BRNO UNIVERSITY OF TECHNOLOGY
VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

FACULTY OF INFORMATION TECHNOLOGY
FAKULTA INFORMAČNÍCH TECHNOLOGIÍ

DEPARTMENT OF INFORMATION SYSTEMS
ÚSTAV INFORMAČNÍCH SYSTÉMŮ

MULTI-AGENT SYSTEM FOR THE PREDICTION OF THE EFFECT OF MUTATIONS ON PROTEIN STABILITY

MULTI-AGENTNÍ SYSTÉM PRO PREDIKCI VLIVU MUTACÍ NA STABILITU PROTEINŮ

TERM PROJECT
SEMESTRÁLNÍ PROJEKT

AUTHOR **Bc. ONDŘEJ DOSEDĚL**
AUTOR PRÁCE

SUPERVISOR **Ing. MILOŠ MUSIL, Ph.D.**
VEDOUCÍ PRÁCE

BRNO 2022

Department of Information Systems (DIFS) Academic year 2021/2022

Master's Thesis Specification

24374

Student: **Doseděl Ondřej, Bc.**
Programme: Information Technology and Artificial Intelligence
Specialization: Information Systems and Databases
Title: **Multi-Agent System for the Prediction of the Effect of Mutations on Protein Stability**
Category: Biocomputing
Assignment:

1. Study the problematics of protein engineering and protein stability.
2. Study various machine learning techniques and their usability for the prediction of the effect of mutations on the protein's stability.
3. Obtain protein stability data from the available sources (protein stability datasets, scientific literature).
4. Filter and clean collected data. The student will have to deal with various issues such as non-fitting indexes, errors and overlaps in the data, their imbalance and low diversity, or various inconsistencies.
5. Split the data into several categories based on their properties (physico-chemical properties, position in the protein, etc.).
6. Calculate/obtain sequence and structural features usable for the training of the predictor such as physico-chemical matrices, conservation, or secondary structure.
7. Train the multi-agent system that would combine general and specialized predictors.
8. Evaluate constructed system on the independent testing dataset.

Recommended literature:

- Mazurenko, S., 2020: Predicting Protein Stability and Solubility Changes Upon Mutations: Data Perspective. ChemCatChem 12: 1-10.
- Stourac, J., Dubrava, J., Musil, M., Horackova, J., Damborsky, J., Mazurenko, S., Bednar, D., 2020: FireProtDB: Database of Manually Curated Protein Stability Data. Nucleic Acids Research 49: D319-D324.
- Musil, M., Konegger, H., Hon, J., Bednar, D., Damborsky, J., 2019: Computational Design of Stable and Soluble Biocatalysts. ACS Catalysis 9: 10331054.

Requirements for the semestral defence:

- First four items of the assignment.

Detailed formal requirements can be found at https://www.fit.vut.cz/study/theses/

Supervisor: **Musil Miloš, Ing., Ph.D.**
Head of Department: Kolář Dušan, doc. Dr. Ing.
Beginning of work: November 1, 2021
Submission deadline: May 18, 2022
Approval date: October 22, 2021

Abstract

Proteins are building blocks of every living organism, as they are responsible for multiple crucial functions. They consist of amino acids chains and these chains can be changed. The change is called mutation. Mutation can happen naturally, or created in laboratory. The aim of this thesis is to present novel methology for determining protein's stability upon mutations. It consists of two models. The first model is multi-agent system which handles classification into two classes, i.e, stabilizing and destabilizing. The best model gained 0.7 ACC and 0.41 MCC. The second part dealt with predicting exact values of ΔΔG where an Extreme Gradient Boosting model was created which managed to gain 1.67 RMSE with 0.53 PCC. New datasets for training and validation, which are truly independent, were also introduced in this thesis.

Abstrakt

Proteiny jsou základním stavebním blokem všech žijících organismů, kde jsou zodpovědné za mnoho důležitých procesů. Jsou složeny z řetězců aminokyselin. Tyto řetězce mohou být jakkoliv změněné. Tomuto procesu se říká mutace a může být samovolná nebo indukovaná v laboratoři. Cílem této práce bylo vytvoření nových modelů pro určení stability proteinů. Skládá se ze dvou modelů. První model je multi-agentní systém pro klasifikaci stability proteinů. Nejlepší multi-agentní systém získal přesnost 0.7 a 0.41 MCC. Druhá část se zabývala predikcí konkrétních hodnot ΔΔG, kde byl vytvořený Extreme Gradient Boosting model, který získal 1.67 RMSE a 0.53 PCC. Součástí této práce byly představené 2 datasety, které jsou na sobě plně nezávislé, použitelné pro trénování a validaci modelů.

Keywords

protein, machine learning, protein stability, classification, regression, mutations, Extreme Gradient Boosting

Klíčová slova

protein, strojové učení, stabilita proteinů, klasifikace, regrese, mutace

Reference

DOSEDĚL, Ondřej. *Multi-Agent System for the Prediction of the Effect of Mutations on Protein*
Stability. Brno, 2022. Term project. Brno University of Technology, Faculty of Information Technology. Supervisor Ing. Miloš Musil, Ph.D.

Rozšířený abstrakt

Proteiny jsou základní stavební a funkční jednotkou všech buněk živých organismů. V organismech plní mnoho různých funkcí například replikaci DNA, transport molekul, regulaci hormonů a katalýzu reakcí. Skládají se z řetězců aminokyselin spojených peptidovou vazbou.

V průběhu buněčného dělení a genové exprese, která zahrnuje procesy replikace DNA, transkripci a translaci, může docházet k chybám, a tedy k vzniku mutací. K těmto procesům dochází spontánně nebo mohou být indukované uměle. Mutace společně s rekombinací a genetickým driftem mají velký dopad na evoluci.

Mutace mohou narušit nebo zvýšit stabilitu proteinu. Stabilní proteiny jsou odolnější vůči extrémním podmínkám a jsou lépe využitelné v biotechnologiích a průmyslu. Stabilita lze počítat dvěma způsoby. První je změření rozdílu teploty tání mezi zmutovaným a původním proteinem. Větší teplota tání znamená větší stabilitu. Druhý způsob je pomocí Gibbsovy volné energie (ΔΔG).

Navrhovaná metoda se dá rozdělit do dvou částí. V první části byl vytvořen multiagentní systém, který řeší, jestli je mutace stabilizovací či destabilizovací. Tento model se skládá z Náhodných stromů a metod podpůrných vektorů. Každý tento model je natrénovaný na konkrétní části datasetu. V druhé části byl vytvořen model, který má předpovědět konkrétní hodnotu ΔΔG.

Pro trénování a validaci těchto modelů byly vytvořené trénovací a validační datasety. Trénovací dataset vznikl extrakcí dat z ProThermDB a jeho rozšíření z Loschmidt laboratoří. Nejdříve se muselo ověřit, že data získaná exportem jsou správná. Jako první došlo k ověření ID z Uniprotu a PDB databází, které tento dataset obsahují. To bylo provedeno pomocí vytvoření řetězce aminokyselin z obou zdrojů a jejich porovnáním. Následně došlo k ujištění, že mutace sedí na řetězec získaný z PDB databáze. Následně došlo k rozšíření dat o index hydrofobicity a objemu. Tato data byla získaná z AAIndex. Pro zjištění, zda je mutace stabilizující nebo ne, jsme se zaměřili na sloupečky pro ΔΔG a ΔTm, kde pozitivní ΔTm znamená stabilizující mutaci. V trénovacím datasetu, byly záporné hodnoty ΔΔG stabilizující, u validačního to bylo obráceně.

Validační dataset vznikl z jiného zdroje a bylo u něho potřeba zajistit, že je určitě nezávislý na trénovacím datasetu. To bylo dosaženo odstraněním všech mutací, které jsou obsažené v trénovacím datasetu. Tj. všechny mutace co obsahují pro stejný protein záměnu na konkrétní pozici jedné aminokyseliny za druhou. Tímto bylo dosaženo, že jsou datasety nezávislé a mohou být využity pro validaci.

Pro ohodnocení datasetů byly pro klasifikaci primárně měřeny tři hodnoty. První je přesnost, druhá je Matthewsův korelační koeficient (MCC) a poslední je Oblast pod trénovací křivkou křivkou. Nejdůležitější je MCC, jelikož je lépe použitelný na nevyvážené datasety. Pro hodnocení regrese se používá Pearsonův korelační koeficient a základ střední kvadratické chyby (RMSE).

Pro klasifikaci byl nejdříve použit jednoduchý model, pro ověření funkčnosti datasetů a schopnosti modelu se na daném datasetu natrénoval. První trénováníí Náhodného stromu ukázalo -0.10 MCC s přesností 0.48. Takové MCC znamená, že neexistuje žádná korelace mezi trénovacími daty. Aktuálně dataset obsahuje jak ΔΔG tak ΔTm mutace. Následně došlo k rozdělení datasetu na dataset obsahující pouze ΔΔG a ΔTm. Dataset obsahující jenom ΔTm znovu natrénoval model s -0.08 MCC a přesností 0.56. Toto značí, že problémy v datech jsou v této části. Když byly použité pouze ΔΔG, došlo k výpočtu 0.35 MCC a přesnost kolem 0.70. Toto jsou dobrá data a můžeme začít trénovat multiagentní systém.

K získání nejlepší kombinace multiagentního systému došlo experimentálně, kde jsme postupně upravovali, jaký model bude použitý na jaký dataset, váhy jednotlivých tříd a váhy výsledných modelů. Nejlepší model, který byl schopný predikovat 2 třídy dosáhl přesnosti 0.7 a 0.4 MCC. Následně došlo k trénování tří tříd, kde byla vytvořena neutrální třída, která vznikla z intervalu -1 až 1 kcal/mol a poté -0.5 až 0.5kcal/mol, jelikož 0.5 kcal/mol bývá chyba metod, které se používají pro měření. S touto změnou měl model nejlepší přesnost a to 0.69 a 0.41 MCC.

Pro trénovaní regrese došlo ke spojení jak trénovací tak validačního datasetu obsahujícího pouze $\Delta\Delta$G hodnoty. Toto spojení bylo kvůli zajištění větší obecnosti modelu, jelikož po filtraci tyto datasety obsahují rozdílné rodiny proteinů, kde se konkrétní hodnoty mohou chovat odlišně. Následně byl tento dataset náhodně rozdělený na trénovací a validační část. Různé modely byly použity pro natrénování. Nejlepších výsledků dosáhl Extreme Gradient Boosting model s 1.67 RMSE a 0.53 PCC.

Tyto hodnoty jsou na úrovni aktuálně nejmodernějších technik. Největší problém u porovnávání rozdílných metod je nedostatek a nevyváženost dat. V ideálním případě, by se měly všechny metody porovnávat na jediném datasetu, který je úplně nezávislý od všech dat, které byly použité pro trénování všech modelů. Jelikož toto není zatím možné, tak může docházet k přetrénovanosti a přeceňování přesnosti jednotlivých modelů.

Tato práce by se následně mohla rozšířit o vytvoření jednoduché webové aplikace, ve které by mohly běžet oba nejlépe natrénované modely pro širší použití veřejnosti a pro další ověření funkčnosti.

Multi-Agent System for the Prediction of the Effect of Mutations on Protein Stability

Declaration

I hereby declare that this Terms's thesis was prepared as an original work by the author under the supervision of Ing. Miloš Musil. I have listed all the literary sources, publications and other sources, which were used during the preparation of this thesis.

. . . .
Ondřej Doseděl
May 16, 2022

Acknowledgements

I would like to thank my supervisor Ing. Miloš Musil, Ph.D for providing all the necessary information and guidance. I would also like to thank all my family, friends and colleagues who supported me through all this journy.

Contents

List of Figures

Chapter 1

Introduction

Proteins are the main building blocks of every living organism, and they perform many different functions such as DNA replication, transporting molecules between parts of living structures, cell signalling and others. They consist of one or more chains of amino acids connected by peptide bonds. Sequence of amino acids and protein spacial structure determines protein function.

The main aim of this thesis is protein stability and the development of the computational system that would be able to predict, whether mutation is stabilizing or not. Stable proteins are able to survive in extreme temperatures, different pH, etc. As proteins are crucial for every living creature, many different methods were introduced to handle protein stability. As mutations can be done in laboratory, in ideal case, all possible mutations would be tested. However, such an approach would be laborious and time demanding. Because of that, there is higher demand for precise computational methods.

In this thesis, a new machine learning method for protein stability is introduced. This method can be divided into two parts. The first part is classification, where the main idea is to use multi-agent system, combining Random Forest and support vector machines classifiers. Each model is trained on a different subset of the training dataset, which was also constructed in this thesis. For regression, different models were tested to find the best performing model on our dataset.

To validate the performance of the classification model, a truly independent validation dataset is presented. As these datasets contain different protein families, for regression purposes, they will be combined into one and then randomly split for training and validation. This was done to create a more robust model, that can better perform on different protein families, as the exact behaviour can differ for each protein family.

1.1 Organization of the Thesis

The Thesis is organized as follows. In Chapter 2, general introduction to proteins is described with some additional details about proteins mutation. In Chapter 3, protein stability is described in more detail with protein folding and the explanation, how protein stability is calculated. Chapter 4 deals with stability prediction and how methods can be divided into different groups. Chapter 5 provides a general introduction to machine learning, with more detailed focus on classification and regression methods. The process of the construction of the training and validation dataset is described in the Chapter 6. The design of the experiments with some implementation details and initial experiments using simple Random Forest are described in Chapter 7. The results of presented experiments and comparison with the state-of-the-art methods can be found in Chapter 8.

Chapter 2

Proteins

Proteins are one of the most versatile molecules in living organisms. They preform various functions such as creation of mechanical structure, transport substances in the blood or lymph throughout the body, DNA replication, catalysis of regulatory or metabolic reactions, immune response, proteins storage and many others. Proteins are polypeptides that contain from thirty to several thousand amino acids. The sequence of amino acids and protein spacial structure determines protein's function. For this reason, amino acids alternation is the driving force for evolution at molecular level.

2.1 Amino acids

The amino acids are the fundamental building block of proteins. Amino acids contain a central carbon atom, which is attached to hydrogen, carboxyl group (COOH) and amino group (NH_2). This is shown in Figure 2.1. Generally, there are more than 500 amino acids, however, only 20 amino acids are biogenic for organism (21^{th} is selenocystein). They differ in side chains (R) which determine chemical properties of amino acids and proteins. One molecule of protein consists of chained amino acids connected using peptide bonds between carboxyl group of one amino acid with the amino group of the other. The chain is ended on one side with carboxyl group and amino group on the other. Based on their properties and structures, amino acids are divided into six groups [53].

1. **Aliphatic side-chains** Glycine (Gly), Alanine (Ala), Valine (Val), Leucine (Leu), Isoleucine(Ile)

2. **Acidic groups with a carboxyl or amine group on the side chain** Asparagine (Asn), Aspartate (Asp), Glutamine (Gln), Glutamate (Glu)

3. **Basic groups with amine group on the side chain** Arginine (Arg), Lysine (Lys)

4. **With aromatic nucleus or hydroxyl group on the side chain** Histidine (His), Phenylalanine (Phe), Serine (Ser), Threonine (Thr), Tyrosine (Tyr), Tryptophan (Trp)

5. **With sulfur on the side chain** Methionine (Met), Cysteine (Cys)

6. **With secondary amine** Prolin (Pro)

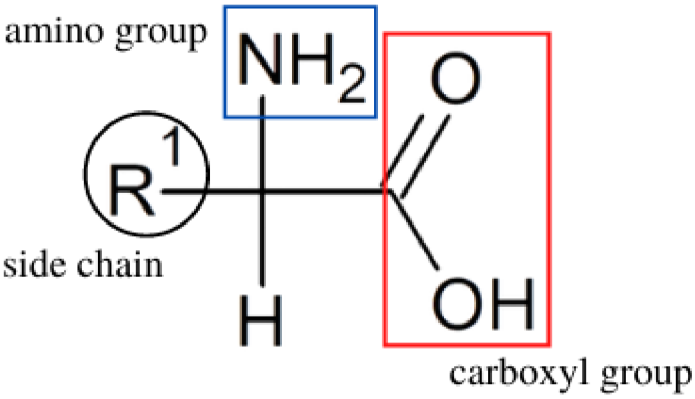


Figure 2.1: Graphical structure of amino acid.

2.2 Transcription and Translation

Central dogma of molecular biology explains the flow between nucleic acids and proteins. This is shown on Figure 2.2. From DNA (deoxyribonucleic acid) to RNA (ribonucleic acid) to protein. DNA is divided up into functional units - genes, which contain information needed to make a functional protein in a process called gene expression.There are differences between DNA and RNA. RNA molecules do not include the base thymine (T), but includes uracil (U).

Gene expression includes two main process, transcription and translation. Regulation of this process have effect on cell structure and function.

DNA replication is a fundamental step in central dogma, because it is essential for cell division during growth. It is producing two identical replicas of DNA from one original DNA molecule.

In transcription, one strand of the gene's DNA is copied into an RNA molecule. It involves rewriting.Transcription of the template strand produces mRNA that nearly matches the other strand of DNA.

Translation takes place inside of ribosomes, which is a molecular machine for building polypeptides. The nucleotide sequence of the mRNA (messenger RNA) is translated into a sequence of amino acids. Nucleotides of the mRNA are read in a group of three (triplets) called codons. One codon is a „start“ codon to signal the start of a polypeptide. There are three variations of „stop“ codon signal that is located on the end of the polypeptide.This set of relationships is known as the genetic code.

Special transfers are reverse transcription and RNA replication. Reverse transcription is a transfer from RNA to DNA. In this process, the enzymes called Reverse Transcriptase are included. RNA replication is the process of copying one RNA to another RNA, which is typical for viruses.

2.3 Structure of proteins

Proteins can be described by four levels of structure, as can be seen in Figure 2.3.

Primary structure is determined by the sequence of amino acids in the peptide bond defining protein.

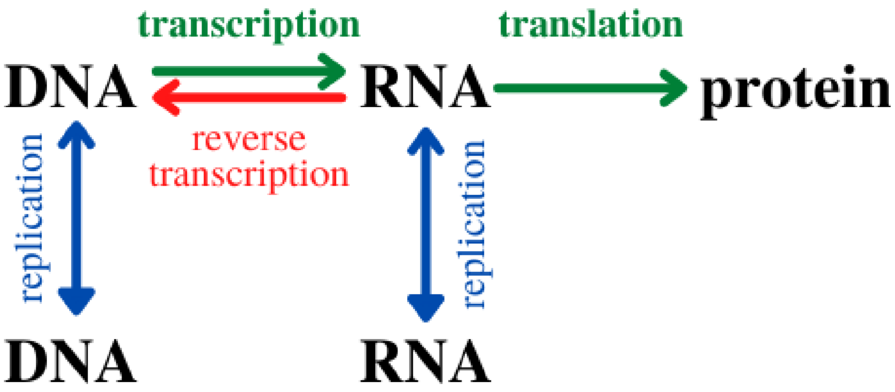


Figure 2.2: The central dogma of molecular biology

.

Secondary structure shows reoccurring molecules, which are maintained by hydrogen bonds between carbonyl oxygens and amino hydrogens of the peptide bonds. Primarily, it is α-helix, β-sheets and random coil, which is not organized. The chain in α-helix is organized into helix and stabilized using hydrogen bonds as shown in Figure 2.4. Approximately 3.6 amino acids are required for one complete turn. The β-structure can be divided into β-strands and β-sheets, where β-sheet consist of several β-strands. β-strands are part of the chain, which is almost fully extended. β-sheets contain 2 sections of the chain in parallel, and they are stabilized using hydrogen bonds.

Tertiary structure is three-dimensional structure in the polypeptide. Order of amino acids and their chemical properties has the biggest impact on the final conformation. The process of protein folding connects secondary structures (α-helix, β-sheet) with turns and random coils to create a specific shape of globular molecule, which is stabilized by ionic interaction, hydrogen bonds, disulphide bonds and others.

Quaternary structure refers to the spatial relationship of the polypeptide chains forming the tertiary structure of a protein. Those are linked to form oligomer molecules.

2.4 Mutation

Mutation is a random or targeted change in the DNA sequence. Mutations have significant impact on process of evolution as if no mutations would happen, evolution would be limited to recombination or reshuffle of already existing genes. New genes that would disadvantage specie would be deleted by evolution. Mutation describes all changes of the genetic information which are not caused by segregation or recombination of already existing genome types. We can distinguish 3 kinds of mutations based on their place of creation [37], i.e., gen mutations, chromosome mutations and genome mutations.

Gen mutations modify information which is stored in genes. They change the nucleotide order. For prediction needs, this is the most important type.

Chromosome mutations change the number of chromosomes or their structure.

The most of the time, **genome mutations** add or reduce a complete set of chromosomes to the genome.

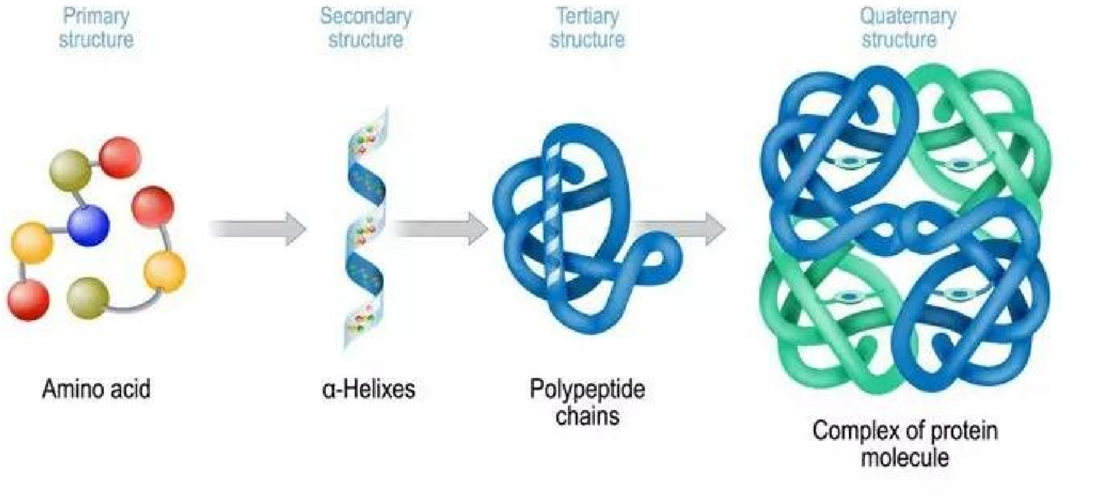


Figure 2.3: Primary, secondary, tertiary and quaternary structure of proteins. Figure taken from [72].

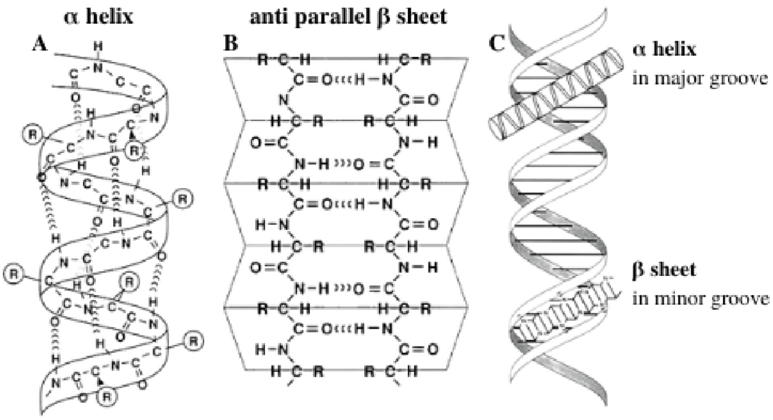


Figure 2.4: Structure of a typical α-helix. Figure taken from [101].

The occurrence of a mutation does not necessary means that it will affect the function of the protein or the viability of the organism. Only a small part of genetic code consists of proteins that encode genes. In humans, only about 1.5% [104] of the genetic code represents the protein encoding genes. Rest of the mutations happens in non encoding areas. However, these mutations can influence the creation of other proteins or stop it completely.

2.4.1 Gen Mutation types

Gen Mutations can be split into 3 basic groups by their mechanism of creation [37].

Substitution changes one or more nucleotides for different ones. However, the length of protein remains the same. This change won't affect transcription or translation. Generally speaking, substitution is less damaging than insertion or deleti on.

Insertion adds one or more nucleotides, which increases the length of the original sequence. The number of inserted nucleotides is very important, as 3 of them would add 1 new amino acid, while one or two would move a whole reading frame. The frame is used when dividing the sequence of nucleotides into non-overlapping triples. These triples equate to individual amino acids.

Deletion is similar to insertion. Deletion is removing one or more nucleotides and therefore, the length of the sequence is changed. As in the previous case, the deletion of the multiple of three nucleotides would cause the smallest change in the resulting sequence of amino acids.

If the mutation occurs in the coding area of the gene, we can differentiate mutations [37] by their final effect on the translated protein as Synonymous, Nonsynonymous, Nonsense and Frameshift mutations.

Synonymous mutation, as the genetic code is degenerated, changes some nucleotides in a codon. However, this will lead to translation to the identical amino acid and spatial arrangement of the protein will stay the same. It would look like no mutation happened. **Nonsynonymous** mutation is the opposite of the above. Changes to nucleotides in the codon will result in the change of amino acids in the protein sequence. **Nonsense** mutations are those which create a STOP codon that terminates translation earlier than expected. **Frameshift** mutations change the reading frame and as a result, it will change amino acids and would lead to earlier identification of STOP codon and the premature termination of translation.

2.4.2 Creation of mutation

Based on their origin, we can split mutations to spontaneous or inducted. Spontaneous are created by error in replication and reparation mechanisms of the DNA. Replication of DNA is extremely precise, and it is assumed that only about 1 mutation will happen in 10^7 nucleotides. There are also self-repairing mechanisms in replication that lower this error rate to $1 : 10^9$ [1].

Inducted mutations are artificially created mutations, where genes are put in contact with mutagens in the environment. These mutagenes can be split into three different groups, i.e., physical, chemical and biological.

Physical mutagenes include electromagnetic radiation, such as ionizing radiation (alfa, beta and also gamma), X-ray and UV light. The degree of damage is directly proportional to the absorbed radiation dose.

Chemical mutagenes are substances that can damage DNA, for example demethylation. Most chemical mutagenes are alkylating agents and azides.

Biological mutagenes are caused by action of transposons, viruses (oncogenic or retroviruses) or bacteria.

Chapter 3

Protein stability

Protein stability is one of the key properties used to determine the applicability of protein under harsh conditions.The stable protein is able to withstand extreme temperatures or the presence of denaturing agents [10]. Furthermore, stable proteins are usually positively correlated with expression yields [35]. Because of that, the interest in the improvement of protein stability is increasing as it can enhance the utility of proteins in various biotechnological, industrial and medical applications.

Stability can be calculated as the difference of intramolecular interactions and conformational entropy [42]. This will determine if the protein will stay in its native folded conformation. Mutations can be used to strengthen or disrupt stability of the protein.

The main aim of this chapter is to describe various physical and biochemical forces that participate in protein folding. The last part of this chapter is describing metrics for calculation of protein stability.

3.1 Stability of folded protein

In 1969, Cyrus Levinthal declared that folding of the protein from primary to tertiary structure cannot be random, due to the high number of degrees of freedom in an unfolded polypeptide chain. According to his estimation, if there would be only a small protein consisting of 101 amino acids with only single bond between each residue and each bond would have only three possible configurations, this would result into $(5 * 10^{47})$ different conformations. If the protein can sample (10^{13}) different bond configurations, it would take (10^{27}) years to sample all possible configurations [123]. This would mean that folding of each protein would take extremely long time, however, this process is almost instant for small proteins, and it takes only a couple of minutes for the most complex proteins. This is called Levinthal's paradox.

While this paradox is in contradiction with the possibility of random protein folding, it is supported by Afinsen's thermodynamic hypothesis. This proves that native structure of globular protein in standard environment is only determined by the amino acid sequence [5]. This means that the process cannot be random. In fact, it has to be deterministic.

Both Levinthal's and Afisnesn's claims acknowledged the existence of powers governing protein folding. These powers can be differentiated on covalent and non-covalent interactions, together with the factor of conformational entropy. Covalent bonds are very strong and stable under standard conditions. Covalent interactions are created by sharing electrons between atoms in the polypeptide chain. Because of that, these interactions are most

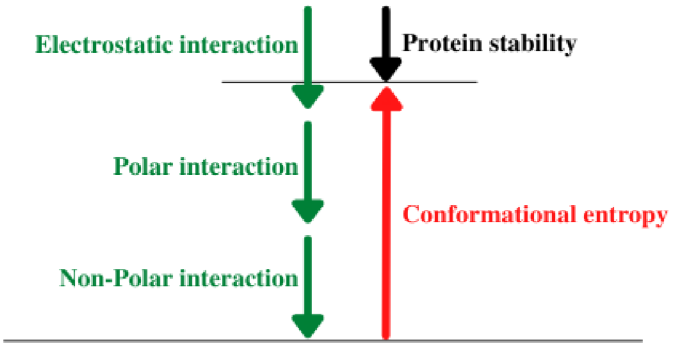


Figure 3.1: Major forces in protein stability. Protein stability can be calculated as the difference of sum of all interactions and the conformational entropy.

important in governing the creation of primary structure of protein. Non-covalent interactions are notably weaker and are the main driving force in the construction of secondary, tertiary and quaternary structures of the protein. They can be also divided into polar, non-polar and electrostatic interactions [42].

Polar interactions can be split into aromatic intersections and hydrogen bonds. Aromatic interactions are created between aromatic rings of aromatic residues (Tyrosine, Tryptophan, Phenylalanine and Histidine) controlled by their πelectrons. The distance must be between 4.5 and 7 Ångström (Å) [6]. Polar residues (Histidine, Cysteine, Threonine, Tryptophan, Aspartic acid, Serine and Tyrosine) are able to share hydrogen attached to an electronegative atom. This can occur at distance around 3 Å. Hydrogen bonds are also very important during the creation of the secondary structures.

Non-polar interactions are crucial for creating tertiary structures. They are weaker, short forces between all atoms in protein. They are not notable beyond 5 Å. Hydrophobic effect also influence tertiary structure, because of the unfavourable entropy of water molecules around hydrophobic residues (Proline, Phenylalanine, Valine, Methionine, Alanine, Leucine and Isoleucine). These residues tend to aggregate, creating the hydrophobic core of the protein. This leads to the increase in the hydrogen boding between water molecules and minimizing the area between non-polar residues and water.

Electrostatic interactions are between cations and anions in charged residues (Lysine, Arginine, Asparagine, Glutamine and Histidine). According to Coulomb's law, their strength decreases with (r^2). They also depend on the environment as they are influenced by salt concentration, pH and permittivity.

Conformational entropy is connected with a number of conformations of the protein's structure. It is a significant contributor to energetic stabilization of the denatured state. It is a countering force to the sum of electrostatic interactions, as can be seen in Figure 3.1. Conformational entropy gain given by secondary structures are way lower than entropy gain by random coils. Due to this reason, proteins with high concentration of the random coils are mostly less stable than proteins with higher number of secondary elements.

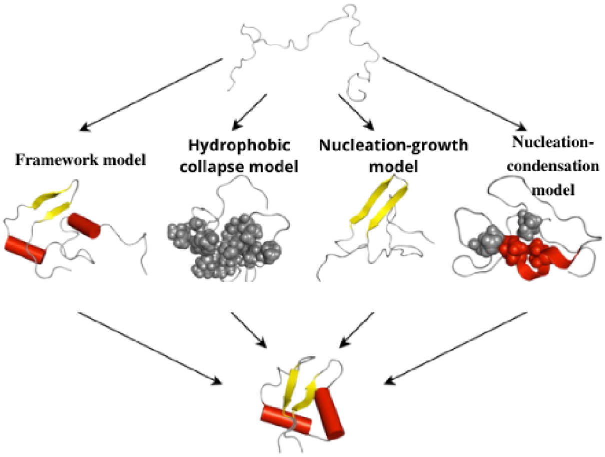


Figure 3.2: The visualization of various protein folding mechanisms. Adapted from [82].

3.2 Protein folding

There are several different ways how to describe the process in which non-covalent interactions transform the polypeptide chain into tertiary structure [82]. The nucleation-growth model was first used to describe folding. It presumed the continuous growth of the tertiary structure. This model was dismissed as folding intermediates were not accounted. After that, some more models were created, as showed in Figure 3.2.

Hydrophobic collapse model means that the protein collapses rapidly around its hydrophobic side-chains. Stable secondary structures start to grow only in the collapsed state.

Nucleation-condensation model suggest the existence of a metastable nucleus. Secondary and tertiary structures are formed in parallel. Folding is triggered when sufficient number of tertiary structure interactions occur. Following this, rapid condensation of native structure occurs.

In the framework model, secondary structure is folded at the beginning of the proteins folding. After that, the coalescence of the secondary structural units into the structure of the native protein occurs.

3.3 Quantification of protein stability

There are several ways how to calculate protein stability. The two most common are Gibbs free energy and melting temperature [42].

3.3.1 Gibbs free energy

Gibbs free energy (or Gibbs energy) is a thermodynamic potential that can be used to calculate the maximum reversible work that may be performed by a thermodynamic system

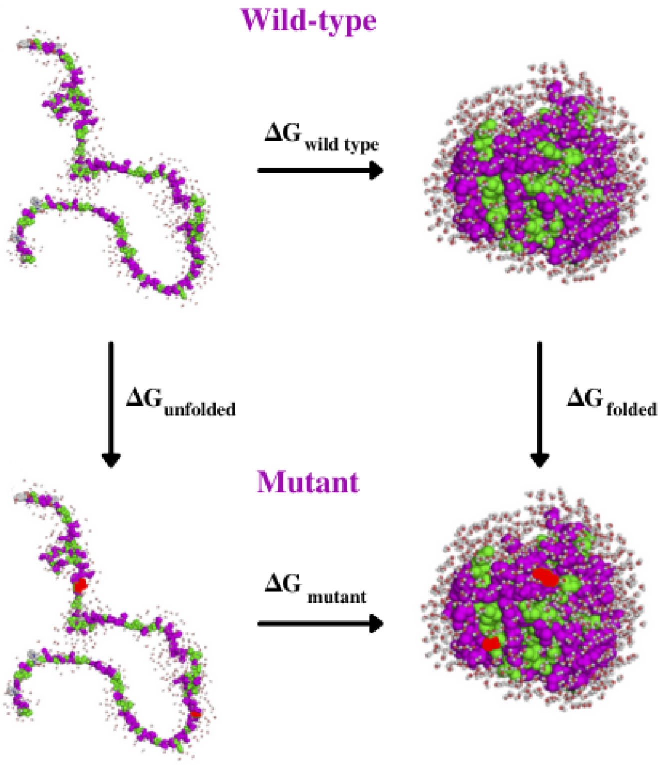


Figure 3.3: Thermodynamic cycle used for computation of ΔΔG. The change of the Gibbs free energy upon mutation is calculated as a difference of energy upon folding of the wild-type and mutant protein. In the figure, the respective mutation sites have been coloured in black for wild-type and red for the mutant protein [80].

at a constant temperature and pressure. It is defined as

$$G = H - TS, \tag{3.1}$$

where H is the enthalpy, T is the temperature, and S stands for the entropy. It is measured in joules in SI. In biology, calories are often used. Stability of protein is measured as the difference between free energies of the folded and unfolded state (ΔG).

$$\Delta G = H_{folded} - H_{unfolded}, \tag{3.2}$$

If we would like to measure the effect of mutation on protein stability, so-called change of Gibbs free energy is measured (ΔΔG). This is the difference between ΔG of the mutated and wild-type protein.

$$\Delta\Delta G = \Delta G_{mutant} - \Delta G_{wild}, \tag{3.3}$$

Usually, the negative value of ΔΔG (kcal/mol) signifies that the mutation improved protein's stability. However, this is not standardized and in some studies the stabilizing values would be positive. This is crucial to check when gathering data for a dataset, as all sources should be using the same calculation. ΔΔG computation is based on thermodynamic cycle showed in Figure 3.3.

3.3.2 Melting temperature

Different way of quantification of protein stability is by measuring the melting Temperature (T_m). The definition is as follows:

$$\Delta G_{folding}(T_m) = 0, \tag{3.4}$$

To put it in other words, it is the temperature at which free energy of the unfolded and folded states is equal, while half of the population is folded and the other is unfolded. Similar to Gibbs free energy $\Delta(T_m)$ means the change of temperature upon mutation. There is strong correlation between melting temperature and Gibbs free energy (Pearson correlation is approximately 0.71 [92]). However, the transformation from one to other is not linear, so there is no simple equation, how to estimate one value from the other.

Chapter 4

Stability Prediction

Ideally, saturation mutagenesis of each residue in the protein sequence would be experimentally validated. Unfortunately, this is an almost impossible task, as these experiments are laborious and costly. A standard protein containing 300 amino acids would have over 5,000 single-point mutations. However, a single-point mutation has negligible effect on stability ($< 2kcal/mol$)[119, 44] and therefore, a combination of mutations is required to have a significant impact on protein stability [17]. However, as mutations can have synergistic or antagonistic effects when combining multiple mutations, the stabilization is not guaranteed to sum up in the additive manner. The mutations are considered to be synergistic if their combined effect on stability is greater than the sum of individual effects. Antagonistic effect is the opposite of synergistic. Usually, synergistic effect means that new physico-chemical interaction was created. Some examples of these interactions are a disulphide bridge between two cysteine residues or a salt bridge between anionic carboxylate and cationic ammonium. Antagonistic effect usually creates clashes between the side chains of the mutated or original residues. This could even completely prevent protein from a successful folding. Usually, antagonistic effects are hard to detect and require further validation.

When 100 potentially stabilizing mutations are applied, almost 5,000 experiments are required to fully evaluate all double-point mutants. The number of experiments is exponentially increasing with every new mutation. Because of that, fast and accurate computational methods are needed for rapid evaluation of protein stability after each mutation. Those tools can be utilized for the prioritization of specific mutations used in laboratory experiments. Generally, the computational methods can be divided into four categories [80]. The first category are Force-field methods, which calculate ΔΔG using models of molecular mechanics.The second is Phylogenetic analysis. This method utilizes evolutionary information obtained from the set of homolog sequences. Methods in the third category are based on Machine learning, where a model is created using stability data gathered from previous experimental validation. The last category consist of Hybrid methods and meta-predictors, where combination of single or more approaches together aim to provide more reliable and robust results.

4.1 Force-field methods

4.1.1 Principles

Calculations do not rely on the availability of experimental data, as they are connected with our current understating of the physico-chemical properties of amino acids and their description. Generally, force-field is a description of all bonded and non-bonded interactions in the protein [76, 83]. These interactions are used in the energy-field equation to estimate potential energy of the system [65]. The most accurate methods are relying on molecular dynamics (MD) or Metropolis Monte Carlo simulations. However, both methods need an enormous amount of computation power and can be used only for a small number of mutations [100]. Many heuristic approaches were created to overcome this bottleneck, however, complex analysis can be done with the usage of simulation-independent stability predictors. These predictors can be divided into three categories [43, 74].

Statistical effective energy functions (SEEFs) are used for rapid analysis, as stability changes can be predicted over the entire sequence of average enzyme in a matter of minutes [31, 32]. The individual terms in the energy-field equations are derived from available datasets. For the overall energy function, each descriptor can be extrapolated using and effective potential [31, 70].

In the **physical effective energy functions (PEEFs)**, the terms of the equations are calculated using the simplification of physical laws. However, the calculations are quite complex and so the computation of the equation for single mutant can take up to several days. However, they are precise and versatile and capable of predicting behaviour under non-standard conditions [2].

Empirical effective energy functions (EEEFs) represent the bridge between SEEFs and PEEFs as they include both statistical and physical terms in energy-field equations, where weights and parameters are used to match experimental data [43, 74]. Derived data are originated from experiments undertaken under normal conditions. Using this, EEEFs are a usable compromise between accuracy and computational time [99], however, they are restricted to the environmental condition of the original experiments [60, 26].

Even though the accuracy of force-field-based method is unsatisfactory, it is still considered to be the most powerful tool for prediction of protein stability. Mostly it is due to imbalances in the force fields, insufficient conformational sampling and the problems with current data sets [60, 30, 28, 110, 29, 59]. Their accuracy is strongly dependent on availability of quality tertiary structure. For proteins without resolved tertiary structure, it relies on the accuracy of the modelling tools. Furthermore, most of the proteins in PDB database ($> 90\%$[107]) are obtained using X-ray crystallography, which might not reflect global minimum of the native state [34] and could be misleading for a comprehensive prediction of stability [26, 62].

4.1.2 Software tools

The Rosetta suite [59] is considered to be the state-of-the-art for predicting protein stability. The suite is very versatile and can be used for many different tasks. It consists of many modules (molecular simulations, stability predictions, ab-initio modelling etc). The Rosetta Design is a general tool used for protein design and contains a protocol for identification of stabilizing mutations in protein. The result is provided in Rosetta energy units, which are automatically converted to $\Delta\Delta G$ values. There is a standalone module developed on Rosetta Design called ddg_monomer, which directly produces $\Delta\Delta G$ [59].

An example of software in PEEFs category is Eris [122] which is implemented using custom Medusa modelling suite. It was tested on large dataset (> 500 samples). Compared to other methods, Eris is not specially trained on stability data, so it can be correctly used for wider range of proteins. Eris models backbone flexibility, which seems to be crucial for predicting stability of small-to-large mutations.

PopMuSic method is one of the most popular, where the force-field equation is calculated using thirteen statistical potentials which are derived from the known database [32]. Dataset used for training and validation consist of 2,648 single point mutations. There is a very similar method called HotMuSic [92], however, it was parametrized for predicting $\Delta(T_m)$ instead of ΔΔG. Five more temperature-dependent potentials were added to the force-field equation. They were extracted from the thermostable and mezostable proteins.

The last category is trying to find the fine line between prediction accuracy and time demands. This is usually done by using both statistically derived terms with calculated force-filed equations. One of the examples is CUPSAT [85] which uses PISCES [114] to obtain tertiary structures. Atom and torsion angles potentials are derived from these structures. Boltzmann's energy is calculated from the radial pair distribution of amino acid atoms. To calculate favourable energy values for the neighbouring orientations of the observed torsion angles combinations, a Gaussian apodization function is applied.

Generally, PEEFs are still more accurate. However, SEEFs are performing well in comparison with most of the machine learning methods and are way faster than PEEFs. Thanks to that, SEEFs and EEFs are mostly used due to acceptable trade-off between computational power and accuracy.

4.2 Phylogenetic Analysis

4.2.1 Principles of Methods Based on Phylogenetic Analysis

These methods take advantage of the information hidden in the set of homolog sequences. The biggest advantage of this approach is that they do not require tertiary sequence. Because of that, they can be employed on the majority of known sequences (about 200 million in Uniprot [7] instead of 100 thousand in PDB [107]). However, they cannot be applied on the families with low representation of sequences in the databases. In the recent years, thanks to the next-generation sequencing methods, these families are shrinking as the number of sequences in databases almost doubles every three years. Most used methods are consensus Design and ancestral sequence reconstructing.

Consensus desing (CD)

In the beginning, a compact multiple-sequence alignment (MSA) is built using a small number of homolog sequences. It allows for computing frequency distribution of every amino acid in the sequence [103]. Only positions where one or just a few amino acids are more prevalent are conserved, as they were not changed very often during the course of evolution. The core assumption of CD is that these conserved position are crucial for protein's function. The most frequent amino acid on these positions have the highest probability to be stabilizing [103, 106, 66, 71, 88, 56]. When designed sequence differs in the conserved regions with the most dominant amino acids, CD can be utilized.

Ancestral sequence reconstruction (ASR)

This method explores evolutionary history of sequence to recreate protein's evolutionary trajectory [47, 116]. The first intention of this method was to study molecular evolution. ASR is widely used in evolutionary biology, as it is able to reconstruct the long-extinct genes and organisms of their ancestors from the current days sequences. The start of ASR is similar to CD as the MSA is constructed from the set of homolog sequences. The difference is that ASR considers evolutionary information from the phylogenetic tree instead of simple analysis in CD. Bayesian interference [51] and maximum-likelihood [121, 102] were designed to interfere the ancestral sequences from MSA.

4.2.2 Software tools

Many tools were build to create both mentioned algorithms more accessible, however, most of them have huge disadvantages. Most of the currently existing methods rely on users to upload their own phylogenetic tree and MSA. Other thing users need to provide when using most of the tools are homolog sequences to identify subset of biologically relevant sequences. The last problem is that the topology must be manually inspected. Because of all of these limitations, for proper usage of the tools in this category, deep knowledge of bioinformatics tools and the biological system of interest is required.

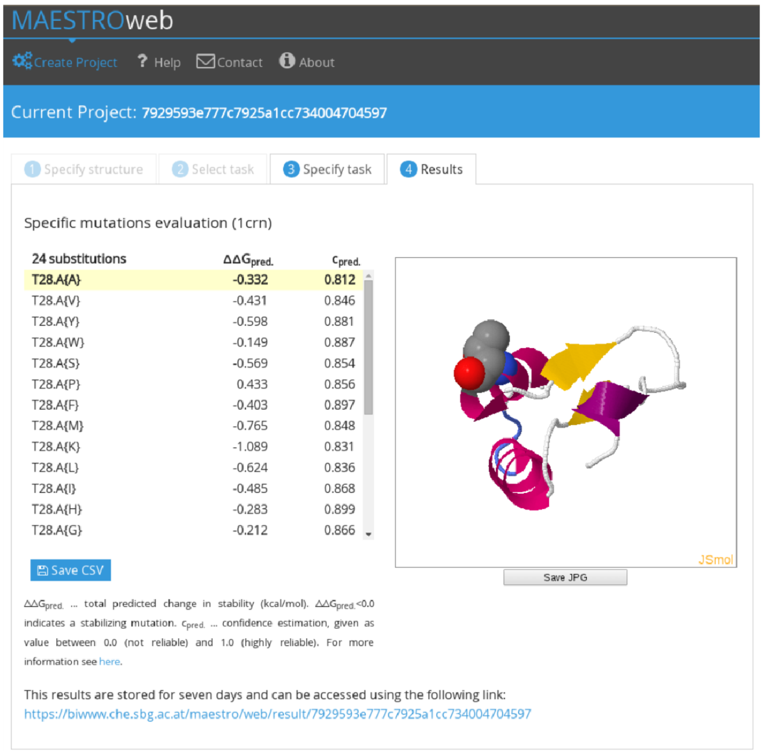
However, a novel approach for fully automatized calculation was presented by Musil et al. [79]. Novel techniques were used to overcome the aforementioned issues. Filters were used to improve the homolog search to check the similarity. New ancestral deviation algorithm [111] was used for the rooting of the phylogenetic tree, and they also presented a new algorithm to replace Fitch's algorithm [36] for ancestral gaps reconstruction. The method was tested by both laboratory experiments and the results published in other studies. The similarity between results was higher than 90%. Thanks to these improvements, the only input that is required from user is protein sequence in FASTA format, so this makes ASR more available for users without depth knowledge of bioinformatics tools.

4.3 Machine Learning

4.3.1 Principles of Methods Based on Machine Learning

Usage of Machine Learning is growing rapidly in past years in every aspect of informational technology and the bioinformatics is no exception. In this section, the utilization of machine learning in protein stability prediction is described. More in-depth description of this topic is described in Chapter 5. For Machine learning algorithms, a correctly sized and balanced dataset is crucial to create and train any useful models. The datasets must not be too small as there could be problem with establishing descriptors during learning. The other problem is with diversity of the data as there is high risk of over training. This would be problematic when the model would be used on new, unknown data. The last problem is regarding size of categories we try to predict. When taking mutations in consideration, if there is 75% of the mutations labelled as deleterious, the model would often tend to classify most of the new data as deleterious as they often appear in the training dataset. There are methods (support vector machines and random forest) that are more resistant to overfitting. [68, 19, 18]. However, neural networks and decision trees are very sensitive to this. Cost-senstive matrices [69] can be used to help with this problem. Furthermore, SMOTE [22] and ADASYN [46] can help with oversampling of the data. For protein stability, a new database was presented in [105], where data are combined from multiple sources and manually checked to verify as many data as needed for creating high efficient and accurate machine learning models.

The other problem occurs when models are validated. Ideally, validation data should also be balanced and truly independent on training data. This is a hard task in protein stability and bioinformatics in general, and k-fold cross-validation has been used as stan-



Current Project: 7929593e777c7925a1cc734004704597

Specific mutations evaluation (1crn)

24 substitutions	$\Delta\Delta G_{pred.}$	$c_{pred.}$
T28.A{A}	-0.332	0.812
T28.A{V}	-0.431	0.846
T28.A{Y}	-0.598	0.881
T28.A{W}	-0.149	0.887
T28.A{S}	-0.569	0.854
T28.A{P}	0.433	0.856
T28.A{F}	-0.403	0.897
T28.A{M}	-0.765	0.848
T28.A{K}	-1.089	0.831
T28.A{L}	-0.624	0.836
T28.A{I}	-0.485	0.868
T28.A{H}	-0.283	0.899
T28.A{G}	-0.212	0.866

$\Delta\Delta G_{pred.}$ ... total predicted change in stability (kcal/mol). $\Delta\Delta G_{pred.}$<0.0 indicates a stabilizing mutation. $c_{pred.}$ ... confidence estimation, given as value between 0.0 (not reliable) and 1.0 (highly reliable). For more information see here.

This results are stored for seven days and can be accessed using the following link:
https://biwww.che.sbg.ac.at/maestro/web/result/7929593e777c7925a1cc734004704597

Figure 4.1: Example of the result and user interface of Maestro web server.

dard method for validating. In this method, k subsets are randomly created and k-1 of them is used for training and last for validating. However, combination of this method with unbalanced dataset increases the risk of overestimation of the system's accuracy [94]. Because of this, cross-validation is no longer accepted in many scientific journals.

4.3.2 Software tools

Tools based on machine learning algorithms are very common, as they do not require deep knowledge of the forces in protein tertiary structure. Predictions are based only on the available experimental data. The most popular methods are based on support vector machines or random forest, thanks to their robustness to the unbalanced data.

In recent years, deep learning was applied to solve this issue. However, this approach is very limited as the datasets are not diverse and big enough to fully utilize the advantages of deep learning. Generalization of the model can be improved with pruning, however, there is no method to help with the dataset size. Until there will be more experimental data, deep learning will hardly be used due to its problem with limited unbalanced datasets.

Better improvements for robustness and reliability is gained by combination of different models into a single multi-agent system. MAESTRO [63] uses a combination of neural networks with linear regression, support vector machines and limited statistical potentials. The outputs from models are averaged into a single prediction. Machine learning can be used to train the arbiter to help with creating optimal weights of the models for best accuracy. An example of the output from MAESTRO web server can be found in Figure 4.1.

Comparison of the presented methods is not easy, as authors usually used different datasets to evaluate the models. This results into bias towards specific proteins and mutation types. This leads to overestimation of the accuracy. The independent comparative studies done by Kellog et al. [59], Potapov et al. [89], and Khan and Vihinem [61] revealed that PEEF based methods outperform tools using only machine learning techniques on the independent dataset. The other problem was revealed in machine learning methods, as their accuracy is highly overestimated [112, 91].

4.4 Meta-predictors and principles of the methods based on hybrid approach

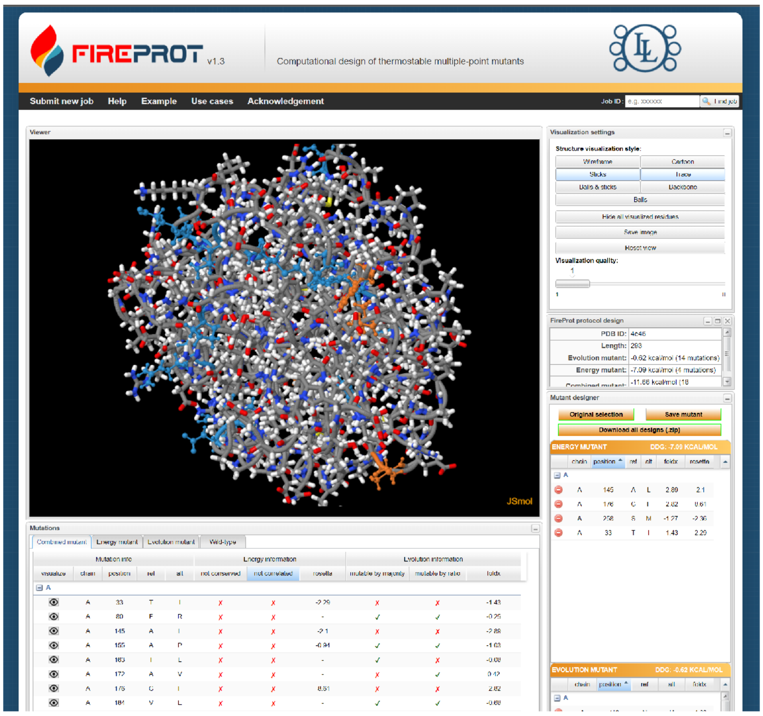
These methods cannot be considered as singular tool but rather combination of different methods, computational strategies and tools. These methods usually incorporate both evolution and energy based approaches. This means that hybrid methods are more robust and reliable. Most of the hybrid methods start with analysis of the highly conserved regions with high correlation within other residues in the MSA [40, 81, 118]. It is based on assumption that these residues are crucial for correct protein's function. Mutations on these position would have higher chance of changing protein's characteristics. In hybrid methods, these positions are excluded from the calculation. This results in smaller computational demand and safer space for designed mutations. Evolution-based and force-field methods has been proven complementary in many proteins as there is only a small overlap of the designed stabilizing methods [12]. Using the combination of both methods, more potentially stabilizing mutations are identified, even though only evolution or energy based approaches would not be able to detect them. Hybrid methods are more robust and complex. Therefore, they are often used to predict more stable multiple-point mutants. These mutants are unattainable by singular approach due to the risk of antagonistic effect.

4.4.1 Software tools

Thanks to all benefits of the hybrid approaches, many research groups are interested in them. However, only tree tools are currently available.

The first method available was The Framework for Rapid Enzyme Stabilization by Computational Libraries (FRESCO) [118]. It is a set of tools and scripts, so an advance knowledge is required for its usage. There are at least two approaches dealing with this issue. A pool of potentially stabilizing mutations are selected based on the predictions from Rosetta and FoldX and then the residues too close to the active sites are filtered out. MD simulations are then utilized to design disulphide bridges and to predict changes in backbone flexibility to remove potentially destabilizing mutations. The result of fresco is a pool of mutations, so it is not fully automated and more effort from the users is required.

The second method is Protein Repair One-Stop Shop (PROSS) which require only 3D structure and sequences of naturally occurring homologs [40]. It starts similarly to FRESCO with Rosetta design calculations to exclude residues too close to active sites. A position-specific substitution matrix is created to remove amino acids that are rarely observed in the homolog sequence [4]. The combinatorial sequence design tool from Rosetta [117] is used to create the optimal combination of mutations with energy function applied to favour most frequent amino acids in the MSA. Using this approach, some neutral or possibly destabilizing mutations can appear in the result [39].



PDB ID:	4e46
Length:	293
Evolution mutant:	-0.62 kcal/mol (14 mutations)
Energy mutant:	-7.09 kcal/mol (4 mutations)
Combined mutant:	-11.66 kcal/mol (18

ENERGY MUTANT DDG: -7.09 KCAL/MOL

	chain	position	ref	alt	foldx	rosetta
A						
⊖	A	145	A	L	2.89	2.1
⊖	A	176	C	I	2.82	0.61
⊖	A	258	S	M	-1.27	-2.36
⊖	A	33	T	I	1.43	2.29

EVOLUTION MUTANT DDG: -0.62 KCAL/MOL

	chain	position	ref	alt	foldx
A					

Mutation info					Energy information			Evolution information		
visualize	chain	position	ref	alt	not conserved	not correlated	rosetta	mutable by majority	mutable by ratio	foldx
A										
👁	A	33	T	I	✗	✗	-2.29	✗	✗	-1.43
👁	A	80	F	R	✗	✗	-	✓	✓	-0.25
👁	A	145	A	I	✗	✗	-2.1	✗	✗	-2.88
👁	A	155	A	P	✗	✗	-0.94	✓	✓	-1.03
👁	A	163	I	L	✗	✗	-	✓	✗	-0.08
👁	A	172	A	V	✗	✗	-	✗	✓	0.42
👁	A	176	C	I	✗	✗	8.61	✗	✗	2.82
👁	A	184	V	L	✗	✗	-	✓	✓	-0.60

Figure 4.2: Example of the result and user interface of FireProt.

The last mentioned is FireProt [81] platform which combines evolutionary and energy based approach using both sequence and structural information. Evolutionary information prohibits the mutations of the important residues and reduce computational time. Both FoldX and Rosetta are used to increase reliability. The risk of antagonistic effect into the mutant is reduced using simple graph based algorithm. As the whole process is fully automated, the only required input from the user is ID from the PDB database. An example of the result and user interface can be seen on Figure 4.2.

4.5 Summary of stability prediction methods

In the previous chapters, only a few methods were described under each section. However, each category have various other methods. These methods were organized into Table 4.1, where each method has its model, input, output, and it can be only single or multiple mutations analyzed. The methods are grouped by same category in order as in this Chapter.

Method	Model	Input	Output	Mutations
PoPMuSiC [32]	SEEF	Structure	$\Delta\Delta G$	Single
FoldX [99]	EEEF	Structure	$\Delta\Delta G$	Single
CUPSAT [85]	Atom potentials Torsion angles	Structure	$\Delta\Delta G$	Single
Rosetta [59]	PEEF	Structure	$\Delta\Delta G$	Single Multiple
ERIS [122]	PEEF	Structure	$\Delta\Delta G$	Single
CC/PBSA [15]	PEEF	Structure	$\Delta\Delta G$	Single
DMutant [48]	Amino acid potentials Torsion angles	Structure	$\Delta\Delta G$	Single
SDM [84]	SEEF	Structure	$\Delta\Delta G$	Single
HotMuSiC [92]	SEEF	Structure	$\Delta(T_m)$	Single
STRUM [93]	SEEF	Structure	$\Delta\Delta G$	Single
AUTO-MUTE [73]	SEEF/ML	Structure	$\Delta\Delta G$/Binary	Single
HotSpotWizard [14]	CA	Seq/struct	hotspots	Single multiple
FastML [9]	ML	MSA+tree	Sequences	Single
RAxML [102]	ML	MSA	Phylogeny	Single
MLGO [49]	ML	MSA+tree	Seq+phylogeny	Single
Ancestors [33]	ML	MSA+tree	Seq+PP	Single
HandAlign [115]	BA	MSA+tree	Seq+PP+phyl.	Single
TreeTime [97]	BA	MSA+tree	Seq+PP+phyl.	Single
PAML [121]	ML	MSA+tree	Seq+PP+phyl.	Single
PhyloBot [45]	ML	MSA+tree	Seq+PP+phyl.	Single
MaxAlike [75]	ML	MSA+tree	Seq+PP+seq.	Single
EASE-MM [38]	SVM	Sequence	$\Delta\Delta G$	Single
MuStab [108]	SVM	Sequence	Binary	Single
ProMaya [113]	RF	Sequence	$\Delta\Delta G$	Single
mCSM [87]	Graph based	Sequence	$\Delta\Delta G$	Single
ELASPIC [120]	SVM+HMM	Structure	$\Delta\Delta G$	Single multiple
MuPro [25]	SVM	Seq./Struct.	$\Delta\Delta G$	Single
I-Mutant2.0 [21]	SVM	Seq./Struct.	$\Delta\Delta G$	Single
TopologyNet [20]	Deep learning	Structure	$\Delta\Delta G$	Single
PROTS-RF [67]	RF	Structure	$\Delta\Delta G$	Single
MAESTRO [63]	M-a system	Structure	$\Delta\Delta G$	Single multiple
IPTREE-STAB [50]	Decision tree	Sequence	Binary	Single
INPS-MD [98]	Sup. Vec. reg.	Sequence	$\Delta\Delta G$	Single
iStable [23]	SVM	Structure	$\Delta\Delta G$	Single
Prethermut [109]	SVM+RF	Structure	$\Delta\Delta G$	Single multiple
FireProt [81]	Evolution+energy	Structure	Mutations+$\Delta\Delta G$	Multiple
PROSS [40]	Evolution+energy	Structure	Mutations	Multiple
FRESCO [118]	Evolution+energy	Structure	Mutations	Multiple

Table 4.1: Summary of multiple software tools available to predict stability of mutations in protein. Methods in the first section use force-filed based approach, in the second section use evolutionary information, in the third section use machine learning and in the last section use hybrid approach.

Chapter 5

Machine learning

The main aim of this chapter is to give overall information regarding machine learning and to describe classification and regression and the methods that can be used.

Machine learning is one of the most important and popular topics over the last few years. One of the main reasons is that machine learning does not require explicit programming and can be used on many different tasks. This field exists for longer time, but has growth just recently, thanks to the increasing computing power and the availability of the large data sets. Machine learning is based on a model and some input data. The data can differ from images and sounds to any kind of text and more. The usage of machine learning can be separated into supervised learning, unsupervised learning and reinforced learning [96].

Supervised learning builds the model from a set of data that contains both the input and their corresponding output. The input data are called training data and contain the set of training examples. Each example has one or more outputs. The main goal of supervised learning is to learn a function that will be able to predict the output from new, unknown input.

Unsupervised learning tries to find structure from the input data, like grouping or clustering. The input data are not labelled, classified or categorized. The mail goal is to identify common values and react on presence of the new data.

Reinforced learning is the most general category. The model interacts with a dynamic environment and tries to achieve some goal. As it is not supervised, there are no rewards that would be given to model if it is getting closer to its goal. The only information for the model comes from interaction with the environment.

Based on the output of the model, required tasks can be split into regression, clustering and classification [3]:

Clustering is an example of unsupervised learning. The aim of clustering is to create the required number of clusters. The values inside one cluster should be as similar as possible, while as different as possible, to items in other clusters.

In the case of classification, the input data are split into finite number of different classes, so the output is always discrete. When training, all input data has to have known output category. This means it is supervised learning. The process can be split into three phases. In the first phase, training data are used to train the model. As it is supervised, all input data must have corresponding output. In this phase, classification rules are created inside the model. In the second phase, the model is tested on the new data. These data have to be different from the testing data. This is important as the precision of the model is calculated from the previously unobserved data. In the last phase, the model is used on the new, completely unknown data in production.

Regression is very similar to classification, so it is supervised learning. The difference between classification and regression is that the output of regression is continuous. The continuous value is a real-value, such as an integer or floating point. In our case, it will be the ΔΔG.

5.1 Calculating accuracy of classification

As the first part of the practical section deals with determining if mutation will be stabilizing or destabilizing, it is a binary classification. Accuracy of classification methods are generally calculated using these metrics [64]:

TPR (True Positive Rate) or sensibility, is the ratio of correctly identified stabilizing mutations to the number of all really stabilizing mutations

TNR (True Negative Rate) or specificity, is the ratio of correctly identified destabilizing mutations to the number of all really destabilizing mutations

FNR (False Negative Rate) and FPR (False Positive Rate) are error rates, and they are sometimes used instead of TPR and TNR. They are calculated as follows:

$$FPR = 1 - TNR \tag{5.1}$$

$$FNR = 1 - TPR \tag{5.2}$$

Normalized Accuracy uses mean value of TNR and TPR to objectively assess success.

$$Acc = \frac{TNR + TPR}{2} \tag{5.3}$$

MCC (Matthews' correlation coefficient) also determines accuracy of classifier and often can be better than normalized accuracy as it also reflects on different cardinality of sets. It can be calculated as follows: [90]

$$MCC = \frac{TPR * TNR - FPR * FNR}{\sqrt{(TPR + FPR) * (TPR + FNR) * (FPR + TNR) * (FNR + TNR)}} \tag{5.4}$$

Receiver Operation Characteristics

In the ideal scenario, specificity and sensibility should be equal to 1, however, it is rarely the case. That is why ROC curve is used. ROC shows the dependency of sensibility to false negative rate (1- TNR). In other words, it shows success of the classifier to distinguish positive and negative parts of a data set. An example of ROC characteristic for binary problem is showed in Figure 5.1. The goal for the curve is to be as close as possible to the point [0,1]. ROC is often supplemented with Area Under Curve (AUC), which helps to quantify and compare distinction of data set parts [64]. In ideal case, AUC is near one.

5.2 Classification methods

There are multiple different methods that can be used to perform classification. They differ in their implementation and some advantages and disadvantages over the other methods.

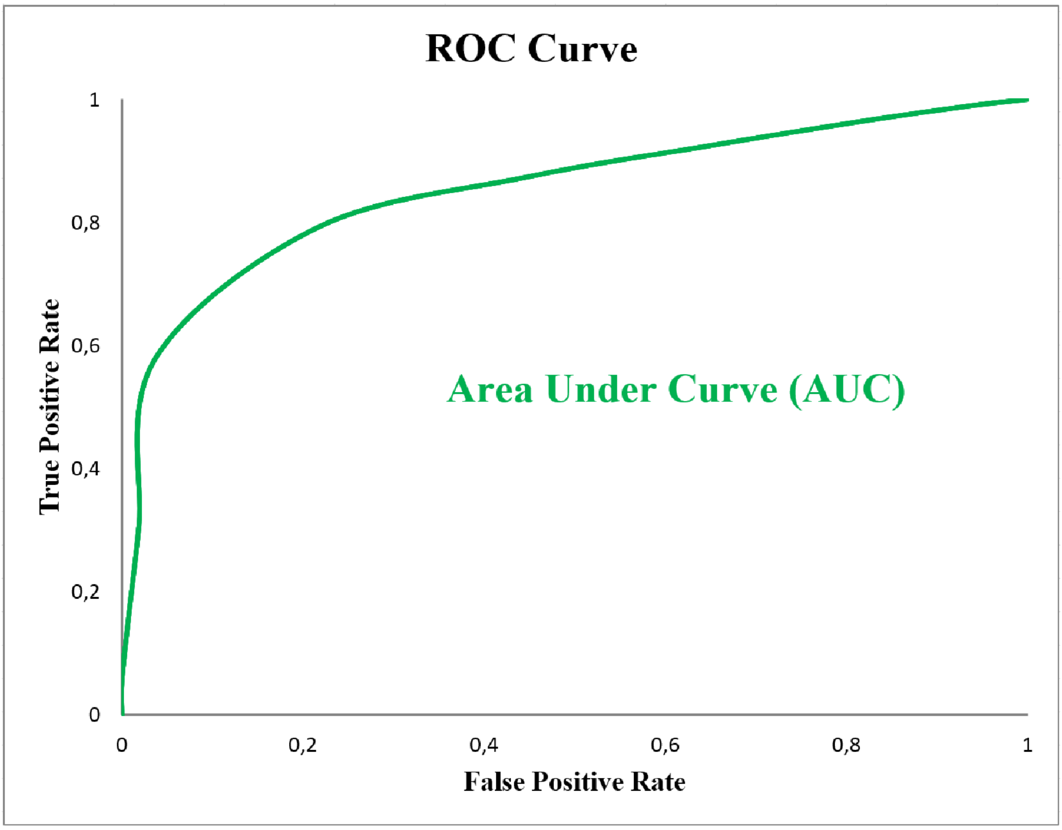


Figure 5.1: Example of ROC characteristic.

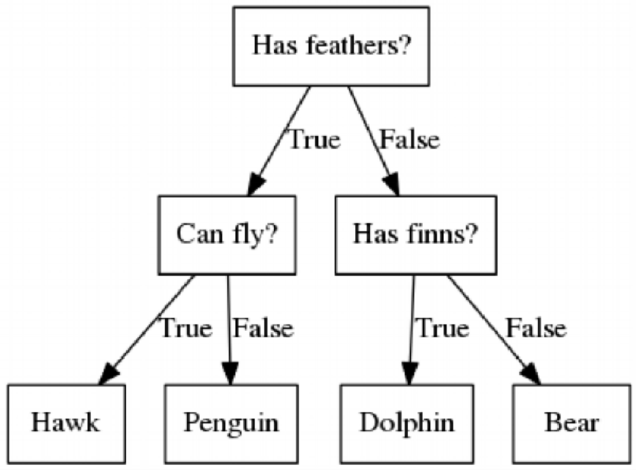


Figure 5.2: Example of decision tree and its rules for categorizing animals.

5.2.1 Methods based on Decision trees

The core of these methods is to create a decision tree from training data and then its application for new data. The core of the algorithm is to correctly detect attributes that have big decision impact to determine the correct category. From this knowledge, a set of rules is created. These rules are compiled as a tree. The individual rules are stored in the tree knots and based on the evaluation of the knot's, the correct path is selected. Categories are stored as leaves. A very simple example can be found on Figure 5.2 After training, the tree is not changed.

One of the common problems is overfitting. This means that the model is too much adapted for the training data and lose its ability to correctly evaluate more general new inputs. Common way how to deal with it is with decision tree pruning. It reduces the size of the decision tree by removing non-critical and redundant sections of the tree.

In practise, random forest is often utilized. In this method, multiple decision trees are constructed and each contains different rules. This also helps with overfitting. The result is then selected by majority selection from all the trees.

5.2.2 Methods based on Bayes' Theorem

These classifiers are called as probabilistic as they determine probability of input with all given classes. The class with the highest probability is then chosen to be the selected one. All methods are based on Bayes' theorem of probability of an event, based on prior knowledge of conditions that might be related to the event [55]. Mathematically it looks like this:

$$P(C_i|X) = \frac{P(X|C_i)P(C_i)}{P(X)}, \tag{5.5}$$

where P(X) is constant, $P(C_i|X)$ is the probability of C_i when we know that X happened.

For each class in training, distribution of attribute values is created. These are then used in validation to select the new input. Naive Bayes' classifiers assume that all categories

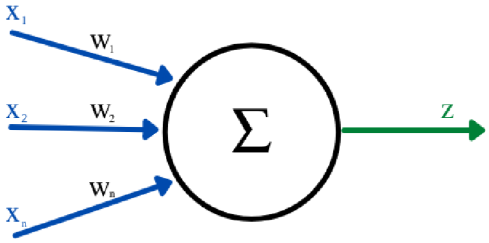


Figure 5.3: Simple Schema of perceptron with n number of X inputs with corresponding weights w and Z is an output.

are independent, however, that is rarely the case. Because of that, these classifiers often lack accuracy. However, there are also Bayesian Networks, which are graphical models where dependency between categories can be modelled. This often improves accuracy and usability. Unfortunately, it is more demanding than Naive Bayes [96].

5.2.3 Methods based on Neural networks

Neural networks have been designed to look like humans' central neural network. They are based on an artificial neuron called perceptron. The input to the perceptron is a weighted vector and a single number is its output. The scheme of perceptron can be found in Figure 5.3. Usually, high number of perceptrons is combined together in one network.

There are plenty of algorithms to learn neural network with different complexity and accuracy. They are generally based on iterative learning and changing the weights of input of individual perceptrons.

Neural networks are used in various applications such as regression and non-supervised training as they are greatly versatile in problem-solving. The core is to find a correct number of layers and perceptrons to have great results. The main disadvantage is the quite long training phase, as generally, it takes more than one iteration through training data. Neural networks are usually not utilized for highly unbalanced data, as it tends to pick the largest group [96].

5.2.4 Linear methods with kernel

Linear methods made decision based on the value of linear combination of the characteristics. Their aim is to divide inputs into groups which can be distinguished (Figure 5.4). The main method in this group is Support-vector machines (SVM).

The aim of SVM is to create as wide gap as possible between individual groups. This helps with the correct determination of the new input into the correct group. Nonlinear distribution is dealt using mapping training samples into new higher dimensional feature space. The classification is then linear in the created feature space.

SVM methods require quite high computational power, however, as they are very accurate and can integrate a combination of multiple attributes, they are often used in bioinformatics [11]. There is a modified version of SVM to support-vector clustering algorithm [13], where the statistics of support vectors developed in SVM are applied for categorization of the unlabelled data.

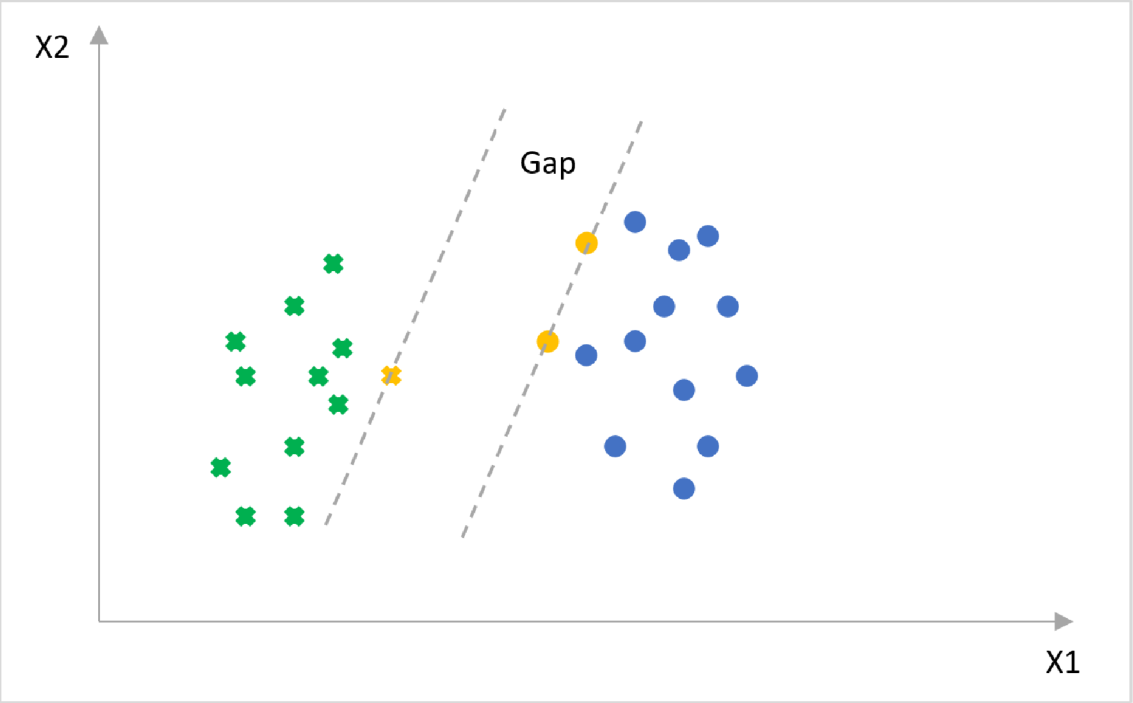


Figure 5.4: Example of gap between two samples using SVM.

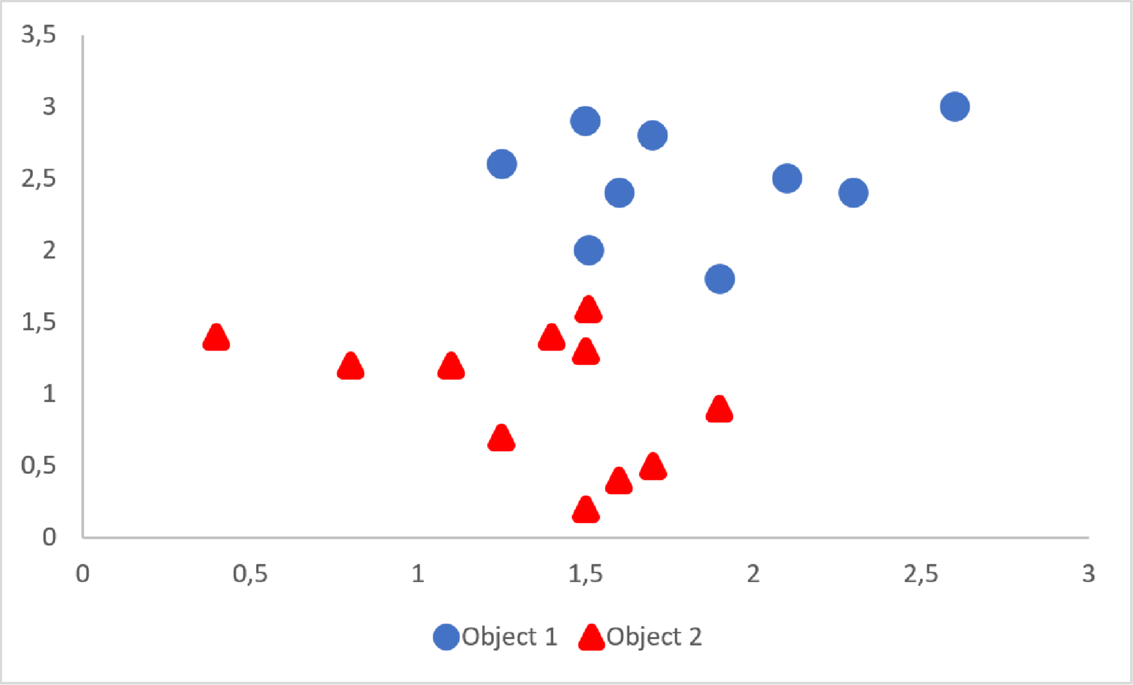


Figure 5.5: Example of usage and voting in KNN.

5.2.5 Non-parametric methods

These methods do not have any parameter, that would have direct impact on the result of training and classification. One of the most known method is K-nearest neighbours algorithm which can be both used for classification and regression. The input consist of the k closest examples from training data set. The output for classification is the class membership that is determined by plurality vote of its neighbours, where the value is set to be the most common of its k nearest neighbours (Figure 5.5).

These methods are fast to learn and, if parameters are correctly selected, very accurate. Disadvantages can be in the combination of different types of parameters and usage of metrics for calculation of the distances. For numerical continuous inputs, euclidean distance can be used. For discrete variables (e.g., text), Hamming distance can be employed. However, it can get pretty complicated for more category types. The last problem is in combination of all distance results of each attribute [78].

5.3 Calculating accuracy of Regression

As the output of regression is continuous, the accuracy of the model is mostly reported as an error. For that, two values are often used. The first one is Root mean squared error:

$$RMSE = \sqrt{\frac{1}{n}\Sigma_{i=1}^{n}\left(\frac{d_i - f_i}{\sigma_i}\right)^2} \tag{5.6}$$

where d_i is the predicted value, f_i is the correct output value and σ_i is a number of times the prediction was made. The main advantage of RMSE is that the unit of the final result is the same as the predicted value.

The other way to validate the model is to use the Pearson correlation coefficient, which is calculated using the following formula:

$$PCC = \frac{n\sum x_i y_i - \sum x_i \sum y_i}{\sqrt{n\sum x_i^2 - (\sum x_i)^2}\sqrt{n\sum y_i^2 - (\sum y_i)^2}} \tag{5.7}$$

where x_i and y_i are the individual sample points and n is the size of the dataset. The values can be between -1 and 1 where 1 is the perfect, almost unrealistic, correlation.

5.4 Regression methods

The most basic type of regression is linear regression. If the data contains more than one independent variable, multiple linear regressions are created. The equation for linear regression is calculated as:

$$y = m * x + c + e \tag{5.8}$$

where m is the slope of the line, c is an intercept, and e represents the error in the model. The best line is calculated by changing m and c. The point is to find the best combination of m and c to have the error as small as possible. As single linear regression is susceptible to outliers, it is not suitable for big datasets.

There are plenty of other methods, including Logistic Regression and Polynomial regression. One of the more robust examples is Gradient boosting regression, where the prediction model is in the form of an ensemble of weaker models. These models are typically decision trees. Gradient boosting often increases the performance of the more simple models such as linear regression, but it is harder to interpret the model. However, there are some existing methods that can help to transform multiple small models into one big model which is easier to interpret.

Chapter 6

Data preparation

The main aim of this chapter is to inform about all the processes that have to be done to obtain valid datasets for training and validation. Both datasets need to be checked to contain truthful information without any errors, as that would lead to incorrect training or validation. The training dataset is generally larger than the one used for validation. Both datasets should be independent, meaning there should not be the same mutation present in both datasets, as that could inflate the accuracy.

6.1 Creation of the training dataset

The raw version of the dataset consists of more than fifteen thousand mutations. The data were extracted from ProTermDB and were expanded with experiments measured in Loschmidt Laboratories.

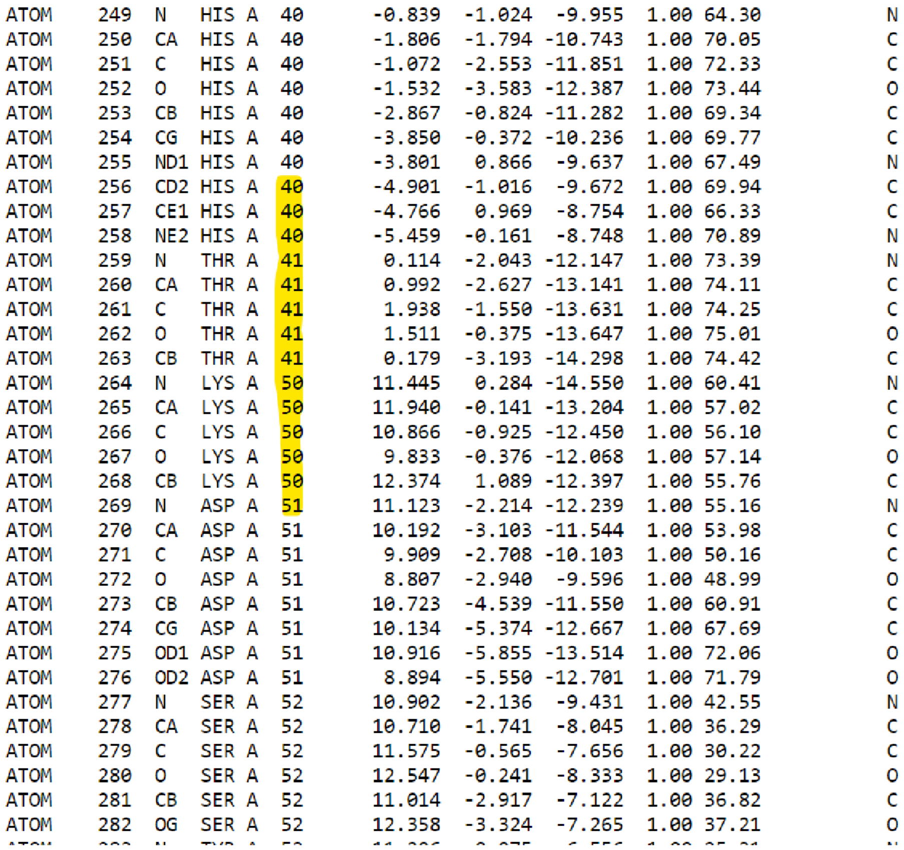
Validating PDB and Uniprot ID

Both PDB ID and Uniprot ID are present in the dataset, however, we need to firstly validate if both IDs are representing the same protein. This was done by creating FASTA sequence from both data-sources. From Uniprot, the FASTA sequence was taken directly. In PDB, there are two options. The first one is to use FASTA directly, the second is to parse PDB file to create the sequence manually. As the main goal is to verify the sequence and after that the mutation, parsing of the PDB file was chosen as the best option.

Parsing of PDB file

PDB file is a textual file format containing information regarding three-dimensional structures of molecules. The file has specific format, and the detailed description can be found in [16]. The most important lines for this purpose are lines starting with *ATOM* and lines starting with *REMARK 465*. The *ATOM* records contain information regarding x, y, z orthogonal Angstrom coordinates for each atom. For our purpose, the important parts are Residue sequence number, residue name and chain identifier as in one PDB files, multiple chains can be stored. For one sequence number, multiple records are stored in PDB file as all data are experimentally created, so in different experiments, different values were measured. However, the amino acid is still the same.

As the data in the PDB file are taken from experiments, there are some problems stored in the PDB file. The first problem is regarding gaps in the chain. An example of gap can be



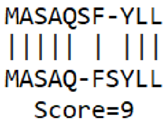
```
ATOM    249  N   HIS A  40      -0.839  -1.024  -9.955  1.00 64.30           N
ATOM    250  CA  HIS A  40      -1.806  -1.794 -10.743  1.00 70.05           C
ATOM    251  C   HIS A  40      -1.072  -2.553 -11.851  1.00 72.33           C
ATOM    252  O   HIS A  40      -1.532  -3.583 -12.387  1.00 73.44           O
ATOM    253  CB  HIS A  40      -2.867  -0.824 -11.282  1.00 69.34           C
ATOM    254  CG  HIS A  40      -3.850  -0.372 -10.236  1.00 69.77           C
ATOM    255  ND1 HIS A  40      -3.801   0.866  -9.637  1.00 67.49           N
ATOM    256  CD2 HIS A  40      -4.901  -1.016  -9.672  1.00 69.94           C
ATOM    257  CE1 HIS A  40      -4.766   0.969  -8.754  1.00 66.33           C
ATOM    258  NE2 HIS A  40      -5.459  -0.161  -8.748  1.00 70.89           N
ATOM    259  N   THR A  41       0.114  -2.043 -12.147  1.00 73.39           N
ATOM    260  CA  THR A  41       0.992  -2.627 -13.141  1.00 74.11           C
ATOM    261  C   THR A  41       1.938  -1.550 -13.631  1.00 74.25           C
ATOM    262  O   THR A  41       1.511  -0.375 -13.647  1.00 75.01           O
ATOM    263  CB  THR A  41       0.179  -3.193 -14.298  1.00 74.42           C
ATOM    264  N   LYS A  50      11.445   0.284 -14.550  1.00 60.41           N
ATOM    265  CA  LYS A  50      11.940  -0.141 -13.204  1.00 57.02           C
ATOM    266  C   LYS A  50      10.866  -0.925 -12.450  1.00 56.10           C
ATOM    267  O   LYS A  50       9.833  -0.376 -12.068  1.00 57.14           O
ATOM    268  CB  LYS A  50      12.374   1.089 -12.397  1.00 55.76           C
ATOM    269  N   ASP A  51      11.123  -2.214 -12.239  1.00 55.16           N
ATOM    270  CA  ASP A  51      10.192  -3.103 -11.544  1.00 53.98           C
ATOM    271  C   ASP A  51       9.909  -2.708 -10.103  1.00 50.16           C
ATOM    272  O   ASP A  51       8.807  -2.940  -9.596  1.00 48.99           O
ATOM    273  CB  ASP A  51      10.723  -4.539 -11.550  1.00 60.91           C
ATOM    274  CG  ASP A  51      10.134  -5.374 -12.667  1.00 67.69           C
ATOM    275  OD1 ASP A  51      10.916  -5.855 -13.514  1.00 72.06           O
ATOM    276  OD2 ASP A  51       8.894  -5.550 -12.701  1.00 71.79           O
ATOM    277  N   SER A  52      10.902  -2.136  -9.431  1.00 42.55           N
ATOM    278  CA  SER A  52      10.710  -1.741  -8.045  1.00 36.29           C
ATOM    279  C   SER A  52      11.575  -0.565  -7.656  1.00 30.22           C
ATOM    280  O   SER A  52      12.547  -0.241  -8.333  1.00 29.13           O
ATOM    281  CB  SER A  52      11.014  -2.917  -7.122  1.00 36.82           C
ATOM    282  OG  SER A  52      12.358  -3.324  -7.265  1.00 37.21           O
```

Figure 6.1: Example of PDB file with gap. In the highlighted section there is residue on position 41 followed by position 50. This means that there is a gap of 9 residues missing from the record.

found in Figure 6.1, where the gap between residues is highlighted on position 41 and 50. This means that 9 residues are missing in the records. These gaps could happen everywhere on the protein chain, so those issues need to be taken in consideration when building the chain.

The other problem is when the database record was created from a low resolution model. This can cause problem while sequencing the chain that residues would be marked as position 5 and 6 as an example. However, later when newer higher-resolution model would be parsed, it would reveal that there are residues between those marked as 5 and 6. Then it is up to the creator of the file to update the results, often creating something like 5a etc.

The last step needed to be done from the PDB file is to transform 3 characters name of the amino acid to one-character, which is used in the FASTA sequence. The last thing was to validate the FASTA sequence for Uniprot and sequence created by us. Due to the nature of the gaps, we cannot simply compare the values, but we need to align them. This is called global alignment, where both chains are compared. In Python, there is a pairwise module in Biopython [27] library which can be utilized. The function used for calculating similarity of two chains add 1 point for same residue on specific location and 0 for gaps or differing amino acids. An example of that can be seen on Figure 6.2. All possible combinations are calculated and the one with the highest score is used. We then divided the number with



```
MASAQSF-YLL
||||| | |||
MASAQ-FSYLL
  Score=9
```

Figure 6.2: Example of alignment of two sequences and their calculated score. Each same letter on position adds 1 to the total score.

the length of the shorter chain to get relative similarity score, where 1.0 is if the chains are identical. As the chains can have different lengths, it is important to take the length of the smaller one.

Using this method, we have noticed that vast majority of the IDs are valid with only few exceptions. However, the next part which needed to be done is to validate the PDB chain on the mutation recorded in the row. This was more crucial as if the mutation would not align with the chain, we would not be able to use any calculated values for that records as they would be invalid.

Validating mutations

Mutations were validated against chains calculated from PDB file and have specific format. *1wq5_A:P28S* is an example of the mutation from the dataset, where *1wq5* is ID of the PDB, A means chain A, P is the original residue, 28 is position of the mutation and S is the new residue. However, for some structures, a PDB file is not created. These mutations needed to be ruled out of the training as there would be no possibility of calculating features. However, this affected only a small portion of the mutations.

Extending dataset from AA index

After validating all necessary information, the next step is to append the dataset. We used data from AA index [58]. Data are stored in custom format. We used 2 different indexes, i.e., Hydrophobicity index [8] and Residue volume [41]. For both of the indexes, the difference between values of the original and new one was used.

Calculating features

The last part needed to prepare was the calculation of some additional features. For this calculation, DSSP[54, 57] module for python was used. This module utilizes PDB files to create models, and then it is able to calculate specific values, such as secondary structure and the area that is accessible to solvent (ASA.) Originally, DSSP is recognizing 7 secondary structures, however, we are mapping it to only 3 different characters. The mapping can be found in Table 6.1. ASA is already presented in the training dataset, however, DSSP values seems to be relative and data exported from ProTermDB seems to be absolute. Therefore, we are calculating the ASA to be used for training later as ASA is missing from the validation dataset, and we need to ensure the same source of data to achieve the best accuracy.

Structure	DSSP code	Our used code
Alpha helix (4-12)	H	H
3-10 Helix	G	H
β-bridge	B	E
Strand	E	E
π-helix	I	C
Turn	T	C
Bend	S	C
None	-	C

Table 6.1: Mapping of 7 characters DSSP code for secondary structures to 3 different characters used for training and validation.

Determining stability

The last part is to create the final result category. We are trying to predict if the mutation is stabilizing or not. For this purpose, there are 3 different columns in dataset (*ΔΔG_(kcal/mol)*, *ΔΔG_H2O_(kcal/mol)*, *ΔT_m*). Positive difference in ΔTm means, that mutation is stabilizing. For ΔΔG, there is no general rule if positive values are stabilizing or not. This had to be checked manually, and we have found out, that negative values are stabilizing in this dataset.

6.2 Construction of the validation dataset

The validation dataset was taken from a different source, however, that does not mean that it is truly independent. For that, it is necessary to exclude all mutations that have been used for training. This has been done by checking if there exist exactly same mutation on the protein. This means what amino acid on what position was changed to which amino acid, to make sure that there are no overlapping mutations.

After we ruled out all overlapping mutations, the dataset was prepared to have the same structure as the training set. Therefore, we appended same data from AA index and what we calculated using DSSP as in training for proper validation.

The last step was to determine the outcome of the mutations. The columns used for this were *ddG* and *dTm (℃)*. The most problematic part was handing ddG as the data are from two different sources. As described in previous chapters, the choice if positive delta is stabilizing or not is upon the author. Fortunately, there were only 316 mutations where ddG was specified from 4 different sources. After manually going through the sources, we have found whether positive values are stabilizing or not.

6.3 Statistics of the datasets

After all the cleaning and removing mutations where we were unable to calculate some value needed for training, we ended up with 8,542 different mutation for training purposes and 691 different mutations for validation.

The training dataset consists of 6,382 destabilizing mutations and 2,160 stabilizing mutations. This has to be taken in consideration during training as we need to use some balancing tools. For validation, the dataset is more balanced, as 364 mutations are destabilizing and 327 are stabilizing.

6.4 Creation of the dataset for regression

For regression, we are only considering ΔΔG values, as we can predict only one. For that, we will be joining both datasets into one big dataset and then in testing, we will be using a function to split this dataset into two parts. This is done due to the completely different protein families represented in each dataset. For classification, it was not necessary, as when we look at the overall spectrum of ΔΔG, it will behave the same. However, for regression, the aim is to be as accurate as possible, while being as robust as possible. For that, the best choice is to combine both datasets to achieve the most accurate results.

Chapter 7

Experiments

The aim of this chapter is to describe the final version of datasets, model and its concrete implementation and design of the experiments. The construction of both datasets was described in Chapter 6. The main goal is to create multi-agent system, where each model would be trained on a different subsection of the dataset, to create a more robust system.

7.1 Dataset splits for Classification

The dataset was split by pH, secondary structure and ASA. This created 9 different splits of the dataset and each would be used to train a different model.

Splitting by secondary structure was used by using mapping, which is described in Table 6.1. This created 3 different subsets.

pH was transformed into 3 different categories. pH lower than 6 was first category, 6-10 was second category, and last category was for all higher than 10. This was done to reduce complexity of the pH as usually.

The last division was using ASA where values lower than 0.2 were called as deeply buried, values between 0.2 and 0.75 were tagged as moderately buried and values higher than 0.75 were marked as exposed. For this division, values calculated using DSSP were used as they are on the scale between 0-1.

One of the aims of the dataset split was to see, if some parts would be more balanced. However, as can be seen in Table 7.1, the splits are also very unbalanced. As can be observed, the size of the datasets are also not balanced, as the biggest dataset consist of 7,109 mutations and the smallest only 56. This needs to be taken in consideration when weighting models in multi-agent system, as a model trained on the smallest dataset might be incorrectly trained due to the size of the training dataset.

7.2 Multi-agent system

The multi-agent system would allow us to use models that would not be usable for the whole datasets. However, in our case, we will be using mostly SVM and Random forests. The exact settings of the models would be changed for experiments to find the most optimal combination of models on datasets, weights of the class in each subset and final weights of the models in the final voting model.

Dataset	Stabilizing Mutations	Destabilizing Mutations	Number of Mutations
Acid	1801	692	2493
Coil	2547	936	3483
Deeply buried	525	205	730
Exposed	2944	1194	4138
Helix	2627	1051	3678
Moderaly buried	651	100	751
Neutral	5122	1987	7109
None	50	6	56
Sheet	1799	698	2497

Table 7.1: Statistics of stabilizing and destabilizing mutations in dataset splits.

7.3 Implementation details

Two main python frameworks were used to implement the models and training. The main one is Scikit-learn [86], which is standard for the machine learning applications. From this package, SVM and RandomForest models were applied to create the classification model. For regression, linear regression, gradient boosting model regressor, HuberRegressor and DecisionTreeRegressor were used. We have also used XGBoost package [24]. For evaluation of the models, various functions, which are inside sklearn package, were utilized to calculate model performance. The other package that was used is called Mlxtend [95], where EnsembleVoteClassifier was used. This model from Mlxtend package was chosen instead of Vote Classifier existing in Scikit-learn as it allows usage of pre-trained models without the need to refit them again, which is mandatory for this thesis as the aim is to train every model on a different section of the dataset.

7.4 Experiments design

Classification

There are 3 main metrics we will be focusing on when validating the model. The first is Accuracy of the prediction, the second one is Matthews' correlation coefficient (MCC) and last one is ROC curve. For the validation of the tested model, all 3 parameters has to be evaluated.

The MCC plays a huge role in predicting, whether the model is getting overtrained or not. Montunaci et al., [77] calculated that for ΔΔG prediction, the maximum of MCC is around 0.7-0.8. Higher values of that would suggest overfitting.

The aim is to firstly validate datasets on single RandomForest with pruning to see if there are some problems with the data such as incorrectly labelled ΔΔG as stabilizing or some other problems with the data. Using this, we would further modify the dataset to get some values. As Random Forest with pruning is great to deal with overfitting, we should not have problems with overfitting on imbalanced datasets, however, we can add class weight matrix to the model to obtain more reliable results. After this, we would slowly start to use the validated dataset to train our voting model with modifications such as changing model type, class weights for specific mode, voting weight of the model and other available functions.

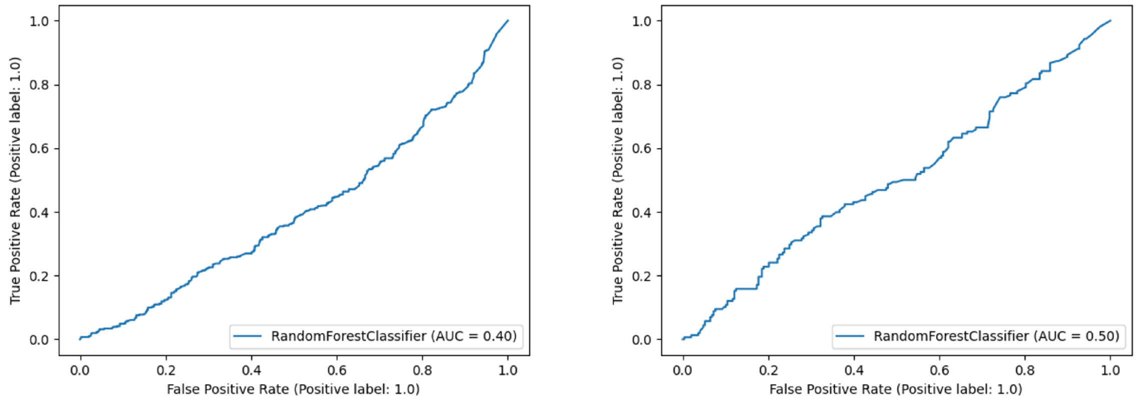


Figure 7.1: **Left** ROC curve of the first attempt suggesting problems with datasets as AUC is near 0.5. **Rigt** ROC curve of the second attempt using only ΔTm values.

Regression

In regression, we are going to measure two metrics. Root Mean Square error (RMSE) and Pearson correlation coefficient (PCC). The main goal is to find the best performing regression predictor on our dataset. The dataset will be divided into training and validation, where the validation part is 0.15 of the overall dataset.

7.4.1 Results of initial classification experiments

The first test was done on the whole training and whole validation dataset with weights set 3-times more on stabilizing mutations to create more balanced dataset. The initial results did not show promising ROC curve as can be observed on Figure 7.1 with Accuracy around 0.48 and MCC -0.10. Those results suggest that there is no correlation in the dataset. This leads to two possibilities. The first is that there is a problem with the dataset, and the second one is that proteins used in the validation dataset have completely different behaviour than those used in the training set.

As the dataset used for that experiment contains both ΔΔG and ΔTm, the next step was to try splitting this datasets into two parts, i.e., containing only ΔΔG or ΔTm.

As determining ΔTm is the same in all publications, the next step was to determine if the problem is in this section. The testing was again done on the RandomForrest classifier with the same weights as in the first experiment. As in the previous experiment, the results showed no correlation at all (MCC -0.08, ACC 0.56 and ROC on Figure 7.1). This suggest that problems are in this part of the dataset.

The last step was to verify if mutations measured by ΔΔG would have the same problem as with ΔTm. That would lead to the conclusion, that proteins used in both datasets behave completely different, which would suggest that values that are widely used are not always accurate. However, when we were using only ΔΔG, AUC near 0.8 (Showed on Figure 7.2), MCC 0.35 and Accuracy 0.70 was achieved. This suggests that ΔΔG values can be used for training and validation on this different values, as these results are similar to Rosseta and other protein stability prediction tools.

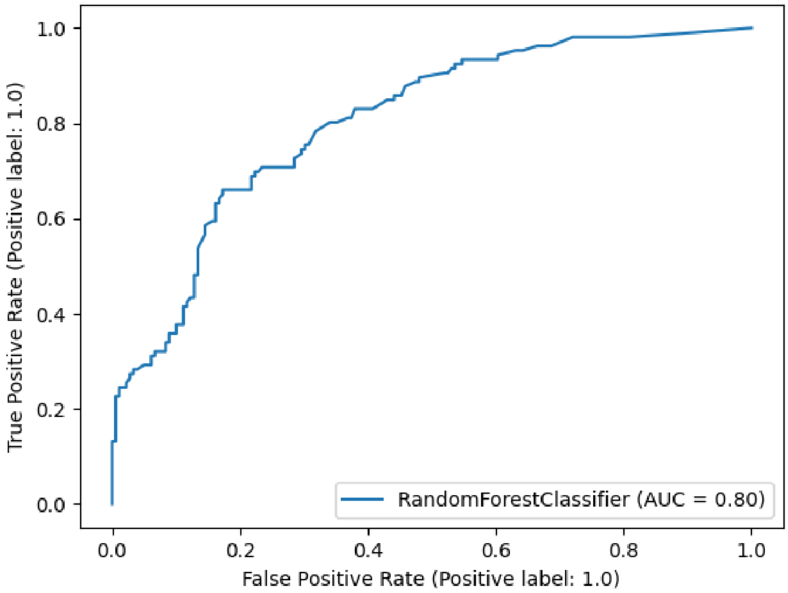


Figure 7.2: ROC curve of the third attempt using ΔΔG values.

7.4.2 Changes to the dataset for final version

The findings using Random Forrest classifier led us to final changes to the dataset. We will be using only ΔΔG for training and validation, which most tools also utilize. This significantly changed the size of the training and validation datasets, where new sizes of splits can be found in Table 7.2.

Overall, training dataset contains of 5,416 different mutation and validating dataset 285, which is still enough to properly train and validate model.

Dataset	**Stabilizing Mutations**	**Destabilizing Mutations**	**Number of Mutations**
Acid	947	342	1289
Coil	1459	501	1960
Deeply buried	299	115	414
Exposed	1608	641	2249
Helix	1425	462	1887
Moderaly buried	56	100	457
Neutral	3083	1024	4107
None	18	2	20
Sheet	1164	405	1569

Table 7.2: Statistics of stabilizing and destabilizing mutations in new dataset splits containing only $\Delta\Delta$G values.

Chapter 8

Results

The aim of this chapter is to describe some experiments that were constructed, and then compare that with existing methods. Detailed data for each experiment can be found in Appendix A.

8.1 Classification into two classes

The first and main part are experiments designed to determine, whether the mutation was stabilizing or destabilizing. We are taking in consideration only the single-point mutations. Stabilizing mutations are marked as 1, destabilizing as 0. The first experiment was created by using 9 different random forests, each trained on a different subset. No weights were employed in this experiment. This experiment was done to create some baseline for experiments using Ensemble classifier. Accuracy of this attempt was 0,63 and MCC 0,11. ROC curve can be seen on Figure 8.1.

Changing to Hard voting from soft voting did not gain noticeable difference (Acc 0.64 with MCC 0.15). So the next step was to add weights. As in most splits of the dataset, destabilizing mutations appear almost three times as much as stabilizing. Therefore, we have added weights which would make the dataset more balanced. This did not lead to a significant performance improvement, as the Accuracy was still around 0.65 and MCC 0,17. ROC curve can be found on Figure 8.1. Looking further into results, we can also see that True positive rate (TPR) for destabilizing mutations is 1.0, while for stabilizing class is around 0.05. This means that the model is predicting a destabilizing class in the majority of cases.

To handle this issue, we have looked into two most imbalanced splits. Moderately buried section has 7 times more destabilizing mutations than stabilizing, and None section has 9 times more. We have changed weights for these two models without gaining much improvement. The next part was to start changing models from Random Forrest to SVM. When applied to both problematic splits, while keeping the same weights for better balancing, the accuracy remained around 0.67. However, the MCC is now around 0.21.

After this, we started to add more SVMs in attempt to find the best combination between Random Forrest and SVM. The best result yield combination of around fifty—fifty split. We were slowly adding SVM to each model and tested, if the accuracy or MCC improved or not. As no further improvement was observed, we reverted those changes. As a result, we achieved 0.70 accuracy and 0.40 MCC. ROC curve can be found on Figure 8.2. The TPR for both classes is 0.69 and 0.71, respectively, which suggest nice balance between

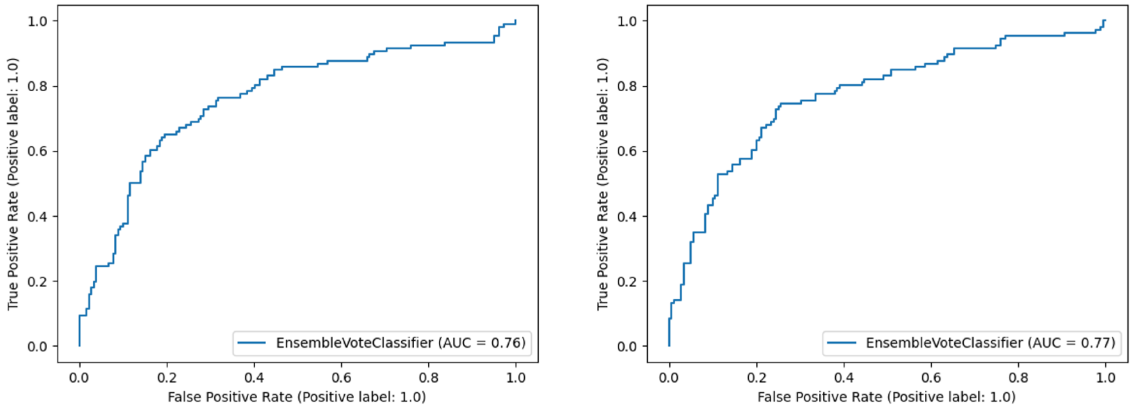


Figure 8.1: **Left** ROC curve of first attempt where no weights were used on models were just Random Forrest. **Right** ROC curve of second attempt where weights of classes were introduced to model.

predicting both classes. We have tried the same settings with soft voting, which resulted in prediction of just one class, which we mark as invalid.

As a next step, we have tried to optimize weights of each model. If the model has only a few mutations to train on instead of few thousands, it has higher risk to provide an incorrect prediction. We have tried to adjust weights based on size of each model, however, the final results were worse or similar than when we did not use any weights. The most accurate model has 0.68 accuracy and 0.38 MCC. Its ROC curve can be found on Figure 8.2.

8.2 Classification into three classes

The next task was to predict three different classes, Stabilizing, destabilizing and neutral. We have tried 2 different intervals for the neutral class. In the first attempt, the neutral class consists of values from -1 kcal/mol to 1 kcal/mol. In the second attempt, we have reduced the class to be from -0.5 kcal/mol to 0.5 kcal/mol. This was made based on the measured experimental error of 0.48 kcal/mol.

The model and strategy is the same as for predicting two classes. This time, we were not measuring the ROC curve, however, we have introduced F1 score.

The results of both attempts are quite similar, which suggest that values from 0.5 to 1 are not significant in the final decision of the classifier. However, what needs to be taken in consideration is that in the validation dataset, the stabilizing class was reduced to 10 when using the bigger interval, while in the smaller one there are 37.

When using the bigger interval, the first attempt looked similar as when predicting only two classes, where ACC is 0.64 and MCC 0.31. After that, we tried to implement the best strategy to contain the highest accuracy and MCC. After few attempts, we have managed to achieve ACC 0.69 and MCC 0.41, which is comparable to predicting two classes. However, the TPR for destabilizing class is 0.74, neutral is 0.69 and stabilizing is only 0.1. This means only one out of ten stabilizing mutations was detected correctly.

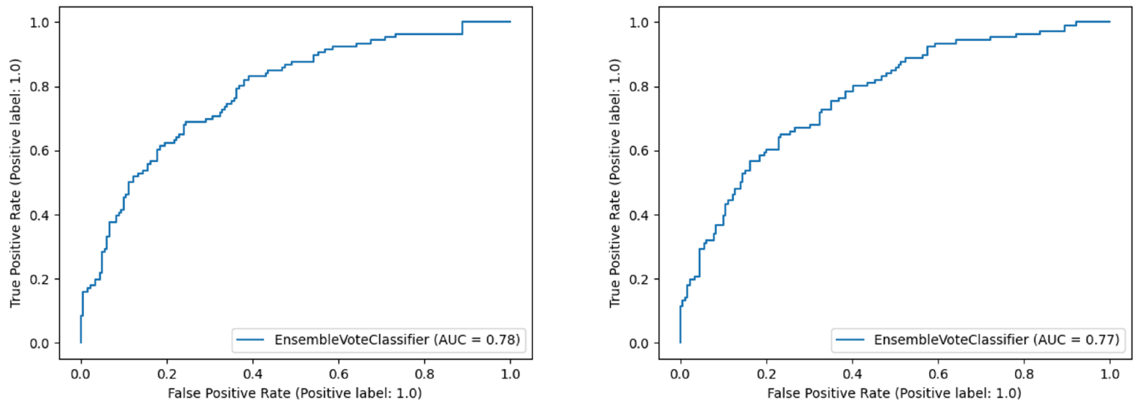


Figure 8.2: **Left** ROC curve of third attempt, which performed the best with MCC of 0.4 and accuracy 0.7. **Right** ROC curve of fourth attempt, which performed best when weights of submodels were applied.

Following these results, we tried to focus the prediction towards the stabilizing mutations, to create the tool more versatile. However, this was a difficult task to achieve as when there was huge focus on stabilizing, we managed to predict every stabilizing mutation correctly. Unfortunately, this lead to decrease in overall accuracy of the model. The best results for the more versatile model had accuracy around 0.53 and MCC 0.31. TPR for stabilizing class was 0.70, destabilizing is 0.70. The biggest decrease of TPR has the neutral class, where the TPR dropped to 0.36.

Knowing this, we have lowered the interval and begin training of the model. Thanks to the results from the larger interval, we did not try to achieve versatility but to keep the highest overall accuracy. This means the model would be better in the prediction of the destabilizing mutations, which would not be produced in laboratory. The best model achieved similar results, with 0.65 accuracy and 0.38 MCC. However, the best TPR for destabilizing class achieved attempt number 6, where we reached 0.8 TPR. The overall performance was 0.61 accuracy and 0.29 MCC.

In the end, the models for predicting into two or three classes predict similarly as the best model in the task of two categories managed to get 0.7 Accuracy and 0.4 MCC, while for the three-class prediction, the best model achieved 0.69 ACC and 0.41 MCC. The main improvements were achieved by switching between SVM and RandomForests and with the class weights for specific model. Surprisingly, the changes of weights of the submodels in the final voting model did not gain an increase in performance, even though there was a model built over an extremely limited dataset.

8.3 Regression

In regression, 5 different models were used (Linear Regression, Huber regressor, Decision Tree Regressor, XGBRegressor and Gradient boosting model). For each model, 19 different experiments were done with different dataset splits. As can be seen in Table 8.1 the best performing model is XGBRegressor, which had an average of RMSE across all attempts about 1.81 and PCC 0.58. It was followed by Gradient boosting regressor, which is the

Model	**Average RMSE**	**Average PCC**
XGBRegressor	1.81	0.58
Linear Regresion	2.13	0.21
Huber regressor	2.12	0.21
Decision Tree	2.35	0.44
Gradient Boosting	1.99	0.43

Table 8.1: Average of RMSE and PCC for all regression models, that were used.

Parameters	**CUPSAT**	**Dmutatnt**	**FoldX**	**I-Mutatnt2.0**	**Imutant 3.0 (sequence)**
Accuracy	0.5	0.56	0.54	0.48	0.52
MCC	-0.01	0.12	0.08	-0.03	0.05

Parameters	**Imutant 3.0 (structure)**	**MUpro**	**MultiMutate**	**SCide**	**SRide**	**Scpread**
Accuracy	0.64	0.37	0.44	0.49	0.49	0.49
MCC	0.27	-0.39	-0.13	-0.03	-0.04	-0.03

Table 8.2: Subset of table containing accuracy and MCC from independent dataset taken from ProTherm to validate accuracy of methods. Full table can be found [61].

only one of the others that managed to have average RMSE below 2. The worst performing model on average was the Decision tree regressor, which had an average RMSE of 2.35.

The best performing model was achieved in attempt 8 using XGBRegressor, which managed to have RMSE of 1.67 with PCC 0.53. The interesting point is that the PCC is almost the lowest with this model, while it shows the highest accuracy. The best model from Gradient boosting is attempt 12, which achieved RMSE of 1.74 with PCC 0.49. It can be noted that only XGBRegressor have shown RMSE under 1.7.

8.4 Comparison with existing methods

8.4.1 Classification

As described in previous chapters, comparison with existing methods is a bit complicated, as each method uses different datasets, and they are often overestimating their results. Many of the existing methods were cross-validated, which resulted in an overtrained model where the actual accuracy of the model is way lower than presented. Table 8.2 is taken from [61], where they prepared a subset from ProTherm consisting of 1,784 mutations obtained from 80 proteins.

When comparing accuracy and MCC with methods showed in Table 8.2, we can see that only Imutant 3.0 (structure) can be compared with our presented method, as the others have MCC negative or close to 0. This shows the problem of overtraining on specific dataset and failing when true independent dataset is used.

Another comparison review was published in 2021 by Iqbal et al. [52], where they created 3 different datasets. S1342 dataset containing 1,342 single point mutations in 130 proteins was extracted from ThermoMutDB. This is the bigger dataset and the main source of the data. The next one is S630, taken from iStable 2.0 which contains 630 mutations

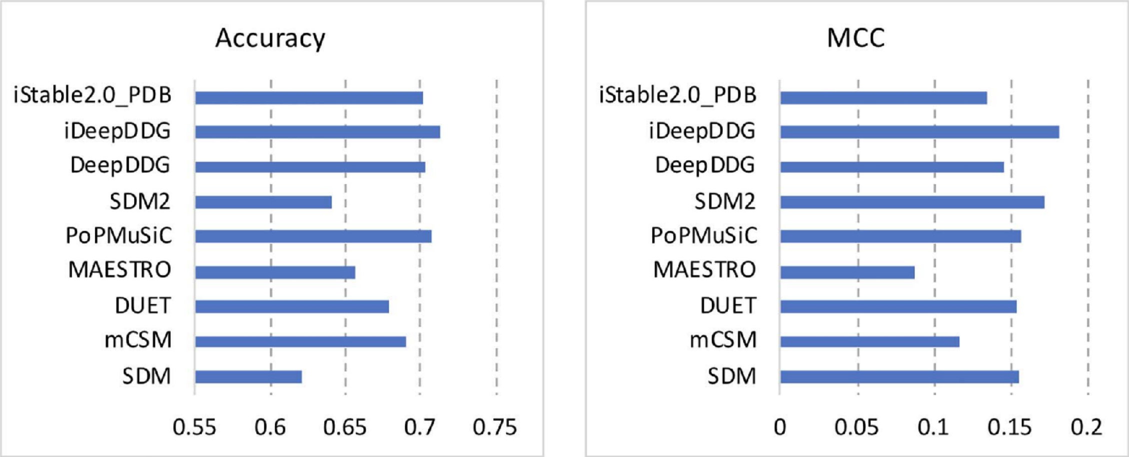


Figure 8.3: Accuracy and MCC of multiple methods on the biggest dataset from [52]. Accuracy of the best methods are around 0.7 with MCC around 0.17.

from 39 proteins. The last one is S268, taken from the most recent ProThemDB. These datasets have some overlaps as they are not independent on each other, however, we can observe how some methods behave differently on different datasets.

The results from the biggest dataset can be found on Figure 8.3. IDeepDDG method is the best performing on the dataset, achieving over 0.7 accuracy and 0.18 MCC. Other tested methods are slightly lower, where the worst performing method is SDM with accuracy around 0.62. However, the method with the lowest MCC is MAESTRO.

Testing on smaller dataset let to increased accuracy and MCC. This can be seen on Figure 8.4, however, it should be noted that this dataset was taken from iStable, so it is best suited for their methods. This is indeed correct as only their method is capable of accuracy higher than 0.8 with MCC around 0.6. However, we should have in mind that the theoretical limit is around 0.7 [77]. The rest of the methods have MCC lower than 0.4.

The results on the smallest dataset show again completely different results regarding MCC as the DUET method is the best performing and showing MCC over 0.5 (Figure 8.5). Accuracy of all methods is between 0.6 and 0.8 where most of them are somewhere around 0.7.

As can be seen from the results, the accuracy of an individual method is fully dependent on the dataset used for validation. However, the best methods are capable of accuracy around 0.7 on multiple different datasets, with MCC around 0.4 on the smaller dataset. This can be used to compare with our independent dataset, where our presented method is capable of achieving 0.7 accuracy and 0.4 MCC. This seems to be in line with the currently used state-of-the-art predictors. For the best comparison, testing all methods on the same dataset, which would be truly independent on the datasets used for training the methods would be the best approach, however, there are currently not enough experimental data available for the testing of this scale.

8.4.2 Regression

As regression deals with the same problems as classification, we will be again using values taken from Iqbal2021. The performance of individual methods is captured in Table 8.3.

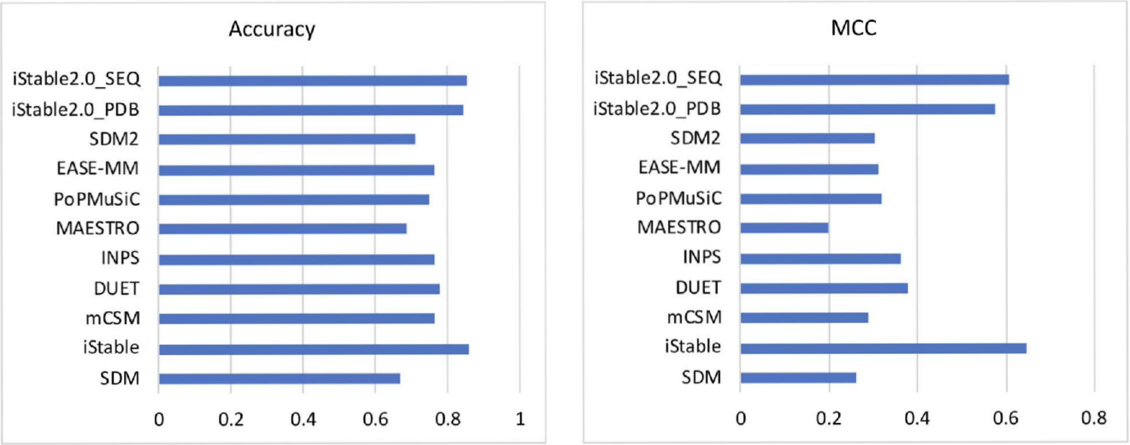


Figure 8.4: Accuracy and MCC of multiple methods on S630 dataset, with accuracy over 0.8 and MCC over 0.6. For many methods, this was the best dataset.

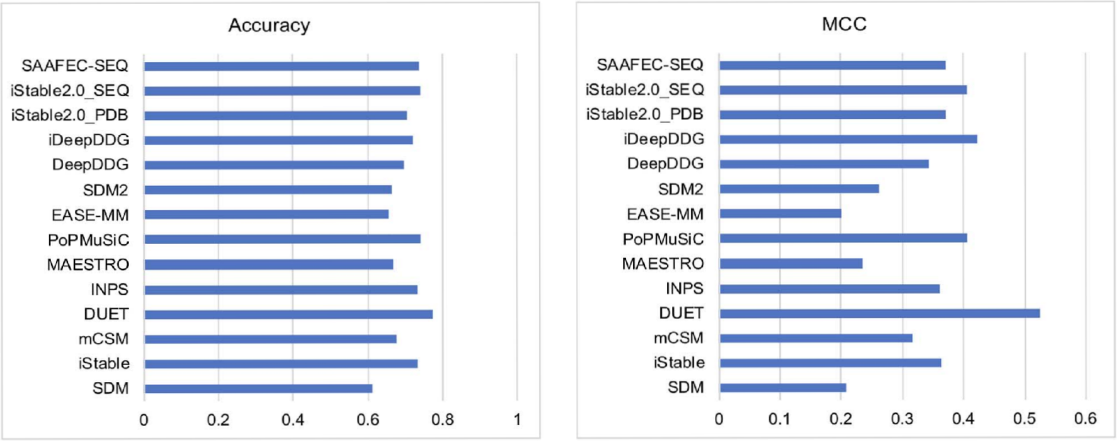


Figure 8.5: Accuracy and MCC of multiple methods on the smallest dataset (S268), where multiple methods were able to have almost 0.8 accuracy with MCC around 0.4. Figure taken from [52].

	S1342		S630		S268	
	PCC	RMSE	PCC	RMSE	PCC	RMSE
SDM	0.30	2.79	0.35	2.24	0.25	2.06
iStable	-	-	0.67	1.73	0.26	2.34
mCSM	0.34	2.63	0,45	1.86	0.30	1.94
DUET	0.37	2.61	0.46	1.87	0.33	1.91
INPS	-	-	0.45	1.92	0.22	2.34
MAESTRO	0.20	2.88	0.33	2.05	0.23	2.30
PopMuSiC	0.36	2.67	0.42	1.92	0.24	2.30
EASE-MM	-	-	0.54	1.77	0.11	2.40
SDM2	0.32	2.70	0.35	2.09	0.23	2.34
DeepDDG	0.51	2.43	*	*	0.19	2.53
iDeepDDG	0.51	2.42	*	*	0.33	1.97
iStable2.0_PDB	0.37	2.67	0.71	1.49	0.39	1.81
iStable2.0_SEQ	-	-	0.70	1.51	0.23	2.26
SAAFEC-SEQ	-	-	-	-	0.17	2.36

Table 8.3: Performance of regression methods. The '-' means that sequence-based predictors were not assessed, and '*' means that the test dataset was not blind. This table was taken from [52].

From this table, we can observe, that depending on the dataset the state-of-the-art methods are achieving RMSE around 1.49 on S630 dataset and 2.88 on the S1342 dataset. The PCC is between 0.11 and 0.71. Overall, it seems that the best performing method is DUET, as it reached the RMS of 1.87 on S630, 1.91 on S268 and 2.61 on S1342. If we compare it to our trained XGBRegressor, which achieved 1.81 RMSE and 0.58 PCC on average, it is similar to this method. However, it should be noted that the methods were not evaluated on the completely independent datasets.

There are two methods that are using the same model as the one utilized in our predictor: iStable2.0 and SAFEC-SEQ. Unfortunately, SAFEC-SEQ could be trained only on S268, where it got RMSE of 2.36. The iStable2.0 shows the best results on S630 with 1.49 RMSE. However it should be noted that S630 is the dataset provided by the authors.

Chapter 9

Conclusion

The main aim of this thesis was to introduce a new method for predicting protein stability. The proposed method contains a multi-agent system, which consist of Support Vector Machines and Random Forests for classification and a single predictive model for regression.

For classification, two independent datasets were constructed for training and validation. The training dataset was taken from ProThermDB with some addition from experimental results measured in the Loschmidt Laboratories. Each of the submodel was trained on a specific subset of the training dataset. The splitting was done by pH, ASA and secondary structure. For the prediction of the protein stability, ΔΔG was used to divide data into two or three classes (stabilizing, destabilizing and neutral).

The final datasets were also used for regression, as we can predict the ΔΔG. However, as both datasets contain different protein families, the training and validation datasets were combined and then randomly split for training and validation. This was done to create a more robust model, as the exact values of ΔΔG can differ based on different protein family.

The best performing model for classification was able to achieve 0.7 accuracy with 0.4 MCC when predicting two classes and 0.69 accuracy with 0.41 MCC for three classes. For regression, the best performing model is Extreme Gradient Boosting model, which shows 1.67 RMSE with 0.53 PCC. These results are comparable with the current state-of-the-art methods.

Nowadays, the major problem in the successful utilization of machine learning methods lies in the small and imbalanced datasets. For the best comparison with other methods, the dataset that would be truly independent of the training sets of all methods should be used. However, this is not possible as there are not enough available data, and therefore, many of the reported results can be overestimated, due to the usage of the same data for training and testing.

Bibliography

[1] Alberts, B. *Essential cell biology.* New York: W.W. Norton & Company, 2019. ISBN 9780393680393.

[2] Alford, R. F., Leaver Fay, A., Jeliazkov, J. R., O'Meara, M. J., DiMaio, F. P. et al. The Rosetta All-Atom Energy Function for Macromolecular Modeling and Design. *Journal of Chemical Theory and Computation.* American Chemical Society (ACS). may 2017, vol. 13, no. 6, p. 3031–3048. DOI: 10.1021/acs.jctc.7b00125.

[3] Alpaydin, E. *Introduction to machine learning.* Cambridge, Mass: MIT Press, 2010. ISBN 9780262012430.

[4] Altschul, S. F., Gertz, E. M., Agarwala, R., Schäffer, A. A. and Yu, Y.-K. PSI-BLAST pseudocounts and the minimum description length principle. *Nucleic Acids Research.* Oxford University Press (OUP). dec 2008, vol. 37, no. 3, p. 815–824. DOI: 10.1093/nar/gkn981.

[5] Anfinsen, C. B. Principles that Govern the Folding of Protein Chains. *Science.* American Association for the Advancement of Science (AAAS). jul 1973, vol. 181, no. 4096, p. 223–230. DOI: 10.1126/science.181.4096.223.

[6] Anjana, R., , Vaishnavi, M. K., Sherlin, D., Kumar, S. P. et al. Aromatic-aromatic interactions in structures of proteins and protein-DNA complexes: a study based on orientation and distance. *Bioinformation.* Biomedical Informatics. dec 2012, vol. 8, no. 24, p. 1220–1224. DOI: 10.6026/97320630081220.

[7] Apweiler, R. UniProt: the Universal Protein knowledgebase. *Nucleic Acids Research.* Oxford University Press (OUP). jan 2004, vol. 32, no. 90001, p. 115D–119. DOI: 10.1093/nar/gkh131.

[8] ARGOS, P., RAO, J. K. M. and HARGRAVE, P. A. Structural Prediction of Membrane-Bound Proteins. *European Journal of Biochemistry.* Wiley. mar 2005, vol. 128, 2-3, p. 565–575. DOI: 10.1111/j.1432-1033.1982.tb07002.x.

[9] Ashkenazy, H., Penn, O., Doron Faigenboim, A., Cohen, O., Cannarozzi, G. et al. FastML: a web server for probabilistic reconstruction of ancestral sequences. *Nucleic Acids Research.* Oxford University Press (OUP). may 2012, vol. 40, W1, p. W580–W584. DOI: 10.1093/nar/gks498.

[10] Babkova, P., Sebestova, E., Brezovsky, J., Chaloupkova, R. and Damborsky, J. Ancestral Haloalkane Dehalogenases Show Robustness and

Unique Substrate Specificity. *ChemBioChem*. Wiley. jun 2017, vol. 18, no. 14, p. 1448–1456. DOI: 10.1002/cbic.201700197.

[11] Baldi, P. *Bioinformatics : the machine learning approach*. Cambridge, Mass: MIT Press, 2001. ISBN 026202506X.

[12] Beerens, K., Mazurenko, S., Kunka, A., Marques, S. M., Hansen, N. et al. Evolutionary Analysis As a Powerful Complement to Energy Calculations for Protein Stabilization. *ACS Catalysis*. American Chemical Society (ACS). aug 2018, vol. 8, no. 10, p. 9420–9428. DOI: 10.1021/acscatal.8b01677.

[13] Ben Hur, A., Horn, D., Siegelmann, H. and Vapnik, V. Support Vector Clustering. *Journal of Machine Learning Research*. november 2001, vol. 2, p. 125–137. DOI: 10.1162/15324430260185565.

[14] Bendl, J., Stourac, J., Sebestova, E., Vavra, O., Musil, M. et al. HotSpot Wizard 2.0: automated design of site-specific mutations and smart libraries in protein engineering. *Nucleic Acids Research*. Oxford University Press (OUP). may 2016, vol. 44, W1, p. W479–W487. DOI: 10.1093/nar/gkw416.

[15] Benedix, A., Becker, C. M., Groot, B. L. de, Caflisch, A. and Böckmann, R. A. Predicting free energy changes using structural ensembles. *Nature Methods*. Springer Science and Business Media LLC. jan 2009, vol. 6, no. 1, p. 3–4. DOI: 10.1038/nmeth0109-3.

[16] Berman, H., Henrick, K. and Nakamura, H. Announcing the worldwide Protein Data Bank. *Nature Structural &amp Molecular Biology*. Springer Science and Business Media LLC. dec 2003, vol. 10, no. 12, p. 980–980. DOI: 10.1038/nsb1203-980.

[17] Bommarius, A. S. and Paye, M. F. Stabilizing biocatalysts. *Chemical Society Reviews*. Royal Society of Chemistry (RSC). 2013, vol. 42, no. 15, p. 6534. DOI: 10.1039/c3cs60137d.

[18] Boughorbel, S., Jarray, F. and El Anbari, M. Optimal classifier for imbalanced data using Matthews Correlation Coefficient metric. *PLOS ONE*. Public Library of Science (PLoS). jun 2017, vol. 12, no. 6, p. e0177678. DOI: 10.1371/journal.pone.0177678.

[19] Breiman, L. Random Forests. *Machine Learning*. Springer Science and Business Media LLC. 2001, vol. 45, no. 1, p. 5–32. DOI: 10.1023/a:1010933404324.

[20] Cang, Z. and Wei, G.-W. TopologyNet: Topology based deep convolutional and multi-task neural networks for biomolecular property predictions. *PLOS Computational Biology*. Public Library of Science (PLoS). jul 2017, vol. 13, no. 7, p. e1005690. DOI: 10.1371/journal.pcbi.1005690.

[21] Capriotti, E., Fariselli, P. and Casadio, R. I-Mutant2.0: predicting stability changes upon mutation from the protein sequence or structure. *Nucleic Acids Research*. Oxford University Press (OUP). jul 2005, vol. 33, Web Server, p. W306–W310. DOI: 10.1093/nar/gki375.

[22] Chawla, N. V., Bowyer, K. W., Hall, L. O. and Kegelmeyer, W. P. SMOTE: Synthetic Minority Over-sampling Technique. *Journal of Artificial Intelligence Research*. AI Access Foundation. jun 2002, vol. 16, p. 321–357. DOI: 10.1613/jair.953.

[23] Chen, C.-W., Lin, J. and Chu, Y.-W. iStable: off-the-shelf predictor integration for predicting protein stability changes. *BMC Bioinformatics*. Springer Science and Business Media LLC. jan 2013, vol. 14, S2. DOI: 10.1186/1471-2105-14-s2-s5.

[24] Chen, T. and Guestrin, C. XGBoost. In: *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, Aug 2016. DOI: 10.1145/2939672.2939785.

[25] Cheng, J., Randall, A. and Baldi, P. Prediction of protein stability changes for single-site mutations using support vector machines. *Proteins: Structure, Function, and Bioinformatics*. Wiley. dec 2005, vol. 62, no. 4, p. 1125–1132. DOI: 10.1002/prot.20810.

[26] Christensen, N. J. and Kepp, K. P. Accurate Stabilities of Laccase Mutants Predicted with a Modified FoldX Protocol. *Journal of Chemical Information and Modeling*. American Chemical Society (ACS). nov 2012, vol. 52, no. 11, p. 3028–3042. DOI: 10.1021/ci300398z.

[27] Cock, P. J. A., Antao, T., Chang, J. T., Chapman, B. A., Cox, C. J. et al. Biopython: freely available Python tools for computational molecular biology and bioinformatics. *Bioinformatics*. Oxford University Press (OUP). mar 2009, vol. 25, no. 11, p. 1422–1423. DOI: 10.1093/bioinformatics/btp163.

[28] Conchúir, S. Ó., Barlow, K. A., Pache, R. A., Ollikainen, N., Kundert, K. et al. A Web Resource for Standardized Benchmark Datasets, Metrics, and Rosetta Protocols for Macromolecular Modeling and Design. *PLOS ONE*. Public Library of Science (PLoS). sep 2015, vol. 10, no. 9, p. e0130433. DOI: 10.1371/journal.pone.0130433.

[29] Das, R. Four Small Puzzles That Rosetta Doesn't Solve. *PLoS ONE*. Public Library of Science (PLoS). may 2011, vol. 6, no. 5, p. e20044. DOI: 10.1371/journal.pone.0020044.

[30] Davey, J. A., Damry, A. M., Euler, C. K., Goto, N. K. and Chica, R. A. Prediction of Stable Globular Proteins Using Negative Design with Non-native Backbone Ensembles. *Structure*. Elsevier BV. nov 2015, vol. 23, no. 11, p. 2011–2021. DOI: 10.1016/j.str.2015.07.021.

[31] Dehouck, Y., Gilis, D. and Rooman, M. A New Generation of Statistical Potentials for Proteins. *Biophysical Journal*. Elsevier BV. jun 2006, vol. 90, no. 11, p. 4010–4017. DOI: 10.1529/biophysj.105.079434.

[32] Dehouck, Y., Kwasigroch, J. M., Gilis, D. and Rooman, M. PoPMuSiC 2.1: a web server for the estimation of protein stability changes upon mutation and sequence optimality. *BMC Bioinformatics*. Springer Science and Business Media LLC. may 2011, vol. 12, no. 1. DOI: 10.1186/1471-2105-12-151.

[33] Diallo, A. B., Makarenkov, V. and Blanchette, M. Ancestors 1.0: a web server for ancestral sequence reconstruction. *Bioinformatics*. Oxford University Press (OUP). oct 2009, vol. 26, no. 1, p. 130–131. DOI: 10.1093/bioinformatics/btp600.

[34] Fan, H. and Mark, A. E. Relative stability of protein structures determined by X-ray crystallography or NMR spectroscopy: A molecular dynamics simulation study. *Proteins: Structure, Function, and Bioinformatics*. Wiley. aug 2003, vol. 53, no. 1, p. 111–120. DOI: 10.1002/prot.10496.

[35] Ferdjani, S., Ionita, M., Roy, B., Dion, M., Djeghaba, Z. et al. Correlation between thermostability and stability of glycosidases in ionic liquid. *Biotechnology Letters*. Springer Science and Business Media LLC. feb 2011, vol. 33, no. 6, p. 1215–1219. DOI: 10.1007/s10529-011-0560-5.

[36] Fitch, W. M. and Margoliash, E. Construction of Phylogenetic Trees. *Science*. American Association for the Advancement of Science (AAAS). jan 1967, vol. 155, no. 3760, p. 279–284. DOI: 10.1126/science.155.3760.279.

[37] Flegr, J. *Uvod do evolucni biologie*. Praha: Academia, 2007. ISBN 9788020015396.

[38] Folkman, L., Stantic, B., Sattar, A. and Zhou, Y. EASE-MM: Sequence-Based Prediction of Mutation-Induced Stability Changes with Feature-Based Multiple Models. *Journal of Molecular Biology*. Elsevier BV. mar 2016, vol. 428, no. 6, p. 1394–1405. DOI: 10.1016/j.jmb.2016.01.012.

[39] Goldenzweig, A. and Fleishman, S. J. Principles of Protein Stability and Their Application in Computational Design. *Annual Review of Biochemistry*. Annual Reviews. jun 2018, vol. 87, no. 1, p. 105–129. DOI: 10.1146/annurev-biochem-062917-012102.

[40] Goldenzweig, A., Goldsmith, M., Hill, S. E., Gertman, O., Laurino, P. et al. Automated Structure- and Sequence-Based Design of Proteins for High Bacterial Expression and Stability. *Molecular Cell*. Elsevier BV. jul 2016, vol. 63, no. 2, p. 337–346. DOI: 10.1016/j.molcel.2016.06.012.

[41] Goldsack, D. and Chalifoux, R. Contribution of the free energy of mixing of hydrophobic side chains to the stability of the tertiary structure of proteins. *Journal of Theoretical Biology*. Elsevier BV. jun 1973, vol. 39, no. 3, p. 645–651. DOI: 10.1016/0022-5193(73)90075-1.

[42] Gromiha, M. M. *Protein Bioinformatics: From Sequence to Function*. ACADEMIC PR INC, october 2010. ISBN 8131222977. Available at: https://www.ebook.de/de/product/10447406/m_michael_gromiha_protein_bioinformatics_from_sequence_to_function.html.

[43] Guerois, R., Nielsen, J. E. and Serrano, L. Predicting Changes in the Stability of Proteins and Protein Complexes: A Study of More Than 1000 Mutations. *Journal of Molecular Biology*. Elsevier BV. jul 2002, vol. 320, no. 2, p. 369–387. DOI: 10.1016/s0022-2836(02)00442-4.

[44] Gumulya, Y. and Reetz, M. T. Enhancing the Thermal Robustness of an Enzyme by Directed Evolution: Least Favorable Starting Points and Inferior Mutants Can Map Superior Evolutionary Pathways. *ChemBioChem.* Wiley. sep 2011, vol. 12, no. 16, p. 2502–2510. DOI: 10.1002/cbic.201100412.

