VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
FACULTY OF INFORMATION TECHNOLOGY

ÚSTAV INFORMAČNÍCH SYSTÉMŮ
DEPARTMENT OF INFORMATION SYSTEMS

PREDIKTOR VLIVU AMINOKYSELINOVÝCH SUBSTITUCÍ NA STABILITU PROTEINŮ

PREDICTOR OF THE EFFECT OF AMINO ACID SUBSTITUTIONS ON PROTEIN STABILITY

DIPLOMOVÁ PRÁCE
MASTER'S THESIS

AUTOR PRÁCE — **Bc. MICHAL FLAX**
AUTHOR

VEDOUCÍ PRÁCE — **Ing. MILOŠ MUSIL**
SUPERVISOR

BRNO 2017

Zadání diplomové práce/19295/2016/xflaxm00

Vysoké učení technické v Brně - Fakulta informačních technologií

Ústav informačních systémů Akademický rok 2016/2017

Zadání diplomové práce

Řešitel: **Flax Michal, Bc.**
Obor: Bioinformatika a biocomputing
Téma: **Prediktor vlivu aminokyselinových substitucí na stabilitu proteinů**
Predictor of the Effect of Amino Acid Substitutions on Protein Stability
Kategorie: Bioinformatika

Pokyny:

1. Nastudujte existující metody a přístupy pro predikci proteinových stabilit
2. Nastudujte existující metody strojového učení (neuronové sítě, SVM, regrese, ...)
3. Sestavte spolehlivý dataset z databáze ProTherm (případně dalších zdrojů)
4. Navrhněte vhodné evoluční, strukturní, statistické a fyzikálně-chemické parametry
5. Otestujte rozličné metody strojového učení a jejich přesnost při predikci proteinových stabilit
6. Vhodné přístupy natrénujte na datech z ProThermu a sestavte konsensuální prediktor
7. Ověřte přesnost predikce konsensuálního prediktoru

Literatura:

- Laimer, J., et al. MAESTRO - multi agent stability prediction upon point mutations. BMC Bioinformatics, 2015.
- Tian, J. et al. Predicting changes in protein thermostability brought about by single or multi site mutations. BMC Bioinformatics, 2010.
- Capriotti, E. et al. A three-state prediction of single point mutations on protein stability changes. BMC Bioinformatics, 2008.
- Dehouck, Y. et al. PopMuSiC 2.1: a web server for the estimation of protein stability changes upon mutations and sequence optimality. BMC Bioinformatics, 2010.
- Bava, K.A. et al. ProTherm, version 4.0: Thermodynamic database for proteins and mutants. Nucleic Acids Research, 2004.

Při obhajobě semestrální části projektu je požadováno:

- První tři body zadání.

Podrobné závazné pokyny pro vypracování diplomové práce naleznete na adrese http://www.fit.vutbr.cz/info/szz/

Technická zpráva diplomové práce musí obsahovat formulaci cíle, charakteristiku současného stavu, teoretická a odborná východiska řešených problémů a specifikaci etap, které byly vyřešeny v rámci dřívějších projektů (30 až 40% celkového rozsahu technické zprávy).

Student odevzdá v jednom výtisku technickou zprávu a v elektronické podobě zdrojový text technické zprávy, úplnou programovou dokumentaci a zdrojové texty programů. Informace v elektronické podobě budou uloženy na standardním nepřepisovatelném paměťovém médiu (CD-R, DVD-R, apod.), které bude vloženo do písemné zprávy tak, aby nemohlo dojít k jeho ztrátě při běžné manipulaci.

Vedoucí: **Musil Miloš, Ing.**, UIFS FIT VUT
Datum zadání: 1. listopadu 2016
Datum odevzdání: 24. května 2017

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
Fakulta informačních technologií
Ústav informačních systémů
612 66 Brno, Božetěchova 2

L.S.

doc. Dr. Ing. Dušan Kolář
vedoucí ústavu

Abstrakt

Tato práce se zabývá predikcí vlivu aminokyselinových mutací na stabilitu proteinů. Pro predikci jsou v této práci využity rozdílné metody strojového učení. Mutace proteinů jsou klasifikovány na mutace, které zvyšují stabilitu proteinů a na mutace, které snižují stabilitu proteinů. Aplikace také predikuje velikost změny Gibbsovy volné energie po mutaci.

Abstract

This paper deals with prediction of influence of amino acids mutations on protein stability. The prediction is based on different methods of machine learning. Protein mutations are classified as mutations that increase or decrease protein stability. The application also predicts the magnitude of change in Gibbs free energy after the mutation.

Klíčová slova

Stabilita proteinů, mutace, aminokyseliny, strojové učení, klasifikace, regrese, support vector machines, umělé neuronové sítě, náhodný les.

Keywords

Stability of proteins, mutation, amino acids, machine learning, classification, regression, support vector machines, artificial neural networks, random forest.

Citace

FLAX, Michal. *Prediktor vlivu aminokyselinových substitucí na stabilitu proteinů*. Brno, 2017. Diplomová práce. Vysoké učení technické v Brně, Fakulta informačních technologií. Vedoucí práce Musil Miloš.

Prediktor vlivu aminokyselinových substitucí na stabilitu proteinů

Prohlášení

Prohlašuji, že jsem tuto diplomovou práci vypracoval samostatně pod vedením pana Ing. Miloše Musila. Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

.....
Michal Flax
19. května 2017

Poděkování

Rád bych poděkoval Ing. Miloši Musilovi za odborné vedení, přístup, rady a materiály vedoucí k vypracování této práce.

Obsah

Kapitola 1

Úvod

Proteiny představují základní stavební kámen všech živých organismů. Pokud tedy chceme porozumět funkci jednotlivých organismů, je nejdříve nutné pochopit mechanismus funkce a vzniku proteinů. Zjištěné poznatky následně mohou najít uplatnění v různých vědních oborech, jako je například lékařství, genetické inženýrství nebo farmakologie.

V proteinech se mohou v průběhu času objevovat mutace, které jsou způsobeny záměnou jedné nebo více aminokyselin za jiné. Tyto mutace jsou základní mechanismus umožňující evoluci organismů. Protože ale jednotlivé mutace mohou funkci organismu ovlivnit v pozitivním i negativním směru, je důležité pochopit vliv konkrétních mutací. Při provádění experimentů v laboratořích, se experimenty setkávají s problémem, kdy vzhledem k velkému počtu aminokyselin a jejich kombinací, je testovaní všech možných mutací časově a finančně náročné. Je tedy snaha, vyvinout počítačové metody, které budou umět co nejpřesněji odhadovat vliv mutací v proteinu v rozumném čase.

Mezi vlastnosti proteinů, které mohou mutace ovlivnit, patří například genová exprese, stabilita proteinů, schopnost proteinů vázat se na správné molekuly konkrétních látek nebo jejich biologická aktivita.

Cílem práce je navrhnout a vytvořit aplikaci, která bude umožňovat predikovat vliv aminokyselinových substitucí na stabilitu proteinů. K tomuto účelu bude využito strojové učení.

Teoretická část této práce je tvořena čtyřmi kapitolami. Kapitola 2 je věnována stručnému popisu proteinů a jejich zasazení do kontextu molekulární biologie. Kapitola 3 popisuje stabilitu proteinů, její význam, co ji ovlivňuje a způsoby jejího vyjádření a měření. Kapitola 4 obsahuje základní popis strojového učení a popis metod, které byly v této práci použity. Kapitola 5 obsahuje přehled již existujících nástrojů pro predikci proteinových stabilit.

Druhá část práce obsahuje popis implementace konsensuálního prediktoru. V rámci této části bude v jednotlivých kapitolách podrobněji rozebrán popis použitých dat, návrh a implementace konsensuálního prediktoru a poslední kapitola bude věnována testování a vyhodnocení přesnosti vytvořeného konsensuálního prediktoru.

Kapitola 2

Úvod do molekulární biologie

Molekulární biologie je věda, která zkoumá buněčné procesy v organismech na jejich molekulární úrovni. Protože k pochopení vlastností a funkce proteinů je nutné pochopit alespoň základy molekulární biologie, bude v této kapitole uveden stručný souhrn základů molekulární biologie.

Nejprve bude stručně popsána funkce DNA a RNA a jejich vztah k proteinům. Následuje popis aminokyselin a poté bude stručně popsána genová exprese jako mechanismus vzniku proteinů. Závěrečná část kapitoly bude věnována proteinům a jejich mutacím.

2.1 DNA a RNA

Kyselina deoxyribonukleová (DNA) a kyselina ribonukleová (RNA) jsou nukleové kyseliny, které se nachází ve většině organismů na zemi, protože ukládají informace nezbytné k formování buněk a jejich interakcí. Chemickým složením se jedná o biopolymery s primární strukturou ve formě dlouhého řetězce nukleotidů. Jednotlivé nukleotidy jsou složeny z pětiuhlíkaté molekuly monosacharidu (ribóza u RNA a deoxyribóza u DNA), fosfátové skupiny a nukleové dusíkaté báze. Nukleové báze se vyskytují ve formě čtyř typů a to purinových bázi adenin (A) a guanin (G) a pyrimidinových bázi thymin (T) a cytosin (C) (v molekulách RNA je thymin nahrazen uracilem (U)).

2.1.1 DNA

Primární účel DNA v organismech je stabilní a dlouhodobé uchovávání genetické informace. Její molekuly mají formu dvoušroubovice složené ze dvou inverzních řetězců nukleotidů, jejichž sekvence popisuje geny, podle kterých je následně během procesu genové exprese syntetizována RNA a z mRNA dále polypeptidové řetězce, které tvoří proteiny. Obecně se ale soudí, že část DNA kódující proteiny je pouze 1.5% celkové DNA a zbylá DNA slouží jako informace pro tvorbu různých typů nekódující RNA, regulace transkripce a translace a dalších funkcí [22].

2.1.2 RNA

Molekuly RNA obsahují oproti DNA kratší sekvence nukleotidů a ve většině případů se vyskytují pouze ve formě samostatného řetězce, což ovlivňuje jejich životnost. Molekuly RNA v organismech nemají pouze funkci kódování proteinů (mRNA) jako mezikrok mezi DNA a proteiny, ale také například různé funkce při procesu translace mRNA. Jako příklad uveďme

tRNA, která umožňuje připojovat jednotlivé aminokyseliny do syntetizovaného polypeptidové řetězce při procesu translace, nebo rRNA, která přestavuje základní stavební složku ribozomů. Další důležité funkce RNA jsou například regulace genové exprese, katalytické funkce, nebo v případě specifických organismů (viry) nosiče genetické informace namísto DNA.

2.2 Aminokyseliny

Aminokyseliny jsou biologické sloučeniny, které tvoří základní stavební složku proteinů. Jsou složeny z aminové (-NH_2) a karboxylové (-COOH) funkční skupiny. Tyto skupiny jsou doplněny postranním řetězcem, jehož chemické složení je unikátní pro každou aminokyselinu a určuje její funkci a vlastnosti.

Do skupiny aminokyselin patří obecně velké množství molekul, ale většina proteinů je složena pouze z následujících 20 aminokyselin: Glycin (G), Alanin (A), Valin (V), Leucin (L), Isoleucin (I), Fenylalanin (F), Tryptofan (W), Tyrosin (Y), Methionin (M), Cystein (C), Kyselina asparagová (D), Kyselina glutamová (E), Histidin (H), Lysin (K), Arginin (R), Asparagin (N), Glutamin (Q), Threonin (T) a Prolin (P). Vzácně se vyskytují také aminokyseliny Selenocystein, Pyrolysin a N-formylmethionin.

Během procesu translace je na základě sekvence kodonů (trojice nukleotidů) v RNA syntetizován jeden nebo více polypeptidových řetězců, které jsou složeny z aminokyselin a následně jsou zformovány do výsledné konformace proteinu. Tvar konformace proteinu je určen způsobem propojení atomových vazeb mezi aminokyselinami, ze kterých je výsledný protein složený. Aminokyseliny odpovídající jednotlivým kodonům jsou uvedeny v tabulce 2.1. V této tabulce je vidět, že aminokyseliny mohou být kódovány více kodony.

		Druhý nukleotid					
		U	C	A	G		
První nukleotid	U	UUU fenylalanin	UCU serin	UAU tyrosin	UGU cystein	U	Třetí nukleotid
		UUC fenylalanin	UCC serin	UAC tyrosin	UGC cystein	C	
		UUA leucin	UCA serin	UAA stop kodon	UGA stop kodon	A	
		UUG leucin	UCG serin	UAG stop kodon	UGG tryptofan	G	
	C	CUU leucin	CCU prolin	CAU histidin	CGU arginin	U	
		CUC leucin	CCC prolin	CAC histidin	CGC arginin	C	
		CUA leucin	CCA prolin	CAA glutamin	CGA arginin	A	
		CUG leucin	CCG prolin	CAG glutamin	CGG arginin	G	
	A	AUU isoleucin	ACU threonin	AAU asparagin	AGU serin	U	
		AUC isoleucin	ACC threonin	AAC asparagin	AGC serin	C	
		AUA isoleucin	ACA threonin	AAA lysin	AGA arginin	A	
		AUG methionin	ACG threonin	AAG lysin	AGG arginin	G	
	G	GUU valin	GCU alanin	GAU kyselina asparagová	GGU glycin	U	
		GUC valin	GCC alanin	GAC kyselina asparagová	GGC glycin	C	
		GUA valin	GCA alanin	GAA kyselina glutamová	GGA glycin	A	
		GUG valin	GCG alanin	GAG kyselina glutamová	GGG glycin	G	

Obrázek 2.1: Přehled jednotlivých kodonů [41]

Proteinogenní aminokyseliny lze rozdělit na dvě skupiny na základě toho, jakou mají tendenci interagovat s okolními molekulami vody. Hydrofobní (nepolární) aminokyseliny mají tendenci umístit se ve vnitřní části proteinu, aby byl minimalizován kontakt s molekulami vody. Hydrofilní (polární) aminokyseliny se naopak vyskytují na povrchu proteinu, kde mohou navazovat atomové vazby na molekuly vody a ostatních látek.

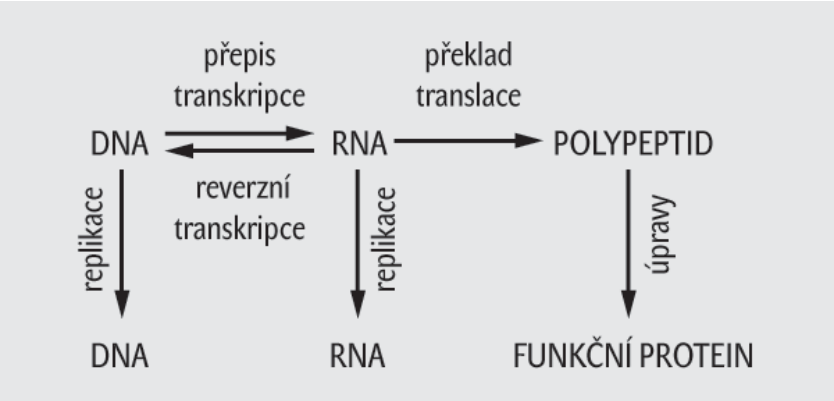
Hydrofobní aminokyseliny se dále rozdělují na skupinu s alifatickým postranním řetězcem, skupinu s aromatickým postranním řetězcem a skupinu jejíchž postranní řetězec obsahuje atomy síry. Výjimku tvoří Glycin, který místo postranního řetězce obsahuje pouze atom vodíku. Hydrofilní aminokyseliny se rozdělují na skupiny s negativním nebo pozitivním nábojem a skupinu bez náboje [19].

Hydrofobní	Atom vodíku	Glycin
	Alifatické	Alanin, Valin, Leucin, Isoleucin
	Aromatické	Fenylalanin, Tryptofan, Tyrosin
	Obsahující síru	Methionin, Cystein
Hydrofilní	Záporný náboj	Kyselina asparagová, Kyselina glutamová
	Kladný náboj	Histidin, Lysin, Arginin
	Bez náboje	Asparagin, Glutamin, Serin, Threonin, Prolin

Tabulka 2.1: Základní rozdělení aminokyselin na základě chemického složení [19]

2.3 Genová exprese

Centrální dogma molekulární biologie tvrdí, že tok informací mezi biopolymery je možný pouze mezi nukleovými kyselinami, nebo nukleovými kyselinami a proteiny. Není tedy možná změna informací obsažených v proteinu a její přenos zpět do genomu [12].



Obrázek 2.2: Přehled jednotlivých kodonů [45]

Genová exprese je proces, který na základě informace uložené v genu vytváří buněčnou strukturu. Jedná se o tok informace ve směru z DNA do RNA v rámci procesu transkripce a následně převod informace z RNA do proteinu v procesu translace. Je vhodné uvést, že existují i geny, které kódují pouze RNA a neprobíhá u nich tedy proces translace na protein. Také je vhodné zmínit, že specifické organismy (retroviry) jsou schopné přepisovat informace z RNA do DNA v procesu označovaném reverzní transkripce.

- **Transkripce** je proces, při kterém dochází k tvorbě RNA na základě sekvence nukleotidů obsažených v DNA. Proces transkripce probíhá tak, že se na dvoušroubovici DNA naváže RNA polymeráza, následně se rozplete dvoušroubovice DNA na samostatné řetězce a vytváří se vlákno RNA komplementární k danému řetězci DNA. Vzniká tak

primární transkript (hnRNA), který dále prochází postranskripčními úpravami a tím vytváří mRNA, která dále podstupuje proces translace. U nekódujících genů vznikají jiné typy RNA, která nepodstupují translaci a v buňce plní další funkce.

U eukaryotických buněk probíhá mezi procesy transkripce a translace sestřih primárního transkriptu. V rámci tohoto procesu jsou z transkribované RNA vystřiženy tzv. introny, které představují části genomu, které se nepřekládají do proteinu a exony, které kódují protein jsou ponechány.

- **Translace** představuje proces, při kterém se vytváří proteiny na základě řetězce mRNA vytvořeného při procesu transkripce. Během translace vzniká postupně polypeptidový řetězec aminokyselin. Pořadí a typ aminokyselin v řetězci je určeno sekvencí nukleotidů v RNA, kdy jednotlivé trojice nukleotidů (tzv. kodony) určují typ aminokyseliny jak je uvedeno v tabulce na obrázku 2.1. Následují postranslační úpravy po kterých vzniká funkční protein. Mezi tyto úpravy patří například odstranění prvního Methioninu na začátku polypeptidového řetězce a signálního peptidu z N-konce polypeptidové řetězce.

