Prediction of the Effect of Point Mutations on Drug-Protein Binding Affinity

Joan Cabot March

MU Bioinf. i Bioest.
Drug Design and Molecular Biology

Supervisor
Jordi Mestres

Responsible Professor
Jorge Valencia Delgadillo

Fecha Entrega
01/2023

FICHA DEL TRABAJO FINAL

Title:	*Prediction of the Effect of Point Mutations on Ligand-Protein Binding Affinity*
Author:	*Joan Cabot March*
Supervisor:	*Jordi Mestres*
Responsible Professor:	*Jorge Valencia Delgadillo*
Submission Date (mm/aaaa):	*01/2023*
Program:	*Joint University Master's Degree in Bioinformatics and Biostatistics (UOC, UB)*
Final Project's Area:	*Drug Design and Molecular Biology*
Language:	*English*
Keywords:	*Mutation, Binding, Polymorphism*

Resumen del Trabajo

En este proyecto se han estudiado 5 metodologías diferentes para la predicción de cambios de afinidad entre fármacos y proteínas debido a polimorfismos puntuales en el ADN. Estas variaciones individuales en el genoma podrían explicar la disparidad de respuesta interindividual a fármacos y, por ende, su estudio presenta gran interés en el ámbito de la medicina personalizada.

Para proceder a realizar las predicciones se apostó por un enfoque totalmente computacional en el que a partir de la estructura 3D de la proteína sin mutar se aplicaban los métodos descritos a continuación para obtener las predicciones.

Las técnicas estudiadas han sido: Molecular Docking con dos metodologías distintas para aplicar la mutación, dos tipos de Machine Learning como son un clasificador y un regresor y, finalmente una técnica conocida como protocolo flex_ddG usando Rosetta como base. Esta última técnica se basa en potenciales que mezclan modelos físicos y de conocimiento.

Los resultados obtenidos con estas predicciones fueron comparados con bases de datos que contenían información sobre la afinidad del complejo proteína-ligando sin mutar y una vez mutado con el objetivo de evaluar los modelos.

En resumen las técnicas que mejores resultados dieron con el dataset testeado fueron el Molecular Docking y el clasificador de Machine Learning que, en general dieron buenos resultados aunque siempre sacrificando exhaustividad o precisión respectivamente.

Abstract

In this project, 5 different methodologies were studied for the prediction of affinity changes between drugs and proteins due to single nucleotide polymorphisms (SNPs) in DNA. These individual variations in the genome could explain the disparity in the interindividual response to drugs and, therefore, their study is of great interest in the field of personalized medicine.

In order to perform said predictions, a fully computational approach was used in which, based on the 3D structure of the unmutated (Wild Type) protein the methods described below were applied to obtain the predictions.

The techniques studied were: Molecular Docking with two different methodologies to apply the mutation, two types of Machine Learning, those being a classifier and a regressor and, finally, a technique known as the flex_ddG protocol using Rosetta. This last technique uses mixed physics and knowledge-based potentials in order to calculate the change in free energy.

The results obtained with these predictions were compared against databases containing information on the affinity of the Wild Type and mutated protein-ligand complex in order to evaluate the models.

In summary, the techniques that gave the best outcome with the tested dataset were Molecular Docking and the Machine Learning classifier, which in general gave good results, although always sacrificing recall or precision respectively.

Table of Contents

Table of Figures

1. Introduction

1.1. Background and Rationale

During drug development, safety is tested in different phases. In pre-clinical studies, the primary goal of safety evaluation is the identification of a safe dose in humans and of safety parameters for clinical monitoring.
Although drug safety evaluation is very rigorous and thorough, pre-marketing clinical trials have intrinsic limitations that do not allow exhaustively evaluating drug safety profiles [1]. These studies are conducted on a limited number of patients that are selected based on strict eligibility criteria not fully representing real-world populations and have limited duration. All of this ends up preventing detection of rare and long-term adverse reactions. In addition, studies have shown that marketed drugs have not been as effective as expected for 40-70% of patients, with clinical practice showing them to have insufficient efficacy [2].

This large variation in drug response has prompted the apparition of pharmacogenomics. This field studies the basis of inter-individual genetic differences and their effect on drug response, including both efficacy and adverse [3]. Studies have shown that most genetic variants in drug-related genes are very rare ($f < 0.1\%$) and thus unlikely to be observed in clinical trials. However, around 80% of patients carry at least one functional variant in the drug targets of the top 100 commonly prescribed drugs in the United States [4].

On the other hand, some drugs are not capable of entering the market due to some severe but rare adverse reaction. This side effect could be potentially caused by a problematic mutation on a secondary target protein and therefore most people would be impervious to this reaction and the drug perfectly good to use when no better alternatives exist. Among these genetic variations, SNPs (Single Nucleotide Polymorphisms) have been wildly correlated with changes in drug response and toxicity [5][6][7] .The variation in drug-response at the protein level and its underlying mechanisms are of significant interest in both the developing of new drugs and the the choice of drug prescription with an estimate of six SNPs affecting five different FDA-approved drugs carried by each individual [8] . Nowadays genetic profiling in clinical practice of drug response affecting genes (and their corresponding proteins) is mostly limited to targets of drugs with very narrow therapeutic windows such as cancer treatment drugs and a number of drug metabolizing enzymes such as cytochromes [9]. Unfortunately, large-scale experimental screening of ligand-binding against protein variants is still time-consuming and expensive. Alternatively, in silico approaches can play a role in guiding or even replacing those experiments. This is where this research may provide some valuable insights by testing different approaches in order to find the most appropriate method when trying to computationally predict changes in drug effectivity or safety.

1.2. Objectives

1.2.1 Main Objectives

1. Identifying mutations with potential to alter drug-target binding affinity.
2. Create models capable of associating location and amino acid change in order to predict increases or decreases in drug response.
3. Evaluating the models against experimental data.
4. Use these models to predict changes on a large number of mutations.

1.2.2 Specific Objectives

1.1 Obtaining a curated benchmark of genetic single point mutations and its experimental effect in drug binding affinity.
1.2 Obtaining the 3D model of the proteins and computationally introducing the corresponding mutations.
2.1 Using Docking, Machine Learning and a Rosetta knowledge-physics based protocol in order to predict the change in Drug-Target binding affinity.
3.1 Evaluate the predicted change in binding affinity against a benchmark consisting of experimental data.
3.2 Compile all this data and create models that can be used for any protein and single point mutation.
4.1 Use the aforementioned models to predict the effect of the mutation on drug response based on location and type of amino acid mutation.

1.3. Sustainability, Ethics and Diversity Impact

Pharmaceutical clinical trials have a history of not always being representative of all demographics [10]. That could be part of the reason why some drugs are later found to be less effective in certain ethnic groups. One of the most well known examples is that of angiotensin-converting enzyme (ACE) inhibitors which use in black populations was associated with poorer cardiovascular outcomes when compared to white populations

[11]. As of today there is no known cause of this disparity but one of the leading hypotheses is a difference in response caused by genetic differences.This project could help not just lead the way in discovering the cause of these differences in drug response but also in proposing alternatives to these populations.

1.4. Chosen Approach

In this thesis four distinct parts can be distinguished. Firstly a data recollection step was performed corresponding to the first two specific objectives. In this step, various databases containing data about the impact of mutations on ligand affinity were analyzed and finally the PLATINUM database [12] extracted from MdrDB [13] was chosen due to its basis on experimental data, in contrast to other databases that did not base their results on experimental measurements or whose entries did not come with associated protein structures as PDB files.

The second part consisted of using various computational techniques in order to make a prediction of the ligand's binding affinity change upon the event of a mutation. Corresponding to the third specific objective, three broad methods were chosen for this purpose: Docking, Machine Learning and Ensemble-Based estimation of changes in protein–ligand binding affinity upon a mutation. Other techniques were considered such as Molecular Dynamics or Steered Molecular Dynamics but were deemed too computationally and time consuming for the limited time available for this project. Nevertheless these discarded techniques have great potential and should be further studied.

The third part consisted of the evaluation of the performances of all generated models. This step corresponds to the fourth and fifth specific objectives and was performed using a combination of python and R. Pretty much any other programming language and data analysis software could have been used, but those were chosen due to the variety of libraries available and their focus on data science.

Finally, corresponding to the sixth specific objective, a use case example was done in which, starting from a list of SNPs in a person's genetic sequence a list of possibly affected drugs is returned based on the predictions from the previously tested models.

1.5. Work Plan

1.1 Obtain a curated benchmark of genetic single point mutations and their experimental effect in drug binding affinity.

Task 1: Download data from the MdrDB database.

1.2 Obtain the 3D model of the proteins and computationally introduce the corresponding mutations.

Task 1: Download Wild Type PDB files from the Protein Data Bank.
Task 2: Use PyMol to introduce the mutations as found in the database.
Task 3: Use FoldX to introduce the mutations as found in the database.

2.1 Use Docking, Machine Learning and a Rosetta knowledge-physics based protocol in order to predict the change in Drug-Target binding affinity.

Task 1: Molecular Docking with Qvina.
Task 2: Machine Learning using Random Forest Classifier and Gradient Boosting Regressor.
Task 3: Use Rosetta's Flex_ddG protocol.

3.1 Evaluate the predicted change in binding affinity against a benchmark consisting of experimental data.

Task 1: Compare the predicted values with the Experimentally determined ones.

3.2 Compile all this data and create models that can be used for any protein and single point mutation.

Task 1: Create a .csv containing all the predictions for all tested entries.
Task 2: Create scripts in order to facilitate the use of these models on further tests.

4.1 Use the aforementioned models to predict the effect of the mutation on drug response based on location and type of amino acid mutation

Task 1: Transform a human BAM file to a list of SNPs present.
Task 2: Get a list of protein and drugs that could be tested (Have adequate PDB file).
Task 3: Combine the 2 previous files and predict the effect of the mutation,

	PEC 1				PEC 2					PEC 3					Final thesis			Defense
	19/9-25/9	26/9-2/10	3/10-9-10	10/10-16/10	17/10-23/10	24/10-30/10	31/10-6/11	7/11-113/11	14/11-20/11	21/11-27/11	28/11-4/12	5/12-11/12	12/12-18/12	19/12-25/12	26/12-1/1	2/1-8/1	9/1-15/1	16/1-22/1
Objective 1.1: Task 1	■																	
Objective 1.2: Task 1		■	■															
Objective 1.2: Task 2			■	■														
Objective 1.2: Task 3			■	■														
Objective 2.1: Task 1					■	■	■											
Objective 2.1: Task 2						■	■	■	■	■	■							
Objective 2.1: Task 3						■	■	■	■	■	■	■	■	■				
Objective 3.1: Task 1								■	■	■	■	■	■	■	■			
Objective 3.2: Task 1														■	■	■		
Objective 3.2: Task 2							■	■	■	■	■	■	■	■	■			
Objective 4.1: Task 1						■	■											
Objective 4.1: Task 2							■	■										
Objective 4.1: Task 3														■	■	■		
Thesis Writing											■	■	■	■	■	■	■	
Presentation																■	■	
Public Defense																		■

1.6. Summary of Obtained Products

Thesis: Written document where the obtained results and conclusions found during the project execution are presented. The final document will contain the following sections: Methodology, Results, Discussion, Glossary, Bibliography and Supplementary Data.

Product: A file containing all the predictions from the different models for the tested entries as well as a script capable of performing predictions on new data.

Presentation: Slides and a 20 minute video describing the most interesting findings and conclusions obtained.

1.7. Brief Descriptions of the Other Chapters.

Background and Rationale: This section provides a brief description of the current situation and issues involving personalized medicine and the disparity in drug response based on genetic differences.

State of the Art: In this section we explore the most recent papers and research involving the computational techniques involving the prediction of the change of ligand binding affinity upon mutations in the target protein.

Materials and Methods: In this section the techniques used in this study are explained as well as the steps taken in order to obtain the data for this project.

Results: In this section the results obtained are shown.

Discussion and Future Work: In this section the results and the most interesting lines of work not explored in this project are discussed.

2. State of the art

This particular field, although small, is rapidly growing and new research continues to be published regularly. Some of the most relevant studies for this particular project as of January 2023 are:

Ammar, A., et al. [14] used molecular docking virtual screening techniques in order to create a database of protein-ligand complexes where the binding sites contained mutations based on the Uniprot variants dataset. The base protein-ligand 3D structures were obtained from the PDBbind database, mutations were applied using FoldX, an energy minimization step was performed using Gromacs and finally the protein-ligand docking was performed using AutoDock Vina. Even though this paper presents very interesting and promising methodologies there was no comparison with experimental data and, therefore, the accuracy of the results cannot be assessed. Furthermore, mutations outside of the binding site were not taken into account.

The same team [15] later on, tried using Machine Learning (ML) techniques such as Random Forest to predict the effect of binding-site mutations on protein-ligand binding affinity. The models were constructed using various protein, ligand, binding site and mutation descriptors and the best results they obtained were: RMSE value within 0.5-0.6 kcal/mol with an R2 value of 0.87-0.90 on an independent test set.
Despite those amazing results the study used the docking results of the previous model as “experimental binding affinities” and, just like the last paper, mutations outside of the binding site were not taken into account.

Yu, Y., et al. [16] used AMBER/20 Molecular Dynamics (MD) on a curated subset of the Platinum Database to predict the change in change of binding free energy (ΔΔG) kcal/mol with simulation run times between 20 and 100 ns, it achieved an accuracy of R=0.517 in systems involving multiple mutations compared with those involving only one mutation R=0.400.
This study achieved worse results than the Machine learning one, this may be caused firstly by the techniques used and secondly by the fact that it used actual experimental change in binding affinity data to test the predictions as well as also accounting for far away mutations.

Aldeghi, et al. [17] performed an analysis of various methodologies including Machine Learning, Molecular Dynamics and the Rosetta flex_ddG protocol [18], a modeling program that uses mixed physics and knowledge-based potentials to predict changes in binding affinity upon mutations for the cancer target Abl kinase.
This study attempted to numerically predict the change in binding affinity obtaining RMSE = 0.62 kcal/mol, R = 0.71. It also classified its predictions in resistant/non resistant mutation using a threshold of ΔΔGexp > 1.36 kcal/mol, i.e a 10-fold drop in

affinity. This model achieved an AUPRC of 0.62 and, at the conventional threshold of $\Delta\Delta$Gcalc > 1.36 kcal/mol, a recall of 0.79 and a precision of 0.48 on the consensus prediction of a MD using A99ff and Rosetta using R15. This study offered a wider look at the different methodologies available and how they compare between them but it was only tested on a specific protein type and there is no way of knowing how well, if at all, these methods will extrapolate to other protein families.

All in all, none of the previous studies were performed on a large dataset of experimentally determined ligand binding affinity changes upon mutation. Some did not use experimental data and some used a very specific set of proteins. Hence, this research aims to fill this gap, expand on the current knowledge and build models trained and tested on a relatively large dataset of protein variants and determine which method (Physics or Information based) performs better under these conditions.

3. Materials and Methods

3.1 Dataset

A dataset containing 1,008 experimentally tested mutations affecting ligand-protein binding affinities known as the PLATINUM database [12] extracted from the Mutation-induced drug resistance DataBase (MdrDB) [13] was filtered out to remove non SNP mutations, entries where no high quality crystallographic 3D structure was available for the wild type protein and entries where no experimental precise ΔΔG data was available. This left us with 635 unique entries. After this initial filter a number of entries were later discarded due to broken ligands, broken PDB files and incompatibilities with various programs and techniques. An analysis of the filtered dataset can be seen at Figure 1.

The dataset contains, among other parameters: PDB code of WT (Wild Type) protein structure, ligand chemical id, original residue, mutant residue, mutation location, protein chain, distance between mutation and ligand and experimental ΔΔG (change in change of free energy between WT and mutant protein/ligand complexes). A full list of PDB Codes in the dataset can be found in Table S4.

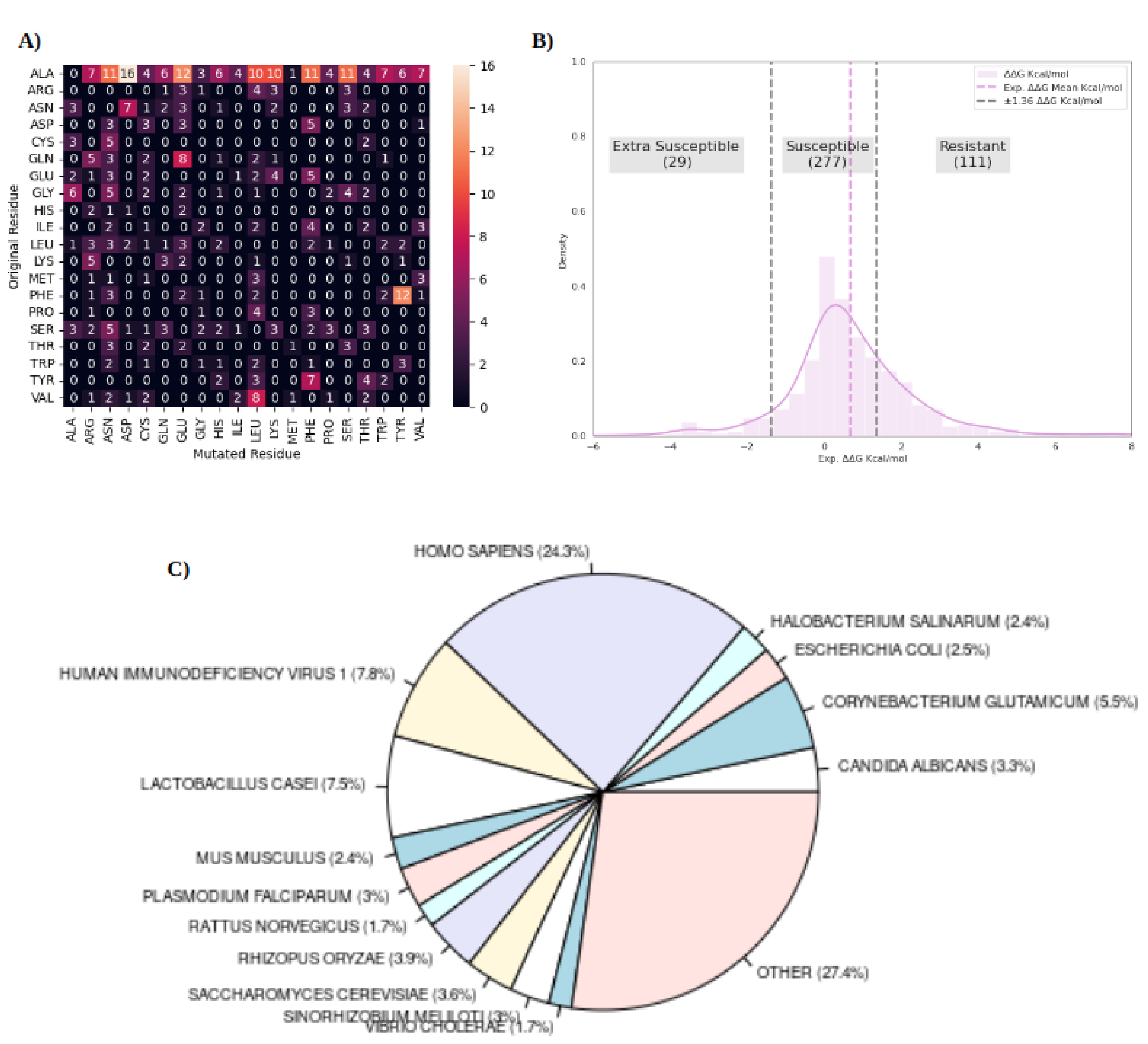

Figure 1: Analysis of the dataset entries. A) Heatmap of residue mutation matrix. B) Distribution of experimental ΔΔG of the dataset, the pink dashed line represents its mean and the dashed gray line represents the 1.36 ΔΔG threshold on which change/no change will be defined. It can be seen that most mutations tend to reduce pK. C) Species distribution of all entries in the dataset, it can be seen that Human samples represent a large portion of all mutations.

3.2 Docking

3.2.1 Dataset

The initial dataset of the PLATINUM database was further filtered by eliminating proteins containing molecular groups incompatible with QuickVina2 [19] such as the iron containing HEME group. The final number of entries was 429.

3.2.2 System Setup

The structures of the WT complexes were taken from the PDB (Protein Data Bank) [20]. Apo structures were generated by discarding the ligand atoms and crystallographic water molecules. Missing atoms in residues were mended and only the reported Chain was kept. All mutant structures were generated using the PyMol Open source [21] Mutagenesis tool or the FoldX BuildModel command having had the structure optimized using the FoldX Optimize command beforehand [22].

The PyMol Mutagenesis tool returns a list of possible rotamers for each mutated residue and an associated "strain" score based on van der Waals clashes. Therefore the most favorable, i.e less strain, rotamer was chosen for each mutation.

FoldX is an empirical force field that was developed for the rapid evaluation of the effect of mutations on the stability, folding and dynamics of proteins and nucleic acids. The core functionality of FoldX, namely the calculation of the free energy of a macromolecule based on its high-resolution 3D structure is therefore applied in order to mutate a chosen residue and minimize the energy of the surrounding amino acids in other to obtain the conformation with the minimum energy.

3.2.3 Docking Setup

The docking procedure and scoring were performed using QuickVina2 [19] referred to as Qvina from now on.

Qvina is a fast and accurate molecular tool based on AutoDock Vina capable of enhancing Vina's computation time via heuristics that prevent unnecessary local searches. Various docking parameters were tried and tested: {--nmodes: [10,15,20,30] --exhaustiveness: [9,20,50,150], --local: [Yes, No]}. Nmodes denotes the number of output conformations to be returned, exhaustiveness is a measure of the time spent on each conformational search and local limits the size of the search box. The best model found consisted of using --nmodes: 15 --exhaustiveness: 20 --local: Yes.

For each mutation four docking and scoring procedures were executed. Firstly the ligand was docked to the WT protein, from now on referred to as rigid docking. Secondly the ligand was docked to the WT protein but the sidechain from the residue where a mutation would be present was allowed to move, this method will be referred from now on as flexible docking. In third place the ligand was docked to the mutant protein (rigid docking). Lastly the ligand was docked to the mutant protein with the mutated residue's sidechain being able to freely move(flexible docking).

In order to evaluate the results both score and RMSD (Root Mean squared distance) between the crystallographic ligand pose and the docked one were calculated for all 15 docking poses for each run. The best docking pose was chosen based on the combination of those two parameters. Poses with a RMSD greater than 3 Angstroms were directly discarded as the docking procedure sometimes returned poses outside of the known binding site and, therefore, those cases were considered an artifact from a faulty docking technique and invalid.

Protein preparation, ligand preparation, and scoring function were performed in Python using the NumPy [23], RDKit [24], Pandas [25] libraries.

3.2.4 Model Evaluation

In order to evaluate the performance of the docking and scoring procedure the best score value (taking into account RMSD) returned by Qvina was evaluated against experimental ΔΔG values in Kcal/mol.

The change in binding affinity predicted was taken to be the difference between the WT score and the mutant protein score following this formula:

$$\Delta Score_{Rigid} = Score_{Rigid}^{WT} - Score_{Rigid}^{Mut}$$

$$\Delta Score_{Flexible} = Score_{Flexible}^{WT} - Score_{Flexible}^{Mut}$$

Then, in order to use this score to binarily classify change or no change in binding affinity it was assumed that an absolute change in experimental ΔΔG of ≥ 1.36 kcal/mol was significant enough to be considered that the binding affinity changed as it represents a 10-fold change in binding affinity. On the other hand it was considered that a percentual change of at least 1.5% in score between WT and mutant protein-ligand complexes was the limit between change and no change.

3.3 Machine Learning

3.3.1 Feature Descriptors

Based on a paper published by Sánchez-Cruz et al. [26] in which a number of molecular features were transformed into numerical descriptors known as Extended Connectivity Interaction Features (ECIF), an adapted technique was tested in order to provide descriptors to feed the machine learning algorithm.

All protein structures were used without additional treatment apart from removing the ligand and side chains. Standardizer was used to add explicit hydrogen atoms (in case they were missing) and to perceive aromaticity in an interpretable way for RDKit for all ligands. JChem, Standarizer, 20.11.0, 2020, ChemAxon (https://www.chemaxon.com) PyMol was finally used in order to introduce the mutations.

In total, 1723 descriptors were calculated for every mutant protein-ligand complex that can be divided into Protein-Ligand interaction, Ligand and Mutation descriptors. A full list of descriptors are available in Supplementary Data 1. Table S1,S2 and S3 respectively

3.3.1.1 Protein-Ligand Interaction Descriptors

Atom types in both ligand and protein were defined considering six atomic features: atom symbol, explicit valence, number of attached heavy atoms, number of attached hydrogens, aromaticity and ring membership. For the ligands, atom types were assigned as interpreted by RDKit (2021.09.4), from the corresponding standard data format (SDF) file. Regarding the proteins, atom types were manually assigned to 22 different atom types in a dictionary-based mapping according to their residue and atom label in the corresponding PDB structural file.

Once all atom types in both protein and ligand were assigned, the number of each possible pair of protein–ligand atom types was counted, with the only additional criterion being a predefined distance cutoff, in this case 6 Angstroms was chosen based on the result of the aforementioned paper [26]. This resulted in 1540 integer-valued features, where each position corresponds to the count of a pair-wise combination of protein–ligand atom types with preserved directionality. For instance, the atom-type pair ‘O;2;1;0;0:0’—‘N;3;2;1;0;0’ means that a protein oxygen atom with an explicit valence of 2, 1 attached heavy atom, without attached hydrogens, no aromaticity and no ring membership interacts with a ligand nitrogen atom with an explicit valence of 3, 2 attached heavy atoms, 1 attached hydrogen, no aromaticity and no ring membership.

3.3.1.2 Ligand Descriptors

Ligand descriptors were obtained from the RDKit (2021.09.4) Chem.Descriptors module, this resulted in 170 numerical features containing information about the ligand, for example, molecular weight, number of rotatable bonds, number of heteroatoms or Balaban’s J index. For more information on Protein-Ligand Interaction and Ligand Descriptors see Sánchez-Cruz et al. research [26]

3.3.1.3 Mutation Descriptors

Mutation Descriptors show the effect of the change in residue upon mutation. This resulted in 13 numerical features based on the change in physicochemical properties. A dictionary of residue properties was extracted from pepdata on github [27].

Additionally another feature, mutation distance from ligand, was calculated using SciPy's [28] distance function to calculate the minimum geometric distance between any atom from the original residue and its closest ligand atom. Some of the mutation features include hydropathy change, volume change, polarity change, accessible surface area change, etc.

3.3.2 Dataset Preparation

The features described in the previous sections were calculated for each of the entries in a curated version of the PLATINUM database; different splits were applied to create the datasets to be used in model training and validation. For preprocessing, the features that had zero variance were removed, in turn reducing the number of descriptors from 1723 to 838. Finally, all values were normalized using the min-max method in order to have a maximum value of 1 and a minimum of 0 for all features.

3.3.3 Data Splitting

This dataset contained a total of 85 different proteins corresponding to 64 different protein families according to PFAM. [29]

Having proteins in the training set that are similar to the ones in the test set can cause leakage issues to the model. Hence, splitting the dataset by protein similarity was used in order to help avoid such a situation. Accordingly no two entries with the same protein family id were allowed to be present in both train and test set. A full list of PDBs for the train, test and validation sets can be found in Table S6.

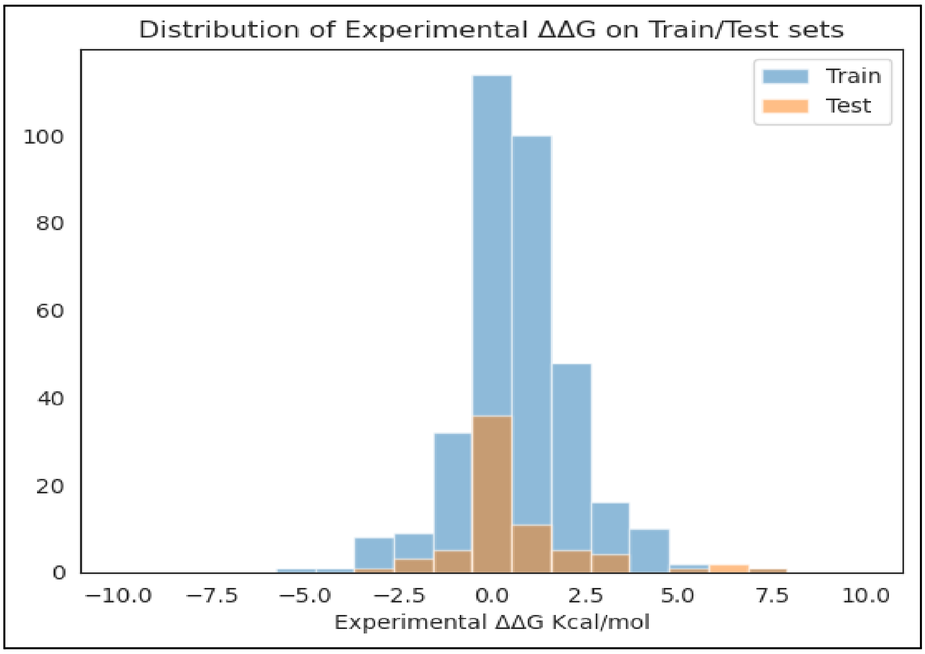

Figure 2: Comparison of experimental mutant ΔΔG distribution between Train and Test set, the blue color stands for the train set and the orange color stands for the test set.

Both train and test splits were divided in a way that a similar percent of samples (~30%, Train: 32.3%, Test: 28.6%) were classified as producing change in binding affinity in each split (Figure 2).

3.3.4 Machine Learning Modeling

As previously mentioned, two different approaches were tested. Firstly a classifier that simply tried predicting possible changes in binding affinity in a binary (Change/No Change) dichotomy. Secondly a regressor model that tried to quantitatively predict ΔΔG .

For the classifier a number of models were tested {Support Vector Machine, K Nearest Neighbors, Random Forest, Decision Tree, Naive Bayes} and in the end Random Forest showed the best performance and was therefore chosen. The following parameters were chosen after a grid search step {'max_depth': 25, 'max_features': 15, 'min_samples_split': 8, 'n_estimators': 32}.

The Gradient Boosting Regressor (GBR) machine learning model was chosen to predict the protein-ligand binding affinities after evaluating its performance when compared to a number of other models {Support Vector Machine, Linear Regression, Stochastic Gradient Descent Regression, XGBoost Regressor}. The following hyperparameters were chosen after tuning using grid search: {'learning_rate': 0.05, 'loss': 'absolute_error', 'max_depth': 8, 'min_samples_split': 5, 'n_estimators': 256}.

A full summary of hyperparameters used can be found in Table S5.

3.3.5 Model Evaluation

A triple approach for model validation was followed. In the first place, a 5-fold cross-validation was applied to all models. Secondly, all models were validated against an independent validation set, the TKI database [30].
Finally, as a third validation approach, a Y-randomization was applied in order to estimate the risk of chance correlations [31]. This validation step was performed by keeping the features fixed in space and randomly shuffling the binding affinity (Y variable) and then retraining the model. The process was repeated ten times, each time with a different randomized dependent variable vector.

The performance of each model was evaluated using four metrics when trying to categorically predict change in binding affinity: Accuracy, Recall, Precision and F1 Score.

When trying to quantitatively predict ΔΔG; Pearson correlation coefficient (r) and Root Mean Square Error (RMSE) were tested for in addition to the previously mentioned metrics.

3.4 Rosetta Calculations

3.4.1 Dataset

In this case a total 406 entries from both the PLATINUM and TKI databases extracted from the MdrDB database were used in order to have a similar N value to the other two techniques as a considerable number of entries from the PLATINUM database were incompatible with this technique due to a variety of factors such as broken PDB files, presence of Heme functional groups, ligand parameterization errors, and joint PDB file creation errors. More work should be done on an individual protein complex level in order to fix these issues.

3.4.2 Flex_ddG protocol

Binding free energy changes were calculated with Rosetta (v2022.43) using the flex_ddg protocol [18]. This method works by sampling multiple conformations of wild-type and mutant proteins with a Monte Carlo algorithm and estimating ΔΔG with the all-atom Rosetta energy function [32], a mixed knowledge-physics based potential, by averaging over the generated wild-type and mutant groups.

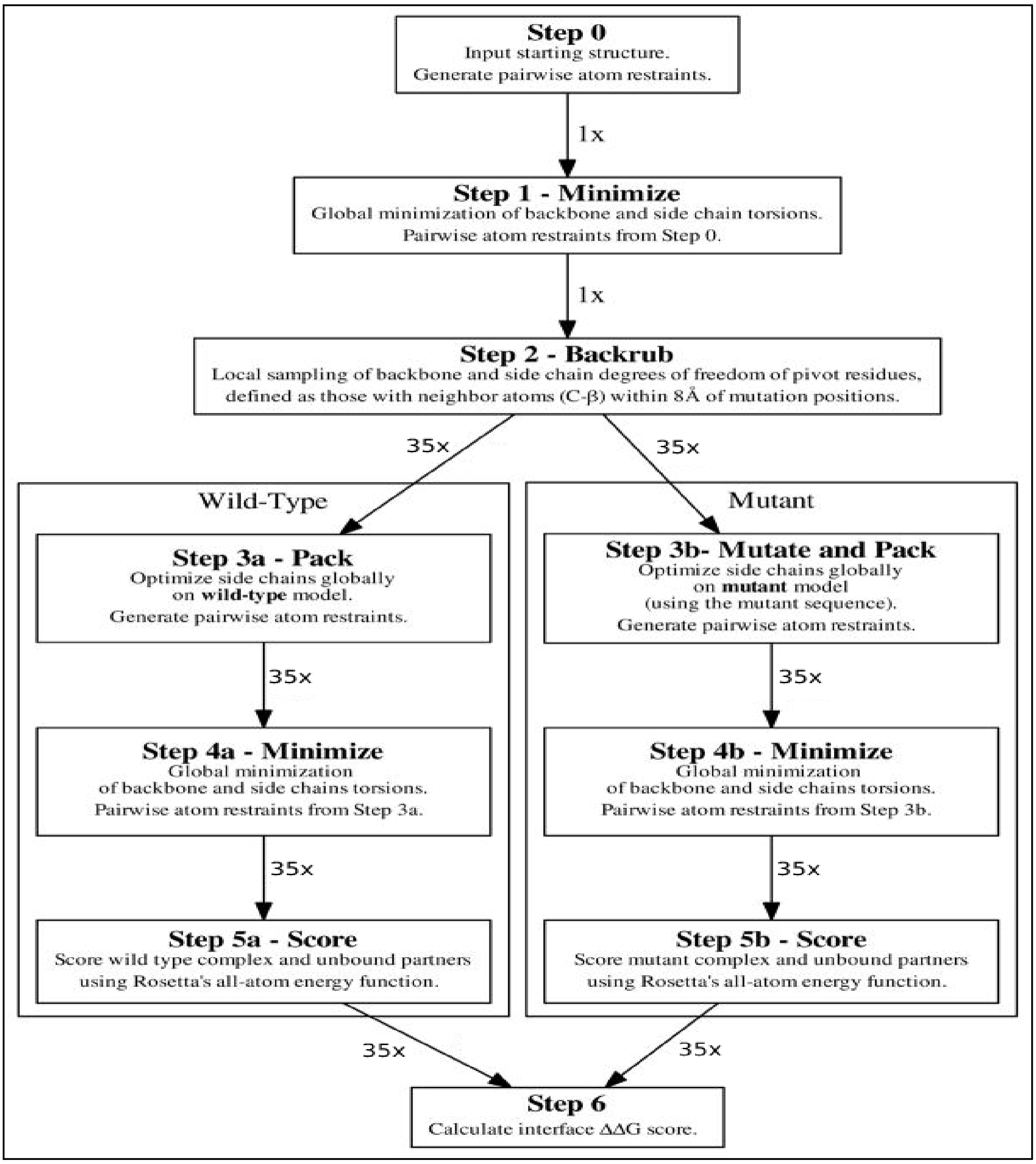
Step 0
Input starting structure.
Generate pairwise atom restraints.
1x
Step 1 - Minimize
Global minimization of backbone and side chain torsions.
Pairwise atom restraints from Step 0.
1x
Step 2 - Backrub
Local sampling of backbone and side chain degrees of freedom of pivot residues,
defined as those with neighbor atoms (C-β) within 8Å of mutation positions.
35x
35x
Wild-Type
Mutant
Step 3a - Pack
Optimize side chains globally
on wild-type model.
Generate pairwise atom restraints.
Step 3b- Mutate and Pack
Optimize side chains globally
on mutant model
(using the mutant sequence).
Generate pairwise atom restraints.
35x
35x
Step 4a - Minimize
Global minimization
of backbone and side chains torsions.
Pairwise atom restraints from Step 3a.
Step 4b - Minimize
Global minimization
of backbone and side chains torsions.
Pairwise atom restraints from Step 3b.
35x
35x
Step 5a - Score
Score wild type complex and unbound partners
using Rosetta's all-atom energy function.
Step 5b - Score
Score mutant complex and unbound partners
using Rosetta's all-atom energy function.
35x
35x
Step 6
Calculate interface ΔΔG score.

Figure 3: Flex ddG protocol flowchart, it follows the path from initial PDB file and mutation resfile to numerical ΔΔG score. Modified from [18]

On its core, the protocol (Figure 3) begins with an initial minimization on backbone φ/ψ and side chain χ torsional degrees of freedom, using the limited-memory Broyden-Fletcher-Goldfarb-Shanno minimizer implementation found in the Rosetta base distribution, with Armijo inexact line search conditions of the input wild-type protein complex crystallographic structure. This first and all following minimizations are performed with harmonic restraints on pairwise atom distances to their values in the input crystal structure and run until convergence.

In the second step, starting from the minimized input structure including both binding partners in the protein-ligand complex, the backrub method begins creating an ensemble of models. To summarize, each backrub move is tackled on a randomly chosen protein segment consisting of anywhere from three to twelve adjacent residues in the neighborhood of any mutated position (C-β atom within 8 Å of the mutant position). All atoms in the backrub segment are rotated locally with respect to an axis defined as the vector between the endpoint C-α atoms. All this process is run at a temperature of 1.2 kT, for up to 50,000 backrub Monte Carlo trials/steps and up to 35 models are generated.

In the third step, for each of the 35 models in the ensemble output generated by the backrub an optimization process is performed for the side chain conformations for the wild-type sequence in which discrete rotameric conformations and simulated annealing are used. Simultaneously the same process is being applied after mutating the protein in order to obtain both the WT and mutant protein.

The fourth step consists of minimizing each of the 35 models for both WT and mutant proteins using the same parameters as the first minimization.

Finally each of the 35 minimized models for both WT and mutant models are scored in complex, and the complex partners are scored individually. The scores of the split, unbound complex partners are obtained simply by separating the complex halves from each other. After that the ΔΔG scores are computed as the arithmetic mean over the different models produced following the following formula:

$$\Delta\Delta G_{Bind} = \Delta G_{Bind}^{Mut} - \Delta G_{Bind}^{WT} =$$

$$= (G_{Complex}^{Mut} - G_{Protein}^{Mut} - G_{Ligand}) -$$
$$- (G_{Complex}^{WT} - G_{Protein}^{WT} - G_{Ligand})$$

This protocol was slightly modified in order to allow it to calculate protein-ligand interaction ΔΔG instead of the usual protein-protein interactions by treating the ligand as an additional protein chain in the PDB file.

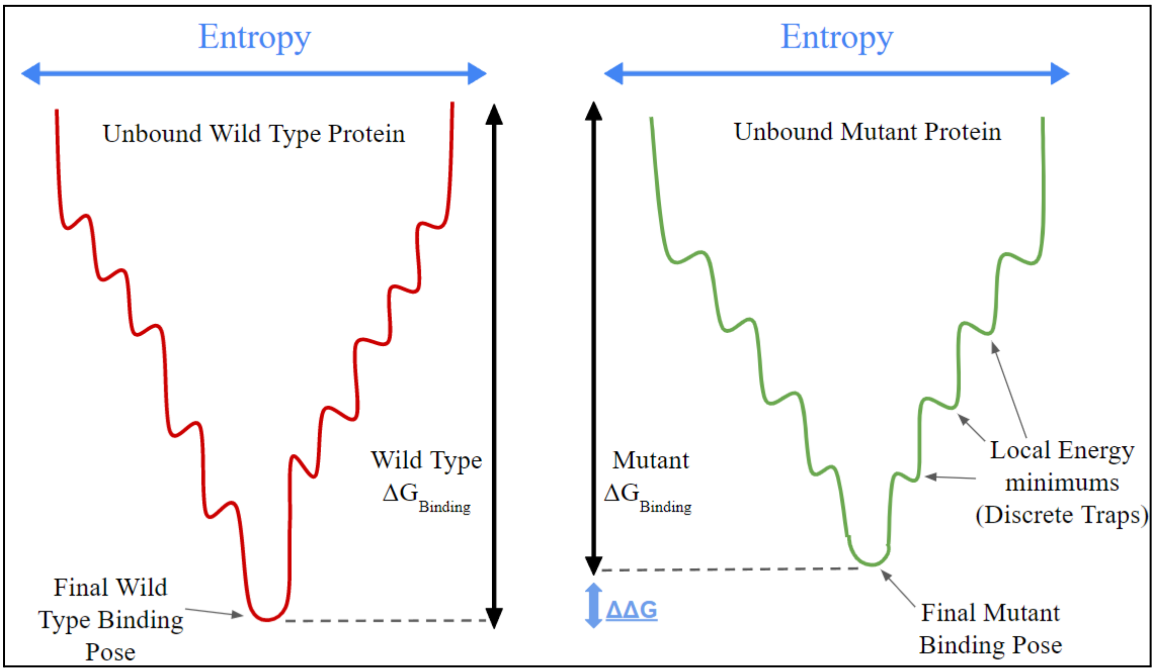

Figure 4: Simplified 2D energy landscape diagram for the Binding process. The x axis represents different ligand binding poses. Energy of the protein/ligand complex structure is measured along the y axis. The unbound states have the highest energy and the final binding pose the least energy.

Explicit hydrogens were added to both protein and ligand using the Gromacs pdb2gmx command [33]. Ligand parameters were obtained with the molfile_to_params.py script provided with Rosetta. Finally, the mutation resfile (an input file that tells Rosetta which protein positions to mutate in the ΔΔG calculation) was generated using python. At each moment in time 12 entries were being run simultaneously and in parallel in order to accelerate the process.

The Rosetta Energy Function 2015 (REF15) scoring function was used. The final ΔΔG estimates were the average values of the model obtained from 35 iterations of the flex_ddg protocol.

3.5 Data Analysis

All data and statistical analysis were performed using a combination of python and R. he python libraries used for this purpose were the NumPy [23], SciPy [28], Pandas [25] and SciKit-Learn [34]. The R packages used for statistical analysis were dplyr [35], corrplot [36], and tidy [37].

All plot generation was performed using python and R with the matplotlib library and ggplot2 package respectively

The main metrics that were evaluated are accuracy, recall, precision, F1 score, Pearson correlation coefficient (r) and root mean square error (RMSE) when applicable. All metrics can be calculated using the following formulas:

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

$$Recall = \frac{TP}{TP+FN}$$

$$Precision = \frac{TP}{TP+FP}$$

$$F1\,Score = 2 \cdot \frac{Precision \cdot Recall}{(Precision + Recall)} = \frac{TP}{TP+\frac{1}{2}(FP+FN)}$$

$$r = \frac{\Sigma\,(x_i - \overline{x})\,(y_i - \overline{y})}{\sqrt{\Sigma\,(x_i - \overline{x})^2\,\Sigma\,(y_i - \overline{y})^2}}$$

$$RMSE = \sqrt{\frac{\Sigma_{i=1}^{N}\,(x_i - \hat{x}_i)^2}{N}}$$

3.6 Use Case Example: BAM file to Problematic Drugs List

As a use case example of the tested techniques we set the goal to provide a list of possible affected drugs, be it at the primary target level, at an off target protein interaction or even at the metabolizer protein level, starting from an individual's direct genetic information. In order to do so, the first step was to obtain a .bam to get a list of SNPs present in its genetic profile.

Later, this list of SNPs was analyzed with the techniques presented in this study in order to provide a list of possibly problematic drugs for that particular individual (Figure 5).

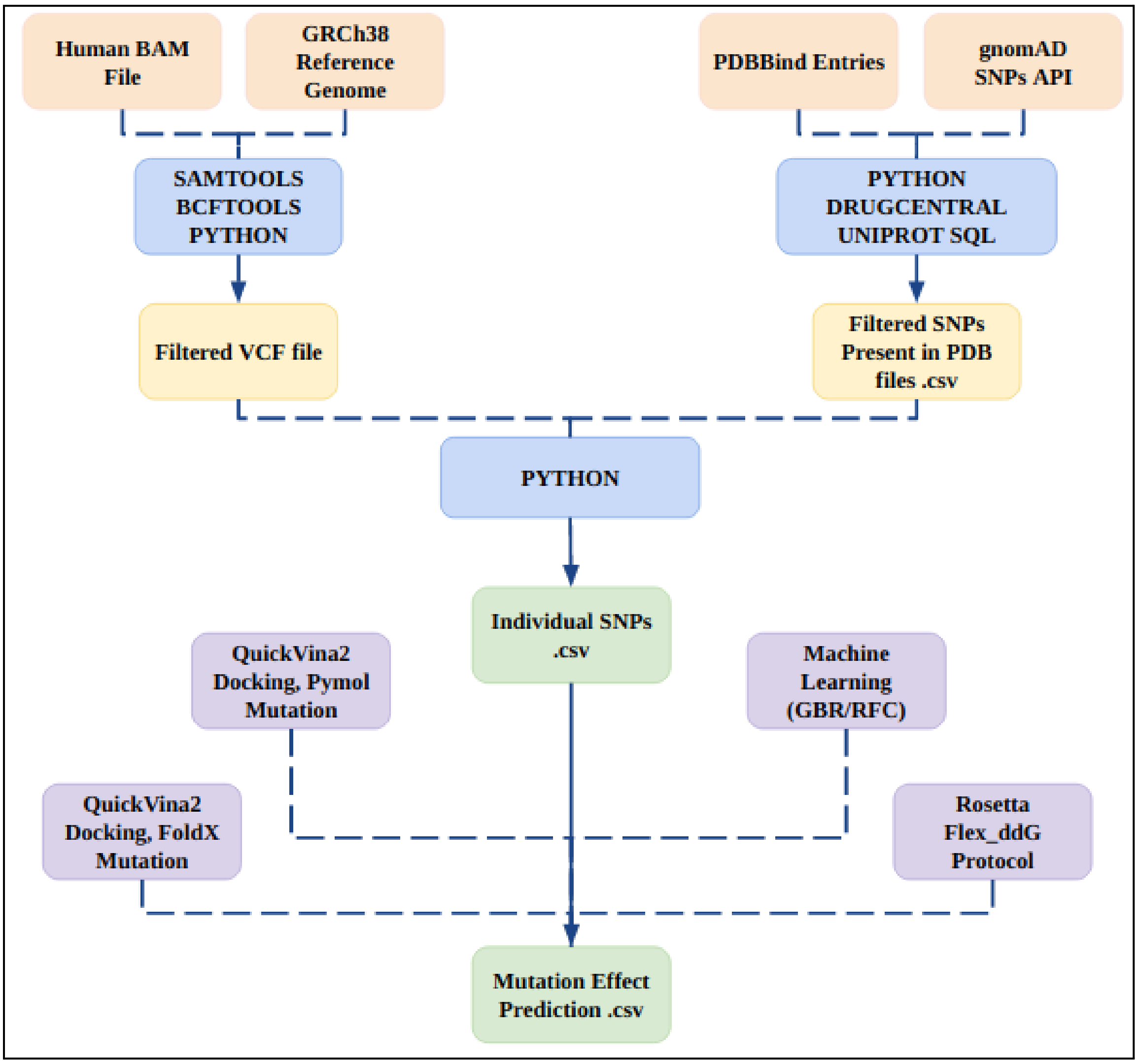
Human BAM File
GRCh38 Reference Genome
PDBBind Entries
gnomAD SNPs API
SAMTOOLS BCFTOOLS PYTHON
PYTHON DRUGCENTRAL UNIPROT SQL
Filtered VCF file
Filtered SNPs Present in PDB files .csv
PYTHON
Individual SNPs .csv
QuickVina2 Docking, Pymol Mutation
Machine Learning (GBR/RFC)
QuickVina2 Docking, FoldX Mutation
Rosetta Flex_ddG Protocol
Mutation Effect Prediction .csv

Figure 5: BAM file to Mutation Prediction Pipeline showing a simplified step by step guide on obtaining all the relevant files

3.6.1 Data Scraping and Recollection

In order to obtain a list of drugs and associated proteins of which we have sufficient data in order to cast predictions, the first step taken was to download the full list of entries from PDBBind [38] as a .tsv file (as of 12/9/2022).

PDBBind was chosen due to both its vast catalog and its curation process which ensures that only the highest quality PDB files are available.

This initial list consisted of 19,943 PDB files which were rapidly filtered in order to only leave those entries in which the ligand was a drug. This filter was performed by cross referencing with DrugCentral [39], a freely available drug database, and only those ligands that were also present in DrugCentral that had an ATC code [40] were retained. After this initial filter 676 PDB files remained.

A second filter consisting of removing all non-human proteins by cross-referencing with the UNIPROT database was performed and, in the end, 235 PDB files remained. Faulty files and chimeric proteins were also removed leaving us with 223 PDB files.

Later on, a full list of SNPs was obtained for each protein by firstly converting the UNIPROT ID to ENSEMBL ID via the Biomart API [41] and then using this id to query the gnomAD API [42]. This process returned 185530 individual SNPs for the 223 PDB files, that were later filtered in order to remove those mutations that occurred in locations of the protein not mapped to the PDB file as well as those SNPs with a reported allele frequency of 0, i.e, not found in any individual. This process left us with 19144 individual SNPs (Figure 6) and the following statistics:

- ➢ Unique PDBs = 223
- ➢ Unique Genes/Proteins = 107
- ➢ Unique Ligands = 129
- ➢ Unique SNPs locations = 15795
- ➢ Average number of locations/PDB = 70.83
- ➢ Primary Target = 115, Secondary = 74, NA = 34 (Probably secondary targets but no data available)

Finally a python script using scipy and numpy was done in order to introduce additional data such as mutation distance from ligand.

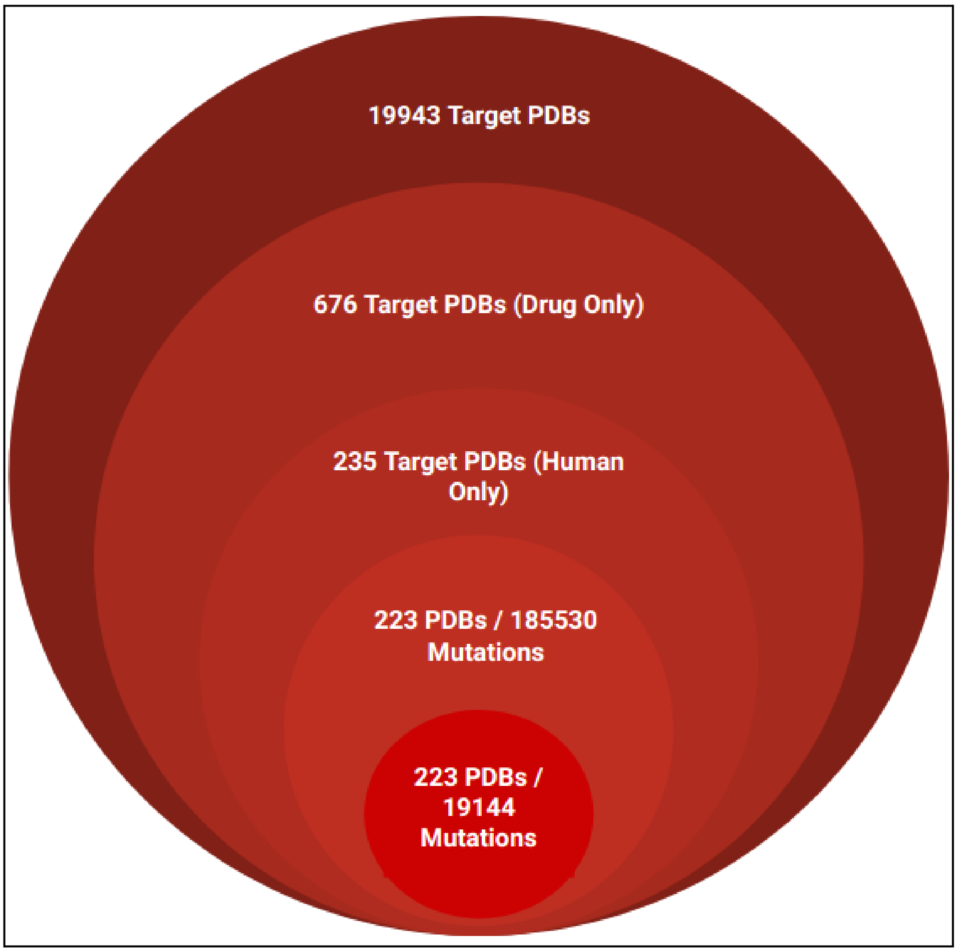
19943 Target PDBs
676 Target PDBs (Drug Only)
235 Target PDBs (Human Only)
223 PDBs / 185530 Mutations
223 PDBs / 19144 Mutations

Figure 6: Full PDBind catalog to Human SNPs affecting human drug-binding protein filters.

3.6.2 BAM file to VCF file

A variant call format file (VCF file) is the output of a bioinformatics pipeline. It specifies the format of a text file used in bioinformatics for storing gene sequence variations. Those sequence variations can be extracted from a BAM file and be various types of mutations including insertions, deletions and SNPs. This last one, SNPs, is the only one we were interested in and therefore all the other mutation types were not looked at.

In order to obtain a full list of all SNPs in an individual's BAM file the following pipeline was followed and turned into a python script.

Firstly BAM files need to be sorted prior to performing any action, this step was performed using the samtools [43] sort command. Once sorted, the file needs to be indexed, this step was performed using the samtools index command.

Once the BAM file was sorted and indexed it could be transformed into a VCF file using the bcftools [44] mpileup command followed by the call command this left us with a VCF text file containing all variations independent of quality or type. In order to filter them, the bcftools command filter was used to only leave variations with more than a 99% certainty and those corresponding to SNPs. Finally the bcftools query command was used in order to remove all data that was not relevant and leaving us with a .tsv file containing chromosome, location, initial nucleotide and mutated nucleotide.

To work with BAM files a reference genome must be provided, in this case the Genome Reference Consortium Human Build 38 patch release 14 (GRCh38.p14) extracted from the ncbi website [45] was used.

3.6.3 SNP List to Change in Binding Affinity Prediction

Once the .tsv file containing all SNPs present in the BAM file was obtained it was combined with the file of variations in PDB files obtained previously. This joining process was done by matching both chromosome, location and initial residue in both files and obtaining the resulting list of variations.

Once the full list of SNPs and their respective PDB files was obtained we could proceed to performing the methods described in the main project in order to cast the predictions for this set of mutations.

4. Results

The performance of all models was evaluated on two distinct but related fields, numerical correlation and categorical classification. In order to evaluate the numerical correlation between the experimental and predicted two parameters were compared root mean-square error (RMSE) and the Pearson's correlation coefficient (r), when applicable. On the other hand, in order to examine the models' classification performance, accuracy, precision, recall and F1-score were compared.

As a baseline a classifier based uniquely on a distinction between mutations in residues involved in the binding (<6Å from ligand) and those far away was able to obtain the following results: Pearson r of 0.274, precision of 0.412, a recall of 0.891, a F1 score of 0.563 and an accuracy of 0.513 with a N of 417 samples.

The first approach tested was the combination of a mutation step followed by a docking and scoring step. Two different methods of applying the mutation were tested, On one hand, mutations were added using the PyMol Open Source Mutagenesis tool followed by the docking procedure executed using Qvina. On the other hand mutations were applied using the FoldX software followed by a docking step performed using Qvina. For each approach different parameters were tested as explained in the methodology section. For simplicity only the best performing model of each approach will be discussed.

The best model using PyMol to introduce the mutations achieved a r value of 0.135, a precision of 0.480, a recall of 0.809, a F1 score of 0.603 and an accuracy of 0.623 with a N value of 417 samples (Figure 7).

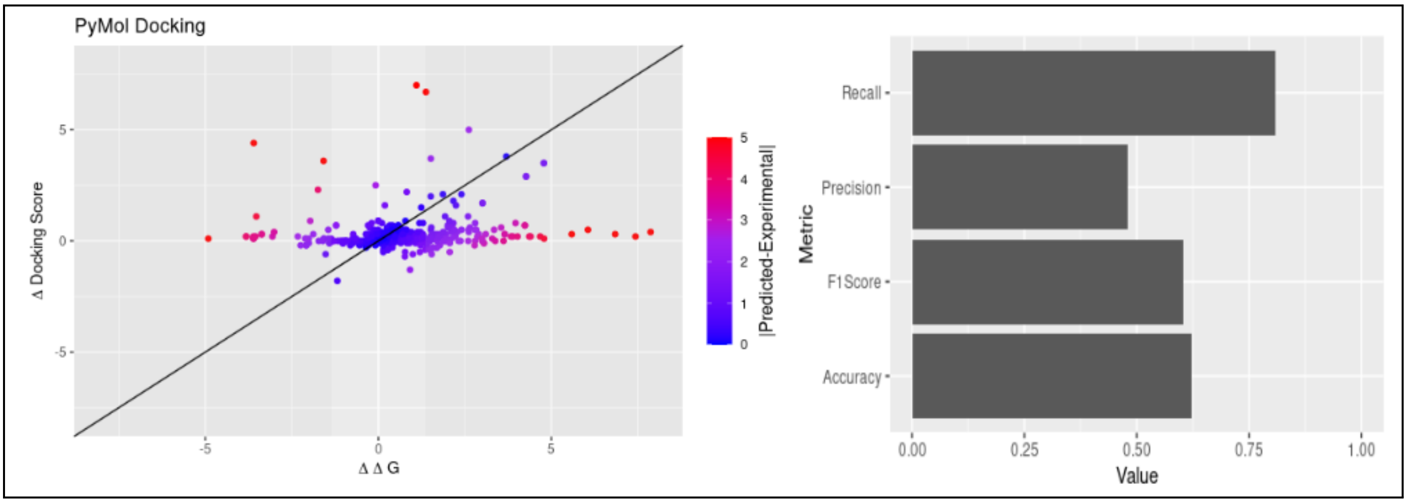

Figure 7: Scatter plot of experimental ΔΔG values versus predicted ΔScore values for PyMol Docking and corresponding classification metrics.

On the other hand, the best model using FoldX was able to achieve a r value of 0.160, a precision of 0.447, a recall of 0.688, a F1 score of 0.542 and an accuracy of 0.600 with a N value of 357 samples (Figure 8).

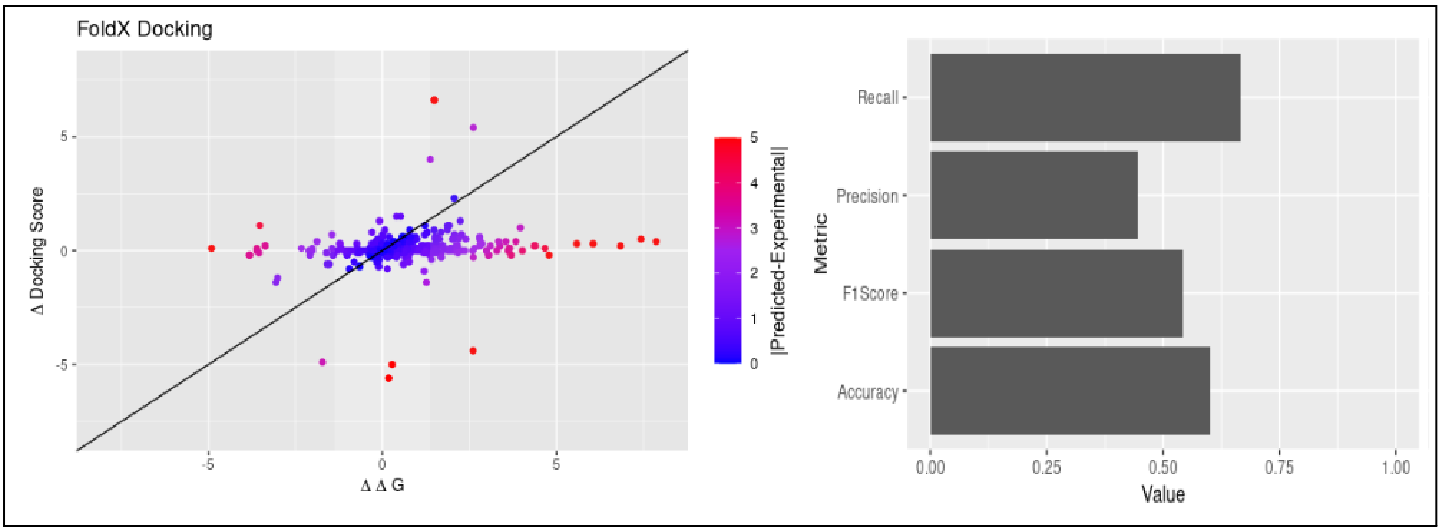

Figure 8: Scatter plot of experimental ΔΔG values versus predicted ΔScore values for FoldX Docking and corresponding classification metrics.

As expected, the docking procedure that induced more accurate results when PyMol was used to introduce the mutation was “Flexible” docking. It can be argued that by leaving the mutated residue free to move the docking step is more capable of mimicking the procedure that happens in a real life environment. On the other hand for the mutations introduced using FoldX the “Rigid” docking procedure achieved better results. This could be caused by the fact that FoldX already changes the position of the residues with a small minimization/optimization step that could render the flexible part of the docking less necessary. All in all no method achieved better than random quantitative correlation, despite this both techniques obtained a good classification performance.

Overall there is no clearly better method between applying the mutation with FoldX or PyMol, both techniques achieved similar results, the biggest difference being computational time required per estimate and number of samples compatible with each technique. Based on this criterion PyMol is the clear winner with an average time per estimate of around twenty times less than FoldX, a slightly better performance and a much bigger number of samples on which the technique can be used. The main reason why some samples produced errors or faulty results when using FoldX was due to the fact that the ligand is not taken into account when performing the optimization step. In return, this can cause the binding pocket to “close” causing clashes when trying the docking step.

In order to test the scalability of the techniques tested another database (TKI) was used. Upon testing the exact same parameters the following results were obtained for the proteins in which PyMol was used to introduce the mutations: r value of 0.127, a precision of 0.181, a recall of 0.428, a F1 score of 0.255 and an accuracy of 0.672 with a N value of 107 samples. On the other hand for those proteins in which FoldX was used to introduce the mutations the following results were obtained: r value of 0.007, a precision of 0.162, a recall of 0.875, a F1 score of 0.274 and an accuracy of 0.559 with a N value of 84 samples. All in all, the performance of both models dropped significantly on this separate dataset. The most probable reason for this drop is that this particular protein (All entries correspond to the same protein, ABL kinase) is not really suited for this technique.

When using machine learning two different approaches were considered, firstly a classifier based on the Random Forest algorithm was tested and the model that achieved the best performance returned a precision of 0.529, a recall of 0.629, a F1 score of 0.599 and an accuracy of 0.826 with a N of 69 entries in the test set and 342 in the train set (411 total entries), when tested against an independent validation set of 125 entries from the TKI dataset the performance dropped drastically, achieving a precision of 0.368 a recall of 0.179 a F1 score of 0.241 with an accuracy of 0.648 (Figure 9).

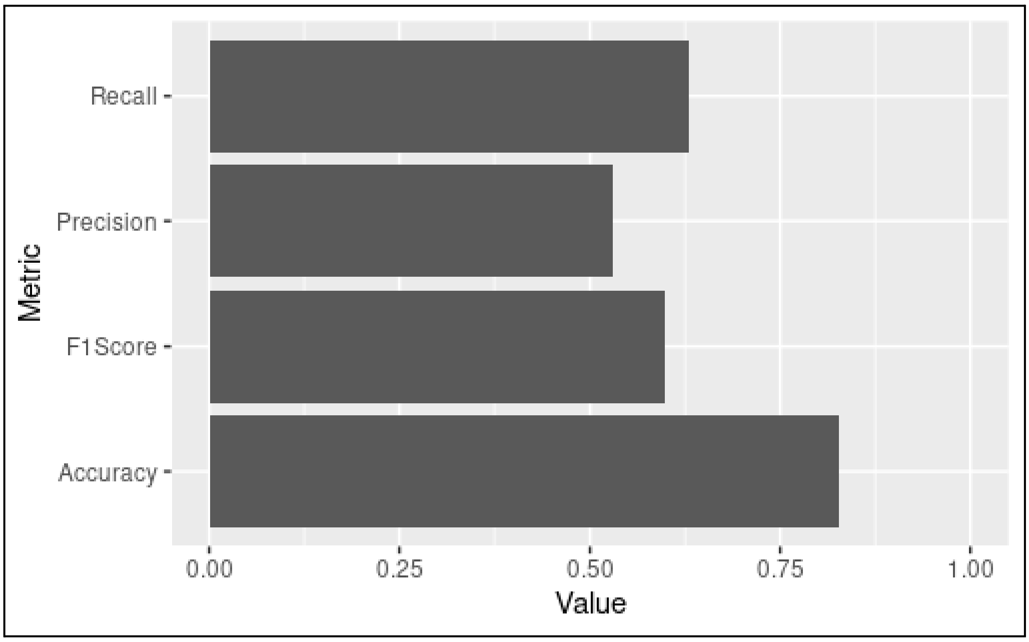

Figure 9: Barplot of classificator metrics for the Random Forest Classifier.

The other approach aimed to first quantitatively predict the ΔΔG numerical value and, later, based on that prediction, classify the mutations in Change/No change of affinity. This second model used a GBR machine learning technique, all in all, it achieved a r value of 0.213, a RMSE of 1.741 Kcal/mol, a precision of 0.211, a recall of 0.364, a F1 score of 0.270 and an accuracy of 0.686 with a N of 69 entries in the test set and 342 in the training set (411 total entries). In turn when tested against the independent validation set it returned an r value of -0.132 a RMSE of 0.956, a precision of 0.053 a recall of 0.500 with a F1 score of 0.096 and an accuracy of 0.848, i.e not much better than random (Figure 10).

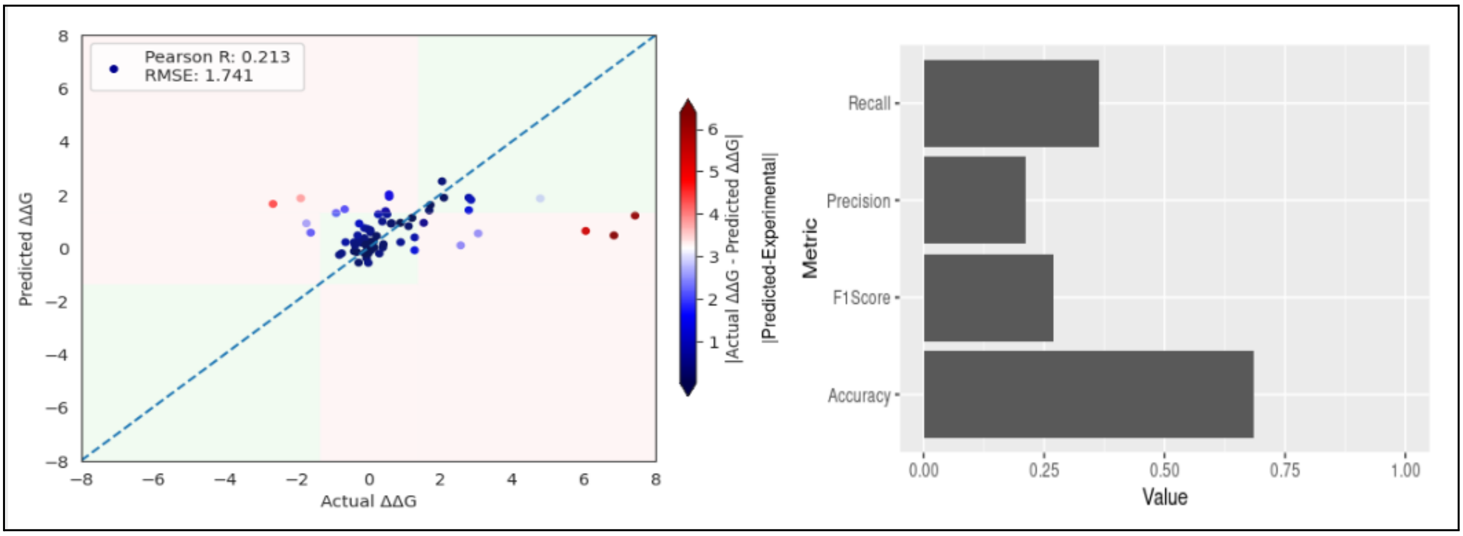

Figure 10: Scatter plot of experimental versus predicted ΔΔG values. The identity is shown as a dashed blue line. Dots are colored based on the difference between experimental and predicted values. The green area represents correct categorical predictions, the red area represents incorrect ones. Classification metrics barplot also included.

An advantage of machine learning as a technique is speed; Once the necessary input features have been computed, a binding affinity estimate can be obtained in fractions of a second. Meanwhile the whole training process needs more or less as much time as 10 FoldX estimates. The most important descriptors for both ML techniques can be seen in Figure 11.

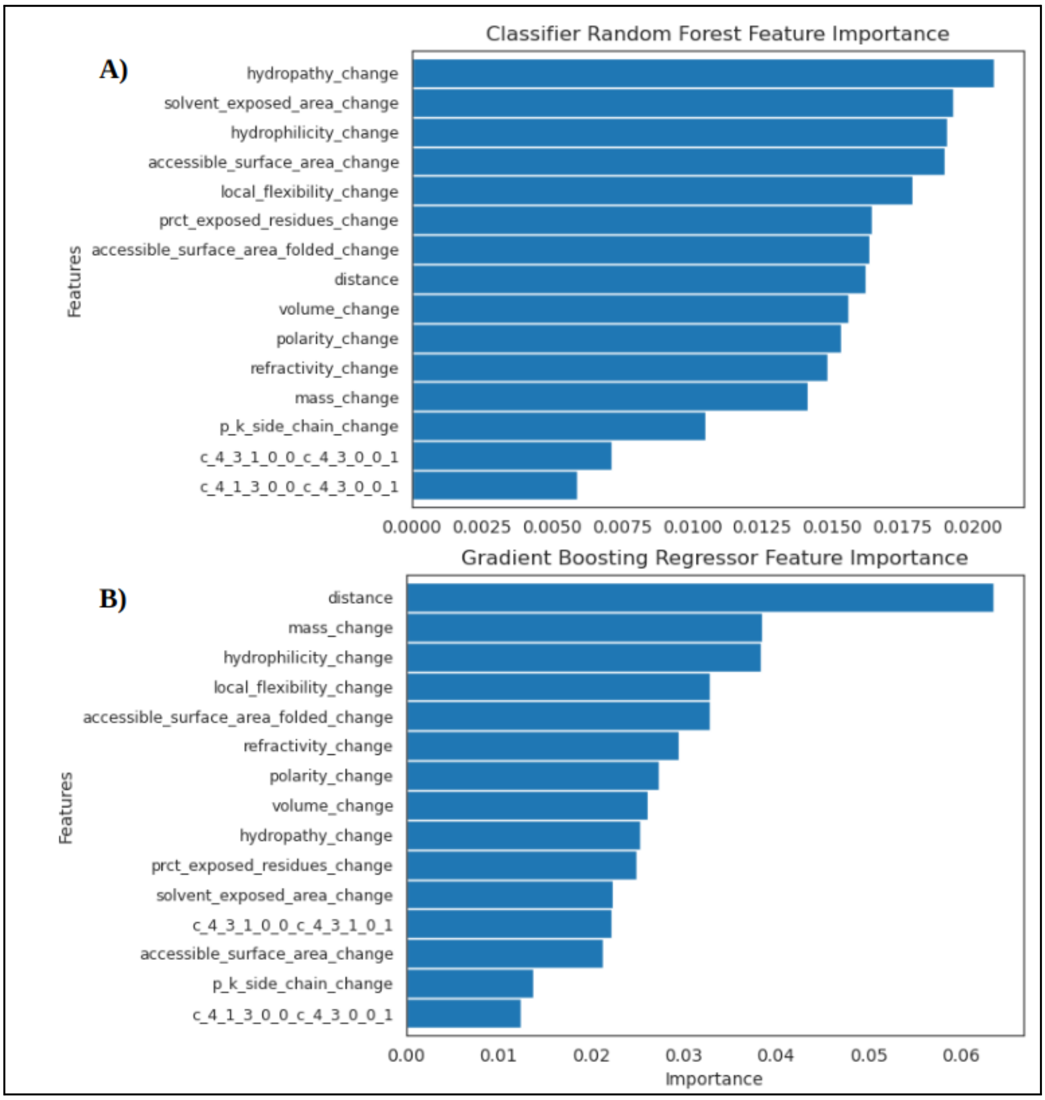

Figure 11: Top 15 features for each of the machine learning models. A) Feature importance for the Random Forest Classifier, in this case no feature is much more important than the next one, mutation descriptors seem to be the most important. B) Feature importance for the Gradient Boosting Regressor, in this case distance between mutation and ligand plays the biggest role.

Finally, when looking at the results from the Rosetta flex_ddG protocol, a knowledge-physics mixed method, we can see a Person's r value of 0.011, a RMSE of 2.067, a precision of 0.623, a recall of 0.323, a F1 Score of 0.425 and an accuracy of 0.714 with an N value of 406 entries (Figure 12).

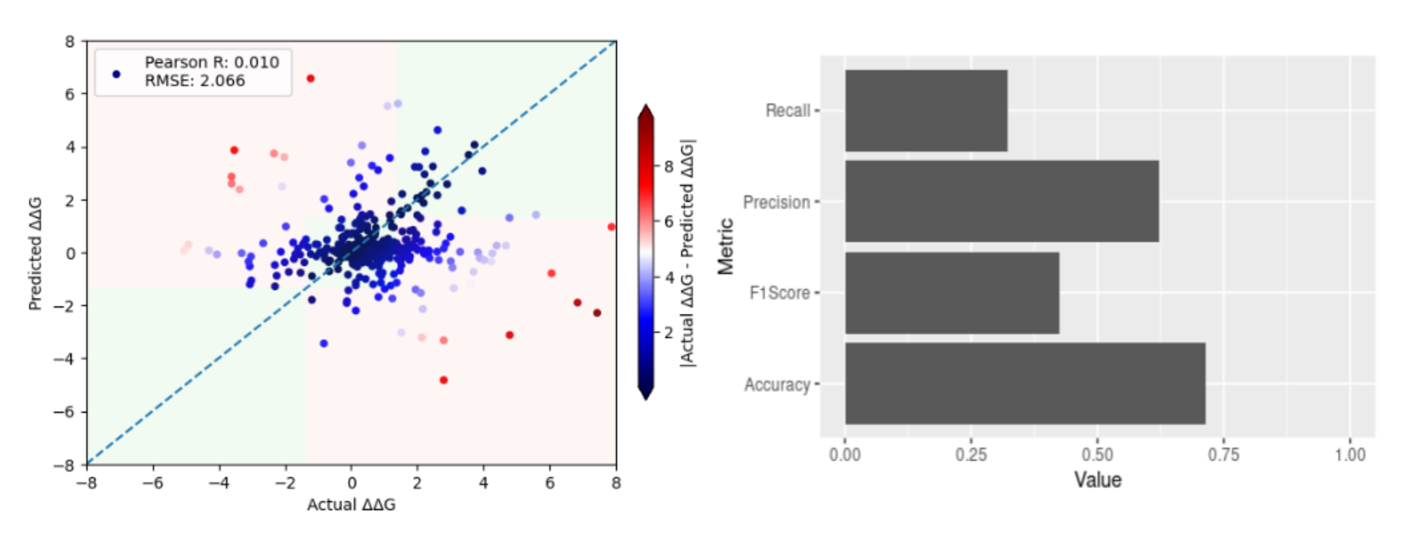

Figure 12: Scatter plot of experimental versus predicted ΔΔG values for the Rosetta flex_ddG protocol. The identity is shown as a dashed blue line. Dots are colored based on the difference between experimental and predicted values. The green area represents correct categorical predictions, the red area represents incorrect ones. Classification metrics barplot also included.

On the other hand, when just looking at the entries from the TKI database we obtained the following results: Pearson's correlation value of 0.453, RMSE of 0.815, a precision of 1, a recall of 0.333, a F1 Score of 0.500 and an accuracy of 0.900 with an N value of 60 entries. This result falls in line with the one achieved by Aldeghi, et al. [17], and, therefore leads to the conclusion that the flex_ddG protocol varies a lot in performance depending on the protein complex tested, and, while it can be suited for protein kinases, its performance cannot be reliably extrapolated to all other protein families.

A confusion matrix and a full summary of the performance of all the techniques studied in this project can be seen in Table 1 and Table 2 respectively. Additionally a graphical representation of the performance on the test and validation sets can be found in Figure 13 and Figure 14 respectively.

		Baseline (<6Å)		PyMol Docking		FoldX Docking		Random Forest Classifier		Gradient Boosting Regressor		Rosetta Flex_ddG	
		Change	**No Change**	**Change**	**No Change**	**Change**	**No Change**	**Change**	**No Change**	**Change**	**No Change**	**Change**	**No Change**
Experimental Data	**Change**	131	16	119	28	84	38	9	8	4	15	43	90
	No Change	187	83	129	141	104	129	4	48	7	44	26	247

Table 1: Confusion Matrix of all the tested techniques against their respective experimental data.

	Aprox. Computational cost per estimate		Performance						
Method	Hardware	Minutes Computing	RMSE Kcal/mol	Pearson Correlation	Precision	Recall	F1 Score	Accuracy	N[1]
Baseline (Distance)	AMD Ryzen 7 5800X 8-Core x2	0.1	n/a	$0.274_{0.183}^{0.361}$	0.411	0.891	0.563	0.513	417
Docking PyMol	AMD Ryzen 7 7 5800X 8-Core x2	0.2	n/a	$0.135_{0.040}^{0.229}$	0.479	0.809	0.602	0.623	417
Docking FoldX	AMD Ryzen 7 5800X 8-Core x2	4.50	n/a	$0.160_{0.058}^{0.259}$	0.447	0.688	0.542	0.600	357
ML Classification	AMD Ryzen 7 5800X 8-Core x2	0.20 (Training)	n/a	n/a	0.692	0.529	0.599	0.826	69 (411)
ML Quantitative	AMD Ryzen 7 5800X 8-Core x2	0.35 (Training)	1.741	$0.213_{0.025}^{0.429}$	0.364	0.211	0.270	0.686	69 (411)
Rosetta Flex_ddG	AMD Ryzen 7 5800X 8-Core x2	1920	2.067	$0.011_{-0.10}^{0.086}$	0.623	0.323	0.425	0.714	406

Table 2 : Summary of the approaches used, their Performance and Computational Cost. For the performance measures, the point estimates from the original samples and their 95% confidence intervals are shown as (x_{lower}^{upper}). N: Sample size. All computing times were calculated using the python time library and averaging over all samples.

[1] The difference in sample size between docking, machine learning techniques and Rosetta is due to incompatibilities between the softwares used and various protein or ligand characteristics such as Heme groups or certain ligand parameters.

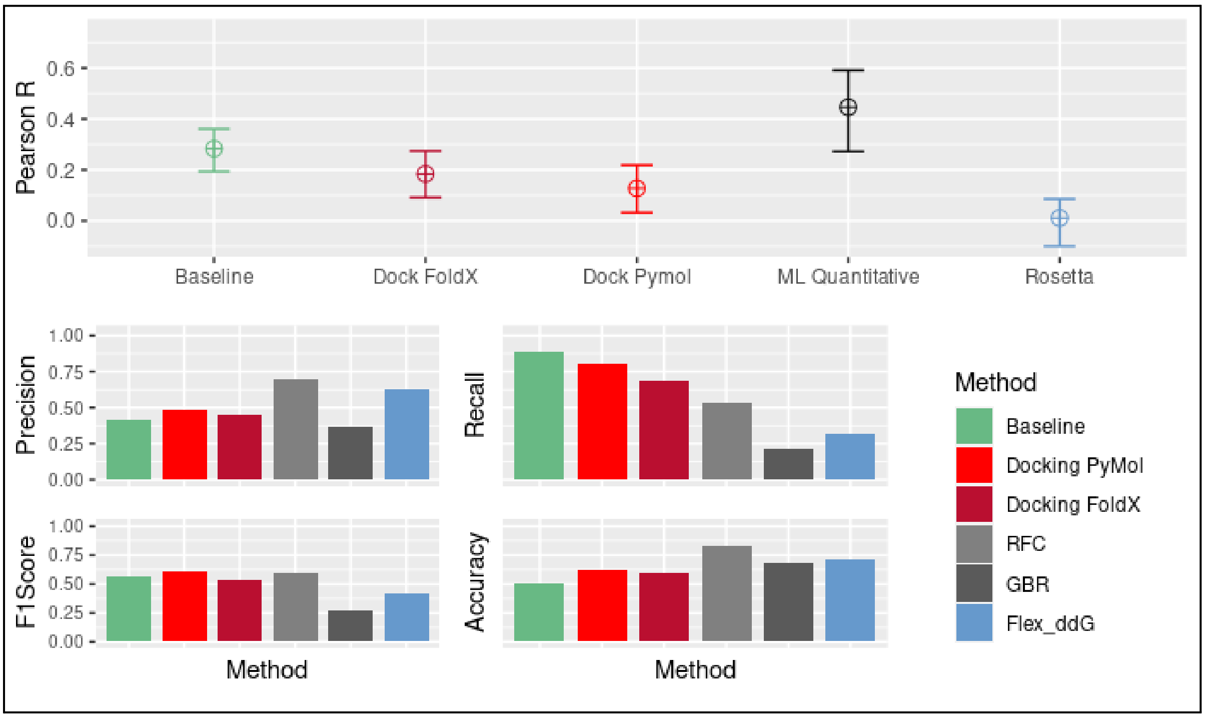

Figure 13: Summary of the performance of the affinity estimates across approaches in the test set in terms of Pearson correlation coefficient with 95% confidence intervals, Precision, Recall, F1 Score and Accuracy.

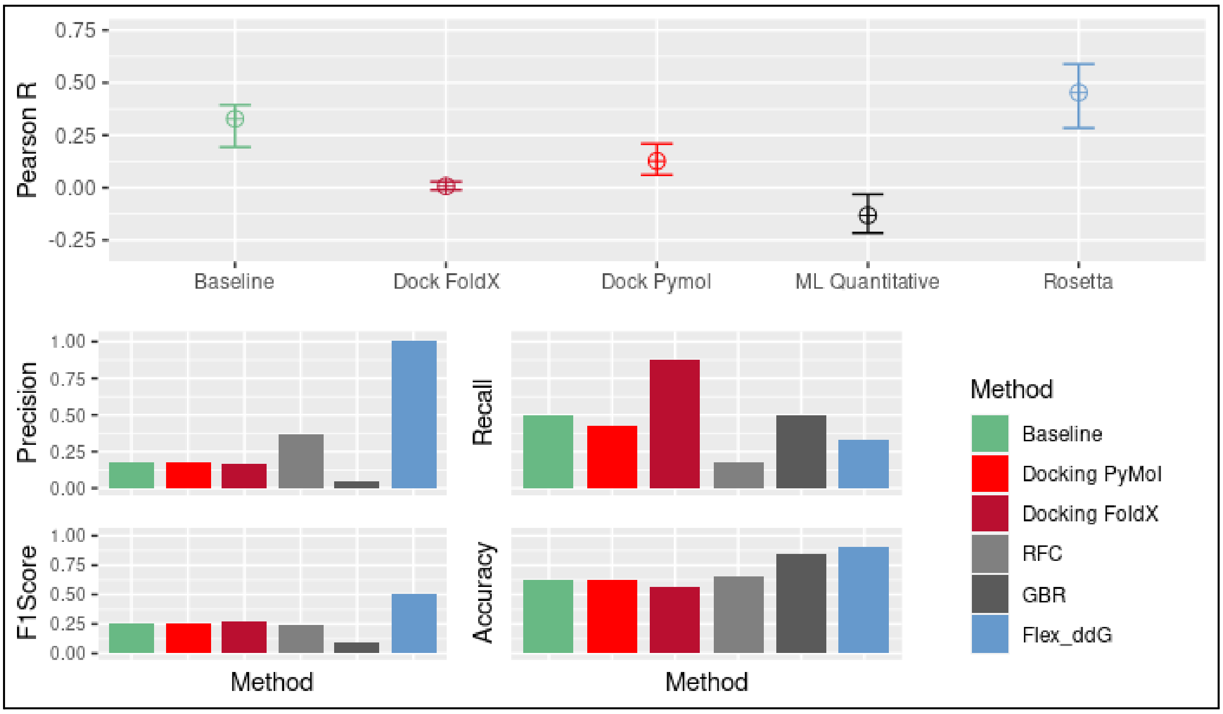

Figure 14: Summary of the performance of the affinity estimates across approaches in the Validation set in terms of Pearson correlation coefficient with 95% confidence intervals, Precision, Recall, F1 Score and Accuracy.

As an example use case of the techniques previously tested, a pipeline that started with a genetic .bam file and ended with a list of drugs probably affected by mutations was developed. A publicly available BAM file GRCh38.illumina.adipose available at (http://www.ensembl.org/info/data/ftp/index.html) was used as a test case by performing the full process on it, see Table 3. Overall the predictions do not really correlate too well between techniques. Some of them predict changes in affinity for mutations that others do not.

To sum up, this technique of directly obtaining an individualized list of problematic drugs from direct genetic data has a promising future ahead, not just when talking about genetic testing companies such as 23andMe but also when talking about personalized medicine in cases like oncologic patients where the genetic makeup of the cancerous cells plays a huge role in drug response.

In addition all those treatments with heavy side effects could be considered on a case by case basis depending on the patients genetic makeup and therefore the propensity and severity of off target side effects.

PDB ID	Uniprot ID	DRUG	Drug Name	Primary Target	chr	loc	variation	Allele Frequency	MUTATION	Distance Ligand	Delta Score Pymol[2]	Delta Score FoldX[3]	DDG GBR[4]	DDG Rosetta[5]	Pred. Dock Pymol	Pred. Dock FoldX	Pred. Rosetta	Pred. GBR	Pred. RFC
2h42	O76074	VIA	sildenafil	TRUE	4	119505895	G,T	0.000191275212	Q743K	28.052009	-0.1	0	1.16829	-0.0241916	Change	No Change	No Change	No Change	No Change
1udu	O76074	CIA	tadalafil	TRUE	4	119505895	G,T	0.000191275212	Q743K	37.918757	-0.1	-0.2	1.52194	0.01140849	Change	Change	No Change	Change	No Change
1uho	O76074	VDN	vardenafil	TRUE	4	119505895	G,T	0.000191275212	Q743K	28.773360	-0.1	0	1.34586	0.12530451	Change	No Change	No Change	No Change	No Change
1udt	O76074	VIA	sildenafil	TRUE	4	119505895	G,T	0.000191275212	Q743K	28.414044	0.1	0	1.20299	0.25720749	Change	No Change	No Change	No Change	No Change
1xoz	O76074	CIA	tadalafil	TRUE	4	119505895	G,T	0.000191275212	Q743K	28.927419	-0.1	-0.1	1.53979	-0.0189451	Change	Change	No Change	Change	No Change
1xp0	O76074	VDN	vardenafil	TRUE	4	119505895	G,T	0.000191275212	Q743K	27.757167	0	0	1.26282	-0.0007672	No Change	No Change	No Change	No Change	No Change
3jwq	O76074	VIA	sildenafil	TRUE	4	119505895	G,T	0.000191275212	Q743K	59.082304	0	0	1.34475	-0.1360560	No Change	No Change	No Change	No Change	No Change
4do5	P17050	DGJ	migalastat	FALSE	22	42067133	G,T	1.31E-05	T161N	50.917239	0	0	0.47962	0.03702604	No Change	No Change	No Change	No Change	No Change
3tt0	P11362	07J	infigratinib	TRUE	8	38415967	T,G	6.58E-06	N586T	10.771609	0	-0.1	0.05416	-0.0402553	No Change	Change	No Change	No Change	Change
2ydo	P29274	ADN	adenosine	TRUE	22	24433456	G,T	6.61E-06	V18L	14.400421	0	0	0.20021	-0.1439321	No Change	No Change	No Change	No Change	No Change
1lbk	P09211	GSH	glutathione	FALSE	11	67585218	A,G	0.3599034772	I105V	13.823928	0	0	0.53629	-0.1034216	No Change	No Change	No Change	No Change	No Change
13gs	P09211	SAS	sulfasalazine	FALSE	11	67585218	A,G	0.3599034772	I105V	2.96163384	0	0	0.55405	0.04819336	No Change	No Change	No Change	No Change	No Change
1lbk	P09211	GSH	glutathione	FALSE	11	67586108	C,T	0.05476952667	A114V	20.97234677	-0.1	0	0.44727	-0.0229505	Change	No Change	No Change	No Change	No Change
13gs	P09211	SAS	sulfasalazine	FALSE	11	67586108	C,T	0.05476952667	A114V	16.60110677	0	0	-0.1568	0.04131340	No Change	No Change	No Change	No Change	No Change
4kn2	P14207	LYA	pemetrexed	FALSE	11	72220933	A,C	0.000105625899	T72P	37.26839201	0	0	0.82079	-0.3698198	No Change	No Change	No Change	No Change	No Change
4kn0	P14207	MTX	methotrexate	FALSE	11	72220933	A,C	0.000105625899	T72P	9.509885225	0.1	0	0.78638	0.53434126	Change	No Change	No Change	No Change	No Change
4kmz	P14207	FOL	folic acid	FALSE	11	72220933	A,C	0.000105625899	T72P	11.27699734	0	-0.1	0.84049	1.08731848	No Change	Change	No Change	No Change	No Change

Table 3. Predictions for all the SNPs found in the GRCh38.illumina.adipose BAM file

[2] Delta Score Pymol: Difference between the Qvina Score of the Wild Type protein-ligand docking and that of the Pymol-mutated docking.
[3] Delta Score FoldX: Difference between the Qvina Score of the Wild Type protein-ligand docking and that of the FoldX-mutated docking.
[4] DDG_GBR: Difference in Binding Free energy between Wild Type and Mutant protein-ligand complexes as returned by the Gradient Boosting Regressor Machine Learning Algorithm.
[5] DDG_Rosetta: Difference between the Wild Type protein-ligand Binding free energy and that of the mutant protein-ligand as obtained by the flex_ddG protocol.

5. Discussion and Future Work

From the results presented, it emerges that the computational approaches tested have the capability of predicting ligand affinity changes upon SNPs but only at a classifier (Change/No Change) level, when trying to quantitatively predict the change of binding free energy all methodologies tested fall short. Different methods, however, show a distinct set of strengths and weaknesses that need to be taken into account in order to use those approaches to their greatest possibilities.

The Molecular Docking based calculations achieved a remarkable performance specially for the process where PyMol was used to introduce the mutations. In this case it achieved a Precision value of 0.479 and a recall of 0.809 .This means that about half of the mutations classified as change producing are in fact so, and 81% of the change producing mutations in the data set have been identified. Another strength that this method showed was the computational time required per estimate, at about 10 seconds per estimate with the previously mentioned setup, being comparable to the machine learning method, albeit less scalable.

The Machine Learning model also showed good results, especially when using a classification algorithm, not so much when trying to numerically predict ΔΔG. The main strength shown by the ML classification model was its accuracy of 0.83 and precision of around 0.70, i.e, about 70% of the mutations classified as change producing are in fact so and 83% of all predictions were correct. The main drawback of this technique is extrapolation. We saw that when looking at a completely independent set (TKI) the accuracy of the results decreased significantly. This problem could be solved with increasing sample size and, especially, diversity of the samples. Despite this, data of this particular type is not easy to come by and poses a real challenge when talking about Machine Learning's usability.

This means that between Docking and Machine Learning we could choose the approach that best suits our goals. For example, if our goal is to identify the majority of mutations that can in fact change the protein-ligand binding affinity and don't mind some false positives, we may use Docking as it has a much better recall. On the other hand, if our goal is to be as precise as possible and only raise a flag for those mutations that actually have an effect, we could use ML as it has a better precision.

When looking at the predictions made with the Rosetta flex_ddG protocol, it showed a poor performance overall, being its best attribute its precision of 62%. The main redeeming factor for this technique is its applicability to specific types of protein complexes such as the Kinases from the TKI database, in this specific subset it showed a much better performance, specially when compared with the other techniques poor performance on that dataset. Overall, when taking into consideration the computational time needed per estimate it is hard to see a use case scenario for this technique.

Although it was outside the scope of this project, in order to improve the performance of the models some tuning could be done. Mainly by trying different approaches, for example for the docking technique different softwares such as rDock or AutoDock could be tested as well as different or even custom scoring functions. Additionally different methods of applying the mutation should be tested as this step is crucial especially when determining the conformation of all nearby residues.

When trying to improve the ML models apart from testing different algorithms and hyperparameters for both the classification as well as the regressor, three main changes could be taken into consideration. Firstly changing the way the mutation is applied to the structure, in this case PyMol was used as it showed the best performance in the docking test but different methods may be used. Secondly testing a different set of descriptors is crucial when trying to improve the model. Despite having hundreds of different descriptors at its disposal only a few of them showed real significance when tested and could be substituted by better descriptors. Lastly and most importantly increasing sample size and sample diversity is crucial in order to get an effective ML algorithm. This change is linked to a necessity to increase experimental studies testing affinity changes upon mutation as well as increasing the amount of high quality 3D crystallographic protein-ligand structures.

Lastly, in order to improve the performance of the Rosetta flex_ddG protocol several parameters could be adjusted such as the number of backrub steps specially for big proteins as well as changing the parametrization of the ligand and the energy function.

In addition to tuning the already tested techniques a new set of models should be tested and were not included in this study due to their heavy computational time requirement and this project's time constraints. Some of this methods include but are not limited to:

- Molecular Dynamics (MD) simulations with a nonequilibrium free energy calculation protocol. This method can be used to sample the isothermal−isobaric ensemble while a specific residue is "alchemically" mutated into another one. From the nonequilibrium work required to transform the residue, it is possible to recover the equilibrium free energy difference and, using a suitable thermodynamic cycle (Figure 11), a rigorous estimate of ΔΔG. Various force field algorithms could be used including: Amber, Charmm, Gromos and OPLS.

- Steered Molecular Dynamics (sMD) is an enhanced sampling method for exploring the force and free energy profiles along a selected direction. The free energy changes as a function of some inter or intramolecular coordinate (such as the distance between two atoms or torsional angles). sMD imitates atomic force microscopy (AFM) experiments and could be used to investigate ligand-receptor interactions. In sMD, an additional external force is applied to pull the ligand out of the target protein binding site along the reaction coordinate ('collective variable', CV).
  The free energy of binding is calculated from the relationship between the force used and the displacement of the ligand. Potential of Mean Force (PMF) (or Free/Gibbs energy profile) is obtained by integrating the average force as a function of constrained distance (Figure 12). It describes the change in free energy along a reaction coordinate. Upon comparing the results for both the Wild Type and mutant protein-ligand complexes a difference in binding affinity could be found.

- Consensus approach to predicting the effect of the mutations. By using a weighted average of the prediction from various models a consensus result could be obtained that outperformed the individual models. In order to properly maximize the accuracy of the consensus prediction a weighting system should be implemented in which each model has different weights based not only on its performance in the whole dataset but also based on the performance in each individual protein, drug, mutation or protein family.

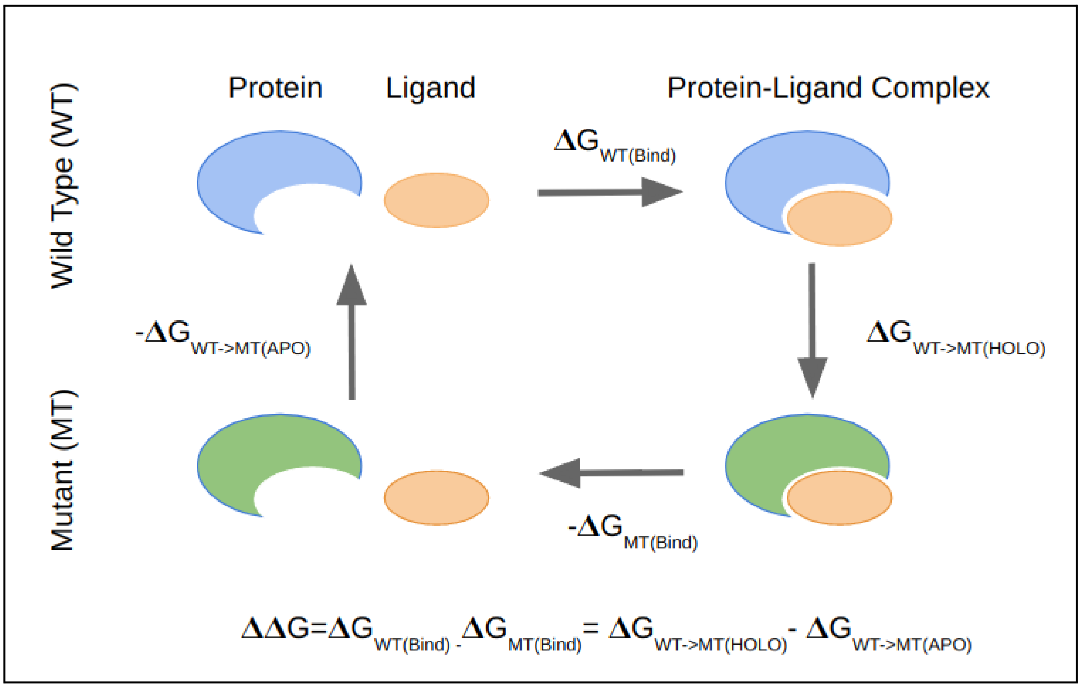

Figure 15: Thermodynamic cycle used in the MD-based free energy calculations.

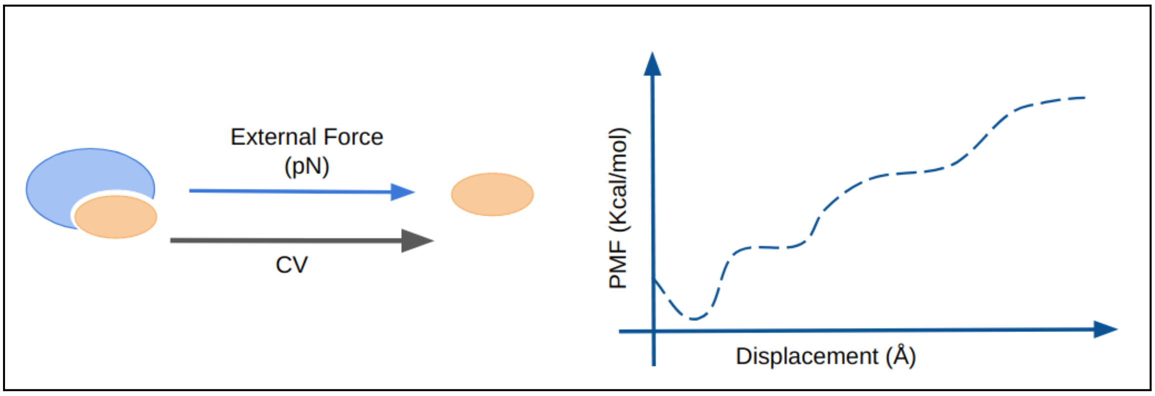

Figure 16: Diagram symbolizing the calculation of PMF using Steered Molecular Dynamics.

6. Glossary

Å: Angstroms (0.1 Nanometers)
A99ff: Amber 99 Force Field
Apo Protein: Unbound Protein
AUPRC: Area Under Precision Recall Curve
CV: Collective Variable
GBR: Gradient Boosting Regressor
Holo Protein: Ligand Bound Protein
MD: Molecular Dynamics
ML: Machine Learning
N: Sample size
ns : Nanoseconds
PDB: Protein Data Bank
PMF: Potential Mean Force
Qvina: Quick Vina 2
R: Pearson correlation coefficient
R15: Rosetta energy scoring function 15
R2 : Squared Pearson correlation coefficient
RF: Random Forest
RMSD: Root Mean Squared Distance
RMSE: Root Mean Squared Error
sMD: Steered Molecular Dynamics
SNPs: Single Nucleotide Polymorphisms
VCF Variant Calling Format
WT: Wild Type
ΔG: Difference in Free energy (in this case difference of free energy upon drug-target interaction)
ΔΔG: Difference in difference of Free energy (in this case difference of free energy upon mutation)

7. Bibliography

1. Singh S, Loke YK. Drug safety assessment in clinical trials: methodological challenges and opportunities. Trials. 13,138 (2012). doi:10.1186/1745-6215-13-138
2. Lahti Jennifer L., Tang Grace W., Capriotti Emidio, Liu Tianyun and Altman Russ B. Bioinformatics and variability in drug response: a protein structural perspective J. R. Soc. Interface. 91409,1437 (2012) http://doi.org/10.1098/rsif.2011.0843
3. Ann K. Daly; Pharmacogenetics and human genetic polymorphisms. Biochem J 1; 429 (3): 435–449 (2010). doi: https://doi.org/10.1042/BJ20100522
4. Schärfe, C.P.I., Tremmel, R., Schwab, M. et al. Genetic variation in human drug-related genes. Genome Med 9, 117 (2017). https://doi.org/10.1186/s13073-017-0502-5
5. Etheridge, A.S., Gallins, P.J., Jima, D., Broadaway, K.A., Ratain, M.J., Schuetz, E., Schadt, E., Schroder, A., Molony, C., Zhou, Y., Mohlke, K.L., Wright, F.A., Innocenti, F.: A new liver expression quantitative trait locus map from 1, 183 individuals provides evidence for novel expression quantitative trait loci of drug response, metabolic, and sex-biased phenotypes. Clinical Pharmacology Therapeutics 107(6):1383-1393 (2020). doi:10.1002/cpt.1751
6. Manish, M., Lynn, A.M., Mishra, S.: Cytochrome p450 2c9 polymorphism: Effect of amino acid substitutions on protein ŕexibility in the presence of tamoxifen. Computational Biology and Chemistry 84,107166 (2020). doi:10.1016/j.compbiolchem.2019.107166
7. Hauser, A.S., Chavali, S., Masuho, I., Jahn, L.J., Martemyanov, K.A., Gloriam, D.E., Babu, M.M.: Pharmacogenomics of GPCR drug targets. Cell 172(1-2):41-5419 (2018). doi:10.1016/j.cell.2017.11.033
8. Pich i Rosello, O., Vlasova, A.V., Shichkova, P.A., Markov, Y., Vlasov, P.K., Kondrashov, F.A.: Genomic analysis of human polymorphisms affecting drug-protein interactions (2017). doi:10.1101/119933
9. Mini E, Nobili S. Pharmacogenetics: implementing personalized medicine. Clin Cases Miner Bone Metab. 6(1):17-24 (2009)
10. Oh SS, Galanter J, Thakur N, et al. Diversity in Clinical and Biomedical Research: A Promise Yet to Be Fulfilled. PLoS Med. 12(12) (2015). doi:10.1371/journal.pmed.1001918
11. G. Ogedegbe, N.R. Shah, C. Phillips, et al. Comparative effectiveness of angiotensin-converting enzyme inhibitor-based treatment on cardiovascular outcomes in hypertensive blacks versus whites. J Am Coll Cardiol, 66:1224-1233 (2015). doi: doi.org/10.1016/j.jacc.2015.07.021
12. Douglas E.V. Pires, Tom L. Blundell, David B. Ascher, Platinum: a database of experimentally measured effects of mutations on structurally defined protein–ligand complexes, Nucleic Acids Research, 43,D1,28:D387–D391 (2015), https://doi.org/10.1093/nar/gku966)

13. Ziyi Yang, Zhaofeng Ye, Jiezhong Qiu, Rongjun Feng, Danyu Li, Changyu Hsieh, Jonathan Allcock and Shengyu Zhang. MdrDB: Mutation-induced drug resistance DataBase. MdrDB: Mutation-induced drug resistance DataBase. https://doi.org/10.1101/2022.10.20.513118
14. Ammar, A., Cavill, R., Evelo, C. et al. PSnpBind: a database of mutated binding site protein–ligand complexes constructed using a multithreaded virtual screening workflow. J Cheminform 14,8 (2022). https://doi.org/10.1186/s13321-021-00573-5
15. Ammar Ammar, Rachel Cavill, Chris Evelo et al. PSnpBind-ML: predicting the effect of binding site mutations on protein-ligand binding affinity, PREPRINT (Version 1) available at Research Square, 24 October (2022) https://doi.org/10.21203/rs.3.rs-2190482/v1
16. Yu, Y., Wang, Z., Wang, L. et al. Predicting the mutation effects of protein–ligand interactions via end-point binding free energy calculations: strategies and analyses. J Cheminform 14,56 (2022). https://doi.org/10.1186/s13321-022-00639-y
17. Aldeghi, Matteo, et al. "Predicting Kinase Inhibitor Resistance: Physics-Based and Data-Driven Approaches." ACS Central Science, 5,8:1468–74 (2019). https://doi.org/10.1021/acscentsci.9b00590
18. Barlow KA, Ó Conchúir S, Thompson S, et al. Flex ddG: Rosetta Ensemble-Based Estimation of Changes in Protein-Protein Binding Affinity upon Mutation. J Phys Chem B. 122(21):5389-5399(2018). doi:10.1021/acs.jpcb.7b11367
19. Amr Alhossary, Stephanus Daniel Handoko, Yuguang Mu, Chee-Keong Kwoh, Fast, accurate, and reliable molecular docking with QuickVina 2, Bioinformatics, 2214–2216;31,13 (2015), https://doi.org/10.1093/bioinformatics/btv082
20. H.M. Berman, J. Westbrook, Z. Feng, G. Gilliland, T.N. Bhat, H. Weissig, I.N. Shindyalov, P.E. Bourne. The Protein Data Bank Nucleic Acids Research, 28: 235-242 (2000). doi: 10.1093/nar/28.1.235
21. Schrodinger, LLC. 2010. The PyMOL Molecular Graphics System, Version X.X
22. Joost Schymkowitz, Jesper Borg, Francois Stricher, Robby Nys, Frederic Rousseau, Luis Serrano, The FoldX web server: an online force field, Nucleic Acids Research, 33,2:W382–W388 (2005), https://doi.org/10.1093/nar/gki387
23. Harris, C.R., Millman, K.J., van der Walt, S.J. et al. Array programming with NumPy. Nature 585:357–362 (2020). doi: 10.1038/s41586-020-2649-2.
24. RDKit: Open-source cheminformatics. https://www.rdkit.org; doi:10.5281/zenodo.7357998
25. McKinney, W. Data Structures for Statistical Computing in Python. In van der Walt, S.; S22 Millman, J., eds., Proceedings of the 9th Python in Science Conference. 51–56; doi: doi.org/10.5281/zenodo.3509134
26. Sánchez-Cruz N, Medina-Franco JL, Mestres J, Barril X. Extended connectivity interaction features: improving binding affinity prediction through chemical description. Bioinformatics. 37(10):1376-1382 (2021). doi:10.1093/bioinformatics/btaa982

27. iskandr, openvax/pepdata, (2018), GitHub repository, https://github.com/openvax/pepdata
28. Virtanen, P., Gommers, R., Oliphant, T.E. et al. SciPy 1.0: fundamental algorithms for scientific computing in Python. Nat Methods 17:261–272 (2020). https://doi.org/10.1038/s41592-019-0686-2
29. Pfam: The protein families database in 2021 J. Mistry, S. Chuguransky, L. Williams, M. Qureshi, G.A. Salazar, E.L.L. Sonnhammer, S.C.E. Tosatto, L. Paladin, S. Raj, L.J. Richardson, R.D. Finn, A. Bateman Nucleic Acids Research (2020) doi: 10.1093/nar/gkaa913
30. Hauser, K., Negron, C., Albanese, S.K. et al. Predicting resistance of clinical Abl mutations to targeted kinase inhibitors using alchemical free-energy calculations. Commun Biol 1, 70 (2018). https://doi.org/10.1038/s42003-018-0075-x
31. Christoph Rücker, Gerta Rücker and Markus Meringer. Y-Randomization and Its Variants in QSPR/QSAR.Journal of Chemical Information and Modeling 47 (6), 2345-2357 (2007) doi: 10.1021/ci700157b
32. Alford RF, Leaver-Fay A, Jeliazkov JR, et al. The Rosetta All-Atom Energy Function for Macromolecular Modeling and Design [published correction appears in J Chem Theory Comput. 12;18(7):4594 (2022)]. J Chem Theory Comput. 13(6):3031-3048 (2017). doi:10.1021/acs.jctc.7b00125
33. H. Bekker, H.J.C. Berendsen, E.J. Dijkstra, S. Achterop, R. van Drunen, D. van der Spoel, A. Sijbers, and H. Keegstra et al., “Gromacs: A parallel computer for molecular dynamics simulations” in Physics computing 92:252–256 Edited by R.A. de Groot and J. Nadrchal. World Scientific, Singapore, (1993)
34. Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, Jake Vanderplas, Alexandre Passos, David Cournapeau, Matthieu Brucher, Matthieu Perrot, and Édouard Duchesnay. Scikit-learn: Machine Learning in Python. J. Mach. Learn. Res. 12: 2825–2830 (2011)
35. Wickham H, François R, Henry L, Müller K (2022). dplyr: A Grammar of Data Manipulation. https://dplyr.tidyverse.org, https://github.com/tidyverse/dplyr
36. Wei T, Simko V (2021). R package 'corrplot': Visualization of a Correlation Matrix. (Version 0.92), https://github.com/taiyun/corrplot
37. Wickham H, Girlich M (2022). tidyr: Tidy Messy Data. https://tidyr.tidyverse.org, https://github.com/tidyverse/tidyr
38. Wang, R.; Fang, X.; Lu, Y.; Wang, S. "The PDBbind Database: Collection of Binding Affinities for Protein-Ligand Complexes with Known Three-Dimensional Structures", J. Med. Chem. 47(12):2977-2980 (2004)
39. Sorin Avram, Cristian G Bologa, Jayme Holmes, Giovanni Bocci, Thomas B Wilson, Dac-Trung Nguyen, Ramona Curpan, Liliana Halip, Alina Bora, Jeremy J Yang, Jeffrey Knockel, Suman Sirimulla, Oleg Ursu, Tudor I Oprea, DrugCentral 2021 supports drug discovery and repositioning, Nucleic Acids Research, 49,D1: D1160–D1169 (2021), https://doi.org/10.1093/nar/gkaa997

40. “Anatomical Therapeutic Chemical (ATC) Classification.” World Health Organization, World Health Organization, www.who.int/tools/atc-ddd-toolkit/atc-classification.
41. Kinsella RJ, Kähäri A, Haider S, Zamora J, Proctor G, Spudich G, Almeida-King J, Staines D, Derwent P, Kerhornou A, Kersey P, Flicek P. Ensembl BioMarts: a hub for data retrieval across taxonomic space. Database (Oxford). Published online Jul 23, 2011
42. Karczewski, K.J., Francioli, L.C., Tiao, G. et al. The mutational constraint spectrum quantified from variation in 141,456 humans. Nature 581:434–443 (2020). https://doi.org/10.1038/s41586-020-2308-7
43. Li H, Handsaker B, Wysoker A, Fennell T, Ruan J, Homer N, Marth G, Abecasis G, Durbin R, and 1000 Genome Project Data Processing Subgroup, The Sequence alignment/map (SAM) format and SAMtools, Bioinformatics 25(16):2078-9 (2009).
44. Li H. A statistical framework for SNP calling, mutation discovery, association mapping and population genetical parameter estimation from sequencing data. Bioinformatics 27(21):2987-93 (2011)
45. Genome reference consortium (no date) National Center for Biotechnology Information. U.S. National Library of Medicine. Available at: https://www.ncbi.nlm.nih.gov/grc (Accessed: December 28, 2022)

8. Supplementary Data

Supplementary Data 1: Additional Tables containing information on PDB splits, hyperparameters and descriptors.

Supplementary Data 2: .csv file containing the results of the tested models.