrome_options.add_argument("--window-size=%s" % window_size)
    chrome_options.add_argument("start-maximized"); # 
https://stackoverflow.com/a/26283818/1689770
    chrome_options.add_argument("enable-automation"); # 
https://stackoverflow.com/a/43840128/1689770
    chrome_options.add_argument("--no-sandbox"); 
#https://stackoverflow.com/a/50725918/1689770
    chrome_options.add_argument("--disable-infobars"); 
#https://stackoverflow.com/a/43840128/1689770
    chrome_options.add_argument("--disable-dev-shm-usage"); 
#https://stackoverflow.com/a/50725918/1689770
    chrome_options.add_argument("--disable-browser-side-navigation"); 
#https://stackoverflow.com/a/49123152/1689770
    chrome_options.add_argument("--disable-gpu")
    try:
        driver = webdriver.Chrome(chrome_options=chrome_options)
        # Grab the web page
        driver.get("http://203.64.84.190:8080/IPTREEr/input.jsp")
        entervalue_service = 
driver.find_element_by_xpath("/html/body/center/font/form/input[2]")
        entervalue_service.click()

        # Second page; Fullfill the required information
        # they are dropdown options: search, select and enter the corresponding 
information:
        mutationfrom = 
Select(driver.find_element_by_xpath("/html/body/center/font/form/select[1]"))
        mutationfrom.select_by_value(mut_from)

        mutationto = 
Select(driver.find_element_by_xpath("/html/body/center/font/form/select[2]"))
        mutationto.select_by_value(mut_to)
        before_3 =
Select(driver.find_element_by_xpath("/html/body/center/font/form/select[3]"))
        before_3.select_by_value(b3)

        before_2 =
Select(driver.find_element_by_xpath("/html/body/center/font/form/select[4]"))
        before_2.select_by_value(b2)

        before_1 =
Select(driver.find_element_by_xpath("/html/body/center/font/form/select[5]"))
        before_1.select_by_value(b1)

        after_1 =
Select(driver.find_element_by_xpath("/html/body/center/font/form/select[6]"))
        after_1.select_by_value(a1)

        after_2 =
Select(driver.find_element_by_xpath("/html/body/center/font/form/select[7]"))
        after_2.select_by_value(a2)

        after_3 =
Select(driver.find_element_by_xpath("/html/body/center/font/form/select[8]"))
        after_3.select_by_value(a3)

        if ph_value!=str(7):
            ph = driver.find_element_by_xpath("/html/body/center/font/form/input[3]")
            ph.clear()
            ph.send_keys(ph_value)
        if temp_value!=str(25):
            temp = driver.find_element_by_xpath("/html/body/center/font/form/input[4]")
            temp.clear()
            temp.send_keys(temp_value)

        search_button = driver.find_element_by_name("submit")
        search_button.click()
        # Third page
        ddg_value = driver.find_element_by_xpath("/html/body/font/p[4]/span")
        ddg = ddg_value.text.strip().split()[0]

        #result.append(link)
        output_file.write(str(pdb)+" "+str(pos)+" "+str(mut_to)+" "+str(temp_value)+"
"+str(ph_value)+" "+str(ddg)+"\n")

        driver.close()
    except NoSuchElementException:
    # In case there were some problem, we put a DDG value of 0 and reported the error.
        output_file.write(str(pdb)+" "+str(pos)+" "+str(mut_to)+" "+str(temp_value)+"
"+str(ph_value)+" "+"0"+"\n")
        print(pdb+" "+mut_from+" -> "+pos+mut_to+" string "+b3+b2+b1+mut_from+a1+a2+a3)
        pass

######################################################
# 1. Preparation of the output file.                 #
######################################################

output_file=open("/home/anarobmr/Documentos/all_programs_database/result_iptreestab.outp
ut","wt")
output_file.write("# PDB POS NEWRES TEMP PH DDG \n")
lines_mutations = []

######################################################
# 2. Classfication of the input information, variable#
# creation.                                          #
######################################################

# Input information classification
pdb=sys.argv[1]
chain = sys.argv[2]
mut_from=sys.argv[3]
pos=sys.argv[4]
mut_to=sys.argv[5]
temp_value=sys.argv[7]
ph_value=sys.argv[6]
b3=sys.argv[8]
b2=sys.argv[9]
b1=sys.argv[10]
a1=sys.argv[11]
```

```
a2=sys.argv[12]
a3=sys.argv[13]

