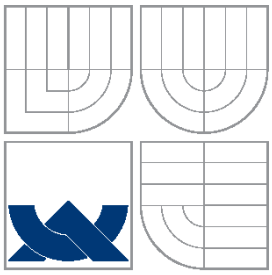


VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV INFORMAČNÍCH SYSTÉMŮ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF INFORMATION SYSTEMS

EVOLUČNÍ STRATEGIE V ÚLOZE PREDIKCE VLIVU AMINOKYSELINOVÝCH MUTACÍ NA STABILITU PROTEINU

PREDICTION OF PROTEIN STABILITY UPON MUTATIONS USING EVOLUTION STRATEGY

DIPLOMOVÁ PRÁCE
MASTER'S THESIS

AUTOR PRÁCE — Bc. DAVID PAVLÍK
AUTHOR

VEDOUCÍ PRÁCE — Ing. JAROSLAV BENDL
SUPERVISOR

BRNO 2014

Abstrakt

Tato práce se zabývá otázkou predikce změn stability proteinů v důsledku aminokyselinových mutací. Cílem je vytvořit meta-klasifikátor, který bude využívat výsledky predikcí vybraných nástrojů, použít evoluční strategii pro přiřazení vah jednotlivým nástrojům a dosáhnout tak větší úspěšnosti predikce než při použití nástrojů samostatně. Bylo vybráno celkem pět dostupných nástrojů, jejichž výsledky predikcí byly váhovány. Jsou zde zkoumány a porovnávány dvě odlišné metody evoluční strategie. První je evoluční strategie s pravidlem 1/5 a druhou je evoluční strategie s autoevolucí řídících parametrů typu 2. Pro trénování a následné ověření úspěšnosti navrženého meta-klasifikátoru byly vytvořeny dvě nezávislé sady mutací. Z provedených experimentů a dosažených výsledků byl zjištěn možný přínos evoluční strategie, ovšem za podmínek pečlivého výběru sady nástrojů a datových sad pro trénování a testování.

Abstract

This master's thesis deals with the matter of predicting the effects of aminoacid substitutions on protein stability. The main aim is to design meta-classifier that combines the results of the selected prediction tools. An evolution strategy was used to find the best weights for each of the selected tools with the aim of achieving better prediction performance compared to that achieved by using these tools separately. Five different and obtainable prediction tools were selected and their prediction outputs were weighted. Two different approaches of evolution strategