BRNO UNIVERSITY OF TECHNOLOGY
VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

FACULTY OF INFORMATION TECHNOLOGY
FAKULTA INFORMAČNÍCH TECHNOLOGIÍ

DEPARTMENT OF INFORMATION SYSTEMS
ÚSTAV INFORMAČNÍCH SYSTÉMŮ

COMPUTATIONAL WORKFLOW FOR THE PREDICTION AND DESIGN OF STABLE PROTEINS

VÝPOČETNÍ METODA PRO PREDIKCI A KONSTRUKCI STABILNÍCH PROTEINŮ

MASTER'S THESIS
DIPLOMOVÁ PRÁCE

AUTHOR **Bc. JOSEF KADLEČEK**
AUTOR PRÁCE

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

BRNO 2022

Brno University of Technology
Faculty of Information Technology

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

Master's Thesis Specification

Student: **Kadleček Josef, Bc.**
Programme: Information Technology and Artificial Intelligence
Specialization: Bioinformatics and Biocomputing
Title: **Computational Workflow for the Prediction and Design of Stable Proteins**
Category: Biocomputing
Assignment:

1. Study the problematics of protein engineering and protein stability.
2. Study various methods for the prediction of the effect of mutations on protein stability, i.e. force-field calculations, machine learning, and evolution-based methods.
3. Study the current state of the FireProt tool.
4. Design a new method that would increase the robustness of the previous solution and extend it with novel approaches for the construction of the stable proteins such as calculation of the saturation mutagenesis using the force-field calculations, flexibility predictions (b-factors), or usage of the position-specific matrices.
5. Implement your solution and parallelize the calculations using HPC.
6. Evaluate the robustness of the implemented tool and the precision of the obtained results.

Recommended literature:

- Mazurenko, S., 2020: Predicting Protein Stability and Solubility Changes Upon Mutations: Data Perspective. ChemCatChem 12: 1-10.
- Musil, M., Stourac, J., Bendl, J., Brezovsky, J., Prokop, Z., Zendulka, J., Martinek, T., Bednar, D., Damborsky, J., 2017: FireProt: web server for automated design of thermostable proteins. Nucleic Acids Research 45: W393-W399.
- Bednar, D., Beerens, K., Sebestova, E., Bendl, J., Khare, S., Chaloupkova, R., Prokop, Z., Brezovky, J., Baker, D., Damborsky, J., 2015: FireProt: Energy- and Evolution-Based Computational Design of Thermostable Multiple-Point Mutants. PLOS Computational Biology 11: e1004556.
- 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

This thesis focuses on enriching computational tools for the prediction of protein mutations for the purpose of enriching their stability. Stable mutants of proteins are necessary for the development of the new drugs. Unfortunately, experimental validation of the stabilization effect of given mutations is costly and time demanding. Therefore, the FireProt tool was developed. FireProt utilizes a combination of both evolutionary and energy-based approaches for identifying potentially stabilizing mutations. This thesis enriches the FireProt tool with the possibility of entering custom mutations for the analysis. It also integrates other prediction software, FireProt ASR, and provides user with an option to select multiple mutational strategies for the detection of stabilizing mutations. Furthermore, it enriches the FireProt tool with another information about proteins residues (for instance relative B-factor) and makes it possible to utilize homology modeling to model the protein's structure based on its sequence. Lastly, it introduces a novel approach for the design of multiple-point mutants.

Abstrakt

Tato práce se soustředí na rozšíření výpočetního nástroje pro predikci mutací proteinů za účelem zvýšení jejich stability. Stabilizace proteinů je nezbytnou součástí návrhu léčiv, a jelikož experimentální vyhodnocení přínosu jednotlivých mutací na stabilitu proteinu je nákladné a zdlouhavé, byl vyvinut nástroj FireProt. FireProt využívá kombinaci evolučních a energetických přístupů pro identifikaci potencionálně stabilizujících mutací. Tato práce rozšiřuje nástroj FireProt o možnost zadat vlastní mutace k analýze, integruje predikční nástroj FireProt ASR a dává možnost uživateli vybrat z několika strategií pro detekci stabilizujících mutací. Tato práce dále rozšiřuje nástroj FireProt o další informace k jednotlivým residuím (jako je například relativní B-faktor), umožňuje využít homologní modelování k vytvoření struktury proteinu na základě jeho sekvence, a v neposlední řadě představuje nový přístup k návrhu vícebodových mutantů.

Keywords

Protein engineering, amino acid mutation, protein stability, stability prediction, multiple-point mutants

Klíčová slova

Proteinové inženýrství, mutace aminokyselin, stabilita proteinů, predikce stability, vícebodové mutace

Reference

KADLEČEK, Josef. *Computational Workflow for the Prediction and Design of Stable Proteins*. Brno, 2022. Master's thesis. Brno University of Technology, Faculty of Information Technology. Supervisor Ing. Musil Miloš, Ph.D.

Rozšířený abstrakt

Proteiny jsou makromolekuly, které hrají důležitou roli v mnoha aspektech našich životů. Slouží jako základní stavební bloky organismů a plní rozličné funkce od enzymů, přes transportní a strukturní proteiny, až po proteiny umožňující pohyb. Proteiny se skládají ze sekvence aminokyselin, respektive z proteinogenních alfa-L-aminokyselin. Každá taková aminokyselina se skládá z karboxylové a aminové funkční skupiny a liší se pouze v navázaném postranním řetězci. Základem pro vznik proteinu je DNA, která se nejprve pomocí mechanismu transkripce přepíše na RNA, a následně dle pravidel genetického kódu přeloží pomocí mechanismu translace na sekvenci aminokyselin. Sekvence aminokyselin (neboli primární struktura proteinu) určuje finální podobu proteinu stejně jako jeho funkci.

Tato práce se zaobírá primárně problematikou stability proteinů z pohledu proteinového inženýrství. Z toho důvodu jsou popsány mechanismy skládání proteinů a jsou rozebrány různé mechanismy jednotlivých mutací: záměny jednotlivých aminokyselin a jejich odstranění nebo vložení. Jelikož u většiny proteinů je známa pouze sekvence, jsou metody pro analýzu stability proteinů rozděleny na ty, kterým stačí znát sekvenci a na metody, které využívají strukturu daného proteinu.

Prvním krokem pro většinu metod využívajících sekvenci jakožto primární zdroj informace je zarovnání sekvencí, neboli MSA (z anglického *multiple sequence alignment*). Toto zarovnání pomáhá nalézt množinu homologních (příbuzných) proteinů, s jejichž pomocí je možné využít metody založené na evoluci daných proteinů.

Metody využívající prostorovou strukturu proteinu často využívají fyzikálně chemické vlastnosti jednotlivých molekul daného proteinu a snaží se namodelovat stav, při kterém bude struktura disponovat nejmenší možnou volnou energií.

V neposlední řadě jsou popsány nástroje pro modelování struktury proteinu na základě jeho sekvence spolu s ohromným pokrokem, který byl v rámci modelování proteinů učiněn.

V této práci jsou popsány nástroje a algoritmy, které se snaží využít jednotlivé principy pro navrhnutí jedno či více-bodových mutací. Hlavním cílem je však vylepšení nástroje FireProt, který je nástrojem hybridním – snaží se kombinovat více přístupů jak na základě evolučního návrhu, tak na základě energetické minimalizace.

Nástroj FireProt využívá evoluční metodiky jako je *back to consensus* a analýza konzervovaných a korelovaných pozic k vyfiltrování mutací, které by mohly poškodit stabilitu proteinu, stejně jako k návrhu mutací s vysokou pravděpodobností přínosu ke stabilitě. Dále používá programy založené na analýze volné energie (FoldX a Rosetta) pro určení konkrétních hodnot jednotlivých mutací spolu s jednoduchým algoritmem pro navrhnutí vícebodového mutantu.

Tato práce představuje vylepšenou verzi nástroje FireProt. Ten byl v předchozích verzích limitován pouze na proteiny se známou strukturou. Proto byl do nástroje FireProt zařazen modul pro modelování struktury proteinu na základě jeho sekvence. Tento modul umožňuje uživateli začít analýzu nad libovolnou sekvencí. Dále umožňuje modelování výsledných multi mutantů pro vizualizaci a další analýzu. Dalším vylepšením je modul pro výpočet relativních B-faktorů, který škáluje jednotlivé pozice proteinu dle jejich flexibility (a tudíž vhodnosti k mutaci).

Dále byl nástroj FireProt rozšířen o možnost zadat uživatelem definované mutace pro rychlou analýzu mutací, které uživatel považuje za přínosné. Další novou možností na straně uživatele je definice šířky prohledávaného stavového prostoru. Uživatel si nyní může zvolit buď méně riskantní strategii, kdy daná mutace musí existovat v jiné sekvenci v rámci zarovnání aspoň s uživatelem definovaným procentním počtem výskytů a zároveň se nesmí změnit náboj původní aminokyseliny vůči nově zmutované. Druhou z možností je více

riskantní strategie, kde nejsou uvalena omezení ani na nutnost existence dané mutace v zarovnaní a můžou vzniknout i mutace měnící náboj. Uživatel zároveň může zvolit oba z přístupů. Největším rozšířením funkcionality je nový přístup k návrhu multi mutantů. Výstupem předchozích výpočtů jednotlivých modulů jsou ohodnocení mutací a párů mutací. Nově je z těchto údajů sestaven neorientovaný graf, kde uzly reprezentují jednotlivé mutace a hrany vztah mezi nimi. Za pomoci algoritmu pro nalezení všech maximálních klik jsou navrženy vícebodové mutanty, které neobsahují navzájem antagonistické mutace. Ty jsou následně ohodnoceny součtem přínosů jednotlivých mutací a je vybrán nejslibnější multi mutant. Uživateli je následně představen jako tabulka s jednotlivými mutacemi a jejich přínosem a zároveň je namodelována jeho struktura pomocí modulu pro modelování. Tyto kroky jsou opakovány pro několik mutačních strategií: pro evoluční přístup návrhu mutací, pro energetický přístup návrhu mutací (riskantní a/nebo méně riskantní) a jejich kombinaci. Tímto způsobem vznikne až pět různých vícebodových mutantů, se kterými může uživatel dále pracovat. Zároveň byly existují moduly upraveny tak, aby co nejvíce využívaly možností výpočetního centra a proběhla standardizace kódu s možností budoucího rozšíření. Na závěr byl také navrhnut API server, který umožňuje volání nástroje FireProt ASR. Nově implementovaný modul v nástroji FireProt využívá tento server k zadání výpočetních úloh do nástroje FireProt ASR a k sesbírání následných výsledků. Díky tomu mohou uživatelé nástroje FireProt z jednoho místa pohodlněji přistupovat k výsledkům obou nástrojů.

Computational Workflow for the Prediction and Design of Stable Proteins

Declaration

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

.....
Josef Kadleček
May 16, 2022

Acknowledgements

Hereby I would love to express my gratitude to the people that made the completion of this thesis possible. In the first place, many thanks belong to my supervisor Ing. Miloš Musil, Ph.D. as he was a patient and helpful mentor. Unforgettable are also members of my family: Josef a Janička (look ma, you are in a thesis) along with my blood brothers Chang and Martin. Merisi together with Kačenka provided valuable insights into making this thesis more readable. Kudos to all of them. Honorable mention belongs to my namesake Josef Švejk for his moral and spiritual guidance. And also to Zdeněk Juříček for bringing brightness into the dimly lit chambers of the faculty of informatics.

Contents

Chapter 1

Introduction

Proteins are macromolecules that play a crucial part in many aspects of our lives. They are the main building blocks of our bodies where they enable many different functions – from blood oxygenation to muscle contraction. Further protein research is an integral part of understanding the protein building mechanisms, which is essential for comprehending the nature of organisms. This knowledge can be further utilized, for example, in drug design development.

Proteins consist of one or more chains of amino acid residues. Based on the sequence of amino acid residues, the protein forms its spatial conformation which later affects its functionality and interactions with other macromolecules. Unfortunately, a lot of proteins are known only by their amino acid sequence and their structure is not yet known. Thus, a significant scientific effort was put into finding how to predict the structure of the folded protein.

The research on protein stability can give us clues, to whether a given protein in given conditions will be stable, i.e. in its folded state. Furthermore, it provides us with an understanding how to change the protein sequence to maintain its function, while increasing protein's stability so it could withstand harsh conditions of the industrial and medical environments.

Firstly, this thesis discusses protein folding mechanisms, building blocks of proteins, their interactions, their connections to the DNA, and how can we improve the attributes of proteins by exchanging their basic building blocks. At the end of the chapter 2 the basic file formats for the exchange of information between bioinformatics applications are described. The next chapter 3 dives deeper into the problematics of protein stability, explaining how to achieve it, why is it needed, and how is it measured. In the chapters 4 and 5, the protein analysis is broken down into the analysis of protein sequence and protein structure, respectively. In the chapter 6, methods of protein structure prediction are discussed. In the chapter 7, the FireProt tool realizing a combined approach to mutation prediction will be presented and its usage and benefit will be discussed together with the basis of its architecture and processing workflow. Lastly, in chapter 8, the benefits of this thesis will be presented as a set of improvements to the FireProt tool, which includes newly implemented usage of B-factors, usage of homology modeling for predicting the protein structure for proteins entered by sequence, providing extended choice for the users (from defining custom mutations to risk definition), and novel approach to complex multi mutations prediction based on graph algorithm.

Chapter 2

Proteins

Proteins are building blocks of every living organism, where they catalyze reactions, transport ions and molecules, act as nutrients, perform the contractions of muscles and last but not least regulate cellular and physiological activities [59]. To further understand how they are capable of doing those tasks and even make changes in them, so they can perform those functions in a different environment, one first needs to understand the underlying principles, upon which proteins are built. This chapter will discuss what is the life circle of protein from being encoded in a DNA to being fully functional in the folded state.

2.1 Amino acids

Amino acids are the building blocks of proteins. They create a polypeptide chain that forms the sequence, also called the primary structure, of any protein.

There are twenty standard amino acids whose chemical properties greatly affect protein's physicochemical properties and structure and therefore protein function. Other two non-standard amino acids occur quite frequently and occur in literature: selenocysteine and pyrrolysine. Further understanding of amino acids is essential for the understanding of protein function and the mechanisms of protein folding. An overview of standard amino acids, including basic categorization, depicted in the Table 2.1.

Amino acids are built from four chemical elements: carbon, hydrogen, nitrogen, and oxygen. Furthermore, two of them contain the fifth element: sulfur. All the amino acids have the same base, illustrated in the Figure 2.1. The main difference lies in the structure of the side chain. In the protein amino group, one amino acid is connected to the carboxyl group of another amino acid via a peptide bond, thus forming a polypeptide backbone.

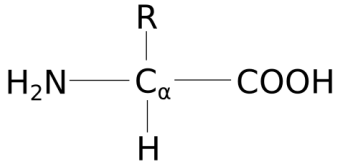

Figure 2.1: Amino acid's base with an amino group on the left, carboxyl group on the right, and side chain on the top. The side chain is the only differing part between various amino acids.

Name			HYDROPHOBIC/ HYDROPHILIC	CHARGE	POLARITY
Alanine	Ala	A	Hydrophobic		Nonpolar
Arginine	Arg	R	Hydrophilic	Positive	
Asparagine	Asn	N	Hydrophilic		Polar
Aspartate	Asp	D	Hydrophilic	Negative	
Cysteine	Cys	C	Hydrophobic		Nonpolar
Glutamine	Gln	Q	Hydrophilic		Polar
Glutamate	Glu	E	Hydrophilic	Negative	
Glycine	Gly	G	Hydrophobic		Nonpolar
Histidine	His	H	Hydrophilic	Positive	
Isoleucine	Ile	I	Hydrophobic		Nonpolar
Leucine	Leu	L	Hydrophobic		Nonpolar
Lysine	Lys	K	Hydrophilic	Positive	
Methionine	Met	M	Hydrophobic		Nonpolar
Phenylalanine	Phe	F	Hydrophilic		Nonpolar
Proline	Pro	P	Hydrophilic		Nonpolar
Serine	Ser	S	Hydrophilic		Polar
Threonine	Thr	T	Hydrophilic		Polar
Tryptophan	Trp	W	Hydrophobic		Nonpolar
Tyrosine	Tyr	Y	Hydrophobic		Polar
Valine	Val	V	Hydrophobic		Nonpolar

Table 2.1: Table representing 20 standard amino acids, describing their full name, three and single-letter abbreviations, the relation to water, if they are charged (and if, which charge), and their polarity in order.

	DNA	RNA	PROTEIN
DNA	DNA replication	Transcription	Direct translation*
RNA	Reverse transciption*	RNA replication*	Translation
PROTEIN	x	x	x

Table 2.2: Information flow between nucleic acid and protein where the left column indicates the source and the top row indicates the destination of the information. Cells marked by the symbol '*' are rare and can be induced either in laboratory conditions or in viruses. Flow in cells marked by 'x' is impossible or not yet known.

2.2 Central dogma of molecular biology

In 1957, Francis Crick first stated that once „information“ has passed into protein, it cannot get out again. To be precise: the transfer of information from nucleic acid to nucleic acid, or from nucleic acid to protein may be possible, but transfer from protein to protein, or from protein to nucleic acid is impossible [30, 29]. A graphical illustration of this information flow can be seen in Table 2.2. For this thesis, two processes are of special interest: the flow of information from DNA to RNA, which is called **transcription**, and the flow from RNA to protein, called **translation**. As such, those processes will be described in detail in the next sections.

2.3 Protein folding

Determining the characteristic of a protein is one of the most frequent problems in biology. Having the structure of the protein facilitates this uneasy task as it provides more insights into the molecular basis of protein function [150, 134].

In „How to fold graciously“ [95], Cyrus Levinthal states that when considering all the conformational possibilities of atoms in the proteins and we know these angles to the precision of a tenth of a radian, there would still be about 10^{300} possible configurations in the model protein. When assessing the calculation presented in „Levinthal's paradox“ [162]. To further elaborate on the conformation of amino acids in a given protein, let's assume that each bond connecting amino acids can have several (for example three) possible states. Let our model protein consist of 101 amino acids (which is fairly low, considering one of the smaller proteins with code 4OEE consisting of 132 residues), this model protein could exist in $3^{100} = 5 * 10^{47}$ possible configurations. If we used „brute force“ to examine every possible configuration at the rate of 3^{13} per second, it would take 10^{27} years to iterate over all the possible configurations. Considering that the proteins in the human body usually fold in rates of seconds or less, we can state with a fairly great confidence, that protein folding cannot be random. This thought experiment is also known as Levinthal's paradox.

Another crucial work regarding the path from sequence to folded state is Anfinsen's thermodynamic hypothesis. So called Anfinsen's dogma was firstly presented in Principles that Govern the Folding of Protein Chains [9]. This postulate states that in their standard physiological environment protein's native structure is determined only by the protein's amino acid sequence. From this, it can be deduced that not only protein folding is not random, even more, it is deterministic. This means that one input sequence will (in the same physiological environment) produce the same 3-dimensional structure.

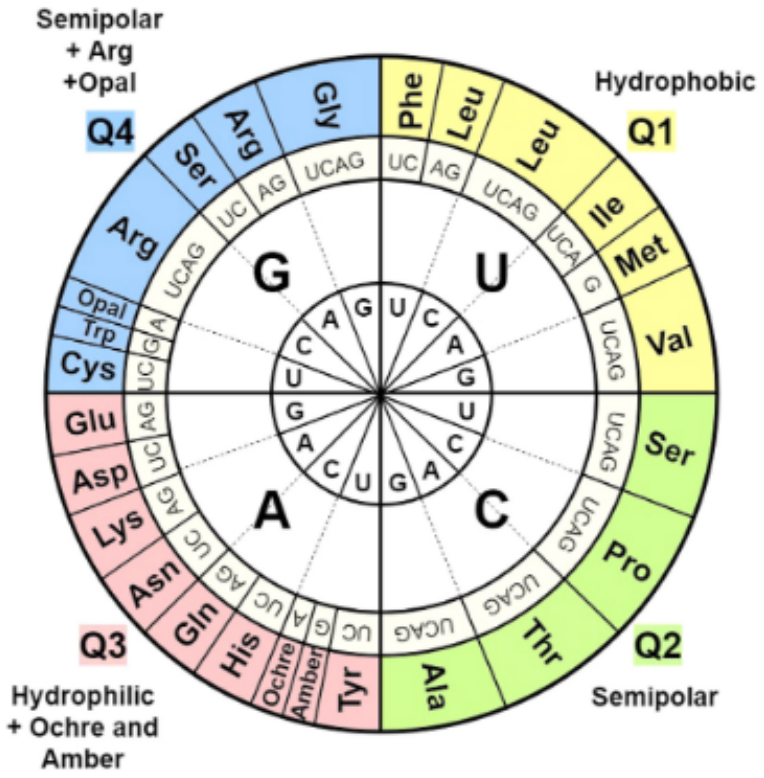

Figure 2.2: Wheel representation of codon usage emphasizes the primary importance of the central codon position (position 2) while determining the type of amino acid. Three letters abbreviation of amino acids are used, for explanation see Table 2.1. The three chain STOP codons are indicated by name (UAA, ochre; UAG, amber; and UGA, opal). Taken over from „Understanding the genetic code“ [132].

2.3.1 Genetic code

Under the term genetic code, one should picture a set of rules used by the living cells to interpret information encoded inside DNA or mRNA to form an amino acid sequence (polypeptide chain). The genetic code is almost identical for all organisms. It maps a sequence of three nucleic acids (codons) into single amino acid. By basic calculation, there are 64 possible codons - four nucleic bases on three-position (4^3), but only 20 standard amino acids and stop codon. Therefore, one amino acid can be coded by multiple sequences. This redundancy is often called „degeneracy of the genetic code“ and was first foreshadowed back in 1961 [28, 160]. The genetic code is also unambiguous - one amino acid can be coded by multiple sequences, but a single sequence (codon) always codes only one specific amino acid as depicted in Figure 2.2.

2.3.2 Transcription

Transcription is the process of copying a specific coding segment of DNA into RNA. Specifically in proteins, we are talking about messenger RNA commonly abbreviated as mRNA. The crucial part of this process is represented by RNA polymerase, which moves over the

DNA strands (with a speed of around 100 nucleotides per second). The transcription of nucleic acid utilizes complementary nucleic bases, where guanine pairs with cytosine and adenine with uracil (uracil in RNA replaces thymine which occurs in DNA).

2.3.3 Translation

Translation is a process in which ribosomes (occurring in the cytoplasm or endoplasmic reticulum) synthesize polypeptide chain (chain of amino acids further forming protein) from mRNA that is created during transcription. The process of transcription and translation is jointly called „gene expression“. Mentioned ribosomes sequentially „read“ codons from mRNA strand - three consecutive nucleotides following the principles of **genetic code** described in section 2.3.1 and utilizing transfer RNA (tRNA) to bring in amino acids that are further forming the resulting polypeptide chain and thus starting to develop resulting protein.

2.4 Protein structure

With the knowledge of how the protein's sequence of amino acids is created, there still lies a question of how does the sequence get its „spacial structure“ - before the proteins are folded. This task can be divided into steps, where protein gradually receives more of its final shape. Those steps are represented by protein structure at a given moment: primary, secondary, tertiary, and quaternary structure [114, 156]. Illustrations of those structures depicted in the Figure 2.3.

Primary structure is the linear order of amino acid residues along the chain. It arises from the covalent linkage of individual amino acids via peptide bonds. Every protein is defined by the sequence of those residues. All subsequent structures (secondary, tertiary, quaternary) are based on this primary level of the structure.

Secondary structure is basically a local conformation of the polypeptide chain (those amino acid residues that are close together in the primary sequence). In globular proteins, the two basic units of secondary structure are α-helix, β-strand, which can be observed in Figure 2.3. α-helix is a conformation, where the strand formed into the right-handed helical structure of approximately 3.6 residues per turn. This conformation is stabilized by hydrogen bonds between overlapping peptide bonds. β-sheets consist of β-strands connected laterally by at least two or three backbone hydrogen bonds, forming a generally twisted, pleated sheet. The β-strands are usually 3 to 10 amino acids long. Similarly as with primary structure, all further (tertiary and quaternary) structures are based on the secondary structure.

Tertiary structure represents the folded polypeptide chain. It can be defined as the spatial arrangement of amino acid residues. It is dependent mainly on the amino acid sequence (primary structure) and chemical properties of given amino acids. For example, hydrophobic amino acids tend to be „hidden“ inside the resulting protein, whereas hydrophilic amino acids tend to move to the protein's surface so their interaction with water molecule is possible.

Quaternary structure is considered when the protein is composed of multiple polypeptide chains (often called subunits). It describes how tertiary structures of those chains are formed together.

The resulting 3-dimensional structure has a great impact on the resulting function of the protein. There are binding sites on the surface of the proteins or inside the protein's

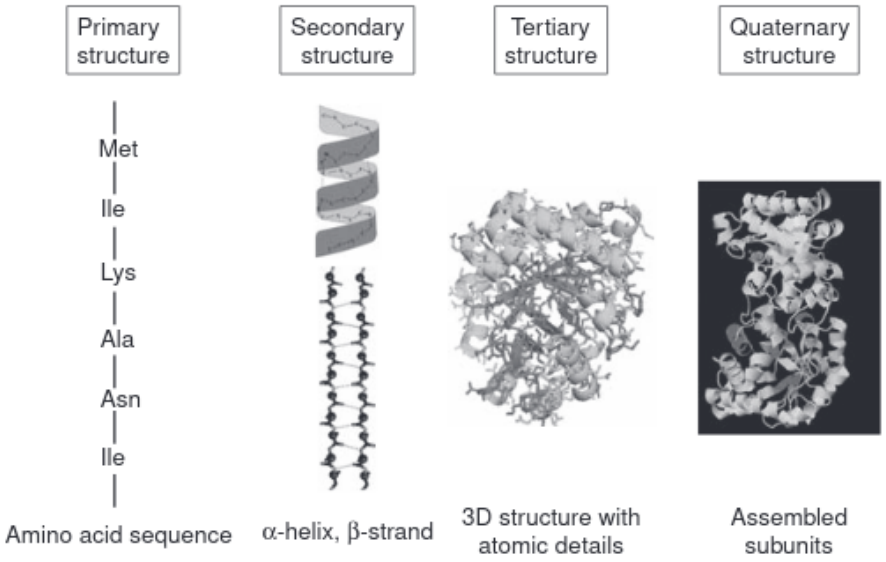

Figure 2.3: Structural organization of proteins. Taken over from „Protein Bioinformatics: From Sequence to Function“ [59].

tunnels. Binding sites are necessary for binding other molecules to the protein – so-called ligands. Those binding sites match with ligand in a „key and lock“ fashion so even a small changes in the protein structure around the binding site can make binding impossible. Thus, mutations around those sites are considered as high risk mutations.

As Adrian A. Nickson and Jane Clarke mentioned [116] we now know four 'classical' folding mechanisms. **Framework model** [83], which implies that local elements of the secondary structure form first, then they diffuse, collide and finally adhere to produce the correct tertiary structure. The **hydrophobic collapse model** [39] implies that a protein collapses rapidly around its hydrophobic side-chains and then rearranges itself. The **nucleation propagation model** [154] states that local interactions form secondary structure, which acts as a base for further folding. Lastly, the **nucleation condensation model** [74] suggests the presence of a metastable nucleus that is unable to trigger folding until a sufficient number of stabilizing long-range interactions is built up. All those models depicted the Figure 2.4.

2.5 File formats

To exchange information between various bioinformatics applications, a few standardized formats for proteins were created. Two of them will be described in this section. One is used for the description of protein sequence: **fasta**, while the second describes protein's structure **pdb**.

Fasta [98] file format is a text-based file format for representing either nucleotide sequences or peptide sequences. The file begins with a description on a single line that is distinguished by the greater-than sign („>“) as the first character. The letters of amino acids of the protein follow (usually with a newline after every eighty characters). Attached example shows the fasta file for protein with pdb code 4OEE:

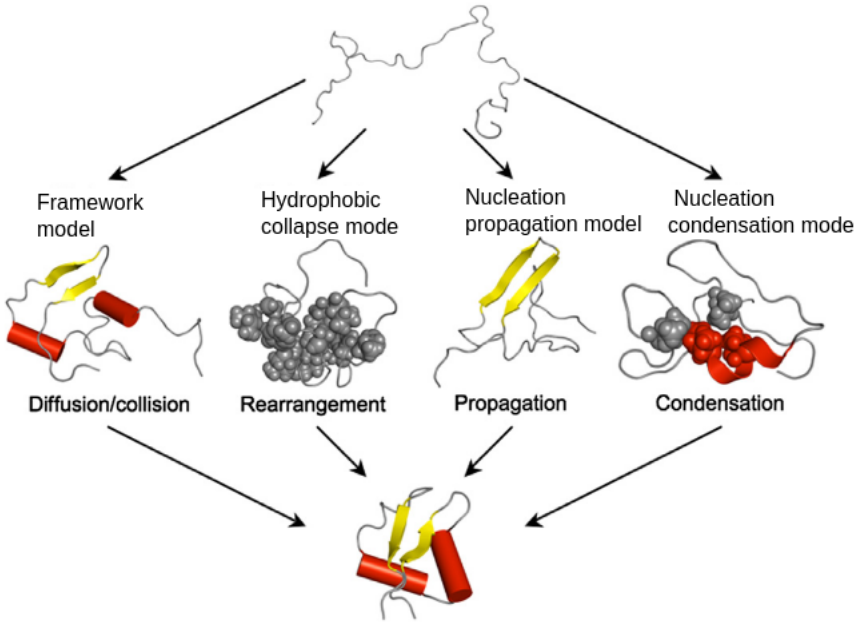

Figure 2.4: Figure representing folding models. Adapted from „Protein Bioinformatics: From Sequence to Function“ [59].

```
>4OEE_1|Chain A|Fibroblast growth factor 2|Homo sapiens (9606)
MAAGSITTLPALPEDGGSGAFPPGHFKDPKRLYCKNGGFFLRIHPDGRVDGVREKSDPHIKLQLQAEERGVVSIKGVSAN
RYLAMKEDGRLLASKSVTDECFFFERLESNNYNTYRSRKYTSWYVALKRTGQYKLGSKTGPGQKAILFLPMSAKS
```

The fasta file format can contain letters from standard amino acids as well as a few special symbols: „-“ for a gap of indeterminate length, and „*“ indicating a translation stop.

Protein Data Bank (**PDB**[1]) file format is a text file format that contains the information about the proteins structure. Its fixed-column width format is limited to eighty columns. Some notable records are

- HEADER uniquely identifies a PDB entry and contains the date when the coordinates were deposited to the PDB archive.
- ATOM records present the atomic coordinates for standard amino acids and nucleotides. They also present the occupancy and temperature factor for each atom.
- HETATM records describe non-polymer or other “non-standard” chemical coordinates, such as water molecules. Atoms presented in HET groups use the HETATM record type.

Example of several lines from the 4OEE pdb file:

```
HEADER PROTEIN BINDING 13-JAN-14 4OEE
TITLE CRYSTAL STRUCTURE ANALYSIS OF FGF2-DISACCHARIDE (S3I2) COMPLEX
...
```

[1]PDB file format description: http://www.wwpdb.org/documentation/file-format-content/format33/v3.3.html

```
REMARK 200 EXPERIMENTAL DETAILS
REMARK 200 EXPERIMENT TYPE : X-RAY DIFFRACTION
REMARK 200 DATE OF DATA COLLECTION : 01-MAR-13
REMARK 200 TEMPERATURE (KELVIN) : 100
....
ATOM 1 N PHE A 12 -8.926 -25.813 18.243 1.00 19.01 N
ATOM 2 CA PHE A 12 -9.267 -24.633 19.032 1.00 13.56 C
ATOM 3 C PHE A 12 -9.487 -24.991 20.502 1.00 17.04 C
ATOM 4 O PHE A 12 -10.186 -25.959 20.816 1.00 17.54 O
...
HETATM 1070 S IDY B 1 -11.480 -10.256 -1.897 1.00 8.59 S
HETATM 1071 C1 IDY B 1 -8.915 -8.089 -2.031 1.00 12.04 C
HETATM 1072 O1 IDY B 1 -7.938 -7.604 -1.116 1.00 13.15 O
...
END
```

2.6 Mutations

A mutation is a change in the primary structure (sequence) of the protein. They are the main factor enabling evolution (different genes have larger „fitness“ and a thus greater chance of evolutionary success). The said mutation is the main tool to increase protein stability - to evolve a more stable protein. This section will focus mainly on gene mutations, where nucleotides in DNA strands are affected.

2.6.1 Types of mutations

Multiple things can occur upon DNA strand that might affect the resulting protein structure:

Insertion of one or more nucleotides into the nucleic acid strand. The greatest impact factor of this change is the number of inserted nucleotides. If the number of inserted nucleotides is not dividable by three (number of nucleotides in codon) reading frame will be altered, thus all following codons will be changed. If the number of inserted nucleotides is the multiplication of three, it will result in additional amino acid(s) in the sequence.

Deletion is quite similar to insertion but instead of adding, we remove certain nucleotides. In the same fashion as in insertion, if the removed count is not a multiplication of three, the reading frame will be shifted.

Substitution is a simple change of nucleotide (or nucleotides) for another one. Sequence length is not altered and the reading frame remains the same. This is the main factor, why substitution tends to be less dangerous than insertion or deletion.

After nucleotide mutation occurs, it doesn't necessarily mean that it will change the protein's structure or even function of the organism. The mutated nucleotide might be located in the noncoding region. The mutation can even be **synonimical**, meaning that due to the degeneracy of the genetic code, the change in nucleic acid sequence doesn't necessarily caus a change in the resulting amino acid. For example, in the sequence GUU resulting in amino acid Valine, we can mutate (change) the second U to any other base (C, A, G) and it would not change the resulting amino acid because all of the sequences GUU, GUC, GUA, GUG are translated as Valine. Other types of mutations, which tend to have an impact on the resulting sequence, are:

Non-synonimical mutations are the opposite of synonymical. It means that the change in the nucleic base will result in the change of the amino acid.

Mutations **encoding stop codon** will prematurely end the translation of the polypeptide chain.

Mutations **changing reading frame** tend to greatly change the resulting polypeptide chain and also prematurely end the translation.

2.6.2 Causes of mutations

Two main causes how mutations occur can be observed:

Spontaneous mutations happen during DNA replication. Even though DNA replication tends to be very precise and it is expected to make less than a single mistake per 10^5 bases, which is followed by a self-correction mechanism that brings the chance to less than $1 : 10^7$, mutations can still happen.

Induced mutations are often induced through external factors. **Physical** mutations are results of the ionizing or ultraviolet radiation and the damage to the DNA strand is dependent on the amount of the radiation. **Chemical** mutation is a result of so-called *genotoxins*. And lastly, **biological** mutations are results of oncogenic viruses.

Chapter 3

Protein stability

Protein stability is a net balance of forces, which determines whether given protein will be in its native folded conformation or a denatured state [59]. Factors affecting protein stability are mainly atomic/group interactions, such as hydrophobic, electrostatic, hydrogen bonding, van der Waals and disulphide bonds. The approximations of their relative contributions depicted in Table 3.1. Contrary to the stabilizing factors, there are many aspects that can cause the protein to desaturate: temperature, pH value, salt type, and concentration, co-solvents, as well as mechanical forces. To increase protein stability means to increase its resistance to said factors. In this thesis, thermodynamic stability (function of the change in free energy) is the main concern as it's a good predictor of whether a protein will be folded [123] and together with melting temperature represents the two most commonly used metrics for the estimation of protein stability [59] .

3.1 Stable mutants

There has been a growing need for the usage of biocatalyst (components that speed up chemical reactions) for the production of fine chemicals and pharmaceuticals in industrial-scale, or for the creation of enzymes that remain stable even at human body temperature [52, 2]. The need for protein stabilization is nothing new. Its origin can be traced as far as to the late eighties of the past century [118]. Unfortunately, most random mutations would be deleterious [158] and a single mutation usually has a small effect on proteins stability. Greater stability arises from small but additive effects distributed over the entire molecule [152].

Free energy	Contribution (%)
Hydrophobic	50.8
Hydrogen bonding	27.1
van der Waals	27.1
Electrostatic	6.4
Disulfide	1.1

Table 3.1: Relative contributions of free energies to protein stability. Adapted from the „Protein Bioinformatics: From Sequence to Function“ [59]

As proteins evolve together with their host to be better suited for the surrounding environment (increase their fitness for the given conditions), proteins of hyperthermophilic and thermophilic organisms (organisms that thrive in hot or extremely hot environments surviving temperatures as high as a hundred degrees Celsius) became a promising resource for learning about protein stability in increased temperatures [158].

Pioneers of stability research observed that the following amino acids are often substituted in thermophilic proteins: Gly for Ala, Ser for Thr, Lys for Arg, Asp for Glu, and Trp for Tyr [10, 60] . Furthermore, in thermophilic proteins, charged residues are present in large numbers, contrary to polar residues which are scarce [51]. When studying hyperthermophilic proteins, it has been found that optimum placement of charged amino acid residues within the protein structure enhances electrostatic interactions [59].

With the utilization of more structural data, it has been found that the main difference between thermophilic and mesophilic proteins in the amino acid composition is mainly on the protein surface, whereas the interior composition is more similar [51]. Another example of utilizing structural information about a thermophilic organism is Baldasseroni et. al. [15], where authors describe how enhanced compactness at subunit interfaces increases the stability of hyperthermophilic enzymes.

This thesis focuses solely on the enhancement of protein stability via protein engineering. Other possible methods like immobilization, Lyophilization, or chemical modifications [123] are out of scope of this thesis.

3.2 Measuring protein stability

As it was mentioned before, there are many ways how to measure protein stability. In this thesis, we will mainly use the change in Gibbs free energy (ΔG) and melting temperature (Tm), which will be discussed in the following sections. Force fields estimating the forces between atoms and molecules inside protein and specific bioinformatics tools and their principles will be also outlined.

The process of measuring protein stability is mainly performed by experiments such as circular dichroism (CD), differential scanning calorimetry (DSC) and absorbance (Abs) [59] as can be seen in Figure 3.1.

Another thermodynamic quantities include enthalpy change (ΔH),entropy change (ΔS), and heat capacity change (ΔC_p) at unfolding. By determining those values, we obtain deeper insight into the forces that affect proteins stability [59].

3.2.1 Gibbs free energy

The Gibbs free energy (Gibbs energy or Gibbs function or free enthalpy (G)) is a thermodynamic potential that measures the maximum of reversible work by a thermodynamic system at a constant temperature and pressure [58].

Gibbs free energy is defined as follows:

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

where T is the temperature (SI unit: kelvin), S is the entropy (SI unit: joule per kelvin) and H is the enthalpy (SI unit: joule). The official SI unit for Gibbs free energy is Joule, however, usage of Calories is more common.

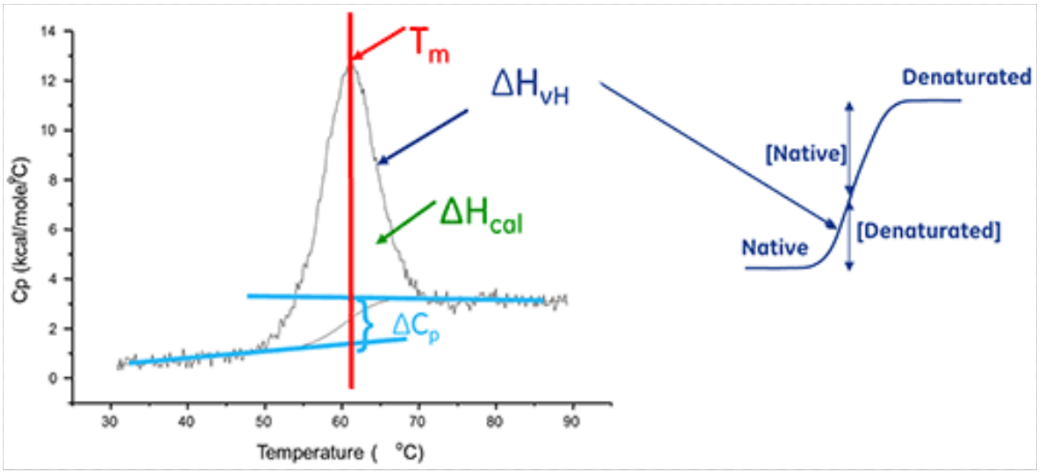

Figure 3.1: Results obtained from differential scanning calorimetry ΔC_p is the difference between heat capacity of native and denatured states. Tm is the melting temperature, ΔH_{cal} is the calorimetric enthalpy and ΔH_{vH} is van't Hoff enthalpy. Source of figure: https://www.malvernpanalytical.com/en/products/technology/microcalorimetry/differential-scanning-calorimetry/

The protein stability is generally represented by the change in the Gibbs free energy upon folding (ΔG).

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

In words: the difference between the free energy of the folded and unfolded state of the protein.

Finally, we need to measure whether mutated protein is more stable than the original (wild type) protein. For this, the following formula will be used:

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

This difference says a lot about change in protein stability. In this case, lesser (more negative) values indicate more stabilizing mutation and positive values indicate destabilizing mutations. Although the reader should focus on the definition of $\Delta\Delta G$, as its format is not standardized and the order of mutant and wild_type can be switched, thus interpretation of results will be reversed.

3.2.2 Melting temperature

Another way to quantify protein stability is by the usage of the melting temperature (Tm). The definition of melting temperature is following:

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

in other words, the temperature at which the free energy of the unfolded and folded protein is equal and half of the protein is unfolded and the other half is folded. Naturally, an increase in said temperature (Tm) means that the protein is more stable. There is a strong correlation between melting temperature (Tm) and Gibbs free energy (ΔG), although it is not linear [1]. Nevertheless, the usage of Gibbs free energy seems to be more common [127].

3.2.3 Force fields

To fully automate computational design for protein stability, we need a computational method, which replaces tiresome manual experimental evaluation. Force fields represent a computational method that is utilized to estimate the forces between atoms within and between molecules. The Force field method is used for calculation of free energy inside the protein and employing the physical and chemical properties of the proteins.

Force fields based methods are usefull for calculating G_{folded} (and analogicaly $G_{unfolded}$) via the formula presented in [124] which goes as follows

$$G_F = G_{hy} + G_{el} + G_{hb} + G_{vw} + G_{ss} \tag{3.5}$$

Where:

G_{hy} = Hydrophobic free energy [45]
G_{el} = Electrostatic free energy
G_{hb} = Hydrogen bonding free energy
G_{vw} = Van der Waals free energy
G_{ss} = Disulphide bonding free energy

Chapter 4

Sequence based identification of stabilizing mutations

The simplest and for more than two decades the only known [161] form of identifying potentially stabilizing mutations is based on the analysis of its amino acid sequence. The main source of information for this operation is obtained through protein evolution. For this purpose, we firstly need to identify homologous sequences and create so-called multiple sequence alignment (MSA). Once the MSA is created, we can utilize various evolution-based methods to find out which mutations might be beneficial.

4.1 Sequence alignment

Sequence alignment is an essential tool for protein structure and function prediction, phylogeny inference and other common tasks in sequence analysis [44]. Because of substitutions (mutations) introduced by evolution, some residues in the protein differ between related proteins. Furthermore, insertions and deletions may occur, and therefore, just calculating letters in the same place is not possible as just single insertion / deletion shifts the reading frame.

Following restrictions are imposed when constructing an alignment [59]:

- All residues should be used in the alignment and all should be in the same order.
- Align one residue from the first sequence with another from the second one.
- A residue can be aligned with a blank symbol (representing evolutionary gap).
- Two blanks (gaps) cannot be aligned.

The alignment of sequences can be either done as **local** when one sequence is notably longer than the other. Moreover, only the area of the match is needed. The second case is **global** alignment where the sequences are of similar length and we need to penalize all of the gaps and **semi-global** which doesn't penalize the gaps at the beginning and the end of the alignment.

Firstly, alignment of only two sequences will be described to better understand how this problem is solved. After that, multiple sequence alignment (MSA) will be described as it utilizes alignment of two sequences.

	C	S	T	P	A	G	N	D	E	Q	H	R	K	M	I	L	V	F	Y	W	
	0	-1	1	0	2	1	1	2	1	2	0	0	2	4	1	5	1	2	-2	5	C
		2	0	-2	0	-1	0	0	0	1	0	0	0	1	0	1	-1	1	1	-1	S
C	9		2	-1	-1	-1	0	0	0	0	0	0	-1	0	-1	1	0	1	1	3	T
S	-1	4		2	-2	-1	-1	0	0	-1	-1	-1	1	1	0	-1	0	0	2	1	P
T	-1	1	5		2	-1	-2	-2	-1	0	0	1	1	0	0	1	0	1	1	2	A
P	-3	-1	-1	7		2	0	-1	-2	0	1	1	0	0	-1	0	-1	1	2	4	G
A	0	1	0	-1	4		3	-1	-1	0	0	1	-1	0	-1	0	-1	0	0	0	N
G	-3	0	-2	-2	0	6		2	-1	-1	-1	0	-1	0	0	0	0	2	1	3	D
N	-3	1	0	-2	-2	0	6		1	0	0	2	2	1	-1	0	0	2	2	4	E
D	-3	0	-1	-1	-2	-1	1	6		0	-2	0	1	1	-1	0	0	1	3	3	Q
E	-4	0	-1	-1	-1	-2	0	2	5		2	-1	0	1	0	-1	0	1	2	2	H
Q	-3	0	-1	-1	-1	-2	0	0	2	5		-1	-1	0	-1	1	0	1	3	-4	R
H	-3	-1	-2	-2	-2	-2	1	-1	0	0	8		1	-2	-1	1	1	2	3	1	K
R	-3	-1	-1	-2	-1	-2	0	-2	0	1	0	5		-2	-1	-1	0	1	2	4	M
K	-3	0	-1	-1	-1	-2	0	-1	1	1	-1	2	5		-1	1	0	0	1	3	I
M	-1	-1	-1	-2	-1	-3	-2	-3	-2	0	-2	-1	-1	5		-1	0	-1	1	2	L
I	-1	-2	-1	-3	-1	-4	-3	-3	-3	-3	-3	-3	-3	1	4		0	1	2	4	V
L	-1	-2	-1	-3	-1	-4	-3	-4	-3	-2	-3	-2	-2	2	2	4		-1	-2	1	F
V	-1	-2	0	-2	0	-3	-3	-3	-2	-2	-3	-3	-2	1	3	1	4		-1	2	Y
F	-2	-2	-2	-4	-2	-3	-3	-3	-3	-3	-1	-3	-3	0	0	0	-1	6		-1	W
Y	-2	-2	-2	-3	-2	-3	-2	-3	-2	-1	2	-2	-2	-1	-1	-1	-1	3	7		
W	-2	-3	-2	-4	-3	-2	-4	-4	-3	-2	-2	-3	-3	-1	-3	-2	-3	1	2	11	
	C	S	T	P	A	G	N	D	E	Q	H	R	K	M	I	L	V	F	Y	W	

Figure 4.1: BLOSUM 62 substitution matrix (Lower) and difference matrix (Upper) obtained by subtracting the PAM 160 matrix position by position. Adapted from Henikoff et. al. [68]

4.1.1 Alignment of two sequences

Substitution matrices

There is a need to reflect that some amino acids are more likely to be substituted with one another than with others - for instance, there is a lower probability that the positively charged amino acid mutates into one with negative charge than to the amino acid that is either positive or neutral. Also with the knowledge that amino acids are determined by codons (triplets) it is more likely that just single nucleotide will be mutated.

To reflect this situation, substitution matrices were designed. After the candidate alignments of two sequences is created, every position is evaluated by the scoring matrix and summed up. Gaps are subtracted from the final score and the alignment with the highest score is selected. Most common substitution matrices are PAM and BLOSSUM [110, 68]. PAM matrices are used to score alignments between closely related sequences and are based on global alignments. Furthermore, higher numbers in matrix naming indicates greater evolutionary distance. On the other hand, BLOSUM matrices are used to score alignment between evolutionary more divergent sequences and are calculated based on the local alignments. In comparison to PAM, higher numbers in matrix naming indicates greater sequence similarity. Therefore, PAM100 and BLOSUM90 can be used for closely related sequences, while PAM250 and BLOSUM45 for divergent ones. Examples of both BLOSUM and PAM matrices can be seen in Figure 4.1.

Alignment calculation

For pairwise alignment, algorithms based on **dynamic programming** were designed. Those algorithms try to simplify the problem by dividing it into subtasks, whose optimal

solutions can be utilized to solve the original task. This solution can be also reused when calculating more complicated tasks. Most notable algorithms, utilizing dynamic programming for pairwise alignment, are Needleman and Wunsch [113] and Smith-Waterman [138]. Those algorithms are guaranteed to find the optimal solution with respect to selected substitution matrix and the gap-scoring scheme. To create optimal alignment with dynamic programming, the table has to be created. Each row corresponds with a character from one sequence and each column corresponds with a character from the second. The table represents the possible alignments as every possible comparison can now be represented as a pathway through the array. Next we need to fill in the array. In the first step, we can either align first characters or put a gap into one of the sequences. In the next step we calculate rest of the table so the values are maximum of three possibilities:

1. value on the left with penalization for gap (we take character from the sequence representing the columns and nothing from the sequence representing rows)
2. value from the above with penalization for gap (we take character from the sequence representing the rows and nothing from the sequence representing columns)
3. value from the upper left diagonal with the score for match / mismatch of the characters

After the whole table is filled, the value in right bottom corner represents the optimal solution. By backtracking the path from bottom right corner to upper left we gain the path representing the optimal alignment.

4.1.2 Multiple sequence alignment

Multiple sequence alignment is often utilized to indicate which regions of each sequence are conserved (unchanging during the course of evolution) and to detect a potential co-evolution between amino acids or nucleotides in the given set of sequences.

Here, we can again distinguish between two cases of alignment: global alignment in which the proteins have maintained a correspondence over the entire sequence length, and local alignment where the alignment consists only from most similar parts of proteins.

The **trivial approach** for finding an alignment of n sequences would require n-dimensional matrix formed in standard pairwise alignment and as such, the complexity would be

$$O(Lenght^{Nseqs})$$

which is not feasible considering hundreds of protein sequences containing hundreds of amino acid. In result, novel algorithms utilizing dynamic programing or some form of heuristics were designed. We can roughly split them into two categories: progressive and iterative methods.

Progressive methods

Progressive methods utilize guided tree that was constructed based on the sequence similarity using UPGMA [61]. From the k sequences, two of them are aligned resulting in $k-1$ sequences. This process is repeated until all of the sequences are aligned. The main disadvantage of this method lies in the optimal selection of sequences to align as the guided tree is not necessarily optimal. Another problem is the relatively slow speed of this method. Examples of software utilizing iterative methods is CLUSTAL and T-COFFE.

Iterative methods

Iterative methods aim to be able to change the alignment during the computation. The main idea behind iterative methods is to realign already aligned sequences when adding a new one. The process is as follows: at first, guided tree is created. Tree is then split into two parts for which two independent alignments are created and then aligned together. If a better score than the previous alignment is achieved, the alignment and the tree are replaced and the method is repeated. An example of software utilizing iterative methods is MUSCLE.

Tools for creating MSA

CLUSTALW [145] was introduced in 1994. The main advantage is its low memory consumption when aligning a small number of extraordinarily large sequences. Clustal Omega [137] is a newer version of CLUSTAL, which uses seeded guided trees and the hidden Markov model engine that focuses on two profiles to generate the alignment.

MAFFT [85, 84] or MUSCLE [43, 42], which find faster and more accurate representations, while offering lower accuracy options as a trade off for computational speed [44].

Other tools for construction of the high-quality MSA are T-COFFEE [117] and PROBCONS [38], which has the highest accuracy score on several benchmarks for the price of larger computational power, especially for larger number of sequences. Furthermore, when structural homologs are available, tools such as 3D-Coffee [11] utilize them to provide outstanding results.

Generally speaking, there is not a single best MSA tool and the user should consider benchmarking of the available solutions for his use case or at least consider the advantages of different tools [44, 37].

4.2 Homolog search

Homology has the precise meaning in the biology as "having a common evolutionary origin", but also carries the loose meaning of "possessing similarity or being matched" [128]. In this thesis the term homologous sequences is used to describe similar sequences which are expected to have a common evolutionary ancestor. When strong similarity is observed between sequences, it is usually a sign that the proteins are related by evolutionary changes. To measure how similar some sequences are, the *percent homology* was historically used. Percent homology indicated the percentage of identical residues between the sequences. However, this metric was replaced by *similarity score*, as it is more rational, and has stronger scientific basis [70]. Furthermore, it doesn't collide with the original meaning of homology as much as percent homology (having a common evolutionary origin is a binary relation and as such, it is hard to describe it in percents [70]).

Another use case, apart of the creation of a set of evolutionarily related sequences, is that homologous sequences can also be useful as a template for modeling protein with previously unknown structure [121]. The most common tools for finding homologous sequences are BLAST and FASTA, both utilizing modifications of original Smith-Waterman alogrithm [138], and profile based HMMER.

BLAST - basic local alignment search tool [8] is widely used to search for homologue sequences.

BLAST starts by finding similar sequences by locating short matches between the two sequences which are performed by seeding: splitting the sequence into words of three characters (word size three is default, but customizable) and utilizing a sliding window with selected size and offset of one. For every word, set of all alternative words is created (by substitutions). All the alternative words are compared with the original word with the usage of a similarity matrix. Next, words under the certain threshold are filtered out. All the remaining words are used to find them in the other sequence or database. If the word is found, the algorithm tries to extend it to both sides to obtain the highest score match. Those sequences are then evaluated. Sequences with low e-value are deemed statistically insignificant and disposed of. Other sequences are then joined together if possible. BLAST suite of programs is still considered as the workhorse for most of the bioinformatic applications [41]. However, other promising tools, utilizing novel approaches including HMMER or commercial tools like PatternHunter [100], are promising good results at a relatively small computational cost.

FASTA [98] works on the similar basis as BLAST. However, it doesn't consider alternative words (only identical matches) and doesn't try to extend those words, only to join the matches together instead. This feature makes FASTA more precise at the cost of computational time.

Tools such as BLAST, FASTA and SSEARCH produce accurate statistical estimates that can be used to reliably infer homology. Searches with protein sequences (BLASTP, FASTP, SSEARCH) or translated DNA sequences (BLASTX, FASTX) are preferred as they are 5- to 10-fold more sensitive than DNA:DNA sequence comparison [120].

HMMER [46] utilizes a method based on the comparison of the profiles of hidden Markov models (HMM) and uses heuristic filters to speed up the evaluation time.

4.3 Evolution based methods

Evolutionary (also called phylogenetic) analysis consists of methods that are utilizing the different information contained in the set of homolog sequences.

In this section, we will focus mainly on the consensus design, briefly mention ancestral reconstruction, and further explain correlated pairs and conservation analysis methods.

4.3.1 Back to consensus

Consensus design (CD), also known as back to consensus, works with multiple sequence alignment (MSA). After it builds given alignment with reasonable amount of homolog sequences, it examines its individual columns. If there is a column where some of the amino acids are more significant than others, there is a chance, that given amino acids are conserved and might have a stabilizing effect [81, 125]. The visual explanation can be seen in the Figure 4.2. Consensus design can be used in those positions, where the amino acid in a given column of the target protein differs from the prevailing amino acids in MSA (regarding the same column). The mutation to this prevailing amino acid is likely to have a stabilizing effect. The success of consensus design may vary, but about 50 % of conserved residues are associated with improved stability, with 10 % being stability neutral and 40% being destabilizing [125]. Another advantage is that consensus design doesn't require large sets of homologs. It has been presented that as few as four homolog sequences can be utilized [119], although nowadays, finding multiple homolog sequences doesn't seem to be the bottleneck and tend to yield better results. Furthermore, when utilizing library-

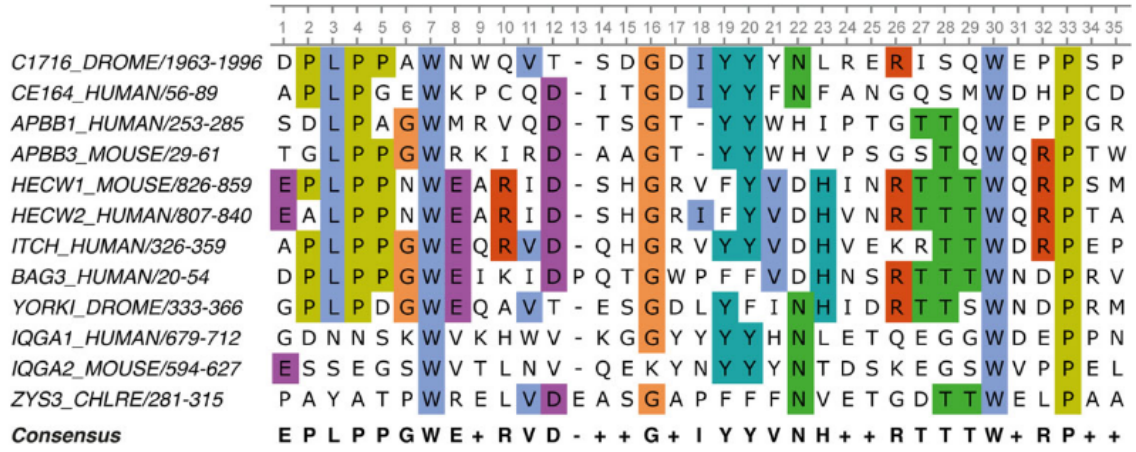

Figure 4.2: Consensus design figure illustrating prevailing amino acid in MSA. In the figure, a dash is a gap, whilst a plus sign is an ambiguous position with no consensus. The most conserved residues are highlighted. Adapted from Porebski and Buckle [125]

based consensus design, there is no need for structurally and functionally related natural homologs [82]. One of the problems that might arise is phylogenetic bias. Works af Christian Jäckel et. al. [82] show that this can be precluded a priori by randomization of just a single parental sequence, followed by functional selection. Due to its low computational demands, this makes consensus design an ideal „first step“ when finding beneficial mutations.

4.3.2 Conserved and correlated positions

As protein usually contains hundreds of amino acids, the knowledge of which positions are crucial for the protein functionality may be useful to filter them out from possible mutations. The main two categories are conserved and correlated positions.

Conserved positions [24] are such positions that preserved mostly the same amino acid during the course of evolution. When the amino acid was not changed throughout evolution, it is assumed that it is crucial for the protein function [27, 72] (as mutating it probably doesn't possess the same evolutionary fitness).

Similarly, **correlated residues** are sets of residues that always change together. Those correlations are usually an indication of probable physical contact in three dimensional structure and are sometimes even used for protein modeling [64]. An example of such interactions are spatially close amino acids with different charge - mutating them to the state with identical charge might break the attraction and thus negatively affect the protein stability [143]. With this knowledge, one can use correlated residues as a filter - mutating only one residue in a way that is not compatible with the others would be unwise.

4.3.3 Ancestral reconstruction

Ancestral sequence reconstruction (ASR) is a method that tries to reconstruct ancient proteins and to retract their evolution in time. ASR starts similarly to CD - it constructs multiple sequence alignment of relevant homolog sequences. The main difference lies in further analysis. While CD only looks on the column of aligned homologs, ASR tries to grasp evolutionary information which lies inside the phylogenetic tree.

Reconstructing a protein evolutionary history is an efficient way to identify structural causes of functional diversity, as it enables a focused identification of causal sequence and structural features [69]. There are several ways how to do that: the phylogenetic relationships among the sequences can be inferred from the sequence using maximum likelihood or Bayesian methods [69]. A number of studies are suggesting that protein's deepest ancestor promises elevated thermostability and even though the descend of stability is not smooth, proteins history still holds valuable information [155]. The main challenges of this method are the selection of the biologically relevant subset of homolog sequences, rooting of the phylogenetic tree, and the reconstruction of the ancestral gaps. There are novel approaches that try to mitigate given concerns and make ASR fully automated [111].

4.4 Software using evolutionary based methods

Only about half of mutations identified by evolution-based approaches are truly stabilizing. In many cases, they improve different aspect of proteins functionality (solubility, activity) [101]. Nevertheless, the evolution-based approaches might be used to filter some potentially interesting mutations for hybrid models.

Consensus design approach is fairly trivial and as such no dedicated software is needed.

For ASR, there are two main approaches - by using maximum likelihood or Bayesian inference. The maximum likelihood is utilized in **FastML** [12], which accounts for uncertainty in ancestral states as it provides not only the posterior probabilities for each character and indel at each sequence position, but also a sample of ancestral sequences from this posterior distribution and a list of the k-most likely ancestral sequences. Another examples are **RAxML** - Randomized Axelerated Maximum Likelihood [140, 139], which utilizes many enhancements for faster construction of phylogeny and **Ancestors** [36], which was the first software to provide the last three steps of the ancestral genome reconstruction procedure in a single workflow (aligning multiple sequences, reconstructing the insertion and deletion history and inferring substitutions).

Bayesian inference is used by **HandAlign** [153], which treats alignments, trees and model parametres as jointly dependent random variables and samples them via Metropolis–Hastings Markov chain Monte Carlo (MCMC). **MrBayes** [131] provides convergence diagnostics and allows multiple analyses to be run in parallel with convergence progress monitored on the fly.

Chapter 5

Structure based identification of stabilizing mutations

This chapter focuses on methods utilizing not only sequential information (order of amino acids) of protein but also spatial conformation (structure) as spatial information is better suited for analyzing physicochemical interactions (free energies described in chapter 3). Additionally, the knowledge of structure is necessary for the energy driven approach trying to find the structure in the conformation with the least free energy.

5.1 Force fields

Computational design of stable proteins based upon the calculation of the energy force tries to take all physicochemical properties of a given protein into consideration and to simulate whether the free energy of folded protein is more advantageous. Naturally, those methods don't require large annotated datasets (like machine learning algorithms do) as they are based upon simulation of real-life physicochemical principles. The most accurate methods in this category are the free energy methods. Unfortunately, free energy calculation cannot be expected to replace other methods that are based on fast optimization algorithms due to its time-demanding constraints [136].

5.1.1 Energy functions

Fortunately, several heuristic approaches were developed to overcome the issue of high time-demands. For a clearer understanding, we can roughly divide energy functions into the following categories: PEEF, SEEF, EEEF [104]. This division is for illustrative purposes as almost all practical methods utilize approaches that would overlap those definitions.

PEEFs - Physical effective energy functions [91] are based on the atomic model of a given structure and thus allow interactions on an atomical level. Those methods try to approximate real-life physical laws. Unfortunately, such approach might still be time-demanding (with current computational power, we are speaking about days of computational time for a single mutation).

SEEFs - Statistical effective energy functions [91] are based on known protein structure. They can be used for fast and accurate [32] detection of stability changes. Statistical energy functions are extracted from databases of known protein structures and do not explicitly model physical molecular interactions. They are also being dependent on folded protein structure [104] .

EEEFs - Empirical effective energy functions [104] are an intermediate stage between PEEFs and SEEFs as they combine physical principles as well as statistical that are further weighted, and parametrized [62]. Example of such energy function can be FOLDEF (for FOLD-X energy function [63]) which can be seen at equation 5.1. The main advantage of this simple energy function is its low demands on computational resources.

$$\begin{aligned}\Delta G = W_{vdw}\Delta G_{cdw} + W_{solvH}\Delta G_{solvH} + W_{solvP}\Delta G_{solvP} + \\ \Delta G_{wb} + \Delta G_{hbond} + \Delta G_{el} + W_{mc}T\Delta S_{mc} + W_{sc}T\Delta S_{sc}\end{aligned} \tag{5.1}$$

Where ΔG_{wdv} stands for the sum of the van der Walls forces of all atoms, ΔG_{solvH} and ΔG_{solvP} are the differences between solvation energies of apolar and polar groups when going from unfolded to the folded state. ΔG_{hbond} is the free energy difference between the intramolecular hydrogen-bond compared with intermolecular hydrogen-bond formation. ΔG_{wb} is the stabilizing free energy of water bridges, ΔG_{el} is the electrostatic contribution of charged groups interactions, ΔS_{mc} is the entropy cost for fixing the backbone in the folded state, and finally the ΔS_{sc} is the entropic cost of fixing a side-chain in a particular conformation.

5.1.2 Software using energy based methods

As mentioned above, there is not a strict line between outlined energy functions as even tools that are considered to be PEEFs utilize statistical approximations as well as SEEFs tools, which utilize some form of physical data. An overview of software accepting structure and a single point mutation as input and calculating the $\Delta\Delta G$ will be presented.

FoldX [135] is an empirical force field that was developed for the evaluation of the effect of mutations on the stability with concern to speed. It uses FOLDEF outlined above. In addition to calculating $\Delta\Delta G$ of the mutation, it allows a calculation of the positions of the protons and the prediction of water bridges, the prediction of metal-binding sites, and the analysis of the free energy of complex formation. It achieves the best results when comparing known structures.

PopMuSiC [33] is using a linear combination of statistical potentials and the force-field equation is constructed using thirteen physical and biochemical terms with approximate values determined from experimental data instead of evaluating every physical term.

Rosetta [92, 31, 5] is a versatile software suite for computational modeling and analysis of protein structures that is freely available for academic use. The suite consists of several modules, including stability predictions, molecular simulations, and ab-initio modeling. A notable part of the suite in regards to mutation effect prediction is a ddg_monomer which is a standalone module for directly predicting $\Delta\Delta G$ for protein stability analysis [5].

5.2 Machine learning

The main advantage of the machine learning method lies in its non-dependence on understanding the physio-chemical principles upon which the proteins work. This is a huge advantage, especially when the underlying principles are not known, if features are hard to extract, or if it would require tiresome laboratory work, which is impossible to replace computationally. Furthermore, by not being dependent on the extracted features, the systems are not limited by them and can for example discover new features, previously

unknown to human understanding. Even though we mentioned the main advantage of the machine learning method as being non-dependent on information extrapolated from data, some methods can work with models that take some form of biological features as an input [144]. The last but not least advantage of machine learning models lies in their speed. Even though the creation of a fitting model can take a huge amount of time, the inference (the process of running data points through a machine learning model) is generally much faster than the usage of complex simulations. The machine learning methods can be loosely divided into two groups:

Unsupervised learning is employed mostly for data clustering and its main advantage lies in its non-dependence on „labels“ (expected outputs). **Supervised learning** on the other hand requires some form of expected output („ground truth“) so it can „learn“ which inputs should produce which outputs and adjust the inner structure of the model accordingly. Machine learning methods utilize large datasets of known proteins and they try to extract statistical facts hidden in the large volumes of data. The sizes of datasets have increased rapidly in the recent years. For example, in years 2016 to 2020 the size of the sequence database Uniprot quadrupled [4] and the size of the structure database PDB increased almost by half in the same time [3]. Even though those datasets are large, they can be rarely fully utilized. Usually, only a small subset of data can be used for a given task, which is then further burdened with unequal representation of individual features. Especially for the problem of protein stability, where one has to find a single protein and then compare it with as close as possible mutants. This dataset still lacks in size due to the lack of experimental data. Only recently, there are efforts to create such datasets [141], even though the dataset focuses solely on single point mutations. Furthermore, some machine learning methods - namely neural networks, have to deal with common problems like overtraining/overfitting and have utilized some approaches to mitigate those problems. In recent years, machine learning methods that utilize structure information were developed and they are superior to other methods that utilize only sequence information. Moreover, with their reliability, they are being competitive with structure-based energy functions [49]. Machine learning methods tend to use a variety of models such as Support Vector Machines [49, 144], decision trees [73], or even more and more popular neural networks [23]. To summarize, in many fields of protein engineering, machine learning techniques represent state-of-the-art methods (for example modeling structure from sequence [80]) and are starting to emerge more in the protein stability domain. Moreover, efforts for creating databases for single point mutations might greatly accelerate further improvements.

5.2.1 Mechine learning based software

Examples of successful machine learning applications include software utilizing the support vector machines like **I-Mutant** [25] or **MuPro** [26] which are capable of calculating the single point mutation $\Delta\Delta G$ for the sequence as well as structure used as an input. **EASE-MM** [50] Evolutionary, Amino acid, and Structural Encodings with Multiple Models utilizes five specialized support vector machine (SVM) models and makes the final prediction from a consensus of two models selected based on the predicted secondary structure and accessible surface area of the mutated residue.

Other tools try to employ a random forest algorithm. Such an example can be **ProMaya** [149] Protein Mutant Stability Analyzer which combines a collaborative filtering baseline model, Random Forests regression and a diverse set of features, or **PROTSRF** [97] which is even capable of calculating multiple point mutations.

Finally a small note: although almost all of the tools for predicting the stabilizing effect of mutations have a strong bias toward destabilizing mutations, machine learning methods tend to be the most affected [126, 147].

5.3 Hybrid approaches

Hybrid approaches try to utilize force-fields calculations as well as evolutionary methods. It has been shown that those methods are complementary [17]. Moreover, in some cases, more than a half of evolution-based stabilizing mutations can be evaluated as destabilizing by force-field calculations [17] and thus be left out by the usage of only evolution-based or only energy-based design.

Another type of regions that are not suited to be the primary target of mutations apart from **conserved regions** are nearby active sites of protein and **correlated residues** are often spotted when analyzing multiple sequence alignment. One might notice that some residues correlate (change together in evolution or between organisms) [112, 56]. This raises the assumption, that the conserved residues are in some way crucial for the function of the protein and are signs of intramolecular interaction. Changing them is a high risk (further discussed in section 7. Methods using a hybrid approach for the design of stable proteins are thus able to leave those regions untouched for the purpose of a faster and safer design. In addition to the previously mentioned benefits of the hybrid approach, they show an improved ability to work with multiple mutations on a single protein. This is important as single point mutations tend to have a smaller effect on overall protein stability [112].

5.3.1 Hybrid approach based software

Not many tools utilizing combined knowledge of evolutionary approach as well as physio-chemical properties are known. The three described approaches PROSS [55], FRESCO [157] and FireProt [112, 16] utilize some form of knowledge mining from the multiple sequence alignment as well as the incorporation of reliable energy-based tools such as Rosetta and/or FoldX.

PROSS Protein Repair One Stop Shop [55] is an automated web-based protein stabilization platform. PROSS starts by creating MSA from which it computes position-specific substitution matrix (PSSM) [7]. The matrix represents the log-likelihood of observing any amino acid at given positions - which is obviously an evolutionary approach that serves as a primary filter for all the possible mutations. Furthermore, it utilizes Rosetta with additional cutoff. This unfortunately misses some stabilizing mutations, however, the results are more reliable as proposed mutations are more likely to be stable. Lastly, Roseta's combinatorial sequence design tool find the potentinally optimal set of stabilizing mutations with energy function favoring amino acids with higher likelihood.

Framework for Rapid Enzyme Stabilization by Computational libraries **FRESCO** [157] can be divided into five steps. The first step is the generation of stabilizing mutations. FRESCO utilizes Rosetta as well as FoldX as the overlap between proposed mutations is only about 50% [157]. Furthermore, the novel Dynamic Disulfide Discovery algorithm which uses molecular dynamics to sample backbone conformational space. After potentially stabilizing mutations are generated, three screening steps follow: filtering out chemically unreasonable mutations, elimination of variants with predicted increases in protein flexibility, and experimental verification of improved T_M and preserved catalytic activity. The

result of those three steps are experimentally verified stabilizing mutations. The last step is combining them altogether to gain highly stabilized mutants.

The last mentioned tool **FireProt** [112, 16] firstly creates MSA for input sequence followed by consensus, correlation, and conservation analysis which are used as a filter for stability predictions utilizing FoldX as well as Rosetta. Furthermore, potentially stabilizing mutations are paired and reevaluated with Rosseta as pair mutations. Lastly, based on previous results, multi mutants are created. An in-depth description of FireProt will be given in chapter 7 together with proposed improvement ideas as an extension of this tool is the main objective of this thesis.

Chapter 6

Protein structure prediction

Protein structure prediction (also known as *protein folding problem*) is a process where we try to predict the spatial conformation (structure) of protein based on its sequence. As discussed in chapter 2.4 there are intermediate steps when aiming for protein's final structure - resolving small substructures as folds and helices.

Through an experimental effort, the structures of around 100,000 unique proteins have been determined [3]. However, this represents a small fraction of the billions of known protein sequences [107]. A steadily growing demand for the prediction of protein structure is the main cause of creating multiple models trying to accomplish this uneasy task. Naturally, with the increasing number of solutions, the need to evaluate and compare those models arise. The accuracy of some modeling servers is continuously evaluated by the *Continuous Automated Model Evaluation (CAMEO)* project [65] and can be used as a source to find the most suitable modeling software as well as an overview of existing solutions.

Commonly used methods can be split into the following categories: homology modeling which finds similar sequence with known structure, fold recognition, ab intio, and combination of multiple methods together with machine learning (mainly neural networks).

6.1 Comparative modeling

Comparative modeling (also known as *homology modeling*) is based on the knowledge that the spatial structures of related proteins are more conserved than their sequences. Therefore, if similarity between two proteins is detectable at the sequence level, structural similarity can usually be assumed [59, 102].

The result of homology modeling varies among the structures as highly-conserved core regions can be modeled much more reliably than variable loop regions or surface residues [134].

Some notable examples of software based on homology modeling are MODELER [150] and SWISS-MODEL [134].

The following sections describe the process of homology modeling divided into four steps: selecting a template, aligning the template with the input sequence, building the model, and evaluating the result.

6.1.1 Template selection

First, a protein with a similar sequence and known structure needs to be identified. This protein(s) will be used as a *template*. The *template* is the main source of information for the modeling. Finding the right template is crucial for a good result. Commonly used

tools for template finding are BLAST and FASTA. For finding more distant homologs, PSI-BLAST can be utilized. We can choose the template from alignment based on the highest identity. Templates with moderate (between 30% and 50%) identity strongly depend on loop modeling as loops among the homologs vary, while the core regions are still relatively conserved and accurately aligned. Templates with high identity (over 50%) tend to yield satisfactory results [134, 150].

6.1.2 Alignment

When using multiple structures as templates, they are superposed (structural alignment is created). Furthermore, alignment of the target sequence and the template is created.

This can be done by conventional multiple sequence alignment tools such as CLUSTAL [75, 137] or finetuned methods that take structural information into account [102]. For example, MOLDER uses a variable gap penalty function that calculates the gap penalty with the account of where it is in structure and chemical properties related to that [102].

There is a huge effort in finding the most fitting alignment possible as current homology modeling methods can't recover from an incorrect alignment.

6.1.3 Model building

There are several methods how to construct the model: modeling by rigid-body assembly [19, 57], modeling by segment matching [78, 96, 146] which relies on the approximate positions of conserved atoms in the template, and modeling by satisfaction of spatial restraints [13, 67, 133] which tries to find the spatial conformation that has the least conflict with the restrains.

Rigid bodies method was the first method designed for comparative modeling. This method begins by selecting the template structures and superposing them. After that, it averages the coordinates of the Cα atoms of the conserved regions, to take into account the different fitness of template structures. Their importance is weighted based on their similarity with input sequence, and significantly deviating atoms are excluded [134]. Next, which atoms of the main chain in the target model can be generated is based on the template, followed by generating the loops. The side chains are modeled based on their conformational preferences and on the conformation of responding template chains. Lastly, some form of energy minimization can be introduced, but generally, it does not dramatically improve the results [134, 86]. For example, the modeling tool utilizing the principles above is COMPOSER [142].

Segment matching is utilizing the knowledge that most hexapeptide segments can be divided into about one hundred classes [146]. Thus, the model can be constructed by using selected subsets of atomic positions from a template and finding coresponding sequences in the input sequence. Cα atoms, which are conserved between the alignment and the template, are often selected as guiding positions.

Methods based on **satisfaction of spatial restraints** try to generate many restraints on the protein conformation (we assume that the corresponding distances and angles between aligned residues in the template and the target structures are similar). Those restraints are then supplemented by stereochemical restraints on bond lengths, bond angles, dihedral angles, and nonbonded atom-atom contacts obtained from a molecular mechanics force field. Further restraints can be obtained by image reconstruction in electron microscopy, crosslinking experiments, fluorescence spectroscopy, etc. An example tool utilizing methods based on satisfaction of restraints is MODELER [150].

All methods mentioned above need to account for loop and side-chain modeling. Differences between homologs as a result of substitutions, insertions, and deletions of residues are the parts that differ between template and input sequence. Those short segments are extremely hard to predict from templates as segments of up to 9 residues can have entirely unrelated conformations in different proteins [105] Those changes are corresponding to exposed loop regions which often connect elements of secondary structure in the folded protein. Loop regions often determine the functional specificity of a given protein as they largely define the active and binding sites. This makes loop modeling a major factor for determining how useful the result can be when analyzing the interaction between proteins and ligands [102]. This problem is out of scope of this thesis and more information can be found in the „Modeling of loops in protein structures“ [47].

6.1.4 Evaluation

A good indication of model quality might be *C-score* which describes the variability of the template structures at the given position as inaccurate alignments between target and template are the most frequent source of errors in models [134].

An examples of the model evaluating matrices are Discrete Optimized Protein Energy (DOPE) [150], which is a scoring function derived from known proteins, or statistically optimized atomic potentials (SOAP) [40] which uses a Bayesian framework.

The errors can occur as the result of the comparative modeling and can be divided into five categories: errors in side-chain packing, distortions and shifts, incorrectly aligned regions, errors in regions without a template, and errors due to misalignments and incorrect templates [102].

6.2 Fold recognition

Fold recognition, also called threading or three-dimensional profile matching is a method of modeling spatial structure from protein sequence. This method possesses similar philosophy to comparison modeling - when we have a similar protein with a known structure we can use it as a guiding template for folding the input sequence [86]. Where the methods start to differ is how they choose the template. Comparative modeling relies on homologs (evolutionarily related sequences) with high identity (preferably over 50%), while fold recognition is not limited to sequences with known homologs as it relies on another principle. It has been found that there are many types of proteins with high structural similarity yet little or no similarity in their sequences (less than 15 percent) [77, 59] as unrelated proteins often adopt similar fold [53]. Thanks to this knowledge, fold recognition methods are much more potent when the sequence doesn't have any known homologs [122].

Another notable difference is in how threading treats the template - it tries to align the input sequence spatially with the template, thus extracting not only sequential information from the template, but also structural alignment. In other words, the input sequence is directly fitted („threaded“) onto the backbone of the template structure in full three-dimensional space incorporating specific pair interactions explicitly [77].

The spatial information fold recognition can utilize energy-based scoring functions and are analogical to minimization by a grid search where the grid points are being calculated on known protein structures [53]. When visualizing this method, a sequence of one protein is being forced (threaded through) to adopt a structure of another. Thus the name threading [22, 54].

For the score calculations, approximations need to be used. Scoring the two sequence alignment can be reasonably solved through dynamic programming, but energy-based scores are not local and alignment with nonlocal functions is an NP-complete problem [53]. Solutions are using an alignment technique that works with nonlocal scoring functions [22]. Two-level dynamic programming is used to optimize interaction for each possible pair of aligned residues [76] and they commonly utilize approximations to energy calculations [54]

When analyzing different software realizing threading, it is noticeable that most of the more effective methods include advanced sequence comparison methods and comparison of predicted secondary structure strings with those of known folds. They tend to combine threading with comparative modeling [59, 53]. When the input sequence is low-homology, the method can rely more on structural information, however, when suitable homologs can be found, homologous information is utilized [122].

6.3 Ab intio

Ab intio aims to be independent of template structure [159] as a poor template may have a negative effect on the whole modeling process, or does not make modeling possible at all [94]. Furthermore, by striving away from the template methods, we can gain more insight into how and why a protein adopts its specific structure [94].

Ab intio modeling strays away from the biological concept of finding similar or even homologous (evolutionary related) structures. Instead, it tries to start from scratch by firstly modeling the secondary structure (alpha helix, beta sheet, beta turn, etc.) of the protein, followed by finding the structure with the lowest free energy. Candidate structures are often generated by models that often run complex simulations, which are guided by composite knowledge-based force fields and often coupled with molecular dynamics simulations [159].

The main three parts of ab intio modelings are: : energy function design, conformational search, and model selection [94, 159].

Energy functions can be roughly divided into two categories [94]: physics-based energy functions (focusing only on the physicochemical properties) and knowledge-based energy functions (which utilize the statistical knowledge from the existing proteins in protein data bank (PDB) such as information about hydrogen bonding, local backbone stiffness of a polypeptide chain, pair-wise residue contact potential, etc).

Conformational search then utilizes the selected energy function to find the global minimum energy structure in a complicated energy landscape. Popular solutions are Monte Carlo and Molecular Dynamics.

Model selection is the last step, where many non-native structure conformations (also called decoys) need to be selected either by the energy-based or the free energy-based approach.

Currently, the bioinformatics and knowledge-based methods outperform methods based on the physicochemical principles in terms of speed and accuracy [94].

The main pitfall of ab intio modeling remains its time complexity, relying on massive computer performance. The progress of the past decade as well as a summary of used methods presented in the „Ab initio protein structure prediction“ [94].

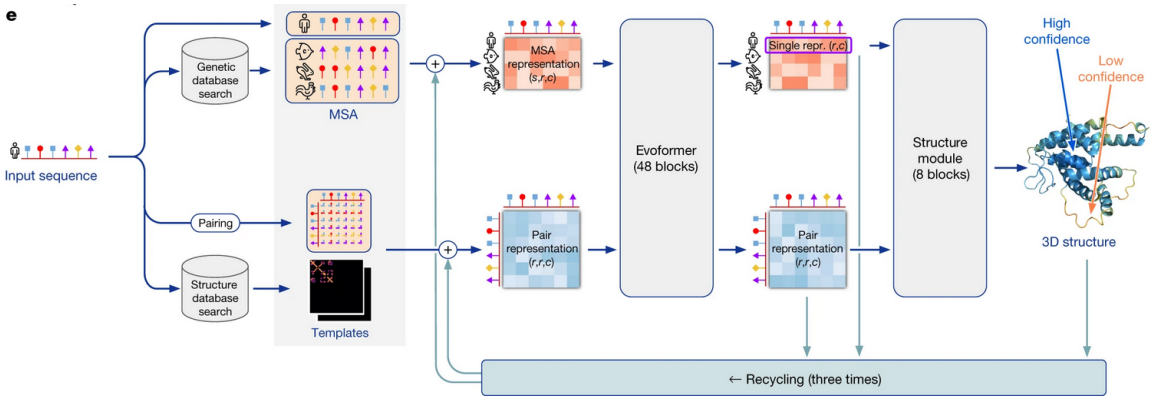

Figure 6.1: The architecture of AlphaFold2, with further description in text. s - number of sequences, r - number of residues, c -number of channels. Adapted from Jumper et. al. [79]

6.4 State of the art

With the increase of new methods to model protein structure, there was a growing need to create some form of unified evaluation and thus Critical assessment of methods of protein structure prediction (CASP) was created. The main idea behind CASP is that registered members of the modeling community are given sequences of proteins that are not yet known (but are about to be) [90]. The results of competitors are then evaluated by a battery of automated methods and by independent assessors [109]. The experiment is double-blinded as participants have no access to the experimental structures and assessors do not know the identity of submiters [89].

CASP13 held in 2018 saw dramatic progress in structure modeling without the use of structural templates [89] as DeepMind's entry, AlphaFold, placed first in the Free Modeling (FM) category [6].

In the most recent CASP14 held in 2020, DeepMind introduced AlphaFold2 whose results were competitive with experimental accuracy for two-thirds of the targets [90]. Additionally, a project to compute the structures of the most structurally challenging proteins coded for the SARS-CoV-2 genome was launched as a part of the competition as provided models may aid in the choice of drug targets, development of vaccine strategies and insights into viral mechanisms [88].

6.4.1 AplhaFold2

AlphaFold is a novel machine learning approach to protein structure modeling that incorporates physical and biological knowledge about protein structure and incorporates multi-sequence alignments into the design of the deep learning algorithm [79].

The architecture of AlphaFold2 depicted in Figure 6.1 . Further descriptions try to grasp the most important nuances of the concept described in the AlphaFold article [79].

Firstly, external databases are queried to create MSA and structural templates with at least lower-similarity sequences. Furthermore, a pair representation of sequence residues is created. Thus, two matrices are created (one of Size Number of sequences to a number of residues for MSA and the second, a square matrix of size equal to the number of residues in the input sequence to represent residue pairs).

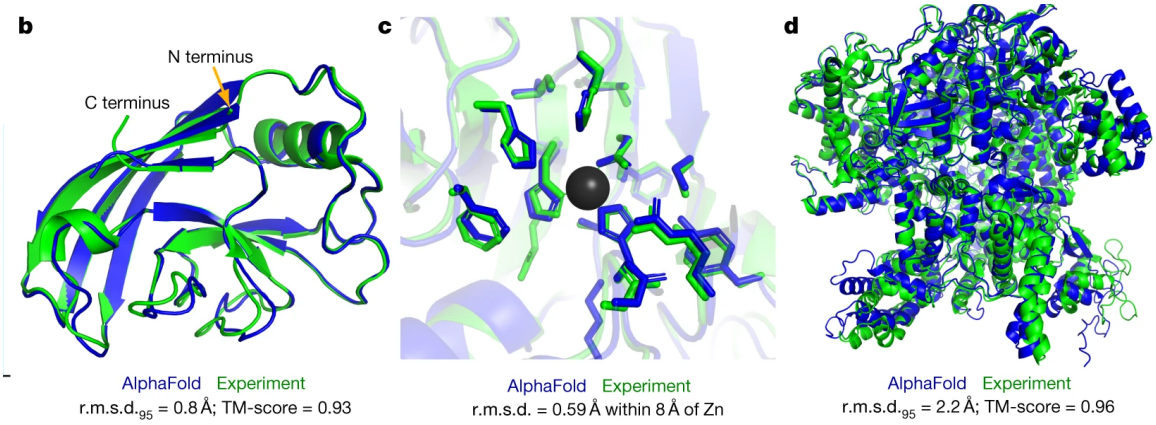

Figure 6.2: AlphaFold2 predictions from CASP14 for the following structures as PDB codes (left to right) 6Y4F, 6YJ1, 6VR4 Alpha fold predictions are in blue whereas the ground truth (experimentally determined structure) is green. Adapted from Jumper et. al. [79]

After that, a 48 layers deep transformer-like [35] neural network is utilized. This neural network has two interacting flows of information: one flow for the MSA (evolutionary) information and the second for pairs (spatial information). Furthermore, two attention schemas are utilized - one that keeps „attention“ to information in the rows (the protein sequence) and one for columns (the different positions of MSA representing the evolutionary information).

Lastly, the structure module transforms the output of the previous evoformer to create a 3D representation represented by quaternions. This is performed by a recurrent neural network that refines the spatial geometry of the proteins. The main idea is to iteratively update the sequence representation and the backbone transformation of the protein so it is gradually adjusted to the final form. An important concept that the network is utilizing is allowing local refinement of all parts of the structure.

Examples of the modeled proteins depicted in Figure 6.2.

Chapter 7

FireProt

FireProt is a web server for the automated design of multiple-point thermostable mutant proteins that combines structural and evolutionary predictions to filter out potentially deleterious mutations [112]. FireProt utilizes physical effective energy function-based tools as well as tools based on statistical potentials for which the energies are derived from frequencies of residues or atom contacts.

FireProt aims to be easy to use as bioinformatics knowledge is not required from users to calculate and analyze the results, yet it is highly configurable, so experienced users can modify values described in the following section. Section 7.1 will describe how FireProt can be used to predict a single point as well as multiple mutations from the user's point of view, and section 7.2 will describe the workflow of FireProt. Both sections will include ideas for improvement which were the main goal of this thesis.

7.1 FireProt WEB

When the need to design a more stable protein arises, one can visit the FireProt web interface available though: https://loschmidt.chemi.muni.cz/fireprotweb/. For inexperienced users, default values are advised and the only necessary input is the PDB code of protein with known structure. This is the first improvement idea discussed in this thesis - with the current advances in protein structure prediction tools it could be possible to incorporate some of the modeling tools and thus, FireProt would be able to work with only sequence as an input. The software architecture of FireProt WEB consists of three blocks: **frontend**, which displays the results and accepts the user input that is then pushed to **backend**. Backend validates the task description, creates a configuration file for the task and ensures that the task is correctly set in the database. Finally, the **core** module loads the job configuration file and starts to perform desired calculations.

When submitting a protein for analysis into FireProt, we talk about creating a FireProt **job**. First, user either uploads the structure in pdb format or sets the PDB ID and the structure is downloaded automatically. The next step regarding job options depicted in the Figure 7.1. Furthermore, the user can set catalytic residues, which can be further utilized to optimize the results (FireProt will try to load them automatically from SwissProt [14]). After the options are set (or left default), user can submit the job. This job is assigned a unique id, which the user should note or bookmark the URL as it will be useful when collecting the results. The runtime strongly depends on the number of residues in the

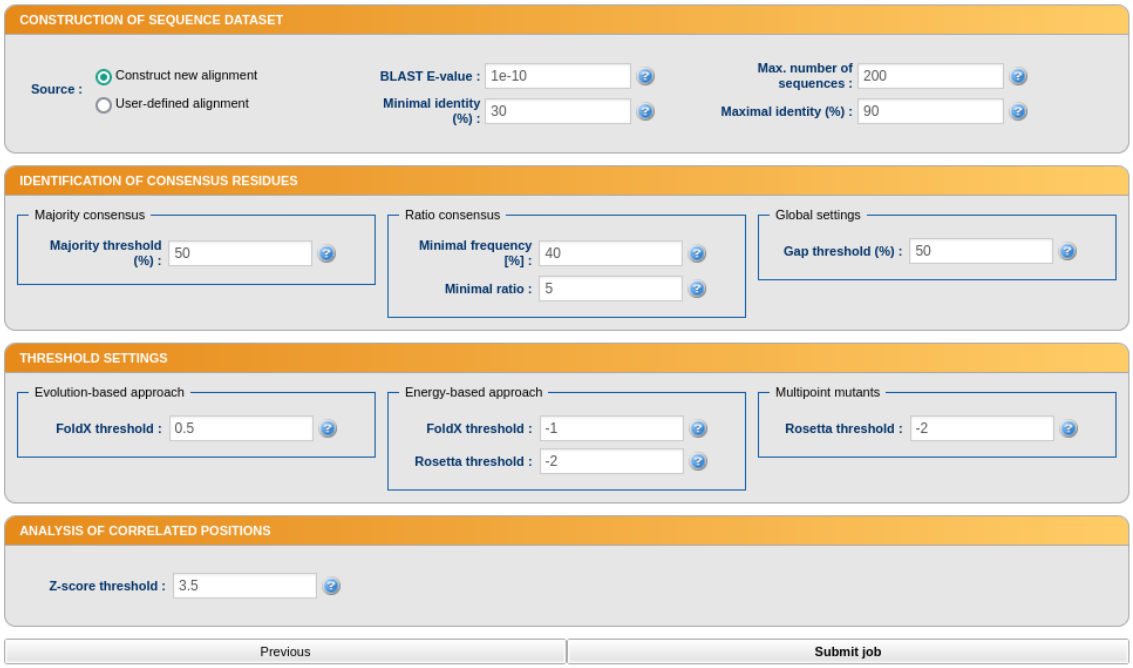

Figure 7.1: Overwiev of FireProt options. The various setting for MSA, identification of consensus residues, and the analysis of correlated positions. Furthermore different thresholds for FoldX and Rosetta can be set (resulting in providing either a lot of low confidence mutations or only a few high confidence). Taken as a screenshot from https://loschmidt.chemi.muni.cz/fireprotweb/ .

target protein. For relatively small-sized protein 4OEE (132 residues), the calculations are performed in about under four hours.

Through the ID generated in previous steps, users can access the FireProt result browser. While the job is still in progress, the browser allows the user to monitor which parts of the calculation were already finished, which are currently running, and which are waiting for the running modules together with the logs for detailed information about the progress of the calculation. After the task is finished, the result browser will look somewhat similar to what depicted in Figure 7.2.

7.2 FireProt core

This section will describe the workflow of the computational core of FireProt. Used tools, software, algorithms, and other implementation details will then be described in their own chapter 8. Furthermore, this section aims to describe the state before the implementation of this thesis and the differences will be discussed in the implementation chapter.

FireProt's workflow depicted in Figure 7.3 where the orange boxes represent modules and the arrows their dependencies. The **preparation module** prepares the structure to be further processed. This consists of steps like renumbering the structure from indexes that start at different numbers or utilizing only standard amino acids. Furthermore, additional user input (like specified catalytic residues) is loaded and mapped onto the structure. The following **MSA module** creates the alignment for each chain which is crucial for further

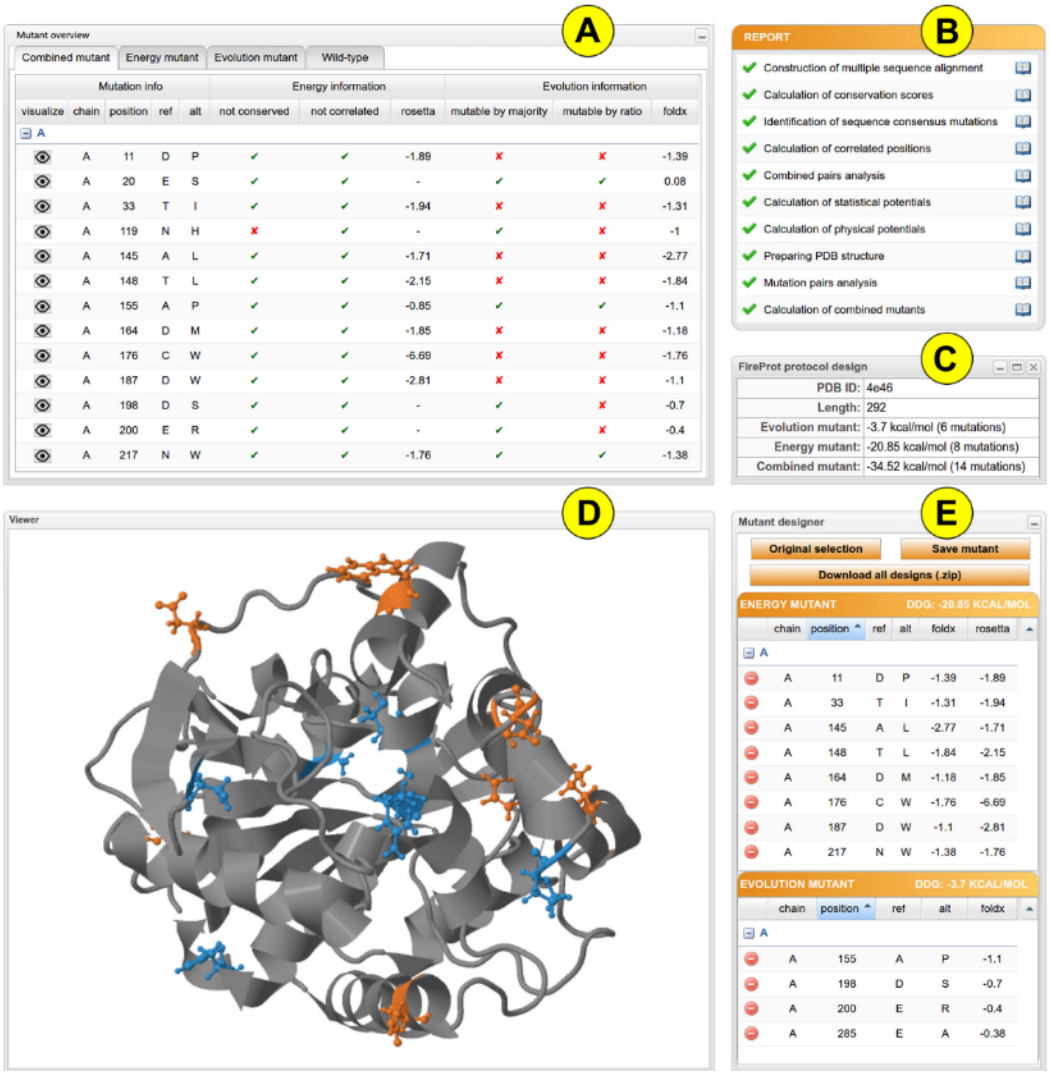

Figure 7.2: FireProt's result browser visualizes results obtained by calculations for 4E46. (A) The Mutant overview panel provides a list of mutations proposed as stabilizing. (B) The Report panel shows the status of calculation in the individual modules (useful when the job is still in progress) (C) The Protocol design panel provides general information about FireProt designs. (D) The JSmol viewer allows interactive visualization of the protein structure. (E) The Mutant designer panel enables manual adjustment of a new combined mutant. Adapted from FireProt [112].

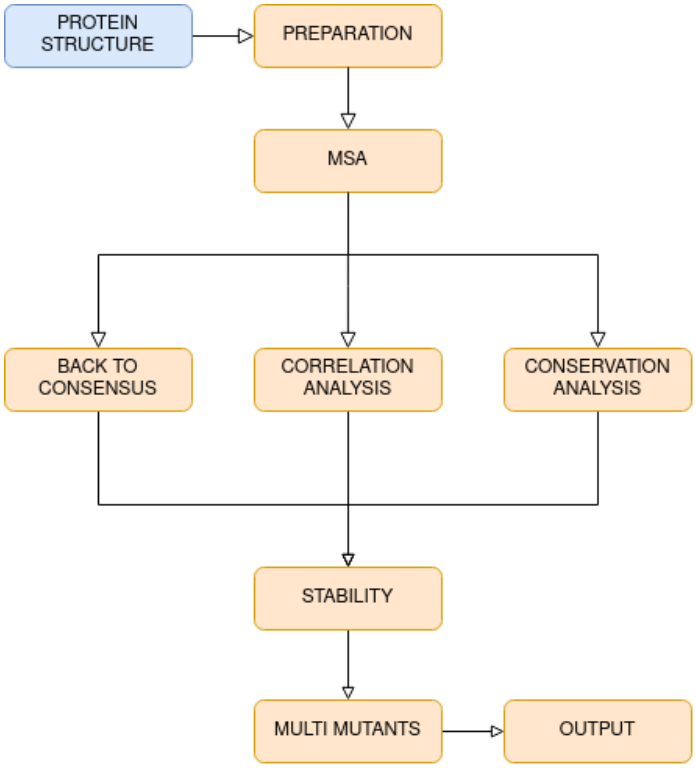

Figure 7.3: Dependency graph describing the workflow of FireProt. Orange boxes are computational modules implemented in Java.

steps as described in previous chapters. Following modules runs in parallel: **back to consensus module**, **correlation analysis module** and **conservation analysis module** which all try to extract the information from the multiple sequence alignment to filter out potentially deleterious mutations and recommend the stabilizing ones. The computationally most demanding module is **stability module** which utilizes FoldX and Rosetta to identify the $\Delta\Delta G$ of selected mutations. For the mutations that are deemed as stabilizing FireProt creates mutation pairs and tries to evaluate $\Delta\Delta G$ even for them. **Multi mutants module** utilizes the information of the stability module to create the most stabilizing mutant. Lastly, the **output module** parses the results from previous modules and creates files necessary for displaying the complete results to the user as well as a pdf report containing the analysis of different approaches for creating the best possible multi mutant.

FireProt results were experimentally verified on three proteins (PDB ID 4E46, 3A76, and 4OEE) and provided respective stabilization of proteins $\Delta T_m = 25, 21$, and $15°C$. Multiple point mutations were validated by the data provided by FRESCO [48] and those mutations were compared with another online protein stabilization tool PROSS [55] which showed similar predictive power (29 correctly identified potentially stabilizing mutations by FireProt and 20 by PROSS).

7.3 FireProt ASR

Another tool from the FireProt suite is a separate webserver for ancestral sequence reconstruction FireProt ASR [111] that is available at https://loschmidt.chemi.muni.cz/fireprotasr/ .

The workflow of FireProt ASR requires no user intervention besides providing the sequence to analyze and in the case of enzymes also set of catalytic residues. On the other hand, a more experienced user can also start by providing an initial set of homolog sequences, MSA, or even a phylogenetic tree instead of a single sequence.

The workflow of FireProt ASR can be split into two phases: the first is the collection of the initial set of homolog sequences and the second phase is the ancestral sequence reconstruction.

In the first phase, catalytic residues are either loaded automatically from SwissProt [20] and the Catalytic Site Atlas [130] or from users input. Next, EnzymeMiner [71] is used to collect an initial set of homolog sequences. If there are no available catalytic residues (the user did not provide them and the database search failed to yield any results), BLAST is used instead. The set of homologous sequences is then filtered by length (homologs length should be of input sequence length +- 20% by default), by identity (homologs identity should be higher than 30% but lower than 90%), and lastly remaining sequences are clustered together by identity (90%) and single representative sequence is selected. Applying these filters produces a diverse set containing hundreds to thousands of homologous sequences. Phylogenetic tree is then constructed utilizing PASTA software suite [106] and Treemer [103]. If the user interaction is desired, the tree is then displayed to the user who can modify it by deselecting some branches or making other manual adjustments.

After the tree is created, the main calculation of ancestral sequence reconstruction can begin. First, a new MSA is constructed from the reduced set of homologous sequences. For the construction of the final phylogenetic tree, best-fitting evolutionary matrix is selected either via the IQTREE package [115] or selected by the user. The phylogenetic tree is then constructed by RAxML [140] performing fifty bootstraps by default and rooted using a minimal ancestor deviation algorithm. The tree together with the evolutionary method and MSA are then used as an input to the Lazarus method [66]. Next, a structure of the query sequence is created utilizing BLASTp for finding the homolog template and ProMod3 for modeling. Finally, an algorithm for ancestral gap reconstruction is utilized to remove the gaps introduced by Lazarus.

The overview of both stages together with intermediate results and used tools depicted in Figure 7.4.

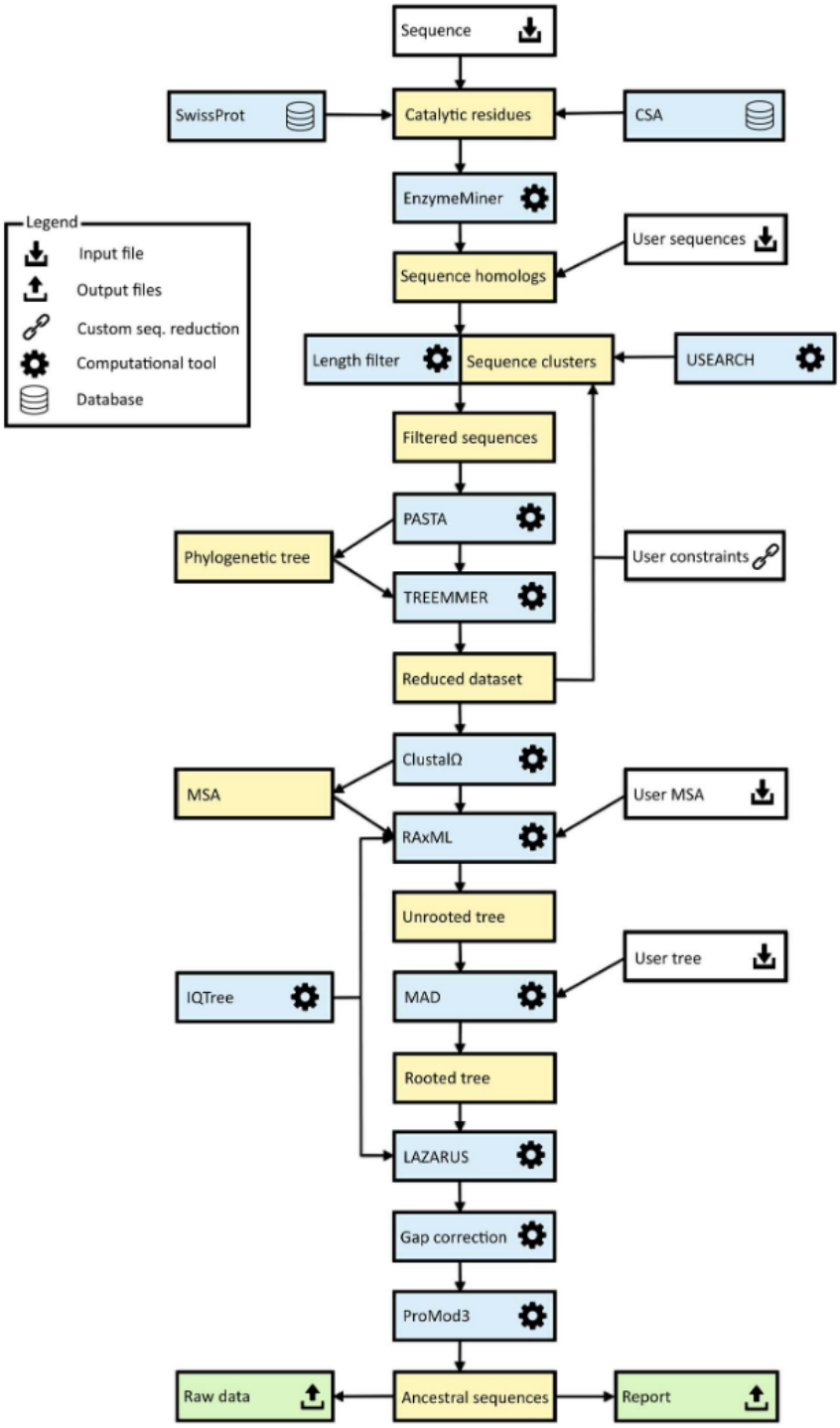

Figure 7.4: Workflow diagram for the FireProtASR. Colour coding: yellow denotes intermediate results and blue denotes computational tools. Grey and green denote inputs and outputs of the calculations, respectively. Adapted from FireProt ASR [111]

Chapter 8

Implementation

This chapter describes the implementation details of a new version of FireProt. It distinguishes between the work that has been already done as described in the original articles [112, 16] and the benefits acquired through the effort as part of this thesis described in section 8.2 and 8.4 respectively. Furthermore, a section 8.5 regarding the notable changes in existing modules is present as those modules were more or less completely rewritten. After the reader is familiar with the purpose of individual modules, features spanning multiple modules are presented in section 8.3. The complete overview of how the workflow changed from the previous version depicted in Figure 8.1.

8.1 Technology stack

FireProt is a modular Java[1] application which uses MariaDB[2] as a platform to store information about submitted jobs. As it needs considerable computational resources, it utilizes Portable Batch System (PBS[3]) to run jobs on a computational cluster. Those jobs are written in BASH[4] and are called from the Java application. As connections to the database and submitting PBS jobs are common for multiple tools developed in Loschmidt Laboratories[5], a library realizing those tasks was developed.

For the integration of ASR [111], which will be thoroughly explained in section 8.4.4, a whole new application was created. It is a Python3[6] application utilizing FastAPI framework[7] for creating CRUD interface to ASR and SQLAlchemy[8] toolset for database manipulation.

8.2 Existing modules

This section will describe modules that were already implemented in FireProt version 1.x and as such are not the work of the author. The improvements of those modules by the author will be noted otherwise.

[1] Java: https://www.java.com/en/
[2] MariaDB: https://mariadb.org/
[3] OpenPBS: https://www.openpbs.org/
[4] Bash: https://www.gnu.org/software/bash/
[5] Loschmidt Laboratories protein engineering group: https://loschmidt.chemi.muni.cz/peg/
[6] Python: https://www.python.org/
[7] FastAPI: https://fastapi.tiangolo.com/
[8] SQLAlchemy: https://www.sqlalchemy.org/

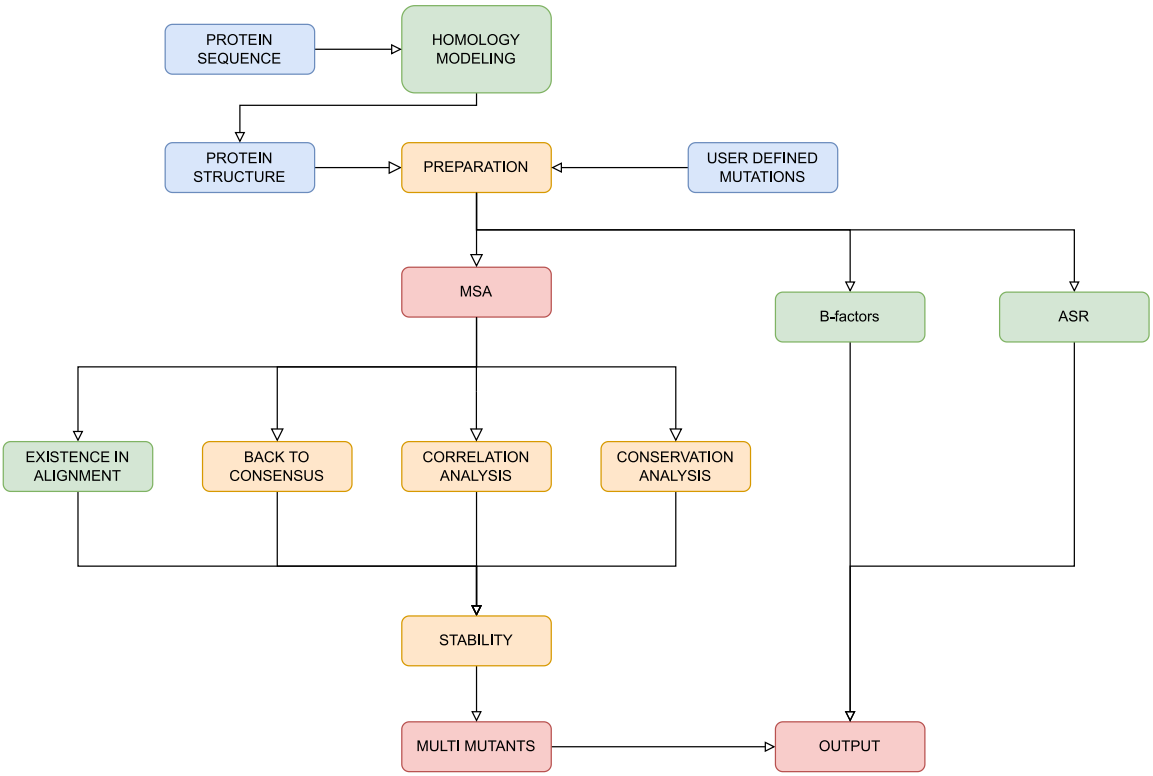

Figure 8.1: Workflow of the new version of FireProt. Green nodes denote newly implemented modules, blue is used for user input or output from the module, yellow nodes are previously implemented modules with small changes, and red modules were heavily reworked.

8.2.1 Preparation module

Preparation module is the first module in the module pipeline. Currently, this module has conditional dependency based on the user's input. Users can start the modeling as before: based on the structure of the selected protein. However, they can now also choose the option to start with an amino acid sequence. If the sequence option is selected, the homology modeling module is run first to create the protein structure as it is necessary for further calculations. The main goal of the preparation module is to perform the following tasks:

Repairing the structure is done by the FoldX's RepairPDB command. The main task of this module is to identify residues which has bad torsion angles, or Van der Waals clashes, or total energy, and recalculates them. RepairPDB[9] command looks for all Asn, Gln and His residues and flips them by 180 degrees to prevent incorrect rotamer assignment in the structure, then it does an optimization of the side chains to eliminate small Van der Waals clashes clashes which will prevent moving side chains in the final step. Lastly, the residues that show bad energies are mutated together with their neighbours to themselves, exploring different rotamer combinations so that the new energy minima is found.

Discarding the HETATMs removes the coordinates of hetero-atoms which were described in section 2.5.

As the pdb file can start with various indexes, and contain nonstandard amino acids and heatatoms **renumbering the structure** file needs to be done. This way, we can work with the array of amino acids and denote the amino acids by their positions in the array.

Next, we **minimize the structure file** through the Rosetta's *minimize_with_cst* command which imposes harmonic constraints on all C-alpha/C-alpha distances, and then minimizes the structure and further creates the constraint file.

Lastly, **mapping indexes** needs to be done. As the structure that went through the preparation has undergone some changes and we still need to pair the original sequence with the current one, MSA between the old and the new sequence is created together with mapping of indexes for the old and new sequence.

Changes necessary for the preparation module consisted of processing user-defined mutations and processing the input regarding the catalytic residues (which were moved from the stability module as catalytic residues can now be sent into the ASR right after the preparation module finishes).

8.2.2 MSA module

For other modules to be able to work, multiple sequence alignment needs to be created. This module has been heavily reworked. Previously, all steps were calculated locally and necessary software needed to be present on the machine running the FireProt. Currently, all the calculations were moved to the computing center, where necessary bash scripts are run together with the portable batch system to perform all the work. The main reason for moving the calculations to the computing center was not the computational speed as the calculation of MSA is nowadays rather quick, but rather a convenient way of decoupling the responsibilities. It made the setup of FireProt easier as running it doesn't require all the MSA dependencies.

If the user uploaded his own MSA, the alignment is parsed, loaded, verified, and mapped to the prepared sequence (as it can differ from the original one). Otherwise, MSA is created

[9] RepairPDB: https://foldxsuite.crg.eu/command/RepairPDB

by running BLAST against uniref90[10] database for each chain. Next, the results are filtered based on three separate filters.

Filtering by identity is done by *usearch* tool which calculates the identity of the input protein with the aligned homologs. Sequences with higher than maximal identity (which users can define or leave to default 90%) are discarded.

Filtering by clustering is also performed by usearch, which identifies the clusters of sequences with high identity (identity over 90% by default) and only selects a single representative sequence to keep the selected sequences variable.

Lastly, **filtering by coverage** is done to ensure less than the maximal number of homolog sequences is preserved in the final set.

8.2.3 Back to consensus module

Back to consensus module is a simple module that tries to figure out the dominant residue across the evolution. In other words, it scans the MSA columnwise. Back to consensus mutations are advised in two cases:

- By majority – if the amino acid is present in more sequences than the majority threshold (user-defined, default 50%).
- By ratio – if the amino acid is at least N times more frequent in MSA than the wild-type one (default is 5).

8.2.4 Conservation analysis module

The conservation analysis module is present to estimate the conservation coefficient of each residue position in the protein based on the Jensen–Shannon entropy. It assigns a mutability parameter to each residue which is then further represented to the user for human analysis.

8.2.5 Correlation analysis module

Correlation analysis module identifies correlated positions employing a consensual decision of following metrics: OMES [24], MI [87], aMIc [93], DCA [151], SCA [99], ELSC [34], McBASC [148], where MI, McBASC, OMES, SCA and ELSC are generated via Fodor package, DCA score by Freecontact tool and aMIc with the tool of the same name.

Next, the Z-score for each tool is calculated in the following steps

- calculate the mean of the distribution of the score produced by every metric
- calculate the variance of the distribution of score produced by every metric
- calculate standard deviation from previously calculated variance use mean & standard deviation to calculate Z-scores

The pairs are deemed correlated if the Z-Score of mentioned metrics is above a certain threshold (user-defined, 3.5 by default).

[10] Uniref: https://www.uniprot.org/help/uniref

8.2.6 Stability module

The stability module is the computationally heaviest module as it utilizes both Rosetta and FoldX to calculate the $\Delta\Delta G$ of all possible mutations that were not deemed as destabilizing by the evolutionary filters.

Mutation pool is created. Every residue can be mutated to every other amino acid (nineteen possibilities). From the pool, those residues which are either catalytic or within the specified distance from catalytic residues are removed instantly. Furthermore, mutations that have a low mutability scale class set during the conservation analysis as described in section 8.2.4 and correlated residues which were described in section 8.2.5 are also removed. Additional filter was implemented and depends whether user chose *low risk* or *high risk* strategy. More in section 8.3.1. This way, two sets of mutations are created: the energy approach mutation set and evolution approach mutation set.

Both mutation sets are processed separately. Firstly, they are run through FoldX and next through Rosetta. After that, mutation pairs are created and evaluated in Rosetta to help to detect conflicting pairs. Those pairs are evaluated for three types of mutation combinations: the energy-driven approach, the evolution-driven approach, and both of them combined.

8.3 Features across multiple modules

Some features cannot be implemented in a single module as they affect the whole computational workflow.

8.3.1 Risk definition

Previous versions of FireProt considered three mutational strategies: back to consensus, evolutionary, and combined. The risk definition was implemented so the users analyzing the protein for enriching its stability can decide whether they want a faster and more safe design of energetic mutations (low-risk strategy), a design that spans a larger search space, consists of potentially less beneficial mutations and will have longer computational time (high risk), or both.

The main task of this module occurs when selecting mutations to evaluate in the stability module. Before the implementation, only mutations that don't change the charge of the amino acids were considered. Now, the low-risk strategy considers only the mutation which doesn't change the charge of amino acids and at the same time exists in alignment (based on the user-defined threshold). On the other hand, the high-risk mutations don't consider the charge nor the existence in alignment. The possible combinations of back to consensus approach and energy-based approach with different risk definition results in the following mutational strategies:

- BTC - back to consensus approach
- ENERGY - energy-based approach with low risk (mutations exist in alignment and don't change charge)
- ENERGY HIGH RISK - energy-based approach with high risk (mutations don't need to exist in alignment and they can change charge)
- COMBINED LOW RISK - a combination of BTC and ENERGY

- COMBINED HIGH RISK - a combination of BTC and ENERGY HIGH RISK

Those strategies are further utilized when designing the potentially most stable multi mutant. In the resulting reports, the user is informed of the results for all selected strategies and each mutation in the high-risk strategy is also annotated with risk (if the mutation would occur even in a low-risk approach).

8.3.2 User-defined mutations

This feature allows the user to enter mutations that the user wants to have evaluated. This can dramatically speed up computational time as it can decrease search space and the most time-demanding task - the stability module - will run FoldX only for desired mutations.

Mutations are entered by the user in a format chain, wild-type amino acid, position, and desired mutations. For example, string „AL72Y“ means that the leucine on position seventy-two on-chain „A“ will be mutated to tyrosine.

When mutations are loaded and validated (only standard amino acids are present, the wild type in a given position matches the entered one) they are later mapped on the minimized structure during preparation.

Lastly, when mutations in the stability module are generated for evaluation, user-defined mutations will be used instead. The output of running FireProt will contain information about all the mutations and residues, but only the user-defined will contain useful information (other mutations will be mostly empty). This is done so the front-end application can display the data in a consistent way and doesn't need to treat user-defined mutations differently.

8.4 New modules

To enrich FireProt functionality and provide a better understanding of the mutated protein, new features and integrations were made. This section describes the novel work and tries to describe its benefits.

8.4.1 Homology modeling module

There were two reasons for creating a homology modeling module: first, users can now design stable proteins not only for proteins with known structure but even for proteins with only the sequence available as the structure will be modeled during the FireProt calculation. The second reason is the modeling of the resulting multi mutants for further visual analysis.

The first idea was to use the new version of AlphaFold, but unfortunately, it was not infrastructurally possible to prepare the environment for AlphaFold runtime. In result, ProMod3 [121] was used instead with the idea that only additional parameters will be passed and it can be replaced by AlphaFold in the future.

The sole input of the module is the protein sequence, which is sent to the computing cluster and the homology modeling bash script is run. This script firstly searches for homologous sequences in the uniref90 database with the BLAST tool. Next, it downloads the protein with the highest identity from the protein data bank. Lastly, it runs ProMod3 which reads a target-template alignment from the alignment and a matching structure from the downloaded protein as a template and produces a gap-less model. Finally, the resulting model is downloaded from the computing cluster.

Figure 8.2: Existence in alignment. The first sequence is the input sequence (stabilized protein) following are example homologs. Circled amino acids are considered when the threshold is set to 25% (at least one sequence from four) and the last line shows which mutations would be recommended with a threshold 50% (at least two sequences contain given residue). Example sequences were selected as a part of alignment for 4OEE against the uniref90 database.

If the homology modeling was run for the input sequence, the workflow can continue by running the preparation module for the modeled structure. In the case of modeling the multi mutant, this is also the last step and output module can be run.

8.4.2 B-Factors

The amino acids that display the highest B-factors are the ones suitable for mutation as they are corresponding to the most pronounced degrees of thermal motion and thus flexibility [129].

To determine the relative flexibility for each residue, all of them are sorted depending on their average B-factor (which is determined based on the data in the pdb file). After that, the residues are ranked ranging from 1–100%. Rankings of 1–25%, 26–75% and 76–100% indicate high, moderate and low levels of relative structural flexibility, respectively. The idea was taken over from HotSpot Wizard [18]. The information about B-factor is not utilized as a filter in FireProt, but rather as an additional information included in the output report for the user to work with.

8.4.3 Existence in alignment

Existence in alignment is a module that analyses the residues in the same positions across the homologous sequences - in other words, it works with columns in MSA. Users can define the minimal occurrence in percent upon which mutations are recommended because they „exist in alignment“. An example depicted in the Figure 8.2.

8.4.4 FireProt ASR

FireProt ASR tool was described earlier in section 7.3. The idea behind the ASR module in FireProt is that the only mandatory input for the FireProt ASR calculation is the sequence and preferably catalytical residues, both of which FireProt already has. It would be more

Figure 8.3: Swagger generated documentation of methods available for FireProt ASR API.

convenient for the user to have a single point of entry for the stability analysis and FireProt could therefore call FireProt ASR from within.

To make this integration possible, a new API webserver called „FireProt ASR API“ was created for FireProt to interact with.

FireProt ASR API

FireProt ASR API is a standalone Python3 application utilizing FastAPI and SQLAlchemy modules. This application loads the environmental variables for connection to the ASR database and maps the ASR job table through ORM mapping to Python objects. The application then listens on a specified port for requests. FastAPI is able to automatically generate Swagger documentation as depicted in the Figure 8.3.

Methods for submitting a new ASR job require a job definition file together with the IP address of the submitter, it then stores the job definition file so it can serve it for the FireProt ASR when requested. It is also capable of automatically generating a necessary config file for the FireProt ASR when requested (the config URL). The method for getting job status is used to check whether the ASR job is waiting to be processed, running, failed, or finished successfully. Lastly, method for checking whether a job with a given id exists is present, so FireProt can prevent creating jobs with duplicate ids and causing errors.

FireProt ASR Module

In the module itself, the main task is to generate an ASR job file for each protein chain, which is represented by a file in XML format, where modules are elements with the default configuration. As catalytic residues are passed into FireProt, they are stripped of chain information (ASR works only on a single chain and as such, information about the chain is redundant). The job then contains an info element, global parameters element, and element for each module: filtering, reconstruction, and reduction.

As protein processed by FireProt can have multiple chains, the pool of ASR jobs is created and submitted in parallel. The status of these jobs is then periodically queried and when any of the jobs are finished, the ASR API is queried for the URL of the results, which are then downloaded and presented in the results of FireProt. FireProt naturally handles all possible errors and logs the information so it can be further analyzed.

8.5 Reworked modules

8.5.1 Multi mutant module

For the construction of the multiple point mutations, it is necessary to define the concept of *antagonisic mutations*: antagonistic mutations occur if their combined $\Delta\Delta G$ is significantly lower than the sum of their individual effects on stability.

The construction of the most promising multi-mutant in the previous version of FireProt followed a simple approach, where the best mutation was selected and the next possible mutations were added by iterating over all possible mutations and picking the ones which were not antagonistic with the mutations already selected.

One of the main goals of this thesis was to design a novel approach that would utilize as much information as possible and provided the most promising multiple point mutations.

Multi mutant search as a graph problem

The final approach is designed to reduce the problem of finding the best possible mutant into a graph problem. The input of the module is the set of all possible mutations with their $\Delta\Delta G$ and all pairs of spatially close mutations with their $\Delta\Delta G$. The $\Delta\Delta G$ value is not determined for the mutations which are far apart (by default more than 10Å), as they are considered to be too distant to affect each other and the $\Delta\Delta G$ of the two mutations is calculated as the sum of $\Delta\Delta G$ of individual mutations. With this, we can create a full graph, where every node represents a single mutation and the edges are the $\Delta\Delta G$ of mutation pairs. Next, the edges representing antagonistic mutations are removed from the graph as depicted in the Figure 8.4.

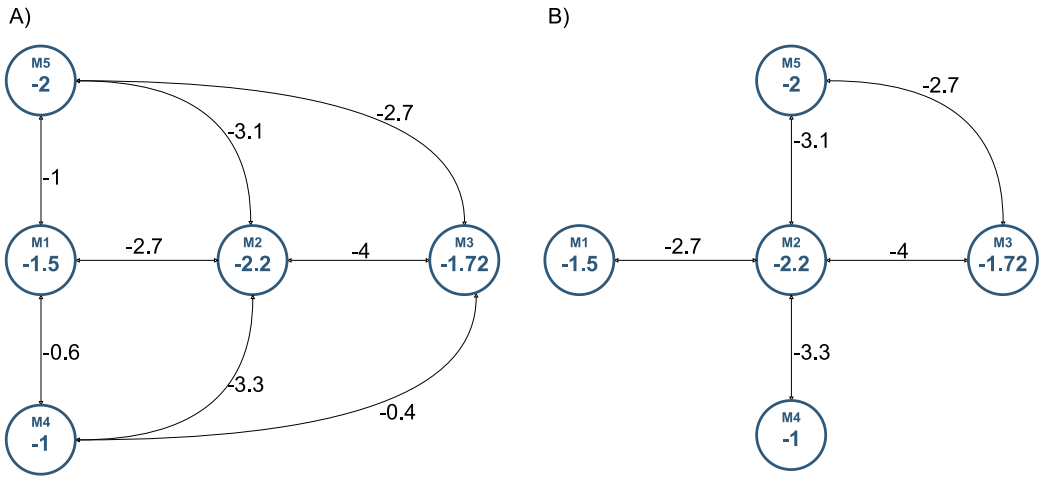

Figure 8.4: The figure shows the construction of a graph for creating multi mutant. The graph follows Rosetta's convention and a smaller value indicates a more stabilizing mutation. In the first graph, A) the relationship of all mutation pairs is described by the values of edges, the edge between M1 and M3 illustrates the missing edge when dealing with two mutations in the same position. Graph B) show the state after removing edges between antagonistic mutations.

The graph obtained as a result of these steps is then representing every possible multi mutant as a complete induced subgraph - clique. In other words, the solution can be

represented as a subset of vertices (mutations) of the given graph, where every two distinct vertices are adjacent (mutations are possible to be together). As the total number of cliques in a complete graph is 2^n, we can further simplify the problem by considering only maximal cliques. Maximal clique is a clique that cannot be extended by including one more adjacent vertex, in other words, a clique which does not exist exclusively within the vertex set of a larger clique. When transferred to the problem of mutli mutant, a clique that is not maximal represents multi mutant that can be extended by adding further mutation. Therefore, maximal clique represents multi mutant which cannot be extended with another mutation. How the number of maximal cliques in the complete graph can be calculated is described by Moon and Mooser [108]. However, generally speaking, we can consider it as $3^{n/3}$, which is a noticeable improvement in the size of the search space and with it related memory usage and the time requirements.

Finding all possible solutions

To find all possible maximal cliques of undirected graph, Bron–Kerbosch algorithm [21] with pivot was employed. The pseudocode of this algorithm is as follows:

Algorithm 1 Bron–Kerbosch

```
1:  procedure BronKerbosch(R, P, X)
2:      if P is empty and X is empty then
3:          R is max clique
4:      end if
5:
6:      for v in P do
7:          BronKerbosch(R ∪ {v}, P ∪ N(v), X ∩ N(v))
8:          P = P \ {v}
9:          X = X ∪ {v}
10:     end for
11: end procedure
```

The algorithm works on the principle of recursive backtracking. R, P and X are three disjoint sets of veriticies, where R represents the currently processed clique (result), P represents the potentional candidates to add to processed clique, and X is an exclusion set representing verticies that are not possible to add into currently processed clique. The function representing the Bron–Kerbosch algorithm is firstly setting R empty (representing empty clique), P containing all verticies of graph, representing that potentialy all of them can be added into the cliq, and X as empty as no verticies need to be excluded yet. The recursion step on line seven calls the function for each vertex in the potentional set as added to the resulting set. The set of verticies that can be added to the clique is represented as neighbours of added vertex v and the same verticies should be excluded if they were added before. Following lines in the pseudocode, the algorithm updates the potentional candidates by removing vertex v and adding it to the excluded set. Furthermore the algorithm can be improved by introducing the pivot as follows:

The idea behind selecting a pivot vertex is that every maximal clique must contain either the pivot vertex or the non-neighboring vertex of the pivot. In other words, if the clique does not contain a non-neighbor of the vertex, then the vertex can be added to the clique. This decreases the search space and the algorithm is then more efficient, especially

Algorithm 2 Bron–Kerbosch with pivot

```
1:  procedure BronKerbosch(R, P, X)
2:      if P is empmty and X is empty then
3:          R is max clique
4:      end if
5:
6:      Select pivot vertex u in P ∪ X
7:      for v in P \ N(u) do
8:          BronKerbosch(R ∪ {v}, P ∪ N(v), X ∩ N(v))
9:          P = P \ {v}
10:         X = X ∪ {v}
11:     end for
12: end procedure
```

on a graph with a lot of non-maximal cliques. In the algorithm, we can see that before iterating over the vertices in the main loop, we select the pivot from the nonresulting sets. In the implementation done as a part of this thesis a selection based upon the number of neighboring nodes was implemented. In result, we only need to make the size of P - the number of neighbors of selected pivot iterations instead of the size of P in the main loop. The decrease of search space, as well as the progress of the algorithm with visualized recursion, depicted in the figure 8.5.

Obtainig final solutions

By applying the Bron-Kerbosch algorithm, all possible solutions were discovered. Every maximal clique represents a single multi mutant. In Figure 8.5, we can see two solutions: blue multi mutant formed by mutations 1, 2 and 3 and orange one formed by mutations 3 and 4. To propose the single best one, their values are estimated simply by summing their $\Delta\Delta G$ and selecting the best one. There is also a possibility to receive multiple different multi mutants as they are all evaluated and ordered by the sum of their $\Delta\Delta G$. Construction of the graphs, finding its maximal cliques, and their evaluation is performed over all selected strategies: for mutations based simply on back to consensus module, high-risk energetic mutations, low-risk energetic mutations, and a combination of mutations from a back to consensus approach with the energetic ones. Depending on the user's choices, the result contains up to five different multi mutants.

8.5.2 Output module

To summarize the results of the preovious modules and prepare them for further display, output module collects the necessary information and concentrates them into two main files. One containing as much information as possible for every single residue of the protein and one containing the resulting multi mutants for selected mutational strategies.

Output for every residue

The per residue output is in a format suitable for frontend processing. It is XML format and the structure is roughly described in Figure 8.6. The root node is *protein* with attributes describing protein pdb id and version of FireProt used for calculation. The protein node

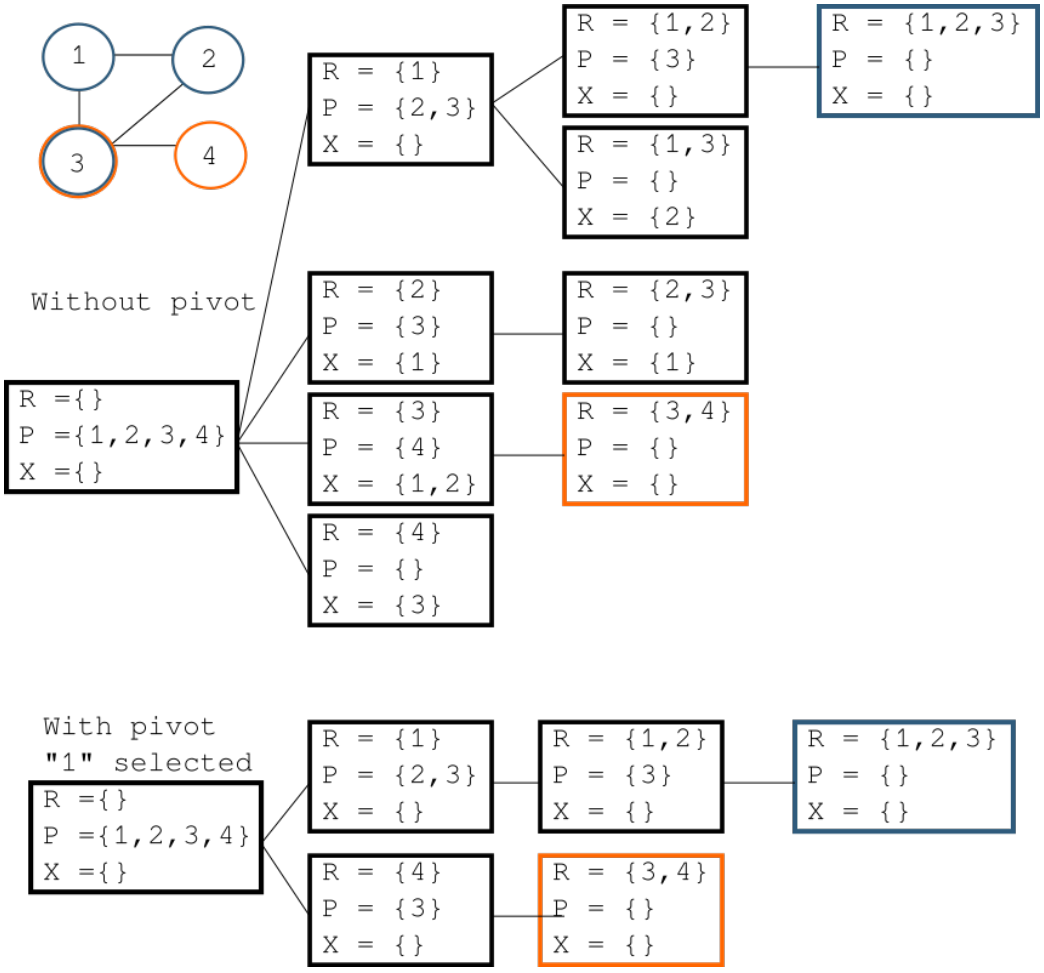

Figure 8.5: Example of Bron Kerbosch algorithm on a simple graph for the unmodified version as well as the version with pivot. The columns represent the depth of recursion and the resulting maximum cliques are colored.

contains *general* node. This node contains additional information regarding the results, namely the length of the protein sequence together with used mutational strategies and mutations obtained by the given approach. Next, under the protein node is *chains* node, which contains the node for every chain. The *chain* nodes contain information about every residue in separate nodes. The *residue* node contains various *mutability* information like B-factor, conservation score, if the residue is conserved or correlated, and the correlation score, along with twenty nodes for every possible mutation. The *mutation* nodes contain information for the *back to consensus* approach (whether it is selected by the majority or by the ratio approach) as well as the energetic approach. The *energy* node contains information about FoldX and Rosetta results (which is empty if it did not pass the previous filter and was considered unsafe). Lastly, every mutation contains multiple mutational strategies (explained in section 8.3.1) and if this strategy recommends the mutation as well as the information if the mutation exists in alignment.

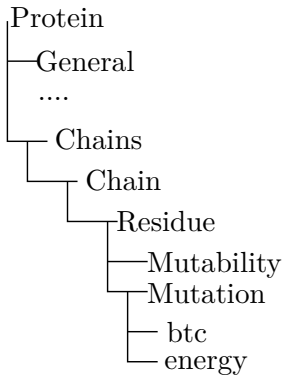

Figure 8.6: Organization of resulting XML containing information about every possible mutation.

Ouput for multi mutants

The generated output for the recommended multi mutants is intended to be human-readable and as such, the PDF file is generated. This file contains a table for every selected mutational strategy (described in section 8.3.1). The table shows information based on the selected strategy. There is always a field for the mutation (wild type residue and the mutant) the index of the mutation, the chain id, b-factor, and the FoldX value of the mutation. The back to consensus approach contains information if the mutation was selected based on the majority or ratio approach. The energy approach contains information on whether the mutation is conserved or correlated and the Rosetta energy calculation. The high-risk mutational strategies contain information on whether the mutation is high risk or would be recommended even at low risk and the combined table contains information that can be found in both approaches. The example table for the combined high-risk approach depicted in the Figure 8.7 as it contains all described information. An example for tables representing individual multi mutants are present in appendix.

COMBINED HIGH RISK mutant: 13 mutations

Chain	Mutation	B-factor	High-risk	EIA	Majority	Ratio	Conserved	Correlated	FoldX [kcal/mol]	Rosetta [kcal/mol]
A	R96M	MOD.	✓	✗			✓	✗	-1.06	-2.33
A	S89E	HIGH	✓	✓	✓	✗	✓	✗	-0.32	
A	L72Y	HIGH	✗	✓	✓	✓	✓	✗	0.25	
A	R109K	HIGH	✗	✓	✓	✗	✓	✗	-0.19	
A	Q43I	HIGH	✓	✗			✓	✗	-1.48	-2.92
A	K15L	HIGH	✓	✗			✓	✗	-1.13	-2.04
A	K124P	HIGH	✓	✗			✓	✗	-1.62	-4.91
A	K108T	MOD.	✓	✗			✓	✗	-1.26	-2.94
A	E67P	HIGH	✓	✗			✓	✗	-1.84	-2.93
A	E88M	HIGH	✓	✗			✓	✗	-1.05	-2.04
A	A46L	MOD.	✓	✗			✓	✗	-1.04	-2.22
A	E47V	MOD.	✓	✓	✓	✗	✓	✗	-0.70	
A	S74Y	MOD.	✓	✗			✓	✗	-1.04	-2.11

Figure 8.7: Resulting multi mutant table for the combined high-risk strategy for a multi mutant of protein with pdb id 4OEE. Meaning of columns explained in the text.

Chapter 9

Conclusion

Firstly, the thesis introduced the fundamental concepts of proteins and further focused on their origin in nucleic acids. The protein origin is described by the definition of protein folding mechanisms. With these principles in mind, the concept regarding improving the proteins stability via amino acid mutations is introduced. For the identification of possibly stabilizing mutants, those approaches were divided into two groups.

At first, the methods concerning sequence-based methods that don't rely on the spatial structure of a given protein but only on its sequence (also called primary structure) were introduced. These methods utilize primary information hidden inside evolutionary close proteins – homologs, which are aligned into the multiple sequence alignment. The methods utilizing the evolutionary approach include back to consensus method, correlation and conservation analysis, and ancestral sequence reconstruction. The second category includes methods based on the analysis of protein spatial structure that mainly utilize the knowledge of force fields for the usage of energy-based methods.

As there is a large number of proteins without known spatial structure, the recent improvement in the field of protein folding was described including the state-of-the-art method AlphaFold.

The bioinformatics tools which can be used for the realization of the discussed approaches were mentioned. Sometimes, when the tools are of high importance, key concepts and algorithms were explained. The most thorough explanation was naturally provided for the FireProt tool as its improvements are the main focus of this thesis.

Most notably, the benefits of this thesis for the FireProt tool were addressed. Firstly, the protein modeling module enabled the users to start the analysis not only for the proteins with known structure, but also with a single sequence as an input. An additional feature available to the users is the definition of the mutations which should be evaluated. Furthermore, the user can now decide between the „high risk“ and „low risk“ strategy of mutation evaluation which now selects if the mutations that do not exist in the multiple sequence alignment and that are affecting the amino acid charges should be considered. The module for the calculation of the multiple sequence alignment is now fully present in the supercomputing center, contrary to before, when all the calculations were performed locally. Newly implemented modules include the calculation of relative B-factor, integration of the FireProt ASR tool through a standalone API server, and a module for calculating whether newly mutated residue already exists in a given percent of cases in the multiple sequence alignment. The notable benefit of this thesis is a novel approach to multiple-point mutant creation. The graph representing every single point mutation (node) and relations between them (edge) is created, and an algorithm for the detection of all the maximal cliques is uti-

lized to find possible multi mutants. All of them are evaluated and the best one is selected. This yields up to five different multi mutants for different mutational strategies (back to consensus, evolutionary approach with two different risk strategies, and the combination of evolutionary approach with the energetic one). Lastly, the output module was redefined to better suit the needs of the frontend application displaying the results and reflecting the implemented changes and added features.

Bibliography

[1] Protein stability curves. *Biopolymers*. 1987, vol. 26. DOI: 10.1002/bip.360261104. ISSN 10970282.

[2] Protein thermostability engineering. *RSC Advances*. 2016, vol. 6. DOI: 10.1039/c6ra16992a. ISSN 20462069.

[3] *PDB Statistics: Overall Growth of Released Structures Per Year* [online]. 2021 [cit. 2021-12-23]. Available at: https://www.rcsb.org/stats/growth/growth-released-structures.

[4] *Uniprot > Current Release Statistics* [online]. 2021 [cit. 2021-12-23]. Available at: https://www.ebi.ac.uk/uniprot/TrEMBLstats.

[5] 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*. ACS Publications. 2017, vol. 13, no. 6, p. 3031–3048.

[6] AlQuraishi, M. AlphaFold at CASP13. *Bioinformatics*. may 2019, vol. 35, no. 22, p. 4862–4865. DOI: 10.1093/bioinformatics/btz422. ISSN 1367-4803. Available at: https://doi.org/10.1093/bioinformatics/btz422.

[7] 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*. december 2008, vol. 37, no. 3, p. 815–824. DOI: 10.1093/nar/gkn981. ISSN 0305-1048. Available at: https://doi.org/10.1093/nar/gkn981.

[8] Altschul, S. F., Gish, W., Miller, W., Myers, E. W. and Lipman, D. J. Basic local alignment search tool. *Journal of Molecular Biology*. 1990, vol. 215, no. 3, p. 403–410. DOI: https://doi.org/10.1016/S0022-2836(05)80360-2. ISSN 0022-2836. Available at: https://www.sciencedirect.com/science/article/pii/S0022283605803602.

[9] Anfinsen, C. B. Principles that govern the folding of protein chains. *Science*. 1973, vol. 181. DOI: 10.1126/science.181.4096.223. ISSN 00368075.

[10] Argos, P. and Rossmann, M. G. Structural comparisons of heme binding proteins. *Biochemistry*. ACS Publications. 1979, vol. 18, no. 22, p. 4951–4960.

[11] Armougom, F., Moretti, S., Poirot, O., Audic, S., Dumas, P. et al. Expresso: automatic incorporation of structural information in multiple sequence alignments

using 3D-Coffee. *Nucleic acids research*. Oxford University Press. 2006, vol. 34, suppl_2, p. W604–W608.

[12] 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. 2012, vol. 40, W1, p. W580–W584.

[13] Aszódi, A. and Taylor, W. R. Homology modelling by distance geometry. *Folding and Design*. Elsevier. 1996, vol. 1, no. 5, p. 325–334.

[14] Bairoch, A. and Boeckmann, B. The SWISS-PROT protein sequence data bank. *Nucleic acids research*. Oxford University Press. 1991, vol. 19, Suppl, p. 2247.

[15] Baldasseroni, F. and Pascarella, S. Subunit interfaces of oligomeric hyperthermophilic enzymes display enhanced compactness. *International journal of biological macromolecules*. Elsevier. 2009, vol. 44, no. 4, p. 353–360.

[16] Bednar, D., Beerens, K., Sebestova, E., Bendl, J., Khare, S. et al. FireProt: energy-and evolution-based computational design of thermostable multiple-point mutants. *PLoS computational biology*. Public Library of Science San Francisco, CA USA. 2015, vol. 11, no. 11, p. e1004556.

[17] 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*. 2018, vol. 8. DOI: 10.1021/acscatal.8b01677. ISSN 21555435.

[18] 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. 2016, vol. 44, W1, p. W479–W487.

[19] Blundell, T., Sibanda, B., Sternberg, M. and Thornton, J. Knowledge-based prediction of protein structures and the design of novel molecules. *Nature*. Nature Publishing Group. 1987, vol. 326, no. 6111, p. 347–352.

[20] Boeckmann, B., Bairoch, A., Apweiler, R., Blatter, M.-C., Estreicher, A. et al. The SWISS-PROT protein knowledgebase and its supplement TrEMBL in 2003. *Nucleic acids research*. Oxford University Press. 2003, vol. 31, no. 1, p. 365–370.

[21] Bron, C. and Kerbosch, J. Algorithm 457: finding all cliques of an undirected graph. *Communications of the ACM*. ACM New York, NY, USA. 1973, vol. 16, no. 9, p. 575–577.

[22] Bryant, S. H. and Lawrence, C. E. An empirical energy function for threading protein sequence through the folding motif. *Proteins: Structure, Function, and Bioinformatics*. Wiley Online Library. 1993, vol. 16, no. 1, p. 92–112.

[23] Cang, Z. and Wei, G. TopologyNet: Topology based deep convolutional and multi-task neural networks for biomolecular property predictions. *PLoS Computational Biology*. 2017, vol. 13. DOI: 10.1371/journal.pcbi.1005690. ISSN 15537358.

[24] Capra, J. A. and Singh, M. Predicting functionally important residues from sequence conservation. *Bioinformatics*. Oxford University Press. 2007, vol. 23, no. 15, p. 1875–1882.

[25] 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. 2005, vol. 33, suppl_2, p. W306–W310.

[26] 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 Online Library. 2006, vol. 62, no. 4, p. 1125–1132.

[27] Cooperman, B. S., Baykov, A. A. and Lahti, R. Evolutionary conservation of the active site of soluble inorganic pyrophosphatase. *Trends in Biochemical Sciences*. 1992, vol. 17. DOI: 10.1016/0968-0004(92)90406-Y. ISSN 09680004.

[28] Crick, F. H., Barnett, L., Brenner, S. and Watts Tobin, R. J. General nature of the genetic code for proteins. *Nature*. 1961, vol. 192. DOI: 10.1038/1921227a0. ISSN 00280836.

[29] Crick, F. Central dogma of molecular biology. *Nature*. 1970, vol. 227. DOI: 10.1038/227561a0. ISSN 00280836.

[30] „Crick, F. H. C. "On protein synthesis„. *"Symposia on the society for Experimental biology number XII: The Biological Replication of Macromolecules. „* "Symposia of the Society for Experimental Biology.„. 1958.

[31] Das, R. and Baker, D. Macromolecular modeling with rosetta. *Annu. Rev. Biochem.* Annual Reviews. 2008, vol. 77, p. 363–382.

[32] Dehouck, Y., Gilis, D. and Rooman, M. A new generation of statistical potentials for proteins. *Biophysical Journal*. 2006, vol. 90. DOI: 10.1529/biophysj.105.079434. ISSN 00063495.

[33] 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*. BioMed Central. 2011, vol. 12, no. 1, p. 1–12.

[34] Dekker, J. P., Fodor, A., Aldrich, R. W. and Yellen, G. A perturbation-based method for calculating explicit likelihood of evolutionary co-variance in multiple sequence alignments. *Bioinformatics*. Oxford University Press. 2004, vol. 20, no. 10, p. 1565–1572.

[35] Devlin, J., Chang, M.-W., Lee, K. and Toutanova, K. Bert: Pre-training of deep bidirectional transformers for language understanding. *ArXiv preprint arXiv:1810.04805*. 2018.

[36] Diallo, A. B., Makarenkov, V. and Blanchette, M. Ancestors 1.0: a web server for ancestral sequence reconstruction. *Bioinformatics*. Oxford University Press. 2010, vol. 26, no. 1, p. 130–131.

[37] Do, C. B. and Katoh, K. Protein multiple sequence alignment. *Functional Proteomics*. Springer. 2009, p. 379–413.

[38] Do, C. B., Mahabhashyam, M. S., Brudno, M. and Batzoglou, S. ProbCons: Probabilistic consistency-based multiple sequence alignment. *Genome research*. Cold Spring Harbor Lab. 2005, vol. 15, no. 2, p. 330–340.

[39] Dolgikh, D. A., Gilmanshin, R. I., Brazhnikov, E. V., Bychkova, V. E., Semisotnov, G. V. et al. α-lactalbumin: compact state with fluctuating tertiary structure? *FEBS Letters*. 1981, vol. 136. DOI: 10.1016/0014-5793(81)80642-4. ISSN 00145793.

[40] Dong, G. Q., Fan, H., Schneidman Duhovny, D., Webb, B. and Sali, A. Optimized atomic statistical potentials: assessment of protein interfaces and loops. *Bioinformatics*. Oxford University Press. 2013, vol. 29, no. 24, p. 3158–3166.

[41] Eddy, S. R. A new generation of homology search tools based on probabilistic inference. In: *Genome Informatics 2009: Genome Informatics Series Vol. 23*. World Scientific, 2009, p. 205–211.

[42] Edgar, R. C. MUSCLE: a multiple sequence alignment method with reduced time and space complexity. *BMC bioinformatics*. BioMed Central. 2004, vol. 5, no. 1, p. 1–19.

[43] Edgar, R. C. MUSCLE: multiple sequence alignment with high accuracy and high throughput. *Nucleic acids research*. Oxford University Press. 2004, vol. 32, no. 5, p. 1792–1797.

[44] Edgar, R. C. and Batzoglou, S. Multiple sequence alignment. *Current opinion in structural biology*. Elsevier. 2006, vol. 16, no. 3, p. 368–373.

[45] Eisenberg, D. and Mclachlan, A. D. Solvation energy in protein folding and binding. *Nature*. 1986, vol. 319. DOI: 10.1038/319199a0. ISSN 00280836.

[46] Finn, R. D., Clements, J. and Eddy, S. R. HMMER web server: interactive sequence similarity searching. *Nucleic acids research*. Oxford University Press. 2011, vol. 39, suppl_2, p. W29–W37.

[47] Fiser, A., Do, R. K. G. et al. Modeling of loops in protein structures. *Protein science*. Cambridge University Press. 2000, vol. 9, no. 9, p. 1753–1773.

[48] Floor, R. J., Wijma, H. J., Colpa, D. I., Ramos Silva, A., Jekel, P. A. et al. Computational library design for increasing haloalkane dehalogenase stability. *ChemBioChem*. Wiley Online Library. 2014, vol. 15, no. 11, p. 1660–1672.

[49] 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*. 2016, vol. 428. DOI: 10.1016/j.jmb.2016.01.012. ISSN 10898638.

[50] 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. 2016, vol. 428, no. 6, p. 1394–1405.

[51] Fukuchi, S. and Nishikawa, K. Protein surface amino acid compositions distinctively differ between thermophilic and mesophilic bacteria. *Journal of molecular biology*. Elsevier. 2001, vol. 309, no. 4, p. 835–843.

[52] Gao, D., Narasimhan, D., Macdonald, J., Brim, R., Ko, M.-C. et al. Thermostable Variants of Cocaine Esterase for Long-Time Protection against Cocaine Toxicity. *Molecular pharmacology*. december 2008, vol. 75, p. 318–23. DOI: 10.1124/mol.108.049486.

[53] Godzik, A. Fold recognition methods. *Methods of biochemical analysis*. 2003, vol. 44, p. 525–546.

[54] Godzik, A. and Skolnick, J. Sequence-structure matching in globular proteins: application to supersecondary and tertiary structure determination. *Proceedings of the National Academy of Sciences*. National Acad Sciences. 1992, vol. 89, no. 24, p. 12098–12102.

[55] 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. 2016, vol. 63, no. 2, p. 337–346.

[56] 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*. 2016, vol. 63. DOI: 10.1016/j.molcel.2016.06.012. ISSN 10974164.

[57] Greer, J. Comparative modeling methods: application to the family of the mammalian serine proteases. *Proteins: Structure, Function, and Bioinformatics*. Wiley Online Library. 1990, vol. 7, no. 4, p. 317–334.

[58] Greiner W., S. H. *Thermodynamics And Statistical Mechanics*. Springer-Verlag, 1995. ISBN 0-387-94299-08.

[59] Gromiha“, M. M. *„Protein Bioinformatics From Sequence to Function“*. „1“th ed. „Elsevier“, „2010“. ISBN „978-81-312-2297-3“.

[60] Gromiha, M. M., Oobatake, M. and Sarai, A. Important amino acid properties for enhanced thermostability from mesophilic to thermophilic proteins. *Biophysical chemistry*. Elsevier. 1999, vol. 82, no. 1, p. 51–67.

[61] Gronau, I. and Moran, S. Optimal implementations of UPGMA and other common clustering algorithms. *Information Processing Letters*. Elsevier. 2007, vol. 104, no. 6, p. 205–210.

[62] 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*. 2002, vol. 320. DOI: 10.1016/S0022-2836(02)00442-4. ISSN 00222836.

[63] 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. 2002, vol. 320, no. 2, p. 369–387.

[64] Göbel, U., Sander, C., Schneider, R. and Valencia, A. Correlated mutations and residue contacts in proteins. *Proteins: Structure, Function, and Bioinformatics*. 1994, vol. 18. DOI: 10.1002/prot.340180402. ISSN 10970134.

[65] Haas, J., Roth, S., Arnold, K., Kiefer, F., Schmidt, T. et al. The protein model portal - A comprehensive resource for protein structure and model information. *Database*. 2013, vol. 2013. DOI: 10.1093/database/bat031. ISSN 17580463.

[66] Hanson Smith, V., Kolaczkowski, B. and Thornton, J. W. Robustness of ancestral sequence reconstruction to phylogenetic uncertainty. *Molecular biology and evolution*. Oxford University Press. 2010, vol. 27, no. 9, p. 1988–1999.

[67] Havel, T. F. and Snow, M. E. A new method for building protein conformations from sequence alignments with homologues of known structure. *Journal of molecular biology*. Elsevier. 1991, vol. 217, no. 1, p. 1–7.

[68] Henikoff, S. and Henikoff, J. G. Amino acid substitution matrices from protein blocks. *Proceedings of the National Academy of Sciences*. National Acad Sciences. 1992, vol. 89, no. 22, p. 10915–10919.

[69] Hochberg, G. K. and Thornton, J. W. Reconstructing Ancient Proteins to Understand the Causes of Structure and Function. *Annual Review of Biophysics*. 2017, vol. 46. DOI: 10.1146/annurev-biophys-070816-033631. ISSN 19361238.

[70] Holman, C. M. Protein Similarity Score: A Simplified Version of the Blast Score as a Superior Alternative to Percent Identity for Claiming Genuses of Related Protein Sequences. *Santa Clara Computer & High Technology Law Journal*. 2004.

[71] Hon, J., Borko, S., Stourac, J., Prokop, Z., Zendulka, J. et al. EnzymeMiner: automated mining of soluble enzymes with diverse structures, catalytic properties and stabilities. *Nucleic acids research*. Oxford University Press. 2020, vol. 48, W1, p. W104–W109.

[72] Howell, N. Evolutionary conservation of protein regions in the protonmotive cytochrome b and their possible roles in redox catalysis. *Journal of Molecular Evolution*. 1989, vol. 29. DOI: 10.1007/BF02100114. ISSN 00222844.

[73] 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*. 2007, vol. 23. DOI: 10.1093/bioinformatics/btm100. ISSN 13674803.

[74] Itzhaki, L. S., Otzen, D. E. and Fersht, A. R. The structure of the transition state for folding of chymotrypsin inhibitor 2 analysed by protein engineering methods: Evidence for a nucleation-condensation mechanism for protein folding. *Journal of Molecular Biology*. 1995, vol. 254. DOI: 10.1006/jmbi.1995.0616. ISSN 00222836.

[75] Jeanmougin, F., Thompson, J. D., Gouy, M., Higgins, D. G. and Gibson, T. J. Multiple sequence alignment with Clustal X. *Trends in biochemical sciences*. Elsevier. 1998, vol. 23, no. 10, p. 403–405.

[76] Jones, D. T., Taylor, W. R. and Thornton, J. M. The rapid generation of mutation data matrices from protein sequences. *Bioinformatics*. Oxford University Press. 1992, vol. 8, no. 3, p. 275–282.

[77] Jones, D. T., Taylort, W. and Thornton, J. M. A new approach to protein fold recognition. *Nature*. Nature Publishing Group. 1992, vol. 358, no. 6381, p. 86–89.

[78] Jones, T. A. and Thirup, S. Using known substructures in protein model building and crystallography. *The EMBO journal*. 1986, vol. 5, no. 4, p. 819–822.

[79] Jumper, J., Evans, R., Pritzel, A., Green, T., Figurnov, M. et al. Highly accurate protein structure prediction with AlphaFold. *Nature*. Nature Publishing Group. 2021, vol. 596, no. 7873, p. 583–589.

[80] Jumper, J., Evans, R., Pritzel, A., Green, T., Figurnov, M. et al. Highly accurate protein structure prediction with AlphaFold. *Nature*. 2021, vol. 596. DOI: 10.1038/s41586-021-03819-2. ISSN 14764687.

[81] Jäckel, C., Bloom, J. D., Kast, P., Arnold, F. H. and Hilvert, D. Consensus protein design without phylogenetic bias. *Journal of Molecular Biology*. 2010, vol. 399. DOI: 10.1016/j.jmb.2010.04.039. ISSN 00222836.

[82] Jäckel, C., Bloom, J. D., Kast, P., Arnold, F. H. and Hilvert, D. Consensus protein design without phylogenetic bias. *Journal of Molecular Biology*. 2010, vol. 399. DOI: 10.1016/j.jmb.2010.04.039. ISSN 00222836.

[83] Karplus, M. and Weaver, D. L. Protein-folding dynamics. *Nature*. 1976, vol. 260. DOI: 10.1038/260404a0. ISSN 00280836.

[84] Katoh, K., Kuma, K.-i., Toh, H. and Miyata, T. MAFFT version 5: improvement in accuracy of multiple sequence alignment. *Nucleic acids research*. Oxford University Press. 2005, vol. 33, no. 2, p. 511–518.

[85] Katoh, K., Misawa, K., Kuma, K.-i. and Miyata, T. MAFFT: a novel method for rapid multiple sequence alignment based on fast Fourier transform. *Nucleic acids research*. Oxford University Press. 2002, vol. 30, no. 14, p. 3059–3066.

[86] Koehl, P. and Levitt, M. A brighter future for protein structure prediction. In:. 1999, vol. 6. DOI: 10.1038/5794. ISSN 10728368.

[87] Korber, B. T., Farber, R. M., Wolpert, D. H. and Lapedes, A. S. Covariation of mutations in the V3 loop of human immunodeficiency virus type 1 envelope protein: an information theoretic analysis. *Proceedings of the National Academy of Sciences*. National Acad Sciences. 1993, vol. 90, no. 15, p. 7176–7180.

[88] Kryshtafovych, A., Moult, J., Billings, W. M., Della Corte, D., Fidelis, K. et al. Modeling SARS-CoV-2 proteins in the CASP-commons experiment. *Proteins: Structure, Function, and Bioinformatics*. Wiley Online Library. 2021, vol. 89, no. 12, p. 1987–1996.

[89] Kryshtafovych, A., Schwede, T., Topf, M., Fidelis, K. and Moult, J. Critical assessment of methods of protein structure prediction (CASP)—Round XIII. *Proteins: Structure, Function, and Bioinformatics*. Wiley Online Library. 2019, vol. 87, no. 12, p. 1011–1020.

[90] Kryshtafovych, A., Schwede, T., Topf, M., Fidelis, K. and Moult, J. Critical assessment of methods of protein structure prediction (CASP)—Round XIV. *Proteins: Structure, Function, and Bioinformatics*. Wiley Online Library. 2021, vol. 89, no. 12, p. 1607–1617.

[91] Lazaridis, T. and Karplus, M. Effective energy functions for protein structure prediction. *Current Opinion in Structural Biology*. 2000, vol. 10. DOI: 10.1016/S0959-440X(00)00063-4. ISSN 0959440X.

[92] Leaver Fay, A., Tyka, M., Lewis, S. M., Lange, O. F., Thompson, J. et al. ROSETTA3: an object-oriented software suite for the simulation and design of macromolecules. In: *Methods in enzymology*. Elsevier, 2011, vol. 487, p. 545–574.

[93] Lee, B.-C. and Kim, D. A new method for revealing correlated mutations under the structural and functional constraints in proteins. *Bioinformatics*. Oxford University Press. 2009, vol. 25, no. 19, p. 2506–2513.

[94] Lee, J., Freddolino, P. L. and Zhang, Y. Ab initio protein structure prediction. In: *From protein structure to function with bioinformatics*. Springer, 2017, p. 3–35.

[95] Levinthal, C. How to fold graciously. *Mössbauer Spectroscopy in Biological Systems Proceedings*. 1969, vol. 24.

[96] Levitt, M. Accurate modeling of protein conformation by automatic segment matching. *Journal of molecular biology*. Elsevier. 1992, vol. 226, no. 2, p. 507–533.

[97] Li, Y. and Fang, J. PROTS-RF: a robust model for predicting mutation-induced protein stability changes. Public Library of Science San Francisco, USA. 2012.

[98] Lipman, D. J. and Pearson, W. R. Rapid and sensitive protein similarity searches. *Science*. 1985, vol. 227. DOI: 10.1126/science.2983426. ISSN 00368075.

[99] Lockless, S. W. and Ranganathan, R. Evolutionarily conserved pathways of energetic connectivity in protein families. *Science*. American Association for the Advancement of Science. 1999, vol. 286, no. 5438, p. 295–299.

[100] Ma, B., Tromp, J. and Li, M. PatternHunter: faster and more sensitive homology search. *Bioinformatics*. Oxford University Press. 2002, vol. 18, no. 3, p. 440–445.

[101] Magliery, T. J. Protein stability: computation, sequence statistics, and new experimental methods. *Current opinion in structural biology*. Elsevier. 2015, vol. 33, p. 161–168.

[102] Martí Renom, M. A., Stuart, A. C., Fiser, A., Sánchez, R., Melo, F. et al. Comparative protein structure modeling of genes and genomes. *Annual review of biophysics and biomolecular structure*. Annual Reviews 4139 El Camino Way, PO Box 10139, Palo Alto, CA 94303-0139, USA. 2000, vol. 29, no. 1, p. 291–325.

[103] Menardo, F., Loiseau, C., Brites, D., Coscolla, M., Gygli, S. M. et al. Treemmer: a tool to reduce large phylogenetic datasets with minimal loss of diversity. *BMC bioinformatics*. Springer. 2018, vol. 19, no. 1, p. 1–8.

[104] Mendes, J., Guerois, R. and Serrano, L. Energy estimation in protein design. *Current Opinion in Structural Biology*. 2002, vol. 12. DOI: 10.1016/S0959-440X(02)00345-7. ISSN 0959440X.

[105] Mezei, M. Chameleon sequences in the PDB. *Protein engineering*. 1998, vol. 11, no. 6, p. 411–414.

[106] Mirarab, S., Nguyen, N., Guo, S., Wang, L.-S., Kim, J. et al. PASTA: ultra-large multiple sequence alignment for nucleotide and amino-acid sequences. *Journal of Computational Biology*. Mary Ann Liebert, Inc. 140 Huguenot Street, 3rd Floor New Rochelle, NY 10801 USA. 2015, vol. 22, no. 5, p. 377–386.

[107] Mitchell, A. L., Almeida, A., Beracochea, M., Boland, M., Burgin, J. et al. MGnify: the microbiome analysis resource in 2020. *Nucleic Acids Research*. november 2019, vol. 48, D1, p. D570–D578. DOI: 10.1093/nar/gkz1035. ISSN 0305-1048. Available at: https://doi.org/10.1093/nar/gkz1035.

[108] Moon, J. W. and Moser, L. On cliques in graphs. *Israel journal of Mathematics*. Springer. 1965, vol. 3, no. 1, p. 23–28.

[109] Moult, J., Fidelis, K., Kryshtafovych, A., Schwede, T. and Tramontano, A. Critical assessment of methods of protein structure prediction (CASP)—round x. *Proteins: Structure, Function, and Bioinformatics*. Wiley Online Library. 2014, vol. 82, p. 1–6.

[110] Mount, D. W. Comparison of the PAM and BLOSUM amino acid substitution matrices. *Cold Spring Harbor Protocols*. Cold Spring Harbor Laboratory Press. 2008, vol. 2008, no. 6, p. pdb–ip59.

[111] 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*. 2021, vol. 22. DOI: 10.1093/bib/bbaa337. ISSN 14774054.

[112] Musil, M., Stourac, J., Bendl, J., Brezovsky, J., Prokop, Z. et al. FireProt: web server for automated design of thermostable proteins. *Nucleic Acids Research*. 2017, vol. 45. DOI: 10.1093/nar/gkx285. ISSN 13624962.

[113] Needleman, S. B. and Wunsch, C. D. A general method applicable to the search for similarities in the amino acid sequence of two proteins. *Journal of Molecular Biology*. 1970, vol. 48. DOI: 10.1016/0022-2836(70)90057-4. ISSN 00222836.

[114] Nelson, D. L. and Cox, M. M. *Lehninger Principles of Biochemistry, Fourth Edition*. Fourth Editionth ed. 2004.

[115] Nguyen, L.-T., Schmidt, H. A., Von Haeseler, A. and Minh, B. Q. IQ-TREE: a fast and effective stochastic algorithm for estimating maximum-likelihood phylogenies. *Molecular biology and evolution*. Oxford University Press. 2015, vol. 32, no. 1, p. 268–274.

[116] Nickson, A. A. and Clarke, J. What lessons can be learned from studying the folding of homologous proteins? *Methods*. 2010, vol. 52. DOI: 10.1016/j.ymeth.2010.06.003. ISSN 10462023.

[117] Notredame, C., Higgins, D. G. and Heringa, J. T-Coffee: A novel method for fast and accurate multiple sequence alignment. *Journal of molecular biology*. Elsevier. 2000, vol. 302, no. 1, p. 205–217.

[118] Pace, C. N., Scholtz, J. M. and Grimsley, G. R. Forces stabilizing proteins. *FEBS Letters*. 2014, vol. 588. DOI: 10.1016/j.febslet.2014.05.006. ISSN 18733468.

[119] Pantoliano, M. W., Whitlow, M., Wood, J. F., Dodd, S. W., Hardman, K. D. et al. Large Increases in General Stability for Subtilisin BPN' through Incremental Changes in the Free Energy of Unfolding. *Biochemistry*. 1989, vol. 28. DOI: 10.1021/bi00444a012. ISSN 15204995.

[120] Pearson, W. R. An introduction to sequence similarity („homology") searching. *Current Protocols in Bioinformatics*. 2013. DOI: 10.1002/0471250953.bi0301s42. ISSN 19343396.

[121] Peitsch, M. C. ProMod and Swiss-model: Internet-based tools for automated comparative protein modelling. In:. 1996, vol. 24. DOI: 10.1042/bst0240274. ISSN 03005127.

[122] Peng, J. and Xu, J. Low-homology protein threading. *Bioinformatics*. Oxford University Press. 2010, vol. 26, no. 12, p. i294–i300.

[123] Polizzi, K. M., Bommarius, A. S., Broering, J. M. and Chaparro Riggers, J. F. Stability of biocatalysts. *Current Opinion in Chemical Biology*. 2007, vol. 11, no. 2, p. 220–225. DOI: https://doi.org/10.1016/j.cbpa.2007.01.685. ISSN 1367-5931. Bioinorganic chemistry / Biocatalysis and biotransformation. Available at: https://www.sciencedirect.com/science/article/pii/S1367593107000221.

[124] Ponnuswamy, P. K. and Gromiha, M. M. On the conformational stability of folded proteins. *Journal of Theoretical Biology*. 1994, vol. 166. DOI: 10.1006/jtbi.1994.1005. ISSN 00225193.

[125] Porebski, B. T. and Buckle, A. M. Consensus protein design. *Protein Engineering, Design and Selection*. 2016, vol. 29. DOI: 10.1093/protein/gzw015. ISSN 17410134.

[126] 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. 2018, vol. 34, no. 21, p. 3659–3665.

[127] Pucci, F., Bourgeas, R. and Rooman, M. Predicting protein thermal stability changes upon point mutations using statistical potentials: Introducing HoTMuSiC. *Scientific Reports*. 2016, vol. 6. DOI: 10.1038/srep23257. ISSN 20452322.

[128] Reeck, G. R., Haën, C. de, Teller, D. C., Doolittle, R. F., Fitch, W. M. et al. „Homology" in proteins and nucleic acids: A terminology muddle and a way out of it. *Cell*. 1987, vol. 50. DOI: 10.1016/0092-8674(87)90322-9. ISSN 00928674.

[129] Reetz, M. T., Carballeira, J. D. and Vogel, A. Iterative saturation mutagenesis on the basis of B factors as a strategy for increasing protein thermostability. *Angewandte Chemie*. Wiley Online Library. 2006, vol. 118, no. 46, p. 7909–7915.

[130] Ribeiro, A. J. M., Holliday, G. L., Furnham, N., Tyzack, J. D., Ferris, K. et al. Mechanism and Catalytic Site Atlas (M-CSA): a database of enzyme reaction mechanisms and active sites. *Nucleic acids research*. Oxford University Press. 2018, vol. 46, D1, p. D618–D623.

[131] Ronquist, F., Teslenko, M., Van Der Mark, P., Ayres, D. L., Darling, A. et al. MrBayes 3.2: efficient Bayesian phylogenetic inference and model choice across a large model space. *Systematic biology*. Oxford University Press. 2012, vol. 61, no. 3, p. 539–542.

[132] Saier, M. H. Understanding the genetic code. *Journal of Bacteriology*. 2021, vol. 201. DOI: 10.1128/JB.00091-19. ISSN 10985530.

[133] Šali, A. and Blundell, T. L. Comparative protein modelling by satisfaction of spatial restraints. *Journal of molecular biology*. Elsevier. 1993, vol. 234, no. 3, p. 779–815.

[134] Schwede, T., Kopp, J., Guex, N. and Peitsch, M. C. SWISS-MODEL: An automated protein homology-modeling server. *Nucleic Acids Research*. 2003, vol. 31. DOI: 10.1093/nar/gkg520. ISSN 03051048.

[135] 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. 2005, vol. 33, suppl_2, p. W382–W388.

[136] Seeliger, D. and Groot, B. L. de. Protein thermostability calculations using alchemical free energy simulations. *Biophysical journal*. 2010, vol. 98. DOI: 10.1016/j.bpj.2010.01.051. ISSN 15420086.

[137] Sievers, F., Wilm, A., Dineen, D., Gibson, T. J., Karplus, K. et al. Fast, scalable generation of high-quality protein multiple sequence alignments using Clustal Omega. *Molecular Systems Biology*. 2011, vol. 7. DOI: 10.1038/msb.2011.75. ISSN 17444292.

[138] Smith, T. F. and Waterman, M. S. Identification of common molecular subsequences. *Journal of Molecular Biology*. 1981, vol. 147. DOI: 10.1016/0022-2836(81)90087-5. ISSN 00222836.

[139] Stamatakis, A. RAxML-VI-HPC: maximum likelihood-based phylogenetic analyses with thousands of taxa and mixed models. *Bioinformatics*. Oxford University Press. 2006, vol. 22, no. 21, p. 2688–2690.

[140] Stamatakis, A. RAxML version 8: a tool for phylogenetic analysis and post-analysis of large phylogenies. *Bioinformatics*. Oxford University Press. 2014, vol. 30, no. 9, p. 1312–1313.

[141] Stourac, J., Dubrava, J., Musil, M., Horackova, J., Damborsky, J. et al. FireProtDB: Database of manually curated protein stability data. *Nucleic Acids Research*. 2021, vol. 49. DOI: 10.1093/nar/gkaa981. ISSN 13624962.

[142] Sutcliffe, M. J., Haneef, I., Carney, D. and Blundell, T. Knowledge based modelling of homologous proteins, Part I: Three-dimensional frameworks derived

from the simultaneous superposition of multiple structures. *Protein Engineering, Design and Selection*. Oxford University Press. 1987, vol. 1, no. 5, p. 377–384.

[143] Taylor, W. R. and Hatrick, K. Compensating changes in protein multiple sequence alignments. *Protein Engineering, Design and Selection*. 1994, vol. 7. DOI: 10.1093/protein/7.3.341. ISSN 17410126.

[144] Teng, S., Srivastava, A. K. and Wang, L. Sequence feature-based prediction of protein stability changes upon amino acid substitutions. *BMC Genomics*. 2010, vol. 11. DOI: 10.1186/1471-2164-11-S2-S5. ISSN 14712164.

[145] Thompson, J. D., Higgins, D. G. and Gibson, T. J. CLUSTAL W: improving the sensitivity of progressive multiple sequence alignment through sequence weighting, position-specific gap penalties and weight matrix choice. *Nucleic acids research*. Oxford university press. 1994, vol. 22, no. 22, p. 4673–4680.

[146] Unger, R., Harel, D., Wherland, S. and Sussman, J. L. A 3D building blocks approach to analyzing and predicting structure of proteins. *Proteins: Structure, Function, and Bioinformatics*. Wiley Online Library. 1989, vol. 5, no. 4, p. 355–373.

[147] Usmanova, D. R., Bogatyreva, N. S., Ariño Bernad, J., 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. 2018, vol. 34, no. 21, p. 3653–3658.

[148] Valencia, A. Multiple sequence alignments as tools for protein structure and function prediction. *Comparative and functional genomics*. Wiley Online Library. 2003, vol. 4, no. 4, p. 424–427.

[149] 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. 2011, vol. 27, no. 23, p. 3286–3292.

[150] Webb, B. and Sali, A. Comparative protein structure modeling using MODELLER. *Current Protocols in Bioinformatics*. 2016, vol. 2016. DOI: 10.1002/cpbi.3. ISSN 1934340X.

[151] Weigt, M., White, R. A., Szurmant, H., Hoch, J. A. and Hwa, T. Identification of direct residue contacts in protein–protein interaction by message passing. *Proceedings of the National Academy of Sciences*. National Acad Sciences. 2009, vol. 106, no. 1, p. 67–72.

[152] Wells, J. A. Additivity of Mutational Effects in Proteins. *Biochemistry*. 1990, vol. 29. DOI: 10.1021/bi00489a001. ISSN 15204995.

[153] Westesson, O., Barquist, L. and Holmes, I. HandAlign: Bayesian multiple sequence alignment, phylogeny and ancestral reconstruction. *Bioinformatics*. Oxford University Press. 2012, vol. 28, no. 8, p. 1170–1171.

[154] Wetlaufer, D. B. Nucleation, rapid folding, and globular intrachain regions in proteins. *Proceedings of the National Academy of Sciences of the United States of America*. 1973, vol. 70. DOI: 10.1073/pnas.70.3.697. ISSN 00278424.

[155] Wheeler, L. C., Lim, S. A., Marqusee, S. and Harms, M. J. The thermostability and specificity of ancient proteins. *Current Opinion in Structural Biology*. 2016, vol. 38. DOI: 10.1016/j.sbi.2016.05.015. ISSN 1879033X.

[156] Whitford, D. *Proteins: structure and function*. John Wiley & Sons, 2013.

[157] 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. 2014, vol. 27, no. 2, p. 49–58.

[158] Wintrode, P. L. and Arnold, F. H. Temperature adaptation of enzymes: Lessons from laboratory evolution. *Advances in Protein Chemistry*. 2001, vol. 55. DOI: 10.1016/s0065-3233(01)55004-4. ISSN 00653233.

[159] Xu, D. and Zhang, Y. Ab initio protein structure assembly using continuous structure fragments and optimized knowledge-based force field. *Proteins: Structure, Function, and Bioinformatics*. Wiley Online Library. 2012, vol. 80, no. 7, p. 1715–1735.

[160] Yanofsky, C. Establishing the Triplet Nature of the Genetic Code. *Cell*. 2007, vol. 128. DOI: 10.1016/j.cell.2007.02.029. ISSN 00928674.

[161] Zeymer, C. and Hilvert, D. Directed Evolution of Protein Catalysts. *Annual Review of Biochemistry*. 2018, vol. 87. DOI: 10.1146/annurev-biochem-062917-012034. ISSN 15454509.

[162] Zwanzig, R., Szabo, A. and Bagchi, B. Levinthal's paradox. *Proceedings of the National Academy of Sciences of the United States of America*. 1992, vol. 89. DOI: 10.1073/pnas.89.1.20. ISSN 00278424.

Appendix A

Contents of the enclosed DVD

Included DVD contains:

- **fireprot** directory containing source code for the FireProt tool
- **fireprotweb-asr-api** directory containing source code for the server designed for integrating the FireProt ASR into the FireProt tool
- **thesis** directory containing this thesis and its source codes
- **results_4OEE** directory containing the complete results for the run of FireProt on protein 4OEE with both high and low risk definition

Appendix B

Example of resulting multi mutants

BTC mutant: 4 mutations					
Chain	**Mutation**	**B-factor**	**Majority**	**Ratio**	**FoldX**
A	S89E	HIGH	✓	✗	-0.32
A	L72Y	HIGH	✓	✓	0.25
A	R109K	HIGH	✓	✗	-0.19
A	E47V	MOD.	✓	✗	-0.70

Table B.1: Back to consensus multi mutant for protein 4OEE. FoldX and Rosetta values are in kcal/mol.

ENERGY HIGH RISK mutant: 9 mutations								
Chain	**Mutation**	**B-factor**	**High-risk**	**EIA**	**Conserved**	**Correlated**	**FoldX**	**Rosetta**
A	R96M	MOD.	✓	✗	✓	✗	-1.06	-2.33
A	K124P	HIGH	✓	✗	✓	✗	-1.62	-4.91
A	K108T	MOD.	✓	✗	✓	✗	-1.26	-2.94
A	E67P	HIGH	✓	✗	✓	✗	-1.84	-2.93
A	Q43I	HIGH	✓	✗	✓	✗	-1.48	-2.92
A	E88M	HIGH	✓	✗	✓	✗	-1.05	-2.04
A	A46L	MOD.	✓	✗	✓	✗	-1.04	-2.22
A	S74Y	MOD.	✓	✗	✓	✗	-1.04	-2.11
A	K15L	HIGH	✓	✗	✓	✗	-1.13	-2.04

Table B.2: High risk energy approach multi mutant for protein 4OEE. FoldX values are in kcal/mol.

COMBINED HIGH RISK mutant: 13 mutations										
Chain	**Mutation**	**B-factor**	**High-risk**	**EIA**	**Majority**	**Ratio**	**Conserved**	**Correlated**	**FoldX**	**Rosetta**
A	R96M	MOD.	✓	✗			✓	✗	-1.06	-2.33
A	S89E	HIGH	✓	✓	✓	✗	✓	✗	-0.32	
A	L72Y	HIGH	✗	✓	✓	✓	✓	✗	0.25	
A	R109K	HIGH	✗	✓	✓	✗	✓	✗	-0.19	
A	Q43I	HIGH	✓	✗			✓	✗	-1.48	-2.92
A	K15L	HIGH	✓	✗			✓	✗	-1.13	-2.04
A	K124P	HIGH	✓	✗			✓	✗	-1.62	-4.91
A	K108T	MOD.	✓	✗			✓	✗	-1.26	-2.94
A	E67P	HIGH	✓	✗			✓	✗	-1.84	-2.93
A	E88M	HIGH	✓	✗			✓	✗	-1.05	-2.04
A	A46L	MOD.	✓	✗			✓	✗	-1.04	-2.22
A	E47V	MOD.	✓	✓	✓	✗	✓	✗	-0.70	
A	S74Y	MOD.	✓	✗			✓	✗	-1.04	-2.11

Table B.3: Multi mutant which is a result of combination of high risk evolutionary approach and back to consensus approach for protein 4OEE. FoldX and Rosetta values are in kcal/mol.