2.4 Proteiny

Proteiny nebo-li také bílkoviny jsou makromolekuly složené z jednoho nebo více polypeptidových řetězců, které se skládají z aminokyselin. Sekvence aminokyselin je pro každý protein jedinečná a určuje jeho vlastnosti. Proteiny v organismu zajišťují většinu jeho procesů, jako je např. replikace DNA, reakce organismu na chemické podněty nebo transport molekul příslušných látek.

Funkce proteinů je určena jejich jejich uspořádáním v prostoru (konformací). Toto uspořádání je určeno pořadím a typem aminokyselin v polypeptidových řetězcích a energetických vlastnostech proteinu. Na molekulární úrovni je formováno pomocí atomových vazeb mezi aminokyselinami. Stabilita konformace proteinů a co ji ovlivňuje bude podrobněji rozebráno v kapitole 3.

Proteiny se dělí do rodin proteinů na základě podobnosti struktury a funkce proteinu.

2.4.1 Struktura proteinů

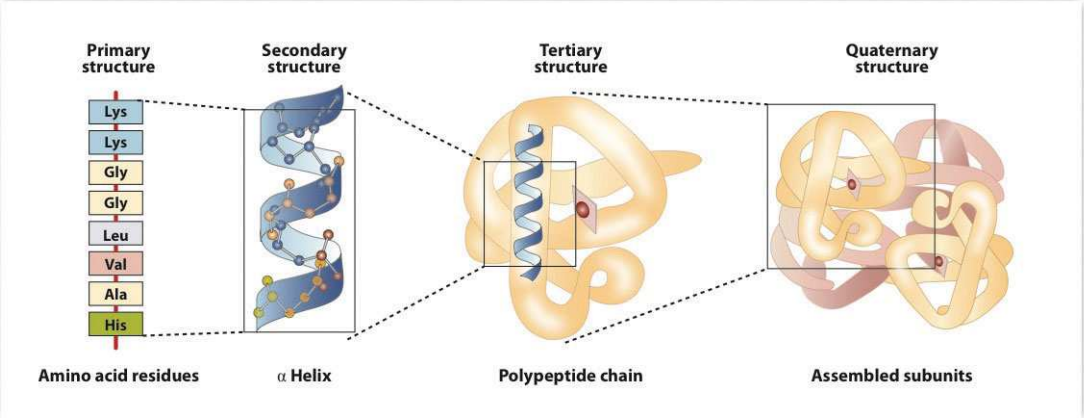
Základní stavební složku proteinu tvoří polypeptidová kostra, na kterou jsou navázány postranní řetězce jednotlivých aminokyselin, které určují funkci proteinu a jeho fyzikální a chemické vlastnosti. Samotná struktura proteinu je mimo jiné určena atomovými vazbami mezi aminokyselinami. Jedná se o vodíkové můstky, iontové vazby a van der Waalsovy síly [7]. Dalším důležitým faktorem je umístění hydrofobních a hydrofilních aminokyselin v rámci struktury proteinu.

Struktura proteinů se popisuje pomocí čtyř úrovní:

- **Primární struktura** je označení pro sekvenci aminokyselin v polypeptidových řetězcích proteinu.
- **Sekundární struktura** popisuje krátké opakující se strukturní vzory, které se tvoří v rámci polypeptidových řetězců a jsou společné pro různé proteiny. Nejčastější vzory jsou α-šroubovice, u které základ pravotočivé šroubovice tvoří polypeptidová kostra

řetězce aminokyselin a postranní řetězce jsou otočeny směrem ven. Další častá struktura je β-skládány list, která je tvořena úseky polypeptidové kostry navázaných rovnoběžně k sobě.

- **Terciální struktura** označuje trojrozměrnou konformaci jednoho polypeptidového řetězce.
- **Kvadrální struktura** popisuje trojrozměrnou konformaci proteinů, které jsou složeny z více polypeptidových řetězců.



Obrázek 2.3: Úrovně struktury proteinů [4]

2.5 Mutace

Mutace je označení pro změnu genetické informace v rámci genomu organismu (existují také epimutace probíhající mimo úroveň genomu v rámci posttranslačních úprav, ale o ty se v této práci nebudeme zajímat). Výsledkem mutace je vetšinou změna posloupnosti aminokyselin ve vytvořeném proteinu, která negativně ovlivňuje funkci organismu, ale existují i mutace s pozitivním vlivem na organismus. Příčina vzniku mutace může být různá. Jedná se například o fyzikální faktory (radioaktivní záření), chemické faktory (změna sloučenin) nebo biologické faktory (infekce).

Mutace se dělí podle způsobu změny genomu:

- **Delece** - ze sekvence genomu je odstraněn jeden nebo více nukleotidů.
- **Inzerce** - do sekvence genomu je vložen jeden nebo více nových nukleotidů.
- **Substituce** - výsledkem mutace je změna nukleotiodu na dané pozici. Výsledkem je většinou změna aminokyseliny, která je kódována kodonem, ve kterém se nachází místo mutace. Může se ale také jednat pouze o tichou mutaci, kdy se kodon změní na jiný kodon, kdy nedochází ke změně aminokyseliny.

Pokud při procesu delece a inzerce není počet smazaných nebo vložených nukleotidů dělitelných třemi, jedná se o tzv. posunové mutace, které mají za následek posun okna

tří nukleotidů pro kódování kodonů a většinou vedou k naprosté ztrátě funkce výsledného proteinu.

Mutace se dále dělí také podle místa mutace a jejich vlivu na DNA a chromozomy.

- **Genová mutace** - je mutace na úrovni DNA. Dělí se na mutace v kódujících oblastech genomu, které mění funkci výsledného proteinu a mutace v nekódujících oblastech genomu, které nemusí způsobit změnu v organismu, pokud tyto nekódující oblasti neobsahují sekvence nukleotidů, které ovlivňují proces genové exprese (promotory, regulační úseky atd.).
- **Genomová mutace** - je označení mutace, která se projevuje změnou počtu chromozomů.
- **Chromozomová mutace** - je mutace, která mění tvar a strukturu chromozomů. Většinou se jedná o mutace se zhoubným vlivem na organismus.

Kapitola 3

Stabilita proteinů

Stabilita proteinů je veličina, která vyjadřuje odolnost proteinů proti denaturaci v nepříznivých podmínkách, jako je například vysoká teplota nebo extrémní hodnoty pH v okolí proteinu. Je to rovnováha sil, která popisuje, jestli se protein bude nacházet v nativním prostorovém uspořádání, nebo jestli bude v denaturovaném stavu. Její velikost je většinou malá, protože vyjadřuje rozdíl dvou sil, které působí proti sobě. Velikost stability je určena množstvím a typem atomových vazeb v proteinu. V nativní konformaci je struktura proteinu stabilizována různými typy atomových interakcí, jako jsou například hydrofobické a elektrostatické interakce, vodíkové vazby, van der Waalsovy síly a disulfidické můstky. V denaturovaném stavu je nejvíce ovlivněna entropickými a neentropickými volnými energiemi [19].

3.1 Atomové interakce ovlivňující stabilitu proteinů

Vybrané atomové interakce a fyzikálně-chemické vlastnosti proteinů, které mají vliv na velikost stability proteinů, jsou uvedeny v následujícím přehledu. Tento přehled byl vytvořen primárně na základě informací, které byly uvedeny v práci [32]. Čím větší množství uvedených interakcí a vazeb se v proteinu nachází, tím je větší stabilita tohoto proteinu.

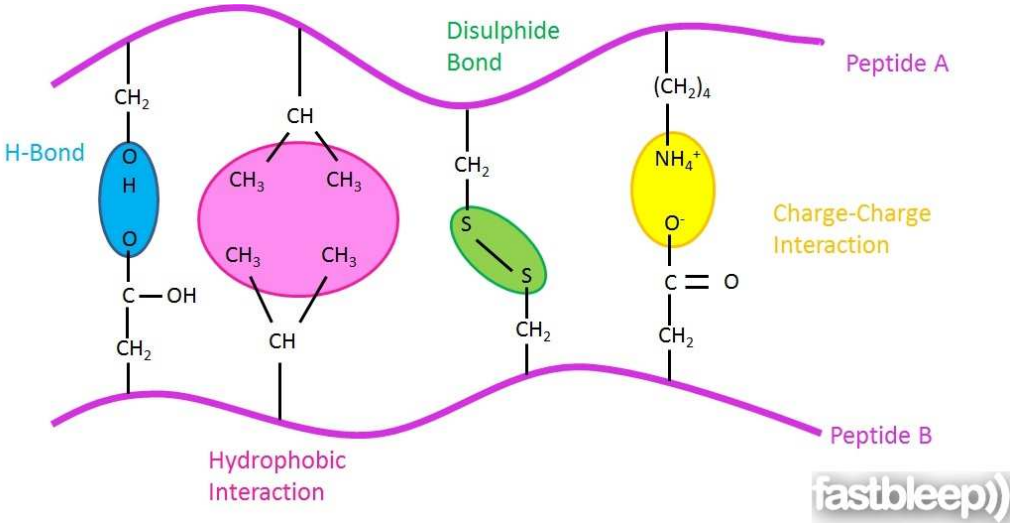
- **Hydrofobicita aminokyselin:** Tato vlastnost je jedna z nejdůležitějších vlastností ovlivňujících stabilitu proteinů a proces jejich sbalení. Aminokyseliny s vysokou hodnotou hydrofobicity (fenylalanin, isoleucin, alanin, valin a další) mají tendenci zaujímat pozici uvnitř proteinu, kde jsou ukryty a chráněny proti roztoku, který protein obklopuje. Naopak nabité a polární aminokyseliny (kyselina aparagová, lysin, arginin atd.) mají tendenci vyskytovat se na povrchu proteinu, kde jejich postranní řetežec může vytvářet atomové interakce s jednotlivými molekulami v okolním roztoku.

- **Vodíkové vazby**: Jedná se o typ atomové interakce, která vzniká, když dva elektronegativní atomy interagují se stejným atomem vodíku. Tento atom vodíku tvoří většinou kovalentní vazbu s jedním z těchto atomů a s druhým atomem vytváří elektrostratickou vazbu. Vodíkové vazby mají klíčovou roli při tvorbě sekundární struktury proteinů, kde jsou vytvářeny mezi jednotlivými atomy v útvarech sekundární struktury proteinu. Těmito útvary jsou například α-šroubovice nebo β-skládáné listy.

- **Konformační entropie rozbalení**: Konformační entropie je typ entropie, která je ve vztahu s počtem možných konformací molekuly. Bylo navrženo, že menší hodnota

konformační flexibility proteinového řetězce v rozbaleném stavu by měla zvyšovat hodnotu stability proteinu, který je sbalený v nativní konformaci [28].

- **Solné můstky**: Tento typ vazby je specialní typ vodíkové vazby, která vzniká mezi dvěmi nabitými aminokyselinami.
- **Iontové vazby:** Jedná se o typ interakcí, které se vytváří mezi dvojící nabitých aminokyselin. Při tvorbě této interakce je jeden nebo více elektronů z prvního atomu přesunuto do druhého atomu a z obou atomů se tak stávají ionty. Vzniklá dvojice iontů je následně k sobě přitahována elektrostatickou silou, protože vzniklé ionty mají opačný náboj.
- **Cation-π interakce**: Jedná se o nekovalentní interakce mezi molekulami, při kterých jsou kationty (např. postranní řetězce aminokyselin lysin a arginin) zarovnány nad střed aromatického prstence jiné molekuly. Mezi aminokyseliny, které obsahují aromatický řetězec, patří fenylalanin, histidin, tyrosin a tryptofan.
- **Disulfidické vazby**: Jedná se o typ vazeb, které se vytváří při procesu oxidace mezi dvojicí cysteinů. Jedná se o kovalentní vazbu mezi dvěma atomy síry.



Obrázek 3.1: Vybrané atomové interakce v proteinech [14].

3.2 Gibbsova volná energie

Gibbsova volná energie je termodynamický potenciál, který vyjadřuje maximální množství reversibilní práce, která může být provedena termonydamickým systémem při konstantním tlaku a teplotě. Je definována vztahem 3.1, ve kterém H je entalpie, T teplota a S entropie [44].

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

Stabilita proteinů je obecně uváděna jako změna Gibbsovy volné energie při procesu skládání proteinu a je označována ΔG. Tato hodnota vyjadřuje rozdíl mezi volnou energií

proteinu sbaleného v nativní konformaci a volnou energií proteinu v denaturovaném stavu. Je tedy definována následujícím vztahem [19].

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

Vliv mutace na stabilitu proteinu, je vyjádřen pomocí rozdílu velikosti ΔG mezi originálním a mutovaným proteinem. Označuje se jako $\Delta\Delta G$ a nazývá se změna Gibbsovy volné energie po mutaci. Je definován vztahem 3.3, kde záporná hodnota $\Delta\Delta G$ vyjadřuje stabilizující mutaci. Nejčastěji používaná fyzikální jednotka pro vyjádření vlivu mutace na stabilitu proteinu je $kCal/mol$.

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

3.3 Metody pro meření stability

V laboratorních podmínkách existují pro měření velikosti stability proteinů různé metody. Jedná se například o diferenční skenovací kalorimetrii, absorbanci, fluorescenci, nukleární magnetickou rezonanci, gelovou filtraci nebo izotermální kalorimetrii. Následuje stručný popis dvou vybraných metod, který byl převzat z publikace [19].

- **Diferenční skenovací kalorimetrie** je jedna z nejčastěji využívaných metod pro studium termodynamiky procesu denaturace proteinu. Tato metoda nevyžaduje vytváření žádných modelů nebo předchozích předpokladů. Princip metody spočívá v měření teplotní kapacity proteinu v roztoku při zvyšující se teplotě.
- **Denaturantem indukované rozbalení** je metoda, která je založena na principu určení rovnovážných dat při procesu rozbalení proteinů v roztoku, který obsahuje denaturant a extrapolaci k nulové koncentraci denaturantu. Velikost stability je následně vypočtena například pomocí lineárně extrapolačního modelu.

Kapitola 4

Strojové učení

Strojové učení je souhrné označení pro počítačové metody, které zpracovávají velké množství reálných dat a hledají vztahy v těchto datech. Protože reálně naměřená data mohou mít velké množství dimenzí a nabývat vysoké složitosti, je manuální hledání jednotlivých vztahů v těchto datech obtížné, ne-li nemožné. Z tohoto důvodu byly vyvinuty metody, které dokáží tento proces zautomatizovat a zefektivnit. Velmi často mají ale také nevýhody, které spočivají například v neschopnosti správně analyzovat nevyvážená data, nemožnosti zpracovat data s velkým množstvím parametrů, požadavku, aby všechny hodnoty měly stejný interval jejich rozsahu nebo potřeby velkého množství experimentálních dat.

V této kapitole bude na začátku uveden stručný úvod do problematiky strojového učení, základní dělení metod na jednotlivé skupiny a jejich účel. Zbytek kapitoly je poté věnován stručnému popisu jednotlivých metod a algoritmů, které byly testovány při vytváření této práce, nebo byly použity v konečné verzi aplikace. Jedná se o metody k-nejblizších sousedů, support vector machines, rozhodovací strom a z ní vycházející metoda náhodný les a neuronové sítě.

4.1 Úvod do strojového učení

Základní typy problémů, pro jejichž řešení se využivají metody strojové učení, je možné rozdělit do dvou skupin. První skupinou je klasifikace, která spočívá v určení, do které z klasifikačních tříd patří klasifikovaný vstupní vektor. Druhým typem problémů je regrese. Při ní nejsou vstupní vektory klasifikovány do jednotlivých tříd, ale jsou k nim vypočteny číselné výstupní hodnoty.

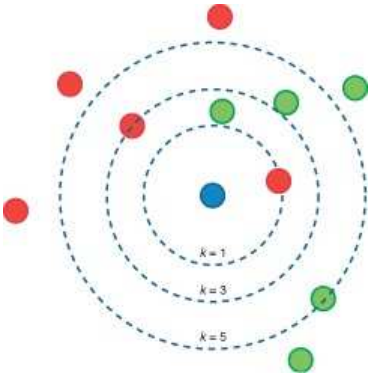
Hlavní přistup pro dělení metod strojového učení je založen na způsobu, jakým se jednotlivé metody učí a nachází vztahy mezi daty.

- **Učení s učitelem**: Metody v této skupině potřebují, aby u dat, které jsou použity pro učení metody, byly uvedeny referenční hodnoty výstupu. Vztahy mezi jednotlivými dvojicemi vstupních a výstupních vektorů jsou následně využity pro správné učení metody. Do této skupiny patří například dopředné neuronové sítě nebo metoda náhodný les.
- **Učení bez učitele**: Při tomto typu učení není nutné znát referenční výstupy. Do této skupiny patří například různé typy metod pro shlukování nebo speciální typy neuronových sítí jako jsou samoorganizující mapy.

4.2 Algoritmus k-nejbližších sousedů

Metoda k-nejbližších sousedů (zkráceně kNN) je jeden z nejjednodušších algoritmů pro klasifikaci. Při učení nejsou prováděny žádné výpočty nebo vytvářeny modely, ale metoda si pouze uloží všechny trénovací vektory, podle kterých bude klasifikovat vstupní vektory.

Proces klasifikace probíhá tak, že pro každý vstupní vektor je nalezeno k nejbližších trénovacích vektorů a vstupní vektor je klasifikován do třídy, která je ve vybraných k vektorech majoritní. Ukázka klasifikace pomocí této metody je uvedena na ilustraci 4.1. Pro jednoduchost předpokládejme, že červené body jsou negativní záznamy a zelené pozitivní. Při hodnotách $k = 1$ a $k = 3$ je vstupní vektor klasifikován jako negativní. Při hodnotě $k = 5$ ale většina z k nejbližších vektorů náleží do pozitivní třídy a vstupní vektor je tedy klasifikován jako pozitivní.



Obrázek 4.1: Ukázka algoritmu k-nejbližších sousedů pro hodnoty $k = 1, 3, 5$ [29].

Volitelnými parametry této metody je počet nejbližších vektorů k a typ vzdálenosti, na základě které jsou určeny nejbližší vektory. Nejčastěji využívaný typ vzdálenosti je Euklidovská metrika, která je definována vztahem 4.1. Je ale možné použít i jiné typy vzdálenosti jako je například Hammingova vzdálenost, Kosinová vzdálenost nebo vzdálenost Manhattan.

$$d(x, y) = \sqrt{\sum_{i=0}^{n}(x_i - y_i)^2} \tag{4.1}$$

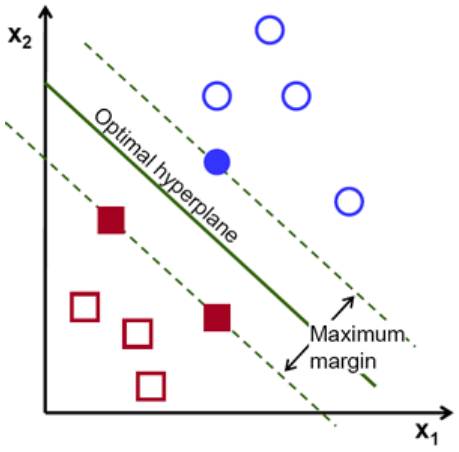
Hlavní nevýhodou této metody je horší schopnost klasifikace dat, u kterých nejsou velikosti klasifikačních tříd rovnoměrně rozloženy. Metoda má u těchto dat tendenci hůře klasifikovat data, které náleží do méně zastoupené klasifikační třídy. Toto chování vyplývá ze způsobu klasifikace u tohoto algoritmu. Protože vstupní vektor je klasifikován do většinové třídy z nejbližších vektorů, může u nevyvážených dat pro větší hodnoty k častěji převažovat chybná klasifikace do více zastoupené třídy.

4.3 Support vector machines

Support vector machines (zkráceně SVM) je metoda, která je založena na principu nalezení nadrovin, které separují jednotlivé klasifikační třídy. Specifickou vlastností této metody je to, že v množině přípustných nadrovin, které úspěšne separují vstupní data, hledá takovou nadrovinu, která je optimální. Ta je vybrána na základě vzdálenosti nejbližších trénovacích

vektorů od této nadroviny. Optimální nadrovina je taková, jejíž vzdálenost od trénovacích vektorů je největší (někdy také označováno jako maximální velikost okolí nadroviny).

Nalezené nadroviny jsou následně reprezentovány váhovým vektorem, jehož prvky tvoří trénovací vektory, které v prohledávaném prostoru leží nejblíže k nadrovinám. Tyto vektory jsou nazývany podpůrné vektory (tzv. support vectors). Ukázka základní lineární verze algoritmu SVM je uvedena na následující ilustraci.



Obrázek 4.2: Ukázka algoritmu SVM s jednou nadrovinou [31].

Nejdříve definujme vztahy, který popisují základní verzi řešeného problému u této metody. Označme trénovací vstupní vektory jako x^t a odpovídající klasifikační třídu pro vstupní vektor x^t jako r^t (předpokládejme značení ± 1). Cílem je nálézt takový vektor w^T a hodnotu w_0 takovou, aby platil následující vztah [1]:

$$r^t(w^T x^t + w_0) \geq 1 \tag{4.2}$$

Důvodem proč ve vztahu 4.2 není uvedeno ≥ 0, ale ≥ 1, je maximalizace velikosti okolí nadroviny, protože větší velikost tohoto okolí zlepšuje schopnost generalizace u této metody.

Pomocí změny hodnoty normy ρ vektoru w můžeme získat nekonečné množství řešení tohoto problému. Definujme tedy $\rho \parallel w \parallel = 1$. Tím umožníme definovat tento problém jako standardní kvadratický optimalizační problém, který je definován následujícím vztahem [1].

$$min \frac{1}{2} \parallel w \parallel^2, \text{ kde } r^t(w^T x^t + w_0) \geq 1 \text{ pro } \forall t \tag{4.3}$$

Pokud trénovací data nejsou lineárně separovatelná, je nutné předchozí vztahy upravit, protože pro neliárně separovatelná data nejsou funkční. Definujme nejdříve proměnou ξ^t, která vyjadřuje velikost odchylky trénovacího vektoru x^t od hranice okolí nadroviny. Následně upravme vztah 4.2 do následujícího tvaru [1].

$$r^t(w^T x^t + w_0) \geq 1 - \xi^t \tag{4.4}$$

Pokud pro trénovací vektor x^t platí, že hodnota $\xi^t = 0$, je vektor klasifikován do správné třídy, pokud platí $0 < \xi^t < 1$, tak je vektor klasifikován korektně, ale leží v okolí nadroviny a pokud platí $\xi^t \geq 1$, je vektor klasifikován do nesprávné třídy. Celková hodnota chyby je následně vyjádřena vztahem $\sum_t \xi^t$. Tato chybová hodnota je přičtena k vztahu 4.3, který

je upraven do následujícího tvaru, ve kterém C označuje trestní faktor [1].

$$min\frac{1}{2} \parallel w \parallel^2 + C\sum_t \xi^t, \text{ kde } r^t(w^T x^t + w_0) \geq 1 \text{ pro } \forall t \tag{4.5}$$

Protože základní verze tohoto algoritmu je vhodná pouze pro lineární data, ale v reálných situacích se data často nachází v nelineárním prostoru, je u tohoto algoritmu často využíván tzv. jádrový trik (anglicky kernel trick). Princip tohoto mechanismu spočívá v mapování vektoru x^t z původního nelineárního d-rozměrného prostoru do nového prostoru, který je k-rozměrný a lineární. Obecně platí, že prostor k může mít mnohem větší rozměr než prostor d [1]. Nad tímto novým lineárním prostorem jsou následně vytvářeny standardní lineární modely. U následujících vztahů budou uvedeny jejich tvary při použití jádrového triku.

Nejdříve definujme bázovou funkci, která mapuje vektor x^t z původního d-rozměrného prostoru na vektor z^t, který leží v novém k-rozměrném prostor [1].

$$z^t = \phi(x^t) \text{ kde } z_j^t = \phi(x_j^t), j = 1, \ldots, k \tag{4.6}$$

Dále definujme diskriminant, ve kterém neuvažujeme samostatnou hodnotu w_0, ale předpokládáme, že platí $z_1 = \phi(x_1) \equiv 1$ [1].

$$g(x) = w^T\phi(x) = \sum_{j=1}^{k} w_j\phi_j(x) \tag{4.7}$$

Definice optimalizačního problému je popsána stejným vztahem jako ve vztahu 4.5, pouze se změní definice jeho ohraničení na následující tvar [1].

$$r^t w^T \phi(x^t) \geq 1 - \xi^t \tag{4.8}$$

Pro zmenšení komplexity problému, je možné převést uvedený optimalizační problém do formy, ve které komplexita problému nezávisí na velikosti rozměru prostoru d, ale na počtu trénovacích vektorů N. Definujme tuto formu problému, jako neohraničený problém, který využívá Lagrangeovy multiplikátory α^t. Parametr μ^t označuje Lagrangeovy parametry, které zajišťují kladnou hodnotu parametru ξ^t [1].

$$L_p = \frac{1}{2} \parallel w \parallel^2 + C\sum_t \xi^t - \sum_t \alpha^t \left[r^t w^T \phi(x^t) - 1 + \xi^t\right] - \sum_t \mu^t \xi^t \tag{4.9}$$

Tento optimalizační problém je potřebné minimalizovat v závislosti na w a w_0 a maximalizovat v závislosti na α^t. Nalezený sedlový bod poté představuje řešení tohoto problému.

Protože tento problém je konvexní a kvadratický, je možné ho řešit pomocí převodu na duální problém. V rámci řešení tohoto duálního problému maximalizujeme hodnotu L_p v závislosti na α^t a s omezením gradientu L_p s ohledem k tomu, že platí, že hodnoty w a w_0 jsou nulové a $\alpha^t \geq 0$. Pokud definujeme potřebné parciální derivace a postavíme je rovny nulové hodnotě, dostaneme následující vztahy [1]:

$$\frac{\partial L_p}{\partial w} = w = \sum_t \alpha^t r^t \phi(x^t) \tag{4.10}$$

$$\frac{\partial L_p}{\partial \xi^t} = C - \alpha^t - \mu^t = 0 \tag{4.11}$$

Po dosazení příslušných parciálních derivací do vztahu 4.8 dostaneme vztah [1]

$$L_d = \sum_t \alpha^t - \frac{1}{2} \sum_t \sum_s \alpha^t \alpha^s r^t r^s \phi(x^t)^T \phi(x^s) \tag{4.12}$$

který je ohraničen limity $\sum_t \alpha^t r^t = 0$ a $0 \leq \alpha^t \leq C, \forall t$. Tento duální problém může být řešen standardními metodami pro kvadratickou optimalizaci. Řešením tohoto problému jsou jednotlivé hodnoty vektoru α^t, kterých je N a odpovídají jednotlivým trénovacím vektorům. Vektorů u kterých platí, že hodnota $\alpha^t = 0$, je většina a představují korektně klasifikované vektory. Vektory s hodnotou $\alpha^t > 0$ leží v okolí nadroviny a jsou označeny a uloženy jako podpůrné vektory. Výsledný vektor w, který reprezentuje diskriminant ve vztahu 4.2 je vypočten jako vážená suma všech vektorů, které byly vybrány jako podpůrné vektory [1].

4.3.1 Jádrové funkce

Zobecněnou formu metody SVM představují metody typu kernel machines. Jejich přístup je založen na nahrazení vnitřního produktu bázových funkcí, $\phi(x^t)^T \phi(x^s)$, jádrovou funkcí $K(x^t, x^s)$, která je definována nad originálním prostorem. Vstupní vektory x^t a x^s tedy nejsou mapovány do nového prostoru, ve kterém je vypočten jejich skalární součin, ale pomocí jádrové funkce je vypočtena jejich vzdálenost přímo v originálním prostoru. Je vhodné uvést, že pro každou jádrovou funkci je možné nalézt odpovídající bázovou funkci pro mapovaní vektorů do nového prostoru. Dosazením jádrové funkce do vztahů pro definici duálního problému 4.12 a pro definici diskriminantu 4.7 dostaneme následující vztahy [1]:

$$L_d = \sum_t \alpha^t - \frac{1}{2} \sum_t \sum_s \alpha^t \alpha^s r^t r^s K(x^t, x^s) \tag{4.13}$$

$$g(x) = w^T \phi(x) = \sum_t \alpha^t r^t \phi(x^t)^T \phi(x) = \alpha^t r^t K(x^t, x) \tag{4.14}$$

Vybrané varianty často používaných jádrových funkcí jsou následující [1]:

- Polynomiální jádrová funkce stupně q, kdy hodnota q je zvolena uživatelem. Pokud zvolíme hodnotu $q = 1$, jedná se o lineární jádrovou funkci, která odpovídá původní verzi algoritmu.
$$K(x^t, x) = (x^T x^t + 1)^q \tag{4.15}$$

- Radiálně-bázová jádrová funkce.
$$K(x^t, x) = exp\left(-\frac{\| x^t - x \|^2}{2s^2}\right) \tag{4.16}$$

- Mahalanobisova jádrová funkce, kde S je kovarianční matice.
$$K(x^t, x) = exp\left[-\frac{1}{2}(x^t - x)^T S^{-1}(x^t - x)\right] \tag{4.17}$$

- Sigmoidní funkce, která je podobná standardní funkci sigmoid používané u neuronových sítí.
$$K(x^t, x) = tanh(2x^T x^t + 1) \tag{4.18}$$

4.3.2 Algoritmy pro učení u metody SVM

V této práci byly při testování různých metod strojového učení použity dva typy algoritmů pro učení metody SVM.

- **LS-SVM**: Least squares support vector machines je varianta metody SVM, která při učení nepoužívá algoritmy pro kvadratickou optimalizaci, ale klasifikační problém řeší pomocí systému lineárních rovnic. Tato metoda je vhodná pro klasifikační problémy o velké velikosti a byla poprvé navrhnuta v práci [40].
- **Sekvenční minimalizační optimalizace**: Tato metoda využívá pro řešení kvadratického programování metodu dělení složitého problému na nejmenší přípustné podproblémy. Iterativně jsou vybírány podproblémy o velikosti 2 a cílová funkce je optimalizována na základě těchto podproblémů. Výhodou této metody je to, že jednoduché podproblémy je možné řešit analyticky a není nutné využívat složité a neefektivní metody pro kvadratickou optimalizaci. V této práci byla testována verze této metody upravená pro regresi, která byla navrhnuta v práci [39].

4.4 Rozhodovací stromy

Rozhodovací stromy je označení pro skupinu modelů, které jsou tvořeny hierarchickou stromovou strukturou a používají se pro klasifikaci dat a regresi. Jejich model je vytvářen na základě trénovací množiny dat a jedná se o strojové učení s učitelem. Jejich struktura je tvořena dvěma typy uzlů. Nelistové uzly obsahují podmíněný výraz, na základě jehož vyhodnocení je určen následující uzel, ve kterém bude vyhodnocování vstupního vektoru pokračovat. Listové uzly obsahují identifikátor klasifikační třídy, do které bude vyhodnocovaný vstupní vektor přiřazen. V případě regrese listové uzly obsahují numerickou hodnotu.

Proces vyhodnocování vstupního vektoru probíhá tak, že je postupně procházena stromová struktura směrem od kořenového uzlu. V každém nelistovém uzlu je vyhodnocen příslušný podmíněný výraz, který určí následující uzel pro vyhodnocení. V okamžiku, kdy algoritmus dospěje do listového uzlu, je vstupní vektor přiřazen do odpovídající třídy, nebo je k němu přiřazena odpovídající numerická hodnota.

4.4.1 Klasifikační stromy

Klasifikační stromy mají v listových uzlech uložen identifikátor třídy, do které bude tento listový uzel klasifikovat vyhodnocované vektory. Model stromu je vytvářen níže popsaným procesem. Označme aktuální uzel m a počet trénovacích vektorů, které byly přiřazeny k tomuto uzlu jako N_m. Dále označme počet trénovacích vektorů v tomto uzlu, které patří do třídy C_i jako N_m^i. Platí vztah $\sum_i N_m^i = N_m$.

Pro jednotlivé klasifikační třídy C_i vypočteme odhad pravděpodobnosti, že vektor vyhodnocovaný tímto uzlem patří do třídy C_i. Vztah pro výpočet pravděpodobnosti je následující [1]:

$$\hat{P}(C_i|x,m) \equiv p_m^i = \frac{N_m^i}{N_m} \tag{4.19}$$

Pokud pro všechny třídy C_i platí, že jim odpovídající pravděpodobnost p_m^i je rovna hodnotě 0 nebo 1, jsou všechny trénovací vektory přípádající k tomuto uzlu klasifikovány korektně. Uzel je následně nastaven jako listový a je označen identifikátorem třídy C_i. Pokud toto neplatí, je tento uzel rozdělen a je vytvořen odpovídající podstrom tohoto uzlu.

Přesnost klasifikace u zpracovávaného uzlu je také možné měřit pomocí jiných metrik, které pro výpočet využívají hodnoty uvedených pravděpodobností. Jedná se například o následující metriky [1].

- **Entropie:**

$$I_m = -\sum_{i=1}^{K} p_m^i log_2 p_m^i \tag{4.20}$$

- **Gini index:**

$$\phi(p, 1-p) = 2p(1-p) \tag{4.21}$$

- **Velikost klasifikační chyby:**

$$\phi(p, 1-p) = 1 - max(p, 1-p) \tag{4.22}$$

Jednotlivé vztahy, s výjímkou entropie, jsou uvedeny pro situaci, kdy jsou klasifikovány dvě třídy, ale je možné je zobecnit pro K tříd. Obecně byly dokázáno, že zmíněné metriky nemají příliš velké rozdíly v přesnosti [1].

Pokud je uzel vybrán pro rozdělení, je potřebné nejdříve určit atribut, který bude v tomto uzlu použit ve vyhodnocovaném výrazu. Protože obecně chceme vytvářet nejmenší rozhodovací stromy, je zvolen atribut, který nejpřesněji klasifikuje trénovací vektory náležející k tomuto uzlu. Pro každý atribut (a v případě numerických atributů pro každou jejich hodnotu, kterou je možné použít jako práh ve výrazu při vyhodnocovaní vektoru) je zjištěna velikost entropie (nebo jiné použité metriky) a je vybrán takový atribut a jeho parametry, u kterých je tato hodnota nejmenší.

Pro ilustraci tohoto procesu, uveďme příslušné vztahy při použití entropie. Označme dělený uzel jako m, N_m trénovací vektory u tohoto uzlu a N_{mj} trénovací vektory, které budou klasifikovány větví j. Počet větví j je $n = 2$ pro numerické atributy a pro diskrétní atributy je roven počtu hodnot n vybraného atributu mezi odpovídajícími trénovacími vektory.

Pro každou potenciálně vytvářenou větev j, následně určíme odhad pravděpodobnosti pro všechny třídy C_i, jak je uvedeno v následujícím vztahu [1]:

$$\hat{P}(C_i|x, m, j) \equiv p_m^i = \frac{N_{mj}^i}{N_{mj}} \tag{4.23}$$

Poté jsou vypočteny dílčí hodnoty entropie pro každou větev j. Tyto hodnoty jsou sečteny a výsledná suma představuje celkovou entropii pro vybraný atribut [1].

$$I_m = -\sum_{j=1} \frac{N_{mj}}{N_m} \sum_{i=1}^{K} p_{mj}^i log_2 p_{mj}^i \tag{4.24}$$

Po skončení výpočtu je vybrán atribut s odpovídajícími parametry, který má nejmenší hodnotu entropie a je použit pro rozdělení uzlu. Tento proces pokračuje rekurzivně až do okamžiku, kdy žádný uzel není potřebné rozdělit na nové uzly a jedná se o základní algoritmus, který využívají algoritmy pro vytváření rozhodovacích stromů jako jsou například algoritmy CART a ID3.

Při procesu tvorby rozhodovacího stromu jsou využívány dvě metody, jejichž cílem je optimalizace velikosti výsledného modelu [1]:

- **Prepruning**: U této metody jsou uzly děleny pouze v případě, že počet trénovacích vektorů pro tento uzel neklesne pod určitou hodnotu. Myšlenka u tohoto přístupu je ta, že dělení uzlu pro příliš málo testovacích vektorů by mohlo zhoršit celkovou generalizaci modelu.

- **Postpruning**: U tohoto přístupu je nejdříve vytvořen menší nezávislý soubor dat. Následně je pomocí zbylé části trénovacích dat vytvořen model rozhodovacího stromu, který klasifikuje všechny trénovací vektory korektně. Po vytvoření modelu, je pro každý podstrom vyhodnocen následující proces. Zvolený podstrom je dočasně nahrazen listovým uzlem a strom je otestován pomocí nezávislého souboru dat, který byl odložen stranou před vytvořením rozhodovacího stromu. Pokud nahrazení podstromu nezhorší celkovou chybu klasifikace u tohoto souboru dat, je dočasný listový uzel změněn na trvalý a nahradí testovaný podstrom, který je oříznut.