[45] Hanson Smith, V. and Johnson, A. PhyloBot: A Web Portal for Automated Phylogenetics, Ancestral Sequence Reconstruction, and Exploration of Mutational Trajectories. *PLOS Computational Biology.* Public Library of Science (PLoS). jul 2016, vol. 12, no. 7, p. e1004976. DOI: 10.1371/journal.pcbi.1004976.

[46] He, H., Bai, Y., Garcia, E. A. and Li, S. ADASYN: Adaptive synthetic sampling approach for imbalanced learning. In: *2008 IEEE International Joint Conference on Neural Networks (IEEE World Congress on Computational Intelligence).* IEEE, Jun 2008. DOI: 10.1109/ijcnn.2008.4633969.

[47] Hochberg, G. K. A. and Thornton, J. W. Reconstructing Ancient Proteins to Understand the Causes of Structure and Function. *Annual Review of Biophysics.* Annual Reviews. may 2017, vol. 46, no. 1, p. 247–269. DOI: 10.1146/annurev-biophys-070816-033631.

[48] Hoppe, C. and Schomburg, D. Prediction of protein thermostability with a direction- and distance-dependent knowledge-based potential. *Protein Science.* Wiley. oct 2005, vol. 14, no. 10, p. 2682–2692. DOI: 10.1110/ps.04940705.

[49] Hu, F., Lin, Y. and Tang, J. MLGO: phylogeny reconstruction and ancestral inference from gene-order data. *BMC Bioinformatics.* Springer Science and Business Media LLC. nov 2014, vol. 15, no. 1. DOI: 10.1186/s12859-014-0354-6.

[50] Huang, L.-T., Gromiha, M. M. and Ho, S.-Y. iPTREE-STAB: interpretable decision tree based method for predicting protein stability changes upon mutations. *Bioinformatics.* Oxford University Press (OUP). mar 2007, vol. 23, no. 10, p. 1292–1293. DOI: 10.1093/bioinformatics/btm100.

[51] Huelsenbeck, J. P., Ronquist, F., Nielsen, R. and Bollback, J. P. Bayesian Inference of Phylogeny and Its Impact on Evolutionary Biology. *Science.* American Association for the Advancement of Science (AAAS). dec 2001, vol. 294, no. 5550, p. 2310–2314. DOI: 10.1126/science.1065889.

[52] Iqbal, S., Li, F., Akutsu, T., Ascher, D. B., Webb, G. I. et al. Assessing the performance of computational predictors for estimating protein stability changes upon missense mutations. *Briefings in Bioinformatics.* Oxford University Press (OUP). may 2021, vol. 22, no. 6. DOI: 10.1093/bib/bbab184.

[53] Jakubke. *Peptides from A to Z : a concise encyclopedia.* Weinheim: Wiley-VCH, 2008. ISBN 9783527317226.

[54] Joosten, R. P., Beek, T. A. H. te, Krieger, E., Hekkelman, M. L., Hooft, R. W. W. et al. A series of PDB related databases for everyday needs. *Nucleic Acids Research.* Oxford University Press (OUP). nov 2010, vol. 39, Database, p. D411–D419. DOI: 10.1093/nar/gkq1105.

[55] JOYCE, J. Bayes' Theorem. In: ZALTA, E. N., ed. *The Stanford Encyclopedia of Philosophy* [https://plato.stanford.edu/archives/fall2021/entries/bayes-theorem/]. Fall 2021th ed. Metaphysics Research Lab, Stanford University, 2021.

[56] JÄCKEL, C., BLOOM, J. D., KAST, P., ARNOLD, F. H. and HILVERT, D. Consensus Protein Design without Phylogenetic Bias. *Journal of Molecular Biology*. Elsevier BV. jun 2010, vol. 399, no. 4, p. 541–546. DOI: 10.1016/j.jmb.2010.04.039.

[57] KABSCH, W. and SANDER, C. Dictionary of protein secondary structure: Pattern recognition of hydrogen-bonded and geometrical features. *Biopolymers*. Wiley. dec 1983, vol. 22, no. 12, p. 2577–2637. DOI: 10.1002/bip.360221211.

[58] KAWASHIMA, S. AAindex: Amino Acid index database. *Nucleic Acids Research*. Oxford University Press (OUP). jan 2000, vol. 28, no. 1, p. 374–374. DOI: 10.1093/nar/28.1.374.

[59] KELLOGG, E. H., LEAVER FAY, A. and BAKER, D. Role of conformational sampling in computing mutation-induced changes in protein structure and stability. *Proteins: Structure, Function, and Bioinformatics*. Wiley. dec 2010, vol. 79, no. 3, p. 830–838. DOI: 10.1002/prot.22921.

[60] KEPP, K. P. Towards a "Golden Standard" for computing globin stability: Stability and structure sensitivity of myoglobin mutants. *Biochimica et Biophysica Acta (BBA) - Proteins and Proteomics*. Elsevier BV. oct 2015, vol. 1854, no. 10, p. 1239–1248. DOI: 10.1016/j.bbapap.2015.06.002.

[61] KHAN, S. and VIHINEN, M. Performance of protein stability predictors. *Human Mutation*. Wiley. mar 2010, vol. 31, no. 6, p. 675–684. DOI: 10.1002/humu.21242.

[62] KUZMANIC, A., PANNU, N. S. and ZAGROVIC, B. X-ray refinement significantly underestimates the level of microscopic heterogeneity in biomolecular crystals. *Nature Communications*. Springer Science and Business Media LLC. feb 2014, vol. 5, no. 1. DOI: 10.1038/ncomms4220.

[63] LAIMER, J., HOFER, H., FRITZ, M., WEGENKITTL, S. and LACKNER, P. MAESTRO - multi agent stability prediction upon point mutations. *BMC Bioinformatics*. Springer Science and Business Media LLC. apr 2015, vol. 16, no. 1. DOI: 10.1186/s12859-015-0548-6.

[64] LARRAÑAGA, P., CALVO, B., SANTANA, R., BIELZA, C., GALDIANO, J. et al. Machine learning in bioinformatics. *Briefings in Bioinformatics*. march 2006, vol. 7, no. 1, p. 86–112. DOI: 10.1093/bib/bbk007. ISSN 1467-5463. Available at: https://doi.org/10.1093/bib/bbk007.

[65] LAZARIDIS, T. Effective energy functions for protein structure prediction. *Current Opinion in Structural Biology*. Elsevier BV. apr 2000, vol. 10, no. 2, p. 139–145. DOI: 10.1016/s0959-440x(00)00063-4.

[66] LEHMANN, M., KOSTREWA, D., WYSS, M., BRUGGER, R., D'ARCY, A. et al. From DNA sequence to improved functionality: using protein sequence comparisons to rapidly design a thermostable consensus phytase. *Protein Engineering, Design and*

Selection. Oxford University Press (OUP). jan 2000, vol. 13, no. 1, p. 49–57. DOI: 10.1093/protein/13.1.49.

[67] Li, Y. and Fang, J. PROTS-RF: A Robust Model for Predicting Mutation-Induced Protein Stability Changes. *PLoS ONE.* Public Library of Science (PLoS). oct 2012, vol. 7, no. 10, p. e47247. DOI: 10.1371/journal.pone.0047247.

[68] Liaw, A. and Wiener, M. C. Classification and Regression by randomForest. 2007, p. 18–22.

[69] Ling, C. and Sheng, V. Cost-Sensitive Learning and the Class Imbalance Problem. *Encyclopedia of Machine Learning.* january 2010.

[70] Liu, H. On statistical energy functions for biomolecular modeling and design. *Quantitative Biology.* Engineering Sciences Press. dec 2015, vol. 3, no. 4, p. 157–167. DOI: 10.1007/s40484-015-0054-x.

[71] Magliery, T. J. Protein stability: computation, sequence statistics, and new experimental methods. *Current Opinion in Structural Biology.* Elsevier BV. aug 2015, vol. 33, p. 161–168. DOI: 10.1016/j.sbi.2015.09.002.

[72] Mandal, S. and Bose, S. HAND SANITIZERS BID FAREWELL TO GERMS ON SURFACE AREA OF HANDS. april 2020.

[73] Masso, M. and Vaisman, I. I. AUTO-MUTE 2.0: A Portable Framework with Enhanced Capabilities for Predicting Protein Functional Consequences upon Mutation. *Advances in Bioinformatics.* Hindawi Limited. aug 2014, vol. 2014, p. 1–7. DOI: 10.1155/2014/278385.

[74] Mendes, J., Guerois, R. and Serrano, L. Energy estimation in protein design. *Current Opinion in Structural Biology.* Elsevier BV. aug 2002, vol. 12, no. 4, p. 441–446. DOI: 10.1016/s0959-440x(02)00345-7.

[75] Menzel, P., Stadler, P. F. and Gorodkin, J. maxAlike: maximum likelihood-based sequence reconstruction with application to improved primer design for unknown sequences. *Bioinformatics.* Oxford University Press (OUP). dec 2010, vol. 27, no. 3, p. 317–325. DOI: 10.1093/bioinformatics/btq651.

[76] Modarres, H. P., Mofrad, M. R. and Sanati Nezhad, A. Protein thermostability engineering. *RSC Advances.* Royal Society of Chemistry (RSC). 2016, vol. 6, no. 116, p. 115252–115270. DOI: 10.1039/c6ra16992a.

[77] Montanucci, L., Martelli, P. L., Ben Tal, N. and Fariselli, P. A natural upper bound to the accuracy of predicting protein stability changes upon mutations. *Bioinformatics.* Oxford University Press (OUP). oct 2018, vol. 35, no. 9, p. 1513–1517. DOI: 10.1093/bioinformatics/bty880.

[78] Murphy, K. *Machine learning : a probabilistic perspective.* Cambridge, Mass: MIT Press, 2012. ISBN 9780262018029.

[79] Musil, M., Khan, R. T., Beier, A., Stourac, J., Konegger, H. et al. FireProtASR: A Web Server for Fully Automated Ancestral Sequence Reconstruction. *Briefings in Bioinformatics.* Oxford University Press (OUP). dec 2020. DOI: 10.1093/bib/bbaa337.

[80] Musil, M., Konegger, H., Hon, J., Bednar, D. and Damborsky, J. Computational Design of Stable and Soluble Biocatalysts. *ACS Catalysis*. American Chemical Society (ACS). dec 2018, vol. 9, no. 2, p. 1033–1054. DOI: 10.1021/acscatal.8b03613.

[81] Musil, M., Stourac, J., Bendl, J., Brezovsky, J., Prokop, Z. et al. FireProt: web server for automated design of thermostable proteins. *Nucleic Acids Research*. Oxford University Press (OUP). apr 2017, vol. 45, W1, p. W393–W399. DOI: 10.1093/nar/gkx285.

[82] Nickson, A. A. and Clarke, J. What lessons can be learned from studying the folding of homologous proteins? *Methods*. Elsevier BV. sep 2010, vol. 52, no. 1, p. 38–50. DOI: 10.1016/j.ymeth.2010.06.003.

[83] Pace, C. N., Scholtz, J. M. and Grimsley, G. R. Forces stabilizing proteins. *FEBS Letters*. Wiley. may 2014, vol. 588, no. 14, p. 2177–2184. DOI: 10.1016/j.febslet.2014.05.006.

[84] Pandurangan, A. P., Ochoa Montaño, B., Ascher, D. B. and Blundell, T. L. SDM: a server for predicting effects of mutations on protein stability. *Nucleic Acids Research*. Oxford University Press (OUP). may 2017, vol. 45, W1, p. W229–W235. DOI: 10.1093/nar/gkx439.

[85] Parthiban, V., Gromiha, M. M. and Schomburg, D. CUPSAT: prediction of protein stability upon point mutations. *Nucleic Acids Research*. Oxford University Press (OUP). jul 2006, vol. 34, Web Server, p. W239–W242. DOI: 10.1093/nar/gkl190.

[86] Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B. et al. Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research*. 2011, vol. 12, p. 2825–2830.

[87] Pires, D. E. V., Ascher, D. B. and Blundell, T. L. mCSM: predicting the effects of mutations in proteins using graph-based signatures. *Bioinformatics*. Oxford University Press (OUP). nov 2013, vol. 30, no. 3, p. 335–342. DOI: 10.1093/bioinformatics/btt691.

[88] Porebski, B. T. and Buckle, A. M. Consensus protein design. *Protein Engineering Design and Selection*. Oxford University Press (OUP). jun 2016, vol. 29, no. 7, p. 245–251. DOI: 10.1093/protein/gzw015.

[89] Potapov, V., Cohen, M. and Schreiber, G. Assessing computational methods for predicting protein stability upon mutation: good on average but not in the details. *Protein Engineering Design and Selection*. Oxford University Press (OUP). jun 2009, vol. 22, no. 9, p. 553–560. DOI: 10.1093/protein/gzp030.

[90] Powers, D. M. W. *Evaluation: from precision, recall and F-measure to ROC, informedness, markedness and correlation*. 2020.

[91] Pucci, F., Bernaerts, K. V., Kwasigroch, J. M. and Rooman, M. Quantification of biases in predictions of protein stability changes upon mutations. *Bioinformatics*. Oxford University Press (OUP). apr 2018, vol. 34, no. 21, p. 3659–3665. DOI: 10.1093/bioinformatics/bty348.

[92] Pucci, F., Bourgeas, R. and Rooman, M. Predicting protein thermal stability changes upon point mutations using statistical potentials: Introducing HoTMuSiC. *Scientific Reports*. Springer Science and Business Media LLC. mar 2016, vol. 6, no. 1. DOI: 10.1038/srep23257.

[93] Quan, L., Lv, Q. and Zhang, Y. STRUM: structure-based prediction of protein stability changes upon single-point mutation. *Bioinformatics*. Oxford University Press (OUP). jun 2016, vol. 32, no. 19, p. 2936–2946. DOI: 10.1093/bioinformatics/btw361.

[94] Rao, R. B., Fung, G. and Rosales, R. On the Dangers of Cross-Validation. An Experimental Evaluation. In: *Proceedings of the 2008 SIAM International Conference on Data Mining*. Society for Industrial and Applied Mathematics, Apr 2008. DOI: 10.1137/1.9781611972788.54.

[95] Raschka, S. MLxtend: Providing machine learning and data science utilities and extensions to Python's scientific computing stack. *The Journal of Open Source Software*. The Open Journal. april 2018, vol. 3, no. 24. DOI: 10.21105/joss.00638. Available at: http://joss.theoj.org/papers/10.21105/joss.00638.

[96] Russell, S. *Artificial intelligence : a modern approach*. Upper Saddle River, N.J: Prentice Hall/Pearson Education, 2003. ISBN 0137903952.

[97] Sagulenko, P., Puller, V. and Neher, R. A. TreeTime: Maximum-likelihood phylodynamic analysis. *Virus Evolution*. Oxford University Press (OUP). jan 2018, vol. 4, no. 1. DOI: 10.1093/ve/vex042.

[98] Savojardo, C., Fariselli, P., Martelli, P. L. and Casadio, R. INPS-MD: a web server to predict stability of protein variants from sequence and structure. *Bioinformatics*. Oxford University Press (OUP). apr 2016, vol. 32, no. 16, p. 2542–2544. DOI: 10.1093/bioinformatics/btw192.

[99] Schymkowitz, J., Borg, J., Stricher, F., Nys, R., Rousseau, F. et al. The FoldX web server: an online force field. *Nucleic Acids Research*. Oxford University Press (OUP). jul 2005, vol. 33, Web Server, p. W382–W388. DOI: 10.1093/nar/gki387.

[100] Seeliger, D. and Groot, B. L. de. Protein Thermostability Calculations Using Alchemical Free Energy Simulations. *Biophysical Journal*. Elsevier BV. may 2010, vol. 98, no. 10, p. 2309–2316. DOI: 10.1016/j.bpj.2010.01.051.

[101] Sinden, R. R. DNA–Protein Interactions. In: *DNA Structure and Function*. Elsevier, 1994, p. 287–325. DOI: 10.1016/b978-0-08-057173-7.50013-4.

[102] Stamatakis, A. RAxML-VI-HPC: maximum likelihood-based phylogenetic analyses with thousands of taxa and mixed models. *Bioinformatics*. Oxford University Press (OUP). aug 2006, vol. 22, no. 21, p. 2688–2690. DOI: 10.1093/bioinformatics/btl446.

[103] Steipe, B., Schiller, B., Plückthun, A. and Steinbacher, S. Sequence Statistics Reliably Predict Stabilizing Mutations in a Protein Domain. *Journal of Molecular Biology*. Elsevier BV. jul 1994, vol. 240, no. 3, p. 188–192. DOI: 10.1006/jmbi.1994.1434.

[104] Steward, C. International Human Genome Sequencing Consortium: Initial sequencing and analysis of the human genome 409, 860921. *Nature.* february 2001, vol. 409, p. 860–921.

[105] Stourac, J., Dubrava, J., Musil, M., Horackova, J., Damborsky, J. et al. FireProtDB: database of manually curated protein stability data. *Nucleic Acids Research.* Oxford University Press (OUP). nov 2020, vol. 49, D1, p. D319–D324. DOI: 10.1093/nar/gkaa981.

[106] Sullivan, B. J., Nguyen, T., Durani, V., Mathur, D., Rojas, S. et al. Stabilizing Proteins from Sequence Statistics: The Interplay of Conservation and Correlation in Triosephosphate Isomerase Stability. *Journal of Molecular Biology.* Elsevier BV. jul 2012, vol. 420, 4-5, p. 384–399. DOI: 10.1016/j.jmb.2012.04.025.

[107] Sussman, J. L., Lin, D., Jiang, J., Manning, N. O., Prilusky, J. et al. Protein Data Bank (PDB): Database of Three-Dimensional Structural Information of Biological Macromolecules. *Acta Crystallographica Section D Biological Crystallography.* International Union of Crystallography (IUCr). nov 1998, vol. 54, no. 6, p. 1078–1084. DOI: 10.1107/s0907444998009378.

[108] Teng, S., Srivastava, A. K. and Wang, L. Sequence feature-based prediction of protein stability changes upon amino acid substitutions. *BMC Genomics.* Springer Science and Business Media LLC. 2010, vol. 11, Suppl 2, p. S5. DOI: 10.1186/1471-2164-11-s2-s5.

[109] Tian, J., Wu, N., Chu, X. and Fan, Y. Predicting changes in protein thermostability brought about by single- or multi-site mutations. *BMC Bioinformatics.* Springer Science and Business Media LLC. jul 2010, vol. 11, no. 1. DOI: 10.1186/1471-2105-11-370.

[110] Trainor, K., Broom, A. and Meiering, E. M. Exploring the relationships between protein sequence, structure and solubility. *Current Opinion in Structural Biology.* Elsevier BV. feb 2017, vol. 42, p. 136–146. DOI: 10.1016/j.sbi.2017.01.004.

[111] Tria, F. D. K., Landan, G. and Dagan, T. Phylogenetic rooting using minimal ancestor deviation. *Nature Ecology & Evolution.* Springer Science and Business Media LLC. jun 2017, vol. 1, no. 1. DOI: 10.1038/s41559-017-0193.

[112] Usmanova, D. R., Bogatyreva, N. S., Bernad, J. A., Eremina, A. A., Gorshkova, A. A. et al. Self-consistency test reveals systematic bias in programs for prediction change of stability upon mutation. *Bioinformatics.* Oxford University Press (OUP). may 2018, vol. 34, no. 21, p. 3653–3658. DOI: 10.1093/bioinformatics/bty340.

[113] Wainreb, G., Wolf, L., Ashkenazy, H., Dehouck, Y. and Ben Tal, N. Protein stability: a single recorded mutation aids in predicting the effects of other mutations in the same amino acid site. *Bioinformatics.* Oxford University Press (OUP). oct 2011, vol. 27, no. 23, p. 3286–3292. DOI: 10.1093/bioinformatics/btr576.

[114] Wang, G. and Dunbrack, R. L. PISCES: a protein sequence culling server. *Bioinformatics.* Oxford University Press (OUP). aug 2003, vol. 19, no. 12, p. 1589–1591. DOI: 10.1093/bioinformatics/btg224.

[115] Westesson, O., Barquist, L. and Holmes, I. HandAlign: Bayesian multiple sequence alignment, phylogeny and ancestral reconstruction. *Bioinformatics.* Oxford University Press (OUP). jan 2012, vol. 28, no. 8, p. 1170–1171. DOI: 10.1093/bioinformatics/bts058.

[116] Wheeler, L. C., Lim, S. A., Marqusee, S. and Harms, M. J. The thermostability and specificity of ancient proteins. *Current Opinion in Structural Biology.* Elsevier BV. jun 2016, vol. 38, p. 37–43. DOI: 10.1016/j.sbi.2016.05.015.

[117] Whitehead, T. A., Chevalier, A., Song, Y., Dreyfus, C., Fleishman, S. J. et al. Optimization of affinity, specificity and function of designed influenza inhibitors using deep sequencing. *Nature Biotechnology.* Springer Science and Business Media LLC. may 2012, vol. 30, no. 6, p. 543–548. DOI: 10.1038/nbt.2214.

[118] Wijma, H. J., Floor, R. J., Jekel, P. A., Baker, D., Marrink, S. J. et al. Computationally designed libraries for rapid enzyme stabilization. *Protein Engineering Design and Selection.* Oxford University Press (OUP). jan 2014, vol. 27, no. 2, p. 49–58. DOI: 10.1093/protein/gzt061.

[119] Wijma, H. J., Floor, R. J. and Janssen, D. B. Structure- and sequence-analysis inspired engineering of proteins for enhanced thermostability. *Current Opinion in Structural Biology.* Elsevier BV. aug 2013, vol. 23, no. 4, p. 588–594. DOI: 10.1016/j.sbi.2013.04.008.

[120] Witvliet, D. K., Strokach, A., Giraldo Forero, A. F., Teyra, J., Colak, R. et al. ELASPIC web-server: proteome-wide structure-based prediction of mutation effects on protein stability and binding affinity. *Bioinformatics.* Oxford University Press (OUP). jan 2016, vol. 32, no. 10, p. 1589–1591. DOI: 10.1093/bioinformatics/btw031.

[121] Yang, Z. PAML: a program package for phylogenetic analysis by maximum likelihood. *Bioinformatics.* Oxford University Press (OUP). 1997, vol. 13, no. 5, p. 555–556. DOI: 10.1093/bioinformatics/13.5.555.

[122] Yin, S., Ding, F. and Dokholyan, N. V. Eris: an automated estimator of protein stability. *Nature Methods.* Springer Science and Business Media LLC. jun 2007, vol. 4, no. 6, p. 466–467. DOI: 10.1038/nmeth0607-466.

[123] Zwanzig, R., Szabo, A. and Bagchi, B. Levinthal's paradox. *Proceedings of the National Academy of Sciences.* Proceedings of the National Academy of Sciences. jan 1992, vol. 89, no. 1, p. 20–22. DOI: 10.1073/pnas.89.1.20.