######################################################
# 3. Internet browser function                       #
######################################################

internet_browser(pdb,mut_from,pos,mut_to,temp_value,ph_value,b3,b2,b1,a1,a2,a3)
output_file.close()
```

Supplementary information 2: Example of the automatization of the request of all possible mutations for a protein. This example uses the stand-alone functionality MUPRO:

```
#!/user/bin/python

##########################################
#                                        #
#     Calculation of all mutations       #
#               Python MUPRO             #
#                                        #
##########################################

import re
import sys
import os

# This option is for the protein in study.
# the name of the PDB code for your protein
pdb_name = sys.argv[1]

# name of the fasta file  as input file for MUPRO
input_file = sys.argv[2]

# chain")
chain = sys.argv[3]

# the initial position (1 or another in case of fragments of proteins)
initial_position = sys.argv[4]

# the temperature (°C)")
temp = sys.argv[5]

# the pH
ph = sys.argv[6]

# Resume of the input data:
print("The input file is "+str(input_file)+"\n")
print("\n")
print("PDB: "+str(pdb_name)+"\n"+"Chain: "+str(chain)+"\n"+"Initial position:
"+str(initial_position)+"\n"+"Temperature: "+str(temp)+" °C\n"+"pH: "+str(ph)+"\n")

# Create tree of folders
print("\n")
print("Creation of folders and directories: mutations and outputs")
print("\n")
os.system("mkdir mutations")
os.system("mkdir outputs")

# Assign paths
mutations_path="./mutations"
outputs_path="./outputs"

######################################################
# 1. Read fasta file to obtain the protein sequence. #
######################################################

with open(input_file,'rt') as input_file:
    # input_file is a fasta file
    lines = [line.strip() for line in input_file]

# Creation of a empty string to join all the aa.
```

```
sequence=""
for line in lines:
    if line[0]!=">":
        sequence+=line

######################################################
# 2. Generation of all possible mutation files.      #
######################################################
i=0
position=int(initial_position)
longitud=len(sequence)

for wt in sequence:

output_file=open(mutations_path+"/"+str(pdb_name)+"_"+str(chain)+"_systematic_mutation_"
+str(wt)+"_"+str(position)+".txt","wt")
    output_file.write(str(pdb_name)+str(chain)+"\n")
    output_file.write(str(sequence)+"\n")
    output_file.write(str(position)+"\n")
    output_file.write(str(wt)+"\n")
    output_file.write("*")
    output_file.close()
    i+=1
    position+=1

input_file.close()

######################################################
# 3. MUPRO calculations with all the files.          #
######################################################
filelist = os.listdir(mutations_path)

mupro_script_systematic = "/home/anarobmr/Escritorio/mupro1.1/bin/predict_regr_all.sh"

for f in filelist:
    os.system("bash "+ mupro_script_systematic +" "+mutations_path+"/"+f+" >
"+outputs_path +"/"+"ddg_values_" + f)
    print(outputs_path +"/"+"ddg_values_" + f+" has been succesfully generated.")

# Deletion of the  temporaly mutation files
os.system("rm -r mutations")

###################################################################
# 4. Merge all the outputs files and extraction of the data.      #
###################################################################

filelist_outputs = os.listdir(outputs_path)

print("\n")
print("Generation of the final output as resultadofinal.txt ...")
print("\n")

final_output = open("mupro_"+str(pdb_name)+".predictions","wt")
final_output.write("PDB CHAIN WT POS NEWRES DDG (kcal/mol) \n")

for f in filelist_outputs:
    # Open each file and extract all lines.
    with open(outputs_path+"/"+f,"rt") as f_in:
        lines = (line.rstrip() for line in f_in)
        lines = list(line for line in lines if line) # we eliminate empty lines.
    # From the name of the file we extract some data: pdb, chain, wt and position.
    file1 = f.split("_")
    pdb = file1[2]
    chain = file1[3]
    wt = file1[6]
    pos = int(re.findall("\d+",file1[-1])[0])
    # Lines with data (start with AA) are processed to extract ddg.
    for line in lines:
        if line.split()[0]=="AA":
            newres=line.split()[2]
            ddg=line.split()[6]
            linea = [pdb,chain,wt,str(pos),newres,ddg]
            # Final result is written.
            final_output.write(" ".join(linea)+"\n")

    print(f+" data has been succesfully extracted and saved in "+str(final_output))
```

```
# Elimination of the  temporaly outputs files
os.system("rm -r outputs")
final_output.close()
```

T