4.4.2 Náhodný les

Metoda Náhodný les byla představena v práci [6]. Základní princip této metody spočívá ve vytvoření více rozhodovacích stromů. Výstupní hodnotou této metody je průměr výstupních hodnot jednotlivých rozhodovacích stromů v případě regrese, nebo majoritní třída v případě klasifikace. Při vytváření jednotlivých rozhodovacích stromů jsou použity náhodně vybrané podmnožiny trénovací datové sady a při procesu dělení jednotlivých uzlů stromů, je u každého uzlu použita náhodná podmnožina vstupních parametrů. Stručně popsaný algoritmus této metody je následující (uvažujme N rozhodovacích stromů):

1. Pro každý N-tý rozhodovací strom je vybrána podmnožina trénovacích dat. K tomuto účelu je použita metoda bagging, která spočívá v náhodném výběru zadaného počtu trénovacích vektorů. Jednotlivé vektory mohou být vybrány vícekrát a tato metoda je použita N-krát. Je tedy vytvořeno N náhodných podmnožin trénovacích dat.

2. Jsou trénovány jednotlivé rozhodovací stromy. N-tý strom je vytvořen na základě odpovídající N-té podmnožiny trénovacích dat. Každý uzel stromu využívá náhodnou podmnožinu vstupních parametrů. Pokud by parametry nebyly vybírány náhodně, je vysoké riziko, že pokud by nějaký parametr klasifikoval lépe než ostatní, byl by při tvorbě rozhodovacích stromů vybírán mnohem častěji a jednotlivé rozhodovací stromy by mezi sebou byly silně korelovány.

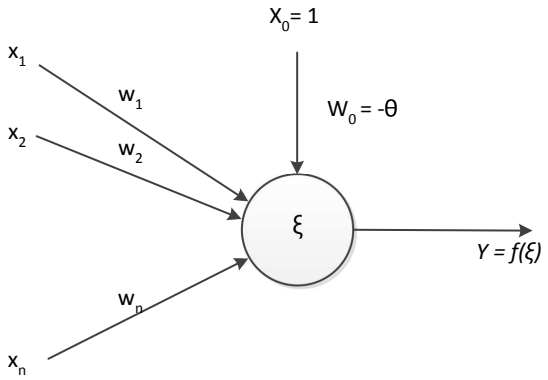
4.5 Neuronové sítě

Umělé neuronové sítě je označení pro rozsáhlé spektrum metod strojového učení, které slouží pro řešení klasifikace, regrese a dalších typů problémů. Společným prvkem pro všechny různorodé typy sítí a typy učících algoritmů je model umělého neuronu, který byl vytvořen na základě inspirace fungováním neuronů v lidském mozku.

Základní model umělého neuronu je zobrazen na ilustraci 4.3 a jeho funkce je popsána vztahy pro výpočet hodnoty vnitřního potenciálu neuronu (4.25) a pro výpočet výstupní hodnoty neuronu (4.26). Vstupní hodnoty umělého neuronu jsou označeny x_i, w_i jsou váhy odpovídající jednotlivým vstupům umělého neuronu, y je výstupní hodnota umělého neuronu, ξ je vnitřní potenciál (vážená suma vstupů), Θ je práh, x_0 pomocný vstup s konstantní hodnotou 1 a f je aktivační funkce [17].

$$\xi = \sum_{i=0}^{n} x_i w_i \tag{4.25}$$

$$y = f(\xi) \tag{4.26}$$



Obrázek 4.3: Umělý neuron [17]

Aktivační funkce umělého neuronu může být různého typu v závislosti na problému, který neuronová síť řeší. Mezi často používané aktivační funkce patří funkce:

- Lineární aktivační funkce.

$$f(\xi) = \xi \tag{4.27}$$

- Funkce sigmoid, která je speciálním typem logistické funkce [26].

$$f(\xi) = \frac{1}{1 + e^{-\xi}} \tag{4.28}$$

- Náběhová funkce, která je upravena pro záporné hodnoty (tzv. Leaky ReLU) [27]

$$f(\xi) = \begin{cases} \xi & \xi > 0 \\ 0.01\xi & \xi \leq 0 \end{cases} \tag{4.29}$$

Jednotlivé varianty neuronových sítí se liší algoritmem, který je použit pro učení sítě a způsobem propojení neuronů. Nejznámější typ neuronové sítě se nazývá dopředná neuronová síť. Struktura tohoto typu sítě je složena z vrstev neuronů a každý neuron je propojen se všemi neurony v předchozí a v následující vrstvě. Standardním učícím algoritmem pro tento typ sítě jsou různé varianty algoritmu zpětného šíření (anglicky Backpropagation).

V této práci byl použit algoritmus resilient backpropagation (zkráceně Rprop), který byl představen v práci [35] a jedná se o jeden z nejrychlejších učících algoritmů. Každá váha v síti má vlastní privátní hodnotu pro aktualizaci váhy, která je aktualizována pouze na základě známénka parciální derivace a ne její velikosti. Pokud se známenko parciální derivace změní oproti znaménku parciální derivace v předchozí iteraci, je tato hodnota vynásobena hodnotou faktoru η^-. V opačném případě je vynásobena hodnotou faktoru η^+. Platí, že hodnoty těchto faktorů podléhají omezením $0 < \eta^- < 1 < \eta^+$.

Kapitola 5

Existující přístupy a nástroje pro predikci stability proteinů

Tato kapitola obsahuje základní přehled principů a rozdělení metod pro predikci změny stability proteinů na základě jejich mutací. V první části bude uvedeno stručné rozdělení metod podle způsobu predikce. Druhá část popisuje vybrané existující nástroje pro predikci stability proteinů. Poslední část kapitoly je poté věnována nástroji MAESTRO [25], který byl primární inspirací pro zadání této práce.

5.1 Rozdělení metod pro predikci stabilit

Základní klasifikace metod pro predikci stabilit je možné provést na základě toho, jaké informace o struktuře proteinu tyto metody pro predikci využívají. Metody lze tedy rozdělit na následující dvě skupiny, přičemž moderní metody kombinují oba přístupy.

- **Sekvenční metody**: tyto metody pro predikci využívají pouze informace o primární a sekundární struktuře vybraného proteinu, tedy informace o aminokyselinách v okolí místa mutace na stejném polypeptidovém řetězci a jejich fyzikálně-chemických vlastnostech jako je např. konzervovanost, korelace aj.
- **Strukturní metody:** v těchto metodách jsou kromě primární a sekundární struktury využity i informace vycházející z terciální struktury proteinu. Pomocí 3D modelu nebo podobných metod, jsou nalezeny aminokyseliny v celkovém okolí mutace a tyto informace jsou využity pro predikci. Dále jsou využívány také strukturní vlastnosti aminokyselin jako je například míra vystavení aminokyseliny rozpouštědlu, označovaná jako ASA.

Pro predikci stability proteinů je možné využít více způsobů. Rozdělení způsobů predikce, které bylo převzato z [19], je stručně popsáno v následujícím seznamu.

- **Fyzikální a chemické vlastnosti aminokyselin:** pro predikci jsou využity vlastnosti substituovaných aminokyselin a proteinů, na základě kterých je vypočten vliv mutace na protein.
- **Efektivní potenciály:** pro predikci jsou využity různé typy potenciálů, které popisují vztahy mezi jednotlivými residui v proteinu. Příklady představují například potenciály pro popis torzních uhlů, vzdálenosti residuí, vlastností kontaktů jednotlivých atomů nebo potenciály založené na znalostech a statistice výskytu konkrétních residuí.

- **Výpočet energetické funkce:** změna stability proteinů může být také vypočtena přímo na základě vlastností proteinu, u nichž byl pozorován vliv na stabilitu proteinu. Jedná se například o vodíkové můstky a další atomové interakce. Příkladem je například nástroj FOLD-X.
- **Strojové učení:** tyto metody pro predikci změny stability využívají metody strojového učení, jakými jsou například neuronové sítě, rozhodovací stromy, algoritmus náhodný les nebo různé varianty metody support vector machines.

5.2 Existující nástroje pro predikci stabilit

V této části jsou stručně popsány existující vybrané nástroje pro predikci stability proteinů.

- **Rosetta** [36] je soubor více nástrojů, které umožňují přesně modelovat makromolekulární struktury na úrovni atomů. Kromě algoritmů pro modelování stability proteinů obsahuje také nástroje pro predikci struktury proteinů, modelování proteinů a další využití.
- **Eris** [46] je metoda, která pro predikci využívá fyzické vlastnosti proteinu a atomové modelování. Hodnota $\Delta\Delta G$ je vyjádřena jako vážená suma Van der Waalsových sil, vodíkových vazeb, rozpustnosti a páteřních statistických energií.
- **FoldX** [37] je nástroj, který změnu hodnoty $\Delta\Delta G$ vypočítává na základě makromolekul pomocí 3D modelu proteinu ve vysokém rozlišení. Kromě hodnoty změny stability umožňuje pro mutace zjistit i vliv na sbalení a dynamiku proteinů, predikci vodíkových můstků a další.
- **PopMusic** [13] je webový server, který predikuje změny termodynamické stability pomocí lineární kombinace statických potenciálů, které jsou zjištěny na základě velikosti plochy daného residua, která je přístupná rozpouštědlu.
- **iPTREE-STAB** [21] je nástroj ve formě webové serveru, který je zaměřen na jednoduchou predikci a proto využívá pouze informace o několika okolních aminokyselinách na polypeptidovém řetězci (okolí ± 3) a informace o mutaci (substituované aminokyseliny, teplota a pH). Pro predikci je použito strojové učení ve formě rozhodovacích stromů.
- **Prethermut** [42] umožňuje predikovat změny stability pro jednobodové i vícebodové mutace. Je založen na použití metod strojového učení support vector machines a náhodný les. Predikce používá informace o struktuře proteinu, které jsou získané pomocí vytvořeného modelu mutovaného proteinu.
- **I-Mutant** [8] je nástroj, který jednotlivé mutace klasifikuje do tří tříd na stabilizující, destabilizující a neutrální mutace. Toto rozložení je dáno skutečností, že experimentálně změřené hodnoty $\Delta\Delta G$ v okolí hodnoty 0 kCal/mol mohou být chybně klasifikovány z důvodu chyby měření. Pro predikci je použita metoda support vector machines a využívají se informace o struktuře proteinu i sekvenci aminokyselin.
- **EASE-AA** [18] je metoda specifická použitou metodou rozdělení dat na testovací a trénovací množiny. Pro testování přesnosti využívá pouze mutace na takových proteinech, které nebyly obsaženy v trénovacích datech, čímž dosahuje objektivnější klasifikace nových mutací. Pro predikci je využita metoda support vector machines.

- **mCSM** [34] pro výpočet změny stability používá grafově založené podpisy, které popisují vzdálenost mezi atomy a tím modelují okolí mutovaného residua. Kromě změny stability vyhodnocuje také interakce mezi proteiny a proteiny a nukleovými kyselinami.

- **Mupro** [9] je další nástroj využívající strojové učení a metodu support vector machines. Pro predikci využívá informace o sekvenci i struktuře proteinu a jeho výhoda spočívá v tom, že umožňuje téměř stejnou přesností predikovat změny stability i u mutací na proteinech, jejichž terciální struktura není k dispozici.

- **CUPSAT** [33] je webový nástroj, který pro predikci hodnoty $\Delta\Delta G$ používá výpočet strukturních atomových potenciálů a potenciálů torzních uhlů. Je specifický tím, že pro vybranou pozici mutace vyhodnotí změnu stability pro každou z 19 možných substitucí aminokyselin. Pro každou z těchto aminokyselin jsou také vyhodnoceny vlastnosti jako je například sekundární struktura, torzní úhly a další strukturní vlastnosti proteinu.

5.3 MAESTRO

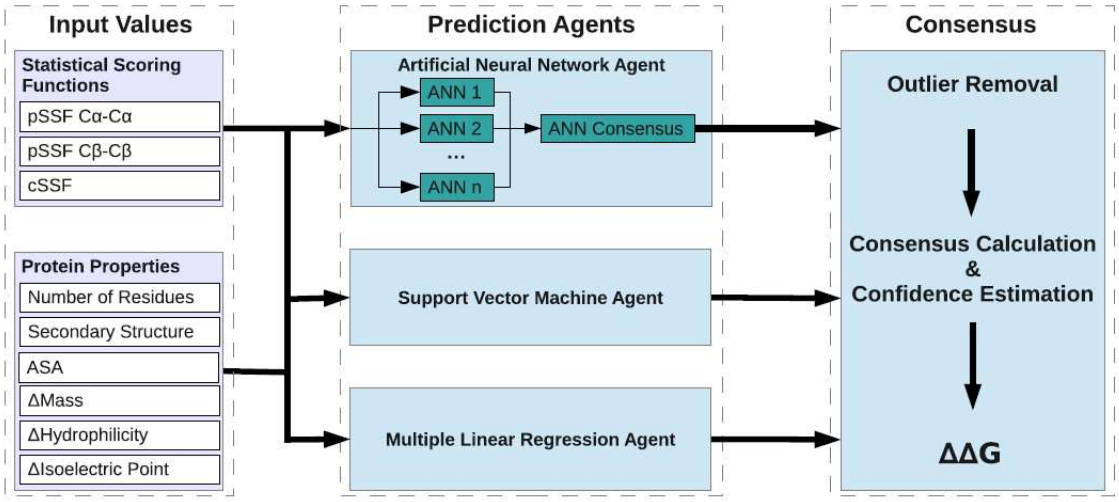
MAESTRO [25] je multiagentní nástroj pro predikci změny stability proteinů pro jednobodové mutace. Predikce využívá informace o struktuře proteinu a systém multiagentního strojového učení. Kromě samotné klasifikace mutací na stabilizující a destabilizující a predikci hodnoty $\Delta\Delta G$ umožňuje také prohledávání vybraného prostoru pro vícebodové mutace a predikci stability disulfidických můstků

Vstupní informace pro predikci jsou rozděleny do dvou skupin a dohromady jde o devět následujících hodnot:

- **Statistická hodnotící funkce** (jinak také potenciály založené na znalostech). První typ představují dvě funkce typu distance-dependent residue pair (pSSFs), které popisují relativní frekvenci pozorované vzdálenosti pro dvojici aminokyselin. Druhý typ je funkce capture solvent exposure of protein residues (cSSFs), která popisuje kontakty jednotlivých atomů v určené vzdálenosti od každého residua.

- **Fyzikální vlastnosti proteinu a aminokyselin** popisující velikost proteinu, dostupnou plochu povrchu a sekundární strukturu. Substituované aminokyseliny jsou popsány pomocí rozdílů velikostí aminokyselin a jejich hydrofobicity a izoelektricity.

Systém multiagentního strojové učení je složen ze sedmi agentů (tři typu neuronová síť, tři typu support vector machines a jeden typu vícenásobné lineární regrese). Tyto agenti jsou dále rozděleni na obecné a specializované, kdy obecní agenti jsou trénování na stabilizujících i destabilizujících mutacích a specializovaní agenti pouze na jedné z těchto tříd. Predikce poté probíhá tak, že skupina obecných agentů klasifikuje mutaci na stabilizující nebo destabilizující a speciální agenti pro odpovídající třídy predikují velikost $\Delta\Delta G$.



Obrázek 5.1: Struktura nástroje MAESTRO [25]

Postup samotné predikce je následující: 1) výpočet statistických funkcí a ostatních vstupních hodnot, 2) spuštění agentů, 3) výpočet konsenzuální predikce. Mezi body 2) a 3) jsou postupně v každé iteraci odebíráni agenti, jejichž výstupní hodnota se nejvíce liší od střední hodnoty a toto probíhá dokud je tato odchylka větší než standardní odchylka a nebo zbývají pouze dva agenti. Výstupní hodnoty zbývajících agentů jsou poté zprůměrovány a určeny jako výstupní hodnota. Dále je vypočtena hodnota důvěry, která je v intervalu $(0.0, 1.0)$ a hodnoty 1.0 nabývá, pokud agenti dosáhnout čistého konsenzu.

Kapitola 6

Návrh aplikace

Cílem práce je navrhnout a vytvořit aplikaci pro predikci vlivu bodových mutací na stabilitu proteinů, která bude využívat různé typy strojového učení a různé parametry proteinů a aminokyselin. První část kapitoly je věnována návrhu vstupních parametrů pro predikci a způsobu jejich výpočtu. Druhá část popisuje navrženou strukturu aplikace a způsob predikce. Poslední část kapitoly je věnována návrhu různých typů strojového učení, které byly následně testovány při vývoji aplikace.

6.1 Návrh parametrů pro predikci

Navržené vstupní parametry pro predikci se dělí na fyzikálně-chemické vlastnosti proteinů a aminokyselin a informace získané z primární, sekundární a terciální struktury mutovaného proteinu. Tento soubor parametrů byl následně rozdělen na čtyři skupiny, jejichž vybrané kombinace byly testovány jako vstupní parametry pro navržené typy strojového učení.

6.1.1 Vlastnosti aminokyselin

První skupina parametrů se skládá ze 48 fyzikálně-chemických, energetických a konformačních vlastností aminokyselin, které byly převzaty z práce od Gromihy a spol. [20]. V této práci byla zkoumána termostabilita 16 proteinových rodin, na základě porovnání uvedených vlastností u mezofilních a termofilních proteinů. Jedná se například o polaritu, kompresibilitu, izoelektrický bod, průměrný počet okolních reziduí, počet atomů v postraním řetězci (mimo atomy vodíku) nebo flexibilitu. Hodnoty molekulní váhy jednotlivých aminokyselin byly nahrazeny mírně přesnější hodnotou z referenční tabulky [3].

Druhou skupinu parametrů tvoří doplňující fyzikálně-chemické vlastnosti aminokyselin. Jedná se o hodnotu indexu elektrického náboje a velikost isotropní povrchové plochy, které byly převzaty z práce [11]. Dále hodnotu hydrofobicity uvedenou v práci [16] a hodnotu relativní hydrofobicity ve formě hydrofobického indexu uvedeného v referenční tabulce [3]. Hodnoty hydrofobicity v tomto indexu byly normalizovány, aby Glycin jako neutrální aminokyselina měl hodnotu 0 a nejvíce hydrofobní aminokyselina hodnotu 100 (Fenylalanin). Dále byly přidány dvě binární hodnoty, které označují, zda aminokyselina patří do aromatické nebo alifatické skupiny. Poslední parametr je velikost van der Waalsova objemu, která byla převzata z tabulky vlastností proteinových aminokyselin v publikaci [38].

6.1.2 Strukturní a statistické parametry

Třetí skupinu parametrů tvoří strukturní a statistické informace, které byly pro jednotlivé proteiny a jejich mutace vypočteny na základě struktury proteinů dostupných z databáze Protein Data Bank (PDB). Predikce je tedy omezena pouze na proteiny, jejichž struktura je v této databázi dostupná. Hodnoty parametrů byly určeny pomocí nástrojů Biopython [10], DSSP [43, 23] a MUSCLE [15].

Navržené strukturní parametry jsou následující: relativní hodnota plochy dostupné solventu, sekundární struktura proteinu, počet atomů uhlíku a frekvence výskytu jednotlivých aminokyselin v okolí místa mutace o velikosti 10Å.

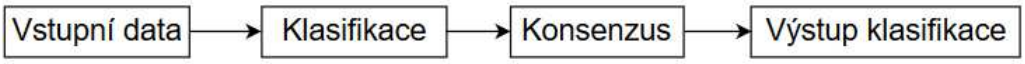
Statistický parametr je hodnota informačního obsahu místa mutace v proteinu. Tato hodnota byla pro každou mutaci vypočtena následujícím postupem. Nejdříve je primární struktura proteinu zarovnána nástrojem online BLAST. Poté je z vrácených sekvencí vytvořeno vícenásobné zarovnání nástrojem MUSCLE a z tohoto zarovnání je vypočten informační obsah pro danou pozici. Vztah pro výpočet byl převzat z nástroje BioPython, ve kterém i označuje pořadí kódu aminokyseliny v abecedě, P_i je frekvence výskytu aminokyseliny na dané pozici ve vícenásobném zarovnání a 0.05 je očekávaná frekvence aminokyseliny [10].

$$IC = \sum_{i=1}^{20} P_i log\left(\frac{P_i}{0.05}\right) \tag{6.1}$$

Čtvrtá skupina parametrů obsahuje informace o primární struktuře proteinu v okolí místa mutace. Jedná se o deset předchozích a deset následujících aminokyselin od místa mutace na polypeptidovém řetězci.

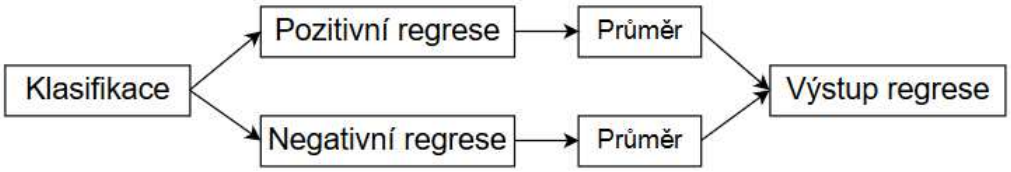
6.2 Struktura aplikace a proces predikce

Aplikace je navržena jako multiagentní systém, který využívá více metod strojového učení a různé kombinace vstupních parametrů u jednotlivých agentů. Před začátkem predikce jsou pro všechny vstupní záznamy vypočteny jednotlivé vstupní parametry. Poté je provedena klasifikace a na základě výsledků klasifikace je predikována hodnota změny Gibbsovy volné energie pomocí regresních agentů. Proces klasifikace a regrese stručně popisují následující ilustrace.



Obrázek 6.1: Proces klasifikace

Před začátkem klasifikace jsou vstupní data nejdříve předzpracována. Následně každý klasifikační agent provede nezávislou klasifikaci. Z výsledků klasifikace jednotlivých agentů je poté pro každou vstupní mutaci vypočten většinový konsenzus, který určuje konečný výsledek klasifikace.



Obrázek 6.2: Proces regrese

Proces regrese probíhá až po ukončení klasifikace. Tento přístup umožnuje využít odlišné regresní agenty pro pozitivně a negativně klasifikované mutace. Každý z těchto speciálních agentů byl trénován na odpovídající podmnožině trénovacího datasetu. Tímto postupem je teoreticky možné dosáhnout lepších výsledků regrese. Po skončení regrese u všech agentů, je vypočten průměr jejich výstupních hodnot a tato hodnota je použita jako konečný výsledek regrese.

6.3 Navržené typy strojového učení a jejich testování

Metody strojové učení pro klasifikaci, které byly navrženy pro testování přesnosti klasifikace stabilizujích a destabilizujích mutací a případné využití v konsensuálním prediktoru, jsou následující:

- **Algoritmus k-nejbližších sousedů** - základní a jeden z nejjednodušších algoritmů pro klasifikaci.
- **Rozhodovací strom** - standardní rozhodovací strom vytvářený pomocí algoritmu C4.5.
- **Náhodný les** - metoda pro klasifikaci, která vytváří více rozhodovacích stromů.
- **Neuronová síť** - standardní neuronová síť, která pro učení používá algoritmus Resilient Backpropagation.
- **Support vector machines** - nelineární verze algoritmu SVM. Pro testování bylo navrhnuto použití dvou typů jádrových funkcí a to lineární a gaussovské. Pro učení bylo navrhnuto použití metody nejmenších čtverců (tzv. LS-SVM).

pro regresi byly navrženy následující dvě metody:

- **Neuronová síť** - viz navržené metody pro klasifikaci.
- **Support vector machines** - nelineární verze algoritmu svm s gaussovskou jádrovou funkcí, která pro učení využívá metodu Sequential minimal optimization upravenou pro regresi.

Proces testovaní uvedených metod byl navržen tak, že každá metoda byla testována s různými kombinacemi vstupních parametrů. Následně byly vybrány vhodné kombinace vstupních parametrů pro jednotlivé metody, které byly použity při konečném testovaní aplikace a porovnávání jednotlivých metod. Podrobný popis testovaných kombinací a procesu testovaní je uveden v kapitole 9.

Kapitola 7

Rozbor použitých dat

Tato kapitola popisuje datovou sadu použitou v této práci. Začátek kapitoly je věnován stručnému představení databáze Protherm, ze které byly převzaty informace o jednotlivých mutacích. Následuje popis datasetu, který byl použit jako výchozí dataset pro tuto práci. Poslední část kapitoly je věnována rozboru dat, které byly k tomuto datasetu přídány.

7.1 Protherm

Protherm je databáze termodynamických parametrů proteinů a jejich mutantů. Obsahuje například informace o experimentálních metodách a jejich podmínkách, informace o sekundární struktuře proteinů, změně Gibbsovy volné energie, změně entalpie, změně teplotní kapacity a o dalších parametrech. Jednotlivé záznamy jsou provázány s databázemi Protein Data Bank, Protein Information Resource, SWISS-PROT a dalšími [5].

7.2 Základní dataset

Protože informace v databázi Protherm nejsou vždy úplné, dostatečně přesné nebo uvedené hodnoty mají rozdílný význam u více záznamů, byl jako základní dataset využit dataset Fireprot [30], který již obsahuje upravené a vyfiltrované záznamy o jednotlivých mutacích a k tomuto datasetu byly následně přidány další atributy.

Nejčastější problémy u záznamů v databázi Protherm byly následující: chybějící hodnota změny Gibbsovy volné energie ($\Delta\Delta G$), nekonzistence mezi jednotkami (kCal/mol, kJ/mol), opačná znaménka, nekonzistence v počtu hodnot Gibbsovy volné energie (některé studie používají tři typy, databáze Protherm rozlišuje pouze dva) a to, že změna Gibbsovy volné energie nebyla měřena s ohledem na původní reziduum, ale na příkladech obsahující Alanin.

Dále byly z datasetu odstraněny záznamy mutací, u kterých se hodnota $\Delta\Delta G$ nacházela v intervalu $(-0.5, 0.5)$. Tento interval byl zvolen na základě velikosti chyby měření, která je udávána 0.48 kCal/mol [24]. Pokud pro danou mutaci existovalo více záznamů, byl ponechán pouze záznam s hodnotou pH nejblíže pH7.

Z datasetu byly použity následující položky:

- Identifikátor proteinu v databázi Protein Data Bank a identifikátor polypeptidového řetězce, na kterém se mutace nachází.
- Pozice mutace v polypeptidovém řetězci.
- Hodnota $\Delta\Delta G$.

7.3 Přidané atributy v datasetu

Ke každé položce v datasetu byly přidány následující atributy:

- Sekundární struktura a relativní velikost plochy dostupné solventu. Hodnoty byly vypočteny pomocí nástojů BioPython [10] a DSSP [43, 23].
- Absolutní hodnota rozdílu hodnot fyzikálně-chemických vlastností substituovaných aminokyselin, které byly popsány v části 6.1.1.
- Informační obsah vypočtený pro danou pozici v polypeptidovém řetezči pomocí vztahu uvedeném v části 6.1.2.
- Počet atomů uhlíku a frekvence výskytu jednotlivých aminokyselin v okolí mutace o velikosti 10 Å. Hodnoty byly určeny pomocí nástroje Biopython.
- Deset předchozích a následujících aminokyselin od místa mutace v primární struktuře proteinu.

Kapitola 8

Implementace

Tato kapitola je zaměřena na technický popis implementace vytvořeného konsensuálního prediktoru. Na začátku kapitoly jsou uvedeny použité technologie, knihovny a nástroje a popis požadavků pro korektní chod vytvořené aplikace. Následuje popis formátu vstupních a výstupních dat, popis grafického uživatelského rozhraní aplikace a popis ovládání aplikace. Poté je stručně popsána vnitřní struktura vytvořené aplikace. Dále je popsán proces přípravy vstupních dat a výpočtu vstupních parametrů pro jednotlivé mutace, za kterým následuje popis předzpracování vstupních dat. Poslední část této kapitoly je věnována rozboru procesu predikce.

8.1 Technické parametry aplikace a použité nástroje

Jako hlavní implementační jazyk byl zvolen programovací jazyk C# a prostředí frameworku .NET ve verzi 4.5.2. Pomocný skript, který slouží pro přípravu vstupních dat a výpočet vstupních parametrů pro predikované mutace, byl vytvořen v jazyce Python ve verzi 3.6.0 v distribuci Anaconda. Pro extrahování informací o struktuře proteinů ze souborů z databáze Protein Data Bank byl použit nástroj Biopython [10] verze 1.68. Pro výpočet sekundární struktury proteinu a velikosti plochy aminokyseliny, která je dostupná rozpouštědlu, byl využit nástroj DSSP [43, 23]. Při výpočtu hodnoty informačního obsahu na pozici predikované mutace v proteinu byly použity nástroje online BLAST [2] a MUSCLE [15].

8.2 Požadavky pro spuštění aplikace

Technické požadavky pro korektní chod aplikace a pro výpočet vstupních parametrů konsensuálního prediktoru jsou následující:

- Framework .NET ve verzi 4.5.2 a vyšší.
- Jazyk Python 3 ve verzi 3.6.0 spustitelný příkazem `python`.
- Nástroj Biopython použitelný v jazyku Python.
- Aplikace DSSP a MUSCLE. Tyto aplikace jsou distribuovány společně s touto aplikací a při instalaci aplikace jsou umístěny v adresáři `Script`, který se nachází v instalačním adresáři.

- Připojení k internetu pro stažení souborů s popisem struktury proteinů z databáze Protein Data Bank a pro korektní funkci výpočtu informačního obsahu, při kterém je využívána online verze nástroje BLAST.

8.3 Formát vstupních a výstupních dat

Vstupní data jsou požadovány v textovém souboru, který bude ve formátu csv a při popisu mutací bude pro oddělení jednotlivých položek používat znak čárky. Příklad formátu vstupních dat je uveden v tabulce 8.1. Na prvním řádku vstupního souboru je vyžadováno uvést hlavičku, která slouží pro identifikaci jednotlivých sloupců v csv formátu a bude mít tvar a pořadí, které je uvedeno v tabulce 8.1. Význam jednotlivých položek je následující:

- **protein** - idenfitikátor proteinu v databázi Protein Data Bank, který je využit pro stažení odpovídajícího souboru popisujícího strukturu tohoto proteinu.
- **chain** - identifikátor polypeptidového řetězce v proteinu, na kterém se mutace nachází a který odpovídá indentifikátoru použitému v databázi Protein Data Bank.
- **mutation** - popis mutace ve formátu XnY, kde X a Y jsou písmena označující substituované aminokyseliny (X - původní aminokyselina, Y - nově vložená aminokyselina) a n je pozice místa mutace v proteinu. Hodnota n je vztažená ke struktuře proteinu, která je uvedená v souboru ve formátu PDB, který je pro mutovaný protein dostupný v databázi Protein Data Bank.

protein,chain,mutation
1a2p,A,L31Q
4wor,A,G45C

Tabulka 8.1: Ukázka vstupních dat

Formát výstupních dat je identický jako formát vstupního souboru. K jednotlivým záznamům ze vstupního souboru jsou pouze přidány položky `stabilization` a `ddg`, které obsahují výsledky predikce. Položka `stabilization` nabývá hodnoty 1 pro mutace, které byly predikovány jako stabilizující a hodnoty 0 pro mutace predikované jako destabilizující. Položka `ddg` obsahuje predikovanou hodnotu $\Delta\Delta G$.

8.4 Ovládání aplikace

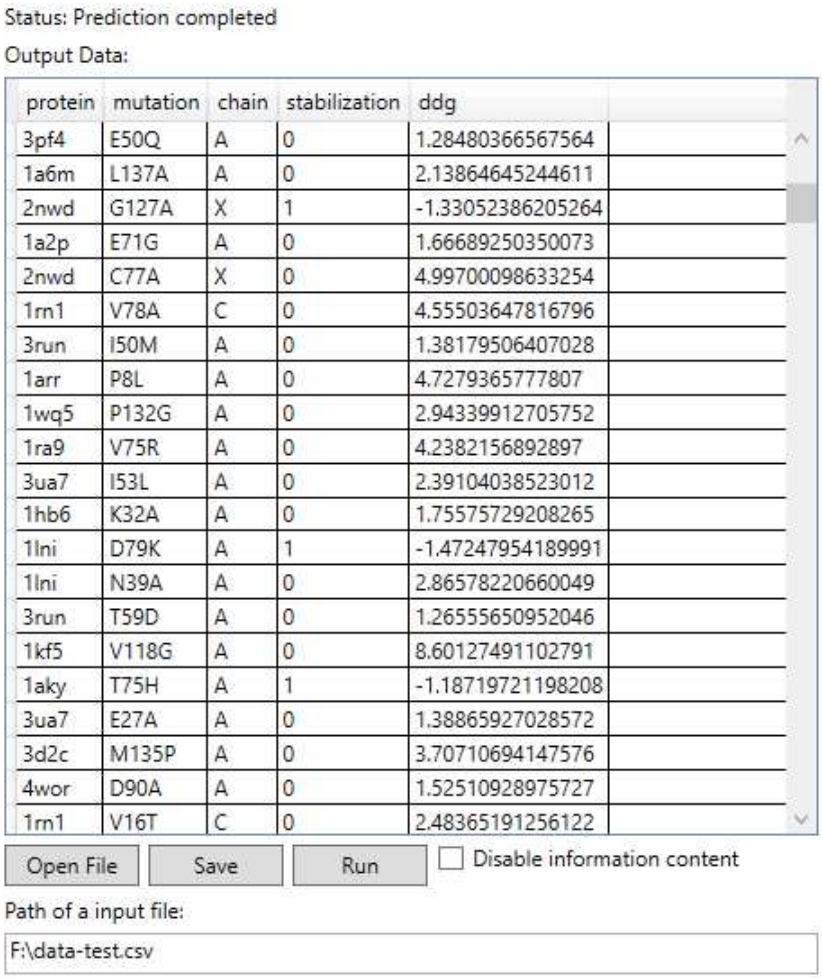
Vytvořený konsensuální prediktor je implementován jako konzolová aplikace, která je ve výchozím stavu spouštěna s doplňujícím grafickým uživatelským rozhraním. Aplikaci je možné spustit také z příkazové řádky jako standardní konzolovou aplikaci, která není ovládána přes grafické uživatelské rozhraní. Při spuštění z příkazové řádky nejsou predikované hodnoty uloženy do výstupního souboru, ale jsou vypsány na standardní výstup.

- **Konzolové rozhraní:** Pro spuštění aplikace bez grafického uživatelského rozhraní je nutné aplikaci spustit s následující parametry:

```
Application.exe -c input_file_path -disable=information_content
```

První parametr `-c` signalizuje aplikaci, aby nevytvářela okno s grafickým uživatelským rozhraním a výstupní informace ve formátu csv vypsala na standardní výstup. Druhý parametr je určený pro zadaní cesty k souboru se vstupními daty. Třetí parametr je nepovinný a pokud je zadán, při spuštení aplikace je deaktivován výpočet informačního obsahu na pozici mutace v proteinu. Důvody pro zavedení tohoto parametru jsou podrobněji rozebrány v podkapitole 8.6.

- **Grafické uživatelské rozhraní:** Při spuštění aplikace bez parametrů je otevřeno jednoduché grafické okno, které obsahuje primárně prostor pro zobrazení tabulky s výstupními daty a základní ovladací prvky pro práci s aplikací. Vzhled grafického uživatelského rozhraní je zobrazen na ilustraci 8.1. Zobrazená data představují část výstupních predikovaných hodnot pro testovací část datasetu.



Obrázek 8.1: Grafické uživatelské rozhraní aplikace

Grafické uživatelské rozhraní umožňuje standardní výběr souboru se vstupními daty, případně manuální zadaní cesty ke vstupnímu souboru. Dále je možné uložit soubor s výstupnými daty z predikce do vybraného adresáře. Tento výstupní soubor bude automaticky pojmenován DP-Flax-output.csv. Tlačítkem `Run` je spuštěna predikce. Zaškrtávací pole `Disable information content` umožňuje deaktivaci výpočtu informačního obsahu na pozici mutace.

8.5 Struktura zdrojového kódu aplikace

Tato podkapitola obsahuje velice stručný popis tříd ve zdrojovém kódu vytvořené aplikace a je v této textové práci uvedena primárně z důvodu, aby bylo možné jednodušeji analyzovat zdrojový kód aplikace.

- **Třída Data:** Objekt této třídy je určený pro uložení dat o jednotlivých mutacích a jejich parametrech. Metody této třídy implementují stahovaní příslušných souborů z databáze Protein Data Bank, předzpracovaní vstupních parametrů pro strojové učení a vytváření vstupních a výstupních datových vektorů pro metody strojového učení.
- **Třída Program:** Tato třída obsahuje jádro aplikace. Implementuje metody pro spuštění predikce, výpočet výsledků konsensuální klasifikace a regrese a výpis výstupních dat.
- **Třída Agent:** Tato abstraktní třída definuje obecné rozhraní pro jednotlivé třídy agentů strojového učení. Všechny třídy, které implementují vybrané metody strojového učení, jsou odvozeny od této abstraktní třídy.
- **Třídy NeuralNetwork, RandomForest, SVM a NeuralNetworkRegression:** Tyto třídy implementují klasifikaci a regresi pomocí odpovídajících metod strojového učení. Pro implementaci jednotlivých metod strojového učení byl použit framework Accord.NET.