Appendix A

Detailed results

Attempt	Class	ACC	MCC	TRP	TNR	PPV	NPV	FPR	FNR	FDR
1	0	0.49	0.00	0.11	0.89	0.51	0.48	0.11	0.89	0.49
	1	0.49	0.00	0.89	0.11	0.48	0.51	0.89	0.11	0.52
2	0	0.49	0.00	0.11	0.89	0.51	0.48	0.11	0.89	0.49
	1	0.49	0.00	0.89	0.11	0.48	0.51	0.89	0.11	0.52
3	0	0.47	-0.05	0.35	0.60	0.48	0.46	0.40	0.65	0.52
	1	0.47	-0.05	0.60	0.35	0.46	0.48	0.65	0.40	0.54
4	0	0.47	-0.11	0.79	0.13	0.49	0.36	0.87	0.21	0.51
	1	0.47	-0.11	0.13	0.79	0.36	0.49	0.21	0.87	0.64
5	0	0.48	-0.10	0.79	0.13	0.51	0.36	0.87	0.21	0.49
	1	0.48	-0.10	0.13	0.79	0.36	0.51	0.21	0.87	0.64
6	0	0.48	-0.07	0.62	0.32	0.51	0.42	0.68	0.38	0.49
	1	0.48	-0.07	0.32	0.62	0.42	0.51	0.38	0.68	0.58
7	0	0.48	-0.12	0.79	0.12	0.50	0.34	0.88	0.21	0.50
	1	0.48	-0.12	0.12	0.79	0.34	0.50	0.21	0.88	0.66
8	0	0.49	-0.02	0.51	0.47	0.52	0.46	0.53	0.49	0.48
	1	0.49	-0.02	0.47	0.51	0.46	0.52	0.49	0.53	0.54
9	0	0.47	-0.01	0.11	0.89	0.52	0.47	0.11	0.89	0.48
	1	0.47	-0.01	0.89	0.11	0.47	0.52	0.89	0.11	0.53
10	0	0.47	-0.14	0.80	0.10	0.50	0.31	0.90	0.20	0.50
	1	0.47	-0.14	0.10	0.80	0.31	0.50	0.20	0.90	0.69
11	0	0.46	-0.16	0.79	0.09	0.49	0.29	0.91	0.21	0.51
	1	0.46	-0.16	0.09	0.79	0.29	0.49	0.21	0.91	0.71
12	0	0.48	-0.13	0.82	0.09	0.50	0.32	0.91	0.18	0.50
	1	0.48	-0.13	0.09	0.82	0.32	0.50	0.18	0.91	0.68
13	0	0.47	-0.13	0.81	0.10	0.50	0.32	0.90	0.19	0.50
	1	0.47	-0.13	0.10	0.81	0.32	0.50	0.19	0.90	0.68
14	0	0.48	-0.12	0.84	0.08	0.50	0.31	0.92	0.16	0.50
	1	0.48	-0.12	0.08	0.84	0.31	0.50	0.16	0.92	0.69

Attempt	Class	ACC	MCC	TRP	TNR	PPV	NPV	FPR	FNR	FDR
15	0	0.46	-0.16	0.81	0.08	0.49	0.28	0.92	0.19	0.51
	1	0.46	-0.16	0.08	0.81	0.28	0.49	0.19	0.92	0.72
16	0	0.48	-0.10	0.82	0.10	0.51	0.35	0.90	0.18	0.49
	1	0.48	-0.10	0.10	0.82	0.35	0.51	0.18	0.90	0.65
17	0	0.47	-0.14	0.82	0.08	0.50	0.29	0.92	0.18	0.50
	1	0.47	-0.14	0.08	0.82	0.29	0.50	0.18	0.92	0.71
18	0	0.48	-0.11	0.82	0.10	0.50	0.33	0.90	0.18	0.50
	1	0.48	-0.11	0.10	0.82	0.33	0.50	0.18	0.90	0.67
19	0	0.48	-0.12	0.81	0.10	0.50	0.33	0.90	0.19	0.50
	1	0.48	-0.12	0.10	0.81	0.33	0.50	0.19	0.90	0.67
20	0	0.48	-0.01	0.08	0.91	0.52	0.47	0.09	0.92	0.48
	1	0.48	-0.01	0.91	0.08	0.47	0.52	0.92	0.09	0.53
21	0	0.32	0.14	0.11	0.97	0.90	0.27	0.03	0.89	0.10
	1	0.32	0.14	0.97	0.11	0.27	0.90	0.89	0.03	0.73
22	0	0.96	0.94	0.96	0.98	0.99	0.89	0.02	0.04	0.01
	1	0.96	0.94	0.98	0.96	0.89	0.99	0.04	0.02	0.11
23	0	0.96	0.94	0.96	0.98	0.99	0.88	0.02	0.04	0.01
	1	0.96	0.94	0.98	0.96	0.88	0.99	0.04	0.02	0.12
24	0	0.96	0.94	0.96	0.98	0.99	0.89	0.02	0.04	0.01
	1	0.96	0.94	0.98	0.96	0.89	0.99	0.04	0.02	0.11
25	0	0.47	-0.14	0.82	0.09	0.50	0.30	0.91	0.18	0.50
	1	0.47	-0.14	0.09	0.82	0.30	0.50	0.18	0.91	0.70
26	0	0.47	-0.14	0.83	0.08	0.50	0.30	0.92	0.17	0.50
	1	0.47	-0.14	0.08	0.83	0.30	0.50	0.17	0.92	0.70
27	0	0.48	-0.13	0.82	0.09	0.50	0.31	0.91	0.18	0.50
	1	0.48	-0.13	0.09	0.82	0.31	0.50	0.18	0.91	0.69
28	0	0.48	-0.13	0.83	0.09	0.50	0.31	0.91	0.17	0.50
	1	0.48	-0.13	0.09	0.83	0.31	0.50	0.17	0.91	0.69
29	0	0.62	0.10	0.89	0.17	0.64	0.50	0.83	0.11	0.36
	1	0.62	0.10	0.17	0.89	0.50	0.64	0.11	0.83	0.50
30	0	0.97	0.95	0.97	0.98	0.99	0.93	0.02	0.03	0.01
	1	0.97	0.95	0.98	0.97	0.93	0.99	0.03	0.02	0.07
31	0	0.63	0.12	0.89	0.19	0.64	0.53	0.81	0.11	0.36
	1	0.63	0.12	0.19	0.89	0.53	0.64	0.11	0.81	0.47
32	0	0.62	0.09	0.89	0.17	0.64	0.49	0.83	0.11	0.37
	1	0.62	0.09	0.17	0.89	0.49	0.64	0.11	0.83	0.51
33	0	0.62	0.10	0.89	0.18	0.64	0.51	0.82	0.11	0.36
	1	0.62	0.10	0.18	0.89	0.51	0.64	0.11	0.82	0.49
34	0	0.60	0.06	0.85	0.19	0.63	0.45	0.81	0.15	0.37
	1	0.60	0.06	0.19	0.85	0.45	0.63	0.15	0.81	0.55
35	0	0.80	0.47	0.89	0.55	0.84	0.66	0.45	0.11	0.16
	1	0.80	0.47	0.55	0.89	0.66	0.84	0.11	0.45	0.34
36	0	0.60	0.06	0.84	0.21	0.63	0.45	0.79	0.16	0.37
	1	0.60	0.06	0.21	0.84	0.45	0.63	0.16	0.79	0.55
37	0	0.57	0.01	0.82	0.18	0.61	0.40	0.82	0.18	0.39
	1	0.57	0.01	0.18	0.82	0.40	0.61	0.18	0.82	0.60

Attempt	Class	ACC	MCC	TRP	TNR	PPV	NPV	FPR	FNR	FDR
38	0	0.28	-0.36	0.02	0.74	0.10	0.31	0.26	0.98	0.90
	1	0.28	-0.36	0.74	0.02	0.31	0.10	0.98	0.26	0.69
39	0	0.29	-0.33	0.02	0.75	0.13	0.31	0.25	0.98	0.87
	1	0.29	-0.33	0.75	0.02	0.31	0.13	0.98	0.25	0.69
40	0	0.71	0.35	0.99	0.24	0.69	0.96	0.76	0.01	0.31
	1	0.71	0.35	0.24	0.99	0.96	0.69	0.01	0.76	0.04
41	0	0.57	-0.01	0.82	0.17	0.61	0.38	0.83	0.18	0.39
	1	0.57	-0.01	0.17	0.82	0.38	0.61	0.18	0.83	0.63
42	0	0.69	0.30	0.97	0.24	0.68	0.81	0.76	0.03	0.32
	1	0.69	0.30	0.24	0.97	0.81	0.68	0.03	0.76	0.19
43	0	0.71	0.34	0.99	0.24	0.69	0.93	0.76	0.01	0.31
	1	0.71	0.34	0.24	0.99	0.93	0.69	0.01	0.76	0.07
44	0	0.59	0.04	0.85	0.18	0.62	0.44	0.82	0.15	0.38
	1	0.59	0.04	0.18	0.85	0.44	0.62	0.15	0.82	0.56
45	0	0.58	0.05	0.80	0.24	0.63	0.43	0.76	0.20	0.37
	1	0.58	0.05	0.24	0.80	0.43	0.63	0.20	0.76	0.57
46	0	0.71	0.34	0.99	0.24	0.69	0.93	0.76	0.01	0.31
	1	0.71	0.34	0.24	0.99	0.93	0.69	0.01	0.76	0.07
47	0	0.70	0.32	0.98	0.23	0.68	0.89	0.77	0.02	0.32
	1	0.70	0.32	0.23	0.98	0.89	0.68	0.02	0.77	0.11
48	0	0.70	0.32	0.98	0.24	0.68	0.86	0.76	0.02	0.32
	1	0.70	0.32	0.24	0.98	0.86	0.68	0.02	0.76	0.14
49	0	0.71	0.34	0.98	0.25	0.69	0.90	0.75	0.02	0.31
	1	0.71	0.34	0.25	0.98	0.90	0.69	0.02	0.75	0.10
50	0	0.67	0.24	0.98	0.15	0.66	0.84	0.85	0.02	0.34
	1	0.67	0.24	0.15	0.98	0.84	0.66	0.02	0.85	0.16
51	0	0.64	0.12	1.00	0.03	0.63	1.00	0.97	0.00	0.37
	1	0.64	0.12	0.03	1.00	1.00	0.63	0.00	0.97	0.00

Table A.1: Table of results when training two classes

Attempt	ACC	MCC	F1	Recall	Precision	TPR
1	0.64	0.32	0.49	0.48	0.76	0.75,0.1,0.58
2	0.62	0.27	0.48	0.46	0.75	0.73,0.1,0.56
3	0.65	0.33	0.49	0.48	0.60	0.76,0.1,0.58
4	0.62	0.28	0.47	0.46	0.58	0.75,0.1,0.54
5	0.61	0.26	0.47	0.45	0.58	0.75,0.1,0.52
6	0.66	0.35	0.51	0.49	0.77	0.72,0.1,0.65
7	0.67	0.37	0.52	0.50	0.78	0.71,0.1,0.69
8	0.66	0.35	0.51	0.49	0.77	0.69,0.1,0.68
9	0.64	0.31	0.49	0.47	0.76	0.75,0.1,0.57
10	0.66	0.36	0.51	0.49	0.78	0.79,0.1,0.59
11	0.62	0.29	0.47	0.46	0.53	0.8,0.1,0.5
12	0.62	0.30	0.45	0.46	0.45	0.67,0.1,0.62
13	0.70	0.42	0.53	0.51	0.80	0.75,0.1,0.7
14	0.63	0.29	0.48	0.47	0.59	0.73,0.1,0.57
15	0.68	0.38	0.51	0.50	0.62	0.74,0.1,0.66
16	0.66	0.35	0.50	0.49	0.56	0.75,0.1,0.63
17	0.66	0.35	0.49	0.48	0.49	0.71,0.1,0.66
18	0.64	0.33	0.49	0.50	0.48	0.73,0.2,0.58
19	0.32	0.24	0.29	0.54	0.57	0.54,1,0.08
20	0.34	0.25	0.31	0.55	0.56	0.57,1,0.1
21	0.34	0.23	0.30	0.55	0.52	0.57,1,0.09
22	0.46	0.22	0.37	0.57	0.48	0.71,0.8,0.21
23	0.65	0.32	0.44	0.44	0.44	0.58,0,0.76
24	0.47	0.24	0.38	0.58	0.49	0.71,0.8,0.23
25	0.32	0.21	0.27	0.54	0.47	0.57,1,0.05
26	0.52	0.29	0.43	0.58	0.49	0.7,0.7,0.35
27	0.52	0.29	0.44	0.58	0.49	0.69,0.7,0.37
28	0.54	0.31	0.45	0.59	0.51	0.71,0.7,0.37
29	0.53	0.31	0.44	0.59	0.50	0.71,0.7,0.37

Table A.2: Detailed results when training three classes with interval between -1 and 1 kcal/mol

Attempt	ACC	MCC	F1	Recall	Precision	TPR	TNR
1	0.61	0.29	0.44	0.46	0.73	0.72,0.03,0.64	0.63,1,0.67
2	0.62	0.34	0.46	0.48	0.75	0.68,0.03,0.74	0.71,1,0.63
3	0.65	0.36	0.47	0.49	0.59	0.79,0.03,0.65	0.63,1,0.73
4	0.64	0.34	0.46	0.48	0.75	0.77,0.03,0.65	0.63,1,0.71
5	0.61	0.29	0.45	0.46	0.62	0.8,0.06,0.53	0.53,1,0.76
6	0.61	0.29	0.44	0.45	0.57	0.82,0.03,0.52	0.53,1,0.76
7	0.61	0.29	0.45	0.46	0.73	0.77,0.06,0.58	0.55,1,0.74
8	0.62	0.31	0.46	0.47	0.63	0.79,0.06,0.58	0.57,1,0.74
9	0.62	0.32	0.45	0.47	0.74	0.74,0.03,0.65	0.64,1,0.68
10	0.64	0.34	0.46	0.48	0.75	0.77,0.03,0.65	0.64,1,0.71
11	0.61	0.30	0.46	0.47	0.62	0.73,0.06,0.64	0.63,1,0.68
12	0.62	0.32	0.45	0.47	0.74	0.7,0.03,0.71	0.69,1,0.64
13	0.65	0.37	0.47	0.49	0.76	0.73,0.03,0.73	0.71,1,0.67
14	0.61	0.29	0.44	0.46	0.73	0.77,0.03,0.58	0.57,1,0.72
15	0.61	0.32	0.45	0.47	0.74	0.67,0.03,0.73	0.71,1,0.62
16	0.62	0.32	0.45	0.47	0.74	0.71,0.03,0.69	0.67,1,0.66
17	0.63	0.34	0.46	0.48	0.58	0.71,0.03,0.72	0.7,1,0.65
18	0.59	0.26	0.43	0.45	0.72	0.7,0.03,0.62	0.62,1,0.65
19	0.61	0.31	0.45	0.47	0.57	0.68,0.03,0.71	0.69,1,0.64
20	0.63	0.36	0.49	0.50	0.75	0.69,0.09,0.74	0.73,1,0.63
21	0.64	0.38	0.50	0.51	0.76	0.7,0.09,0.75	0.74,1,0.64
22	0.63	0.35	0.49	0.50	0.62	0.67,0.09,0.76	0.75,1,0.62
23	0.62	0.34	0.48	0.49	0.75	0.67,0.09,0.74	0.72,1,0.63
24	0.63	0.35	0.47	0.49	0.53	0.72,0.06,0.7	0.7,0.99,0.67
25	0.64	0.36	0.49	0.50	0.67	0.7,0.09,0.74	0.71,1,0.66
26	0.65	0.39	0.50	0.51	0.63	0.73,0.09,0.73	0.73,1,0.68
27	0.63	0.34	0.47	0.49	0.75	0.71,0.06,0.71	0.68,1,0.67
28	0.64	0.37	0.48	0.50	0.65	0.71,0.06,0.74	0.71,1,0.66
29	0.64	0.36	0.48	0.50	0.65	0.71,0.06,0.73	0.7,1,0.67
30	0.65	0.38	0.49	0.51	0.76	0.7,0.06,0.78	0.75,1,0.64
31	0.64	0.37	0.49	0.50	0.63	0.7,0.09,0.74	0.73,1,0.65
32	0.64	0.38	0.50	0.51	0.76	0.68,0.09,0.78	0.76,1,0.63
33	0.64	0.37	0.48	0.50	0.60	0.71,0.06,0.75	0.73,1,0.66
34	0.63	0.36	0.49	0.50	0.76	0.67,0.09,0.77	0.75,1,0.62

Table A.3: Detailed results when training three classes with interval between -0.5 and 0.5 kcal/mol

Attempt	MAE	MSE	RMSE	PCC	model
1	1.24	3.53	1.88	0.56	XGBRegressor1
2	1.22	3.15	1.78	0.62	XGBRegressor2
3	1.20	3.28	1.81	0.60	XGBRegressor3
4	1.16	3.33	1.82	0.61	XGBRegressor4
5	1.25	3.51	1.87	0.60	XGBRegressor5
6	1.23	3.84	1.96	0.55	XGBRegressor6
7	1.25	3.53	1.88	0.55	XGBRegressor7
8	1.17	2.81	1.68	0.53	XGBRegressor8
9	1.21	3.14	1.77	0.62	XGBRegressor9
10	1.16	2.89	1.70	0.60	XGBRegressor10
11	1.22	3.15	1.77	0.53	XGBRegressor11
12	1.19	3.43	1.85	0.60	XGBRegressor12
13	1.28	3.56	1.89	0.56	XGBRegressor13
14	1.22	3.32	1.82	0.57	XGBRegressor14
15	1.16	2.88	1.70	0.62	XGBRegressor15
16	1.17	3.04	1.74	0.61	XGBRegressor16
17	1.21	3.53	1.88	0.52	XGBRegressor17
18	1.27	3.72	1.93	0.58	XGBRegressor18
19	1.16	2.81	1.68	0.61	XGBRegressor19
20	1.41	4.35	2.08	0.17	LinearRegression1
21	1.51	4.91	2.22	0.24	LinearRegression2
22	1.46	4.20	2.05	0.29	LinearRegression3
23	1.45	4.47	2.11	0.21	LinearRegression4
24	1.58	5.35	2.31	0.21	LinearRegression5
25	1.43	4.24	2.06	0.20	LinearRegression6
26	1.45	4.07	2.02	0.26	LinearRegression7
27	1.42	4.36	2.09	0.22	LinearRegression8
28	1.50	5.01	2.24	0.22	LinearRegression9
29	1.41	3.79	1.95	0.20	LinearRegression10
30	1.40	4.08	2.02	0.16	LinearRegression11
31	1.49	4.74	2.18	0.22	LinearRegression12
32	1.47	4.68	2.16	0.14	LinearRegression13
33	1.49	5.22	2.29	0.19	LinearRegression14
34	1.47	4.17	2.04	0.25	LinearRegression15
35	1.52	5.12	2.26	0.18	LinearRegression16
36	1.53	4.91	2.22	0.24	LinearRegression17
37	1.47	4.31	2.08	0.25	LinearRegression18
38	1.52	4.71	2.17	0.23	LinearRegression19
39	1.45	4.19	2.05	0.25	HuberRegressor1
40	1.45	4.46	2.11	0.15	HuberRegressor2
41	1.54	5.08	2.25	0.16	HuberRegressor3
42	1.51	4.69	2.17	0.25	HuberRegressor4
43	1.47	4.34	2.08	0.22	HuberRegressor5
44	1.49	4.60	2.15	0.21	HuberRegressor6
45	1.45	4.41	2.10	0.25	HuberRegressor7
46	1.45	4.17	2.04	0.27	HuberRegressor8
47	1.42	4.63	2.15	0.23	HuberRegressor9
48	1.40	3.93	1.98	0.23	HuberRegressor10

Attempt	MAE	MSE	RMSE	PCC	model
49	1.41	4.01	2.00	0.20	HuberRegressor11
50	1.44	4.45	2.11	0.20	HuberRegressor12
51	1.52	4.91	2.21	0.21	HuberRegressor13
52	1.40	4.20	2.05	0.20	HuberRegressor14
53	1.46	4.40	2.10	0.22	HuberRegressor15
54	1.49	4.43	2.10	0.23	HuberRegressor16
55	1.48	5.01	2.24	0.22	HuberRegressor17
56	1.50	4.50	2.12	0.19	HuberRegressor18
57	1.50	5.13	2.26	0.24	HuberRegressor19
58	1.52	6.31	2.51	0.41	DTRegressor1
59	1.47	5.72	2.39	0.44	DTRegressor2
60	1.61	5.78	2.40	0.46	DTRegressor3
61	1.48	5.22	2.28	0.45	DTRegressor4
62	1.48	5.36	2.32	0.48	DTRegressor5
63	1.55	5.99	2.45	0.41	DTRegressor6
64	1.54	6.07	2.46	0.41	DTRegressor7
65	1.49	5.62	2.37	0.50	DTRegressor8
66	1.49	5.27	2.29	0.42	DTRegressor9
67	1.51	5.36	2.31	0.43	DTRegressor10
68	1.42	4.89	2.21	0.45	DTRegressor11
69	1.48	5.49	2.34	0.43	DTRegressor12
70	1.55	5.97	2.44	0.42	DTRegressor13
71	1.45	5.28	2.30	0.44	DTRegressor14
72	1.53	5.41	2.33	0.47	DTRegressor15
73	1.51	5.75	2.40	0.49	DTRegressor16
74	1.51	5.18	2.28	0.46	DTRegressor17
75	1.46	5.18	2.28	0.45	DTRegressor18
76	1.44	5.13	2.27	0.39	DTRegressor19
77	1.40	4.02	2.01	0.46	GBRegressor1
78	1.37	4.14	2.03	0.43	GBRegressor2
79	1.37	4.14	2.04	0.44	GBRegressor3
80	1.36	4.00	2.00	0.43	GBRegressor4
81	1.41	4.47	2.11	0.43	GBRegressor5
82	1.37	3.88	1.97	0.40	GBRegressor6
83	1.38	3.80	1.95	0.36	GBRegressor7
84	1.34	3.96	1.99	0.39	GBRegressor8
85	1.35	4.02	2.01	0.45	GBRegressor9
86	1.40	3.79	1.95	0.41	GBRegressor10
87	1.49	4.70	2.17	0.43	GBRegressor11
88	1.28	3.03	1.74	0.50	GBRegressor12
89	1.43	4.01	2.00	0.45	GBRegressor13
90	1.40	4.43	2.10	0.41	GBRegressor14
91	1.31	3.37	1.84	0.48	GBRegressor15
92	1.36	3.97	1.99	0.46	GBRegressor16
93	1.35	3.74	1.93	0.47	GBRegressor17
94	1.41	4.31	2.08	0.43	GBRegressor18
95	1.39	3.96	1.99	0.39	GBRegressor19

Table A.4: Detailed results of regression