8.6 Příprava vstupních parametrů pro strojové učení

Před začátkem klasifikace a regrese jsou nejdříve staženy potřebné datové soubory z databáze Protein Data Bank. Poté jsou ke každé mutaci, která je uvedena ve vstupním souboru, vypočtěny vstupní parametry pro strojové učení. Všechny datové soubory, které jsou staženy a vytvořeny během procesu přípravy vstupních parametrů, jsou uloženy do adresáře DP-Flax-predictor, který je vytvořen ve standardním uživatelském systémovém adresáři Dokumenty. Proces přípravy vstupních dat je možné rozdělit na dvě části:

1. **Stažení dat z databáze Protein Data Bank** - aplikace nejdříve vytvoří seznam všech proteinů, ve kterých se nachází jednotlivé mutace ze vstupniho souboru. Pro tyto proteiny jsou následně staženy soubory ve formátu PDB, které obsahují popis makromolekulární struktury proteinu a jsou uloženy do výše uvedeného adresáře v adresáři Dokumenty. Pokud už soubor s popisem struktury vybraného proteinu v adresáři existuje, není pro tento protein soubor opakovaně stahován.
2. **Spuštění skriptu pro přípravu vstupních dat** - aplikace spustí samostatný skript vytvořený v jazyce Python, který načte vstupní soubor a ke každé uvedené mutaci vypočitá jednotlivé vstupní parametry. Výsledný soubor je uložen pod jmeném preparedData.csv do výše uvedeného adresáře. Vytvořený skript je možné rozdělit do tří částí:
   (a) Výpočet vstupních parametrů založených na fyzikálně-chemických vlastnostech aminokyselin, které jsou podrobněji popsány v části 6.1.1. Pro každou vstupní mutaci jsou vypočteny absolutní hodnoty rozdílu hodnot fyzikálně-chemických

vlastností substituovaných aminokyselin. Hodnoty fyzikálně-chemických vlastností aminokyselin, které byly použity pro výpočet, je možné nálézt v souboru AAproperties.csv, který je dostupný v instalačním adresáři aplikace v podadresáři Script.

(b) Výpočet informačního obsahu na pozici mutace v proteinu. Velikost informačního obsahu je vypočtena podle vzorce 6.1, který byl převzat z nástroje Biopython. Nejdříve je pro každý protein ze vstupního souboru extrahována primární struktura tohoto proteinu z odpovídajícího staženého souboru formátu PDB. Tato sekvence aminokyselin proteinu je nástrojem online BLAST zarovnána s databází NCBI. Ze všech sekvencí aminokyselin, která tento nástroj nalezl, je vytvořeno vícenásobné zarovnaní. K výpočtu vícenásobného zarovnaní byl použit externí program MUSCLE. Z tohoto vícenásobného zarovnání je vypočten informační obsah pro konkrétní pozici v sekvenci proteinu. Soubory vytvořené nástroji BLAST a MUSCLE jsou ukládány ve výše uvedeném adresáři a pokud už pro zvolené proteiny tyto soubory existují, nejsou znovu vytvářeny.

Protože výpočet v nástrojích BLAST a MUSCLE je při větším počtu proteinů časově náročný, je možné výpočet informačního obsahuje deaktivovat, jak je uvedeno v části 8.4, která popisuje ovládání aplikace. Pro každou mutaci poté bude hodnota informačního obsahu nastavena na standardní hodnotu 4.124174935, která byla vypočtena jako průměrná hodnota z použitého datasetu.

(c) Výpočet strukturních a sekvenčních parametrů. Pomocí nástroje Biopython jsou na základě molekulární struktury proteinu z databáze Protein Data Bank vypočteny informace popisující strukturu a sekvenci proteinu v okolí místa mutace. Pro výpočet plochy dostupné rozpouštědlu a určení typu sekundární struktury byl použit nástroj DSSP.

8.7 Předzpracovaní vstupních dat

Před začátkem výpočtu u klasifikačních a regresních metod strojového učení jsou vstupní parametry u jednotlivých mutacích předzpracovány. Předzpracovaní vstupních dat se skládá ze dvou částí. V první části předzpracování dat jsou hodnoty vstupních parametrů normalizovány. Ve druhé části jsou poté vytvořeny vstupní datové vektory pro strojové učení, jejichž velikost je daná počtem a typem vybraných vstupních parametrů, protože při testovaní a implementaci aplikace byly testovány různé kombinace vstupních parametrů.

- **Normalizace dat:** Hodnoty všech vstupních parametrů, které jsou vyjádřeny číselnou hodnotou a nejsou reprezentovány binárními hodnotami, jsou transformovány ze svého původního rozsahu hodnot do rozsahu hodnot v intervalu $(0, 1)$. Jedná se o fyzikálně-chemické vlastnosti aminokyselin, informační obsah na pozici mutace v proteinu a počet atomů uhlíku v okolí mutace o velikosti 10 Å. Tato transformace je nutná z důvodů, že rozsah možných hodnot u jednotlivých vstupních parametrů se liší až o tři řády a pro metody strojového učení není vhodné, pokud jednotlivé hodnoty ve vstupním datovém vektoru mají velmi odlišnou velikost. Pokud by tato transformace nebyla provedena, je možné, že by metody strojového učení přikládaly vstupním parametrům s velkými hodnotami větší význam než ostatním parametrům.

- **Vytvoření vstupních vektorů a kódování vstupních parametrů:** Při tvorbě vstupních datových vektorů pro metody strojového učení jsou nejdříve zakódovány

substituované aminokyseliny v mutaci. Kódování jednotlivých aminokyselin je implementováno formou binárního vektoru s dvaceti hodnotami, ve kterém je pozice odpovídající konkrétní aminokyselině nastavena na hodnotu 1 a ostatní pozice v tomto vektoru mají nulovou hodnotu. Dvojice substituovaných aminokyselin v proteinu je tedy reprezentována 40 vstupními parametry, z nichž dva mají hodnotu 1 a zbytek je nulový.

Jednotlivé aminokyseliny, které se nacházejí v primární sekvenci proteinu v okolí místa mutace, jsou reprezentovány stejným formátem binárních vektorů, který je popsaný v předchozím odstavci. Pro zvolenou velikost okolí mutace v této aplikaci, která je deset předchozích a deset následujících aminokyselin od místa mutace, se tedy jedná o 400 vstupních parametrů. Pokud se mutace nachází na začátku nebo na konci polypeptidového řetězce proteinu, je u binárních vektorů, které reprezentují neexistující aminokyseliny, nastaveno všech dvacet hodnot nulových.

Tvar sekundární struktury proteinu, ve kterém se mutace nachází, je reprezentován binárním vektorem se sedmi položkami. Každá pozice odpovídá jednomu tvaru sekundární struktury, do kterých klasifikuje program DSSP. Tyto typy jsou následující: α-šroubovice, izolovaný β-můstek, β-skládáný list, Π-šroubovice, vodíkem vázaná otočka a ohyb.

Ostatní vstupní parametry jsou do výsledného vstupního vektoru vloženy bez dodatečného kódovaní, protože jsou reprezentovány pouze jedinou číselnou hodnotou. Výsledné vektory se vstupními daty pro metody strojového učení mohou mít rozdílnou velikost, protože různé metody využívají různé kombinace vstupních parametrů.

8.8 Postup predikce

Proces predikce je v implementovaném konsensuálním prediktoru rozdělen na dvě hlavní části, kterými jsou klasifikace a regrese. Před klasifikací aplikace vytvoří objekty pro vybrané metody strojového učení a jejich parametry načte ze souborů, které jsou uloženy v instalačním adresáři aplikace. Tyto souboru obsahují uložené serializované objekty, které byly vytvořeny při procesu trénovaní metod strojového učení a obsahují natrénované modely. Proces trénování a testovaní jednotlivých metod strojového učení a jejích kombinací je podrobně popsán v kapitole 9. Zde pouze uveďme, že výsledný konsensuální prediktor využívá pro klasifikace tři neuronové sítě, jednu instanci metody support vector machines a jednu instanci metody náhodný les. Pro regresi jsou využity čtyři neuronové sítě, dvě pro regresi u mutací klasifikovaných jako stabilizující a dvě u mutací klasifikovaných jako destabilizující. Postup predikce je následující:

1. **Klasifikace:** Každý z nezávislých klasifikátorů postupně klasifikuje všechny mutace ze vstupního souboru a uloží výsledky klasifikace pro jednotlivé mutace.

2. **Výpočet většinového konsensu:** Pro každou vstupní mutaci je vyhodnocena klasifikace z dílčích klasifikátorů a je vypočten většinový konsensus.

3. **Regrese:** V tomto kroku jsou klasifikované mutace vyhodnoceny neuronovými sítěmi určenými pro regresi. První dvojice sítí predikuje hodnotu $\Delta\Delta G$ pouze pro stabilizující mutace a druhá dvojice pro destabilizující mutace. Výsledná hodnota $\Delta\Delta G$ je vypočtena jako průměrná hodnota z výstupních hodnot odpovídajících neuronových sítí.

Kapitola 9

Testování prediktoru

Tato kapitola je zaměřena na popis testování jednotlivých metod strojového učení a jejich přesnosti při predikci stability proteinů. Dále obsahuje podrobnosti návrhu vybraných kombinací strojového učení, které byly uvažovány pro konsensuální predikci a naměřené výsledky přesnosti u těchto kombinací metod.

Na začátku kapitoly je popsáno rozložení dat v použitém datasetu a jeho rozdělení na část, která byla použita pro výběr vhodných metod strojového učení a jejich trénování a testování a část, která byla použita pro finální vyhodnocení přesnosti konsensuálního prediktoru. Poté je popsán podrobný popis procesu testování jednotlivých metod strojového učení. Následuje popis naměřených výsledků přesnosti metod pro klasifikaci i regresi, vyhodnocení těchto výsledků a návrh vhodných kombinací metod pro konsensuální predikci. Konec kapitoly je věnován popisu finální verze konsensuálního prediktoru a ověření přesnosti prediktoru na testovací části datasetu.

9.1 Rozdělení datové sady na trénovácí a testovací část

Použitý dataset, který byl podrobněji popsán v kapitole 7, obsahuje 1520 záznamů. Z toho 1234 záznamů tvoří destabilizující mutace proteinů a 286 záznamů popisuje stabilizující mutace proteinů. Stabilizujích mutací v datasetu je tedy necelých 20% z celkového počtu mutací.

Dataset byl rozdělen na část určenou pro testování a trénování jednotlivých metod strojového učení a testovací část určenou pro finální ověření přesnosti konsensuálního prediktoru. Testovací část dat obsahovala přibližně 20% z celkového počtu záznamů. Při rozdělování datasetu byl u obou částí přibližně zachován poměr stabilizujících a destabilizujících mutací z originálního datasetu, ale samotný výběr mutací v obou kategoriích byl proveden náhodně.

Trénovací část datasetu se skládála z 1200 položek s rozložením 959 destabilizujích a 241 stabilizujích mutací. Testovací část datasetu obsahovala 320 záznamů, ze kterých 275 popisovalo destabilizující mutace a 45 stabilizující mutace.

9.2 Testovaní metod strojového učení

Při testování byla použita metoda křížové validace, která se používá pro výběr a validaci modelů pro predikci. U křížové validace bylo použito rozdělení dat pro trénování na deset částí. Z důvodu náhodného rozdělení dat do jednotlivých částí a nevyváženosti použitého

datasetu, byla tato metoda vždy vyhodnocena v deseti opakováních při každém měření (výjimku tvoří metody k-nejbližších sousedů a rozhodovací strom, u kterých byl proces zopakován dvacetkrát). Naměřené hodnoty z jednotlivých opakování byly zprůměrovány a tím bylo dosaženo přesnějšího vyhodnocení. Pro každou metodu bylo testováno šest různých kombinací vstupních parametrů, které jsou dále popsány v podkapitole 9.2.1.

9.2.1 Testované kombinace vstupních parametrů

Pro testování jednotlivých metod strojového učení bylo navrženo šest různých kombinací vstupních parametrů. Podrobný popis vstupních parametrů a jejich rozdělení do jednotlivých skupin je uveden v kapitole 6.1.

První skupina parametrů obsahuje fyzikálně-chemické vlastnosti aminokyselin převzaté z práce [20]. Druhá skupina obsahuje doplňující fyzikálně-chemické vlastnosti aminokyselin. Třetí skupina vstupních parametrů obsahuje informace, které byly získány na základě struktury proteinu získané z databáze Protein Data Bank. Čtvrtá skupina parametrů popisuje primární strukturu proteinu v okolí mutace (neboli sekvenci okolních aminokyselin). Ve zbylé části kapitoly bude pro zjednodušení popisu využíváno číselné označení jednotlivých skupin vstupních parametrů (1, 2, 3 a 4). Navrhnuté kombinace vstupních parametrů jsou:

- **Kombinace 1, 2, 3, 4** - kombinace všech navrhnutých vstupních parametrů.
- **Kombinace 2, 3, 4** - kombinace druhé skupiny fyzikálně-chemických vlastností aminokyselin a strukturních a sekvenčních parametrů.
- **Kombinace 3, 4** - pouze parametry popisující strukturu a sekvenci proteinu v okolí mutace.
- **Kombinace 1, 3** - první skupina fyzikálně-chemických vlastností aminokyselin a informace o struktuře proteinu.
- **Kombinace 2, 3** - druhá skupina fyzikálně-chemických vlastností aminokyselin doplněná o informace popisující strukturu proteinu.
- **Kombinace 3** - pouze skupina vstupních parametrů, která obsahuje strukturní informace.

Kombinace, které nezahrnují čtvrtou skupinu vstupních parametrů (sekvence okolních aminokyselin), byly pro testování vybrány také z důvodu, že v použitém datasetu je velký nepoměr počtu mutací u jednotlivých proteinů (některé proteiny jsou uvedeny jednou, pro jiné jsou popsány desítky mutací). Z tohoto důvodu je teoreticky možné, že primární struktura proteinu by mohla zlepšit přesnost prediktoru při testování na dostupném datasetu, ale zhoršit schopnost generalizace prediktoru. Tím by se teoreticky zhoršila přesnost predikce u proteinů, které nebyly obsaženy v použitém datasetu. V dalších částech kapitoly budou navržené kombinace vstupních parametrů označovány kombinacemi čísel, které byly popsány v předchozím seznamu. Jedná se o kombinace čísel 1234, 234, 34, 13, 23 a 3.

9.2.2 Metriky meřené při testování

Pro měření přesnosti jednotlivých metod byly při testování použity metriky, které byly vybrány s ohledem na nevyvážený poměr stabilizujících a destabilizujících mutací v použitém datasetu:

- **Senzitivita** je metrika, která vyjadřuje podíl pozitivních záznamů, které byly správně klasifikovány jako pozitivní.

$$Senzitivita = \frac{TP}{TP + FN} \tag{9.1}$$

- **Specificita** je metrika, která vyjadřuje podíl správně klasifikovaných negativních záznamů.

$$Specificita = \frac{TN}{TN + FP} \tag{9.2}$$

- **Prediktivní hodnota pozitivního testu** (dále PPV) vyjadřuje podíl záznamů, které byly klasifikovány jako pozitivní a ve skutečnosti jsou také pozitivní. Tato metrika byla použita z důvodu menšího počtu stabilizujích mutací v použitém datasetu. Proto bylo vhodné sledovat, zda vyšší přesnost klasifikace stabilizujích mutací neznaméná také velký nárust nesprávně klasifikovaných destabilizujících mutací.

$$PPV = \frac{TP}{TP + FP} \tag{9.3}$$

- **F1 skóre** je metrika, která je definována jako harmonický průměr senzitivity a prediktivní hodnoty pozitivního testu. Byla vybrána z důvodu, že jde o vhodnou metriku v případě, kdy je počet pozitivních záznamů menší jak počet negativních záznamů. Také se jedná o vhodnou metriku, pokud potřebujeme sledovat, zda vyšší přesnost správné klasifikace pozitivních záznamů neznamená horší klasifikaci negativních záznamů.

$$F1\ skóre = \frac{2TP}{2TP + FP + FN} \tag{9.4}$$

- **Matthewsův korelační koeficient** (dále MCC) je metrika, která využívá všechny položky v klasifikační matici. Tato metrika je velice robustní a je vhodná i pro situace, kdy klasifikované třídy mají velmi rozdílnou velikost.

$$MCC = \frac{TP \cdot TN + FP \cdot FN}{\sqrt{(TP + FP)(TP + FN)(TN + FP)(TN + FN)}} \tag{9.5}$$

- **Celková přesnost** klasifikace byla měřena, protože se jedná o základní metriku. Tato metrika je velmi nevypovídající v případech, kdy jsou klasifikační třídy nevyvážené a je v této práci uváděna primárně ze zvyklostních důvodů a pro sledovaní celkové přesnosti byla použita metrika MCC.

$$Celková\ přesnost = \frac{TP + TN}{TP + TN + FP + FN} \tag{9.6}$$

Při porovnávání testovaných metod strojového učení a jejich kombinací, je v následující části práce používána primárně metrika MCC, protože se jedná o nejvhodnější metriku pro nevybalancované klasifikační třídy. Sekundárně byly sledovány metriky F1 skóre, senzitivita a specificita.

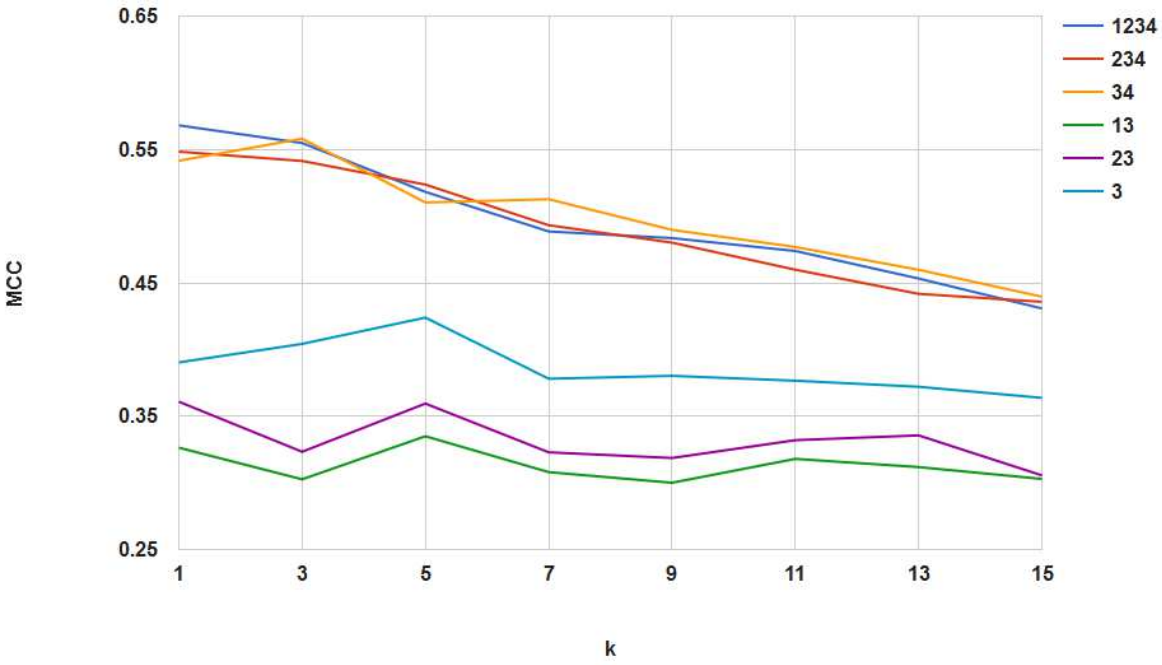
Pro měření velikosti chyby u metod strojového učení pro regresi byla použita střední čtvercová chyba (dále jen MSE).

9.2.3 Výsledná přesnost klasifikačních metod

V této části jsou uvedeny naměřené hodnoty sledovaných metrik pro testované metody strojového učení a vybrané kombinace vstupních parametrů. U každé metody je popsáno, jaké parametry a algoritmy byly použity při testování této metody. Testování metody a naměřené výsledky jsou u každé metody podrobně popsány. Následně je vždy uvedeno, jaká kombinace vstupních parametrů byla u dané metody zvolena jako nejvhodnější a finální. V části 9.3 je uvedeno srovnání jednotlivých metod strojového učení, trénovaných již pouze na zvolené kombinaci vstupních parametrů.

Algoritmus k-nejbližších sousedů

Tento algoritmus patří mezi nejjednodušší metody pro klasifikaci dat. Pro výpočet vzdálenosti jednotlivých vektorů byla v této práci použita standardní euklidovská vzdálenost, která je definována vztahem 4.1. Pro určení vhodného počtu nejbližších vektorů k, které tato metoda využívá pro klasifikaci vstupního vektoru do odpovídající třídy, byly pro každou kombinaci vstupních parametrů otestovány všechny liché hodnoty k z intervalu $(1, 15)$.



Obrázek 9.1: Hodnoty MCC pro $k = \{1, 3, 5, 7, 9, 11, 13, 15\}$

Po vyhodnocení naměřených hodnot metriky MCC pro vybrané hodnoty parametru k byla jako konečná hodnota tohoto parametru zvolena hodnota $k = 3$. Pro hodnotu $k = 1$ byla naměřena mírně lepší přesnost, ale nevýhoda spočívá v tom, že při této hodnotě parametru k je u tohoto algoritmu zvýšené riziko přeučení. U hodnot $k > 3$ algoritmus postupně ztrácí schopnost správně klasifikovat stabilizující mutace a v závislosti na tom se postupně zhoršuje hodnota metriky MCC. Toto chování je způsobeno rozdílem ve velikosti klasifikačních tříd v použitém datasetu. Při zvyšování počtu nejbližších vektorů k, se proto častěji stává, že z k nejbližších vektorů jich většina patří do skupiny destabilizujících mutací a tím se zvyšuje pravděpodobnost nesprávné klasifikace u stabilizujících mutací.

Kombinace parametrů	1234	234	34	13	23	3
Senzitivita	0.5270	0.5205	0.5195	0.3629	0.3826	0.4417
Specificita	0.9555	0.9521	0.9592	0.9056	0.9067	0.9206
F1 Skóre	0.6185	0.6085	0.6178	0.4174	0.4362	0.5025
MCC	0.5548	0.5414	0.5581	0.3026	0.3233	0.4042
PPV	0.7487	0.7324	0.7624	0.4914	0.5075	0.5832
Celková přesnost	0.8695	0.8655	0.8709	0.7966	0.8014	0.8244

Tabulka 9.1: Algoritmus k-nejbližších sousedů (k = 3)

Z naměřených hodnot u tohoto algoritmu vyplývá, že tato metoda dosahuje největší přesnosti klasifikace při použití kombinací vstupních parametrů, které obsahují informace o sekvenci proteinu v okolí mutace. Tyto tři kombinace vstupních parametrů mají přibližně stejnou přesnost. Při vynechání informací o sekvenci proteinu, dochází k značnému poklesu přesnosti klasifikace u této metody. Celkově metoda dosahuje největší přesnosti z porovnávaných metod. Nevýhodou je teoretická možnost horší generalizace u proteinů, které nebyly obsaženy v trénovacím datasetu. Tohoto teoretického závěru bylo dosaženo na základě úvahy zahrnující způsob klasifikace u tohoto algoritmu, reprezentaci a kódování vstupních parametrů, které popisují sekvenční informace v aplikaci a rozložení proteinů a pozicí mutací v jednotlivých proteinech v použitém datasetu. Jako konečná kombinace vstupních parametrů byla pro tuto metodu zvolena kombinace vstupních parametrů s označením 34.

Neuronová síť

Pro klasifikaci byla použita standardní dopředná neuronová síť, která obsahovala jednu skrytou vrstvu s 25 umělými neurony. Jako aktivační funkce byla použita standardní funkce sigmoid a pro inicilizaci vah před začátkem trénování sítě byla použita metoda Nguyen-Widrow. Jako učící algoritmus byl zvolem algoritmus Resilient backpropagation, který byl stručně popsán na konci podkapitoly 4.5 a jeho výhodou je vysoká rychlost učení. Síť byla trénována, dokud velikost celkové chyby neklesla pod hodnotu 2.5, nebo celkový počet epoch učení sítě nepřesáhl hodnotu 5000.

Kombinace parametrů	1234	234	34	13	23	3
Senzitivita	0.5553	0.5544	0.5564	0.5091	0.5058	0.5162
Specificita	0.9181	0.9200	0.9216	0.8991	0.9028	0.9081
F1 Score	0.5882	0.5921	0.5956	0.5328	0.5345	0.5484
MCC	0.4947	0.5003	0.5048	0.4230	0.4269	0.4454
PPV	0.6295	0.6363	0.6411	0.5591	0.5669	0.5858
Celková přesnost	0.8447	0.8466	0.8483	0.8207	0.8231	0.8294

Tabulka 9.2: Neuronová síť

Z naměřených hodnot metriky MCC je možné určit, že neuronová síť dosahuje největší přesnosti při použití kombinací vstupních parametrů, které obsahují informace o sekvenci aminokyselin v proteinu. Z těchto kombinací je mírně přesnější kombinace s označením 34, která neobsahuje informace o fyzikálně-chemických vlastnostech aminokyselin. Tato kombinace byla určena jako finální kombinace vstupních parametrů pro neuronovou síť.

V porovnání s ostatními testovanými metodami klasifikuje neuronová síť lépe stabilizující mutace a nedochází k výraznému poklesu přesnosti klasifikace u destabilizujících mutací.

Rozhodovací strom

Pro vytvoření rozhodovacího stromu byl využit algoritmus C4.5. Z naměřených výsledků pro tuto metodu lze odvodit, že v porovnání s ostatními testovanými metodami, klasifikuje rozhodovací strom lépe stabilizující mutace. U destabilizujících mutací však tato metoda dosahuje nejhorších výsledků mezi testovanými metodami a celková přesnost klasifikace je proto nízká. Nejlepší přesnosti klasifikace je dosaženo pro kombinaci vstupních parametrů 34, která obsahuje pouze informace o sekvenci a struktuře proteinu a která byla vybrána jako konečná kombinace vstupních parametrů pro tvorbu rozhodovacího stromu.

Kombinace parametrů	1234	234	34	13	23	3
Senzitivita	0.5351	0.5564	0.5587	0.5525	0.5595	0.5668
Specificita	0.8791	0.8899	0.8998	0.8667	0.8730	0.8852
F1 Score	0.5307	0.5581	0.5708	0.5304	0.5419	0.5603
MCC	0.4117	0.4476	0.4662	0.4072	0.4228	0.4484
PPV	0.5267	0.5603	0.5838	0.5105	0.5258	0.5544
Celková přesnost	0.8100	0.8230	0.8313	0.8036	0.8100	0.8213

Tabulka 9.3: Rozhodovací strom

Náhodný les

Algoritmus náhodný les interně pro tvorbu jednotlivých rozhodovacích stromů používá metodu C4.5, která je v použitém frameworku identická s metodou pro vytváření samostatného rozhodovacího stromu. Počet rozhodovacích stromů v náhodném lese byl nastaven na 20.

Kombinace parametrů	1234	234	34	13	23	3
Senzitivita	0.4436	0.5000	0.5083	0.4436	0.5055	0.5046
Specificita	0.9496	0.9569	0.9503	0.9380	0.9443	0.9426
F1 Score	0.5394	0.5983	0.5958	0.5247	0.5850	0.5823
MCC	0.4694	0.5360	0.5267	0.4425	0.5101	0.5058
PPV	0.6888	0.7450	0.7202	0.6433	0.6957	0.6889
Celková přesnost	0.8480	0.8652	0.8615	0.8387	0.8561	0.8547

Tabulka 9.4: Náhodný les

Tato metoda dosahuje největší přesnosti klasifikace u kombinací vstupních parametrů s označením 234 a 34. Pro konečné použití této metody však byla zvolena kombinace parametrů 23, pro kterou byla změřena pouze mírně horší přesnost klasifikace v porovnání s předchozí dvojící kombinací. Primárním důvodem výběru této kombinace vstupních parametrů bylo zvýšení diverzity v konečném konsensuálním prediktoru. Specifické chování bylo u této metody pozorováno při použití první skupiny vstupních parametrů, která značně snižuje přesnost klasifikace u této metody. V porovnání s ostatními metodami má metoda

náhodný les vysokou přesnost klasifikace u destabilizujících mutací a současně nedochází k výraznému zhoršení klasifikace stabilizujících mutací.

Support vector machines

U metody support vector machines (dále SVM) byly testovány dvě varianty jádrových funkcí. Jednalo se stardardní lineární jádrovou funkci a gaussovskou jádrovou funkci. Váha stabilizujích mutací pro proces učení byla nastavena na větší hodnotu než váha destabilizujících mutací, aby byl algorismus schopen lépe pracovat s nevyváženými daty z použitého datasetu. Pro učení modelu byla použita metoda nejmenších čtverců, která je označována jako LS-SVM.

Kombinace parametrů	1234	234	34	13	23	3
Senzitivita	0.5154	0.5091	0.5021	0.4635	0.4842	0.4772
Specificita	0.9125	0.9210	0.9254	0.8756	0.8725	0.8717
F1 Score	0.5531	0.5584	0.5583	0.4733	0.4862	0.4802
MCC	0.4528	0.4638	0.4668	0.3446	0.3578	0.3506
PPV	0.5970	0.6183	0.6288	0.4837	0.4883	0.4833
Celková přesnost	0.8328	0.8383	0.8404	0.7928	0.7945	0.7925

Tabulka 9.5: Support vector machine - Lineární jádrová funkce

Kombinace parametrů	1234	234	34	13	23	3
Senzitivita	0.4456	0.4369	0.4402	0.4544	0.4714	0.4788
Specificita	0.9615	0.9691	0.9687	0.8879	0.8769	0.8795
F1 Score	0.5574	0.5602	0.5627	0.4781	0.4806	0.4889
MCC	0.5015	0.5150	0.5167	0.3563	0.3534	0.3641
PPV	0.7444	0.7807	0.7797	0.5047	0.4903	0.4997
Celková přesnost	0.8579	0.8622	0.8626	0.8008	0.7954	0.7990

Tabulka 9.6: Support vector machine - Gaussovská jádrová funkce

U obou testovaných jádrových funkcí měla metoda SVM nejvyšší přesnost klasifikace při použití kombinací vstupních parametrů 234 a 34. U obou variant jádrových funkcí bylo pozorováno výrazné snížení přesnosti klasifikace, pokud je vynechána sekvence proteinu. Pro finální srovnání metod strojového učení byla u obou variant jádrových funkcí vybrána kombinace vstupních parametrů 34.

Při použití gaussovské jádrové funkce měla tato metoda velmi vysokou přesnost klasifikace destabilizujících mutací, ale menší schopnost klasifikovat korektně stabilizující mutace. Při využití lineární jádrové funkce měla metoda vyšší přesnost klasifikace stabilizujích mutací, ale současně došlo k výraznému snížení přesnosti klasifikace pro destabilizující mutace. Z tohoto důvodu měla metoda SVM při použití lineární jádrové funkce nižší celkovou přesnost než při použití gaussovské jádrové funkce.

9.2.4 Výsledná přesnost regresních metod a jejich kombinací

Pro predikci hodnoty $\Delta\Delta G$ byly pro testování a případné použití ve výsledném konsensuálním prediktoru navrhnuty dvě metody strojového učení pro regresi. Jednalo se o neuronovou

síť a modifikaci metody support vector machines určenou pro regresi. Přesnost regrese byla testována pro tyto dvě uvedené metody a dále pro vybranou trojici jejich kombinací. Podrobný popis parametrů testovaných metod je uveden v následujím seznamu:

- **Neuronová síť**: Byla použita dopředná neuronová síť s jednou vrstvou skrytých neuronů, která se skládala z pěti skrytých neuronů. Jako aktivační funkce byla použita funkce sigmoid. Pro inicializaci vah byla použita metoda Nguyen-Widrow a pro učení neuronové sítě byl použit algoritmus Resilient backpropagation.

  Proces učení byl ukončen, pokud celková chyba sítě klesla pod určenou hodnotu nebo bylo dosaženo 5000 epoch učení. Maximální hodnota celkové chyby byla určena následujícím výpočtem: *velikost trénovacího datasetu*/500.

  Výstupy u trénovacích vektorů byly před začátkem trénování neuronové sítě transformovány do intervalu $(0, 1)$, protože obor hodnot aktivační funkce sigmoid je $(0, 1)$. Výstupní hodnoty sítě byly transformovány zpět do původních rozsahů, které byly určeny na základě reálných hodnot $\Delta\Delta G$ v použitém datasetu. Pro stabilizující mutace byl tento rozsah $(-11, 0)$ a pro destabilizující mutace $(0, 31)$.

- **Support vector machines**: Pro testování metody support vector machines pro regresi byla použita gaussovská jádrová funkce. Pro učení metody byla zvolena modifikace algoritmu sekvenční minimální optimalizace určená pro regresi.

Přesnost regrese byla testována samostatně pro stabilizující a destabilizující mutace. Tento přístup teoreticky umožňuje dosáhnout větší přesnosti regrese a také použití odlišných metod strojového učení pro regresi u stabilizujících a destabilizujících mutací a tím zvýšení diverzity prediktoru.

Testováno bylo pět kombinací navrhnutých metod strojového učení. U kombinací, které byly složeny z více metod, byly konečné výstupní hodnoty regrese vypočteny jako průměr dílčích výsledků regrese. Testované kombinace metod byly následující: samostatná neuronová síť (NS), samostatná metoda support vector machines, dvojice neuronových sítí, neuronová síť a metoda support vector machines a dvojice neuronových sítí s metodou support vector machines.

Kombinace parametrů	1234	234	34	13	23	3
NS	0.8303	0.9097	0.9854	1.0436	1.0994	1.0830
SVM	0.8062	0.8157	0.8198	0.8719	0.8610	0.8754
dvojice NS	0.8235	0.8637	0.9017	1.0356	1.0542	1.1076
SVM, NS	0.8025	0.8081	0.8062	0.8998	0.8979	0.9266
SVM, dvojice NS	0.7969	0.8121	0.8414	0.9416	0.9458	0.9674
Kombinace parametrů	1234	234	34	13	23	3
NS	5.5887	6.2176	6.1881	4.7311	4.7965	4.7377
SVM	5.5766	5.7021	5.8214	5.7398	5.7115	5.7847
dvojice NS	5.4522	5.8029	5.7743	4.6168	4.5331	4.6699
SVM, NS	5.3776	5.4992	5.4927	4.9259	4.8629	4.8486
SVM, dvojice NS	5.2322	5.4406	5.5290	4.7147	4.6805	4.8139

Tabulka 9.7: Velikost střední čtvercové chyby pro stabilizující a destabilizující mutace.

První z předchozích tabulek obsahuje hodnoty střední čtvercové chyby pro regresi u stabilizujících mutací. Druhá tabulka obsahuje hodnoty chyby pro destabilizující mutace.

U destabilizujích mutací byla naměřena vyšší velikost chyby. Tento výsledek plyne z většího počtu hodnot $\Delta\Delta G$ u destabilizujících mutací, které se velmi lišily od průměrné hodnoty. U neuronových sítí také z maximální velikosti celkové chyby sítě, která byla určena na základě počtu odpovídajících mutací v trénovacím datasetu a pro destabilizující mutace byla zhruba pětkrát větší.

Největší přesnosti regrese pro stabilizující mutace bylo dosaženo při použití kombinace vstupních parametrů 1234, u které všech pět testovaných kombinací metod strojového učení dosahovalo podobné velikosti celkové chyby. Z těchto kombinací byla největší přesnost naměřena u kombinace dvou neuronových sítí a metody support vector machines.

U destabilizujících mutací bylo největší přesnosti regrese dosaženo u kombinace dvou neuronových sítí, které používaly kombinaci vstupních parametrů s označením 23. Dále bylo zjištěno, že všechny kombinace obsahující neuronovou síť měli podobnou přesnost u všech kombinací vstupních parametrů, které neobsahovaly informace o sekvenci proteinu.

Pro finální verzi regrese v konsensuálním prediktoru byla vybrána kombinace dvou neuronových sítí pro predikci hodnoty $\Delta\Delta G$ u stabilizujících i destabilizujících mutací. U stabilizujích mutací měla tato kombinace pouze mírně horší přesnost než kombinace metody support vector machines a dvojice neuronových sítí. Druhým faktorem pro toto rozhodnutí bylo to, že implementace metody support vector machines pro regresi byla v použitém frameworku Accord.NET obsažena pod GPL licencí (framework samotný má licenci LGPL), z čehož plyne, že v případě použití této třídy by pod licencí GPL musela být uvolněna i tato práce.

9.3 Kombinace metod pro konsensuální klasifikaci a jejich testování

Před výběrem klasifikačních metod, které byly použity pro návrh testovaných kombinací metod pro konsensuální klasifikaci, byla nejdříve vypočtena míra korelace výsledků klasifikace mezi všemi testovanými metodami pro klasifikaci. Zjištěné hodnoty korelace jsou uvedeny v tabulce 9.8 a jsou vyjádřeny ve formě fí koeficientu. Při výpočtu korelace byla použita metoda křížové validace, která byla desetkrát zopakována. Výsledné hodnoty koeficientu fí představují průměrnou hodnotu tohoto koeficientu ze všech opakovaní. Míra korelace byla měřena z důvodu, že u konsensuální klasifikace je vhodné, aby výsledky dílčí klasifikace u použitých metod nebyly příliš podobné.

Pro testované metody bylo v tabulce 9.8 a v dalších tabulkách v této podkapitole zvoleno následující označení: kNN - algoritmus k-nejbližších sousedů, NS - neuronová síť, RS - rozhodovací strom, NL - náhodný les, SVM-L (SVM-G) metoda support vector machines s lineární (resp. gaussovskou) jádrovou funkcí.

	kNN	NS	RS	NL	SVM-L	SVM-G
kNN	1	0.2548	0.1774	0.6175	0.2292	0.2825
NS	0.2548	1	0.3847	0.2264	0.5078	0.5417
RS	0.1774	0.3847	1	0.1775	0.4113	0.4866
NL	0.6175	0.2264	0.1775	1	0.2122	0.2448
SVM-L	0.2292	0.5078	0.4113	0.2122	1	0.7599
SVM-G	0.2825	0.5417	0.4866	0.2448	0.7599	1

Tabulka 9.8: Fí koeficient vyjadřující míru korelace mezi klasifikací u testovaných metod

Při měření hodnoty korelace, byla u všech metod použita kombinace vstupních parametrů s označením 34, která při testovaní samostatných metod dosahovala vysoké přesnosti klasifikace. Výjímku tvoří metoda náhodný les, u které byla z důvodu větší diverzity konsesuální klasifikace zvolena kombinace vstupních parametrů 23.

Z naměřených hodnot koeficientu fí plyne, že výsledná klasifikace u metod k-nejbližších sousedů a náhodný les je relativně podobná. Dále je možné říci, že oba testované typy jádrové funkce u metody support vector machines klasifikovaly většinu trénovacích vektorů podobně. Pro doplnění informací je dále vhodné uvést, že průměrná hodnota korelace pro dvojici neuronových sítí byla 0.2571.

Konsensuální klasifikace

Na základě míry korelace mezi jednotlivými metodami a přesnost klasifikace jednotlivých metod byla zvolena následující trojice metod strojového učení: metoda náhodný les, která při testovaní dosáhla dobré přesnosti klasifikace v porovnání s ostatními metodami. Protože tato metoda má velkou míru korelace s algoritmem k-nejbližších sousedů, nebyl tento algoritmus navrhnut pro konsensuální klasifikace. Druhou metodou byla neuronová síť. Třetí metodou byla metoda support vector machines s gaussovskou jádrovou funkcí, která z testovaných metod klasifikovala nejlépe destabilizující mutace.

Byly navrženy tři kombinace těchto metod, které byly testovány v deseti opakovaních metodou křížové validace. Všechny kombinace obsahovaly lichý počet klasifikátorů, aby nedocházelo k nejednoznačnostem při výpočtu většinového konsensu. První kombinací je trojice neuronových sítí. Druhou kombinaci tvoří neuronová síť, náhodný les a metoda support vector machines. Třetí kombinace je identická s druhou kombinací, pouze místo jedné neuronové sítě obsahuje trojici neuronových sítí.

Kombinace parametrů	3x NS	NS + SVM-G + NL	3x NS + SVM-G + NL
Senzitivita	0.5550	0.4732	0.5334
Specificita	0.9277	0.9680	0.9485
F1 Score	0.6024	0.5913	0.6137
MCC	0.5160	0.5429	0.5434
PPV	0.6592	0.7883	0.7228
Celková přesnost	0.8529	0.8687	0.8651

Tabulka 9.9: Naměřené hodnoty metrik pro konsensuální klasifikaci

Z naměřených hodnot sledovaných metrik lze pro testované kombinace metod strojového učení odvodit následující informace. První kombinace metod, kterou je trojice neuronových sítí, klasifikuje nejlépe stabilizující mutace, ale má menší úspěšnost při klasifikaci destabilizujících mutací. Druhá kombinace metod klasifikuje nejpřesněji destabilizující mutace, ale v porovnání s dvojicí ostatních kombinací má malou přesnost při klasifikaci stabilizujících mutací. Třetí testovaná kombinace metod představuje rovnováhu mezi první a druhou kombinací. Stabilizují mutace klasifikuje pouze mírně hůře než první kombinace. Přesnost u klasifikace destabilizujích mutací je zhruba ve středu intervalu mezi přesností klasifikace destabilizujích mutací u první a druhé kombinace. Třetí kombinace vykazuje také nejlepší naměřené hodnoty u metriky F1 Score, která vyjadřuje přesnost při klasifikaci stabilizujících mutací a u metriky MCC, která byla v této práci použita pro měření celkové přesnosti klasifikace.

Pro finální implementaci konsensuálního prediktoru byla zvolena třetí kombinace metod, která je tvořena trojící neuronových sítí, metodou support vector machines a metodou náhodný les. Tato kombinace metod strojového učení byla zvolena jako kompromis mezi přesností klasifikace stabilizujících a destabilizujích mutací. V porovnání s ostatními testovanými kombinacemi metod dosahuje také nejlepší hodnoty metriky MCC.

9.4 Finální přesnost konsensuálního prediktoru

Finální implementace konsensuálního prediktoru využívá pro klasifikaci trojici neuronových sítí, jednu instanci metody support vector machines s gaussovskou jádrovou funkcí a jednu instanci metody náhodný les. Výsledné hodnoty klasifikace jsou vypočteny jako většinový konsensus z výstupních hodnot pěti nezávislých klasifikátorů. Pro regresi jsou v konsensuálním prediktoru využity čtyři neuronové sítě. Dvě z těchto neuronových sítí jsou specializovány na predikci hodnoty $\Delta\Delta G$ pro mutace, které byly klasifikovány jako stabilizující a dvě pro mutace, které byly klasifikovány jako destabilizující. Výsledná predikována hodnota $\Delta\Delta G$ je pro každou mutaci vypočtena jako průměr výsledných hodnot odpovídající dvojice neuronových sítí.

Pro vyhodnocení přesnosti konečného konsensuálního prediktoru byl použit testovací dataset, který byl vyčleněn z originálního datasetu před testováním a trénováním jednotlivých metod. Podrobný popis testovacího datasetu je uveden v části 9.1. Výsledné hodnoty klasifikace, které jsou uvedeny ve formě standardní matice záměn, jsou obsaženy v tabulce 9.10. Výsledná střední čtvercová chyba regrese byla pro testovací dataset 9.0559.

	Stabilizující	Destabilizujicí
Predikce - stabilizující	28	32
Predikce - destabilizující	17	243

Tabulka 9.10: Výsledná přesnost klasifikace konsesuálního prediktoru

Na základě naměřených hodnot, které jsou uvedeny v předchozí maticí záměn, byly vypočteny zvolené metriky, které jsou popsány v části 9.2.2. Jejich hodnoty jsou uvedeny v tabulce 9.11.

Senzitivita	0.6222
Specificita	0.8836
F1 Score	0.5333
MCC	0.4506
PPV	0.4667
Celková přesnost	0.8469

Tabulka 9.11: Hodnoty použitých metrik vypočtené z tabulky 9.10

Kapitola 10

Závěr

Primárním cílem této diplomové práce bylo otestovat různé metody strojového učení pro klasifikaci i regresi, navrhnout a otestovat vhodné parametry proteinů a aminokyselin a z vhodných kombinací metod strojového učení a navržených parametrů proteinů vytvořit konsensuální prediktor.

Testované metody strojového učení byly pro klasifikaci metody support vector machines, rozhodovací strom, algoritmus náhodný les, který vychází z rozhodovacích stromů, neuronové sítě a algoritmus k-nejbližších sousedů. Pro regresi byly testovány neuronové sítě a varianta metody support vector machines pro regresi.

Pro nalezení vhodných parametrů proteinů pro predikci byly nejprve navrženy rozdílné kombinace fyzikálně-chemických, strukturních a statistických parametrů proteinů. Pro každou metodu strojového učení byly testovány všechny navržené kombinace parametrů, aby byly nalezeny optimální parametry u každé testované metody. Bylo zjištěno, že nejvýznamnější vliv na přesnost predikce mají parametry, které popisují strukturu a sekvenci proteinu. Fyzikálně-chemické vlastnosti aminokyselin měly u většiny testovaných metod zanedbatelný vliv na výslednou přesnost.

Z testovaných metod strojového učení byly jako nejvhodnější metody pro klasifikaci mutací vybrány metody náhodný les, support vector machines a neuronové sítě. Poté byly testovány vybrané kombinace těchto metod, aby byla nalezena vhodná kombinace pro konsensuální klasifikaci. Jako nejvhodnější kombinace byla určena trojice neuronových sítí doplněná metodou support vector machines a metodou náhodný les. Tato kombinace představovala optimální vyvážení přesnosti klasifikace stabilizujících a destabilizujících mutací. Finální konsensuální klasifikace byla otestována pomocí testovacího datasetu, pro který dosáhla 62% přesnosti klasifikace u stabilizujících mutací a 89% přesnosti klasifikace pro destabilizující mutace.

Pro regresi byla jako nejvhodnější kombinace zvolena dvojice neurových sítí pro stabilizující mutace a dvojice neuronových sítí pro destabilizující mutace. Střední čtvercová chyba finální regrese byla pro testovací část datesetu 9.0559.

Výsledným produktem této práce byla aplikace pro predikci vlivu jednobodových mutací v proteinech na stabilitu proteinů. Vytvořený prediktor klasifikuje mutace na stabilizující a destabilizující a také predikuje hodnotu $\Delta\Delta G$. Aplikace byla implementována ve formě konsensuálního prediktoru, který využívá různé metody strojového učení s různými vstupními parametry.

Literatura

[1] Alpaydin, E.: *Introduction to Machine Learning second edition*. The MIT Press, 2010, ISBN 978-0-262-012343-0.

[2] Altschul, S. F.; Gish, W.; Miller, W.; aj.: Basic Local Alignment Search Tool. *Journal of Molecular Biology*, 1990.

[3] Amino Acids Reference Chart.
URL http://www.sigmaaldrich.com/life-science/metabolomics/learning-center/amino-acid-reference-chart.html

[4] Ball, D. W.; Will, J. W.; Scot, R. J.: *The Basics of General, Organic, and Biological Chemistry, v. 1.0*. Flat World Knowledge, 2011, ISBN 978-1-4533-2788-3.

[5] Bava, K. A.; Gromiha, M. M.; Uedaira, H.; aj.: ProTherm, version 4.0: thermodynamic database for proteins and mutants. *Nucleic Acids Research*, ročník 32, 2004.

[6] Breiman, L.: Random Forests. *Machine Learning*, ročník 45, 2001.

[7] Bruce, A.; Alexander, B. D. J.; aj., L. J.: *Základy buněčné biologie*. Espero Publishing, 1998, ISBN 80-902906-2-0.

[8] Capriotii, E.; Fariselli, P.; Rossi, I.; aj.: A three-state prediction of single point mutations on protein stability changes. *BMC Bioinformatics*, 2008.

[9] Cheng, J.; Randall, A.; Baldi, P.: Prediction of protein stability changes for single site mutations using support vector. *Proteins: Structure, Function, and Bioinformatics*, 2006.

[10] Cock, P. J. A.; Antao, T.; Chang, J. T.; aj.: Biopython: freely available Python tools for computational molecular biology and bioinformatics. *Bioinformatics*, ročník 25, 2009: s. 1422–1423.

[11] Collantes, E. R.; Dunn, W. J.: Amino Acid Side Chain Descriptors for Quantitative Structure-Activity Relationship Studies of Peptide Analogs. *Journal of Medicinal Chemistry*, ročník 38, č. 14, 1995: str. 2705–2713.

[12] Crick, F.: On Protein Synthesis. *The Symposia of the Society for Experimental Biology 12*, 1958.

[13] Dehouck, Y.; Kwasigroch, J. M.; Gilis, D.; aj.: PoPMuSiC 2.1: a web server for the estimation of protein stability changes upon mutation and sequence optimality. *BMC Bioinformatics*, 2011.

[14] Devine, R.: Proteins: Binding and interactions. [cit. 2017-05-06].
URL https://www.fastbleep.com/biology-notes/40/1175

[15] Edgar, R. C.: MUSCLE: multiple sequence alignment with high accuracy and high throughput. *Nucleic Acids Res*, ročník 32, č. 5, 2004.

[16] Eisenberg, D.; Schwarz, E.; Komaromy, M.; aj.: Analysis of membrane and surface protein sequences with the hydrophobic moment plot. *Journal of Molecular Biology*, 1984.

[17] Flax, M.: *Akcelerované neuronové sítě*. Bakalářská práce, Vysoké učení technické v Brně, Fakulta informačních technologií, 2015.

[18] Folkman, L.; Stanic, B.; Sattar, A.: Towards sequence-based prediction of mutation-induced stability changes in unseen non-homologous proteins. *BMC Genomics*, 2014.

[19] Gromiha, M. M.: *Protein Bioinformatics From Sequence to Function*. Elsevier Inc., 2010, ISBN 978-8-1312-2297-3.

[20] Gromiha, M. M.; Oobatake, M.; Sarai, A.: Important amino acid properties for enhanced thermostability from mesophilic to thermophilic proteins. *Biophysical Chemistry*, 1999.

[21] Huang, L.; Gromiha, M.: iPTREE-STAB: interpretable decision tree based method for preong protein stability changes upon mutations. *Bioinformatics*, 2007.

[22] International Human Genome Sequencing Consortium: Initial sequencing and analysis of the human genome. *Nature*, 2001.

[23] Kabsch, W.; Sander, C.: Dictionary of protein secondary structure: pattern recognition of hydrogen-bonded and geometrical features. *Biopolymers*, ročník 22, 1983.

[24] Khatun, J.; Khare, S. D.; Dokholyan, N. V.: Can Contact Potentials Reliably Predict Stability of Proteins? *Journal of Molecural Biology*, 2004.

[25] Laimer, J.; Hofer, H.; Fritz, M.; aj.: MAESTRO - multi agent stability prediction upon point mutations. *BMC Bioinformatics*, 2015.

[26] Šíma, J.; Neruda, R.: *Teoretické otázky neuronových sítí*. matfyzpress, 1996, ISBN 80-85863-18-9.

[27] Maas, A. L.; Hannun, A. Y.; Ng, A. Y.: Rectifier Nonlinearities Improve Neural Network Acoustic Models. 2013.

[28] Matthews, B. W.; Nicholson, H.; Becktel, W. J.: Enhanced protein thermostability from site-directed mutations that decrease the entropy of unfolding. *Proceedings of the National Academy of Sciences of the United States of America*, ročník 84, č. 19, 1987: str. 6663–6667.

[29] Mitchell, J. B. O.: Machine learning methods in chemoinformatics. *Wiley interdisciplinary reviews: Computational Molecular Science*, ročník 4, č. 5, 2014.

[30] Musil, M.; Stourac, J.; Bendl, J.; aj.: FireProt: web server for automated design of thermostable proteins. *Nucleic Acids Res*, 2017.

[31] OpenCV 2.4.13.2 documentation: Introduction to Support Vector Machines. [cit. 2017-05-02].
URL http://docs.opencv.org/2.4/doc/tutorials/ml/introduction_to_svm/introduction_to_svm.html

[32] Pace, C. N.; Scholtz, J. M.; Grimsley, G. R.: Forces Stabilizing Proteins. *FEBS Letters*, 2014.

[33] Parthiban, V.; Gromiha, M.; Schomburg, D.: CUPSAT: Prediction of protein stability upon point mutations. *Nucleic Acids Research*, 2006.

[34] Pires, D. E. V.; Ascher, D. B.; Blundenll, T. L.: mCSM: predicting the effects of mutations in proteins using graph-based signatures. *Bioinformatics*, 2014.

[35] Riedmiller, M.; Braun, H.: RPROP - A Fast Adaptive Learning Algorithm. Technická zpráva, Proc. of ISCIS VII), Universitat, 1992.

[36] RosettaCommons: Rosetta Software.
URL https://www.rosettacommons.org/

[37] Schymkowitz, J.; Borg, J.; Stricher, F.; aj.: The FoldX web server: an online force field. *Nucleic Acid Research*, ročník 33, 2005.

[38] Simpson, R. J.: Protein and Proteomics A Laboratory manual. Table: Properties of amino acids.
URL
http://www.proteinsandproteomics.org/content/free/tables_1/tables.html

[39] Smola, A. J.; Schölkopf, B.: A tutorial on support vector regression. *Statistics and Computing*, ročník 14, č. 3, 2004.

[40] Suykens, J. A. K.; Vandewalle, J.: Least Squares Support Vector Machine Classifiers. *Neural Processing Letters*, ročník 9, č. 3, 1999.

[41] Teplá, M.: Nukleové kyseliny - Translace. [cit. 31.12.2016].
URL http://www.studiumbiochemie.cz/translace.html

[42] Tian, J.; Wu, N.; Chu, X.; aj.: Predicting changes in protein thermostability brought about by single- or multi-site mutations. *BMC Bioinformatics*, 2010.

[43] Touw, W. G.; Baakman, C.; Black, J.; aj.: A series of PDB related databases for everyday needs. *Nucleic Acids Research*, ročník 43, 2015.

[44] Voet, D.; Voet, J. G.; Pratt, C. W.: *Fundamentals of Biochemistry: Life at the Molecular Level, 3rd Edition*. John Wiley & Sons, Inc., 2008, ISBN 0470129301.

[45] Vondrejs, V.: Co je to gen? *Vesmír*, ročník 91, 2 2012, [cit. 31.12.2016].
URL http://casopis.vesmir.cz/clanek/co-je-to-gen-%282%29

[46] Yin, S.; Ding, F.; Dokholyan, N. V.: Eris: an automated estimator of protein stability. *Nature Methods*, ročník 4, 2007.

Přílohy

Příloha A

Obsah CD

`/Data`

`data.csv` - použitý dataset.

`data-train.csv` - trénovací část datasetu.

`data-test.csv` - testovací část datasetu.

`/Source` - zdrojové soubory aplikace pro kompilaci nástrojem Visual Studio 2015. V podadresářích, do kterých Visual Studio uloží zkompilované soubory (`/DP-Flax/bin/Debug` a `/DP-Flax/bin/Release`) jsou tyto podadresáře a soubory, které jsou nutné pro správný chod aplikace:

`/Script`

`AAproperties.csv` - soubor obsahující hodnoty použitých fyzikálně-chemických vlastností aminokyselin.

`dssp.exe` - program DSSP.

`muscle.exe` - program MUSCLE.

`prepareData.py` - skript pro přípravu dat.

`/Agents` - v tomto adresáři jsou uloženy soubory, které obsahují serializované objekty reprezentující natrénované metody strojového učení.

`/Source-text` - zdrojové soubory textové zprávy pro program LaTeX.

`DP-Flax.pdf` - pdf soubor obsahující textovou zprávu pro tuto diplomovou práci.

`setup.exe` - instalační soubor této aplikace - varianta .exe.

`Setup.msi` - instalační soubor této aplikace - varianta .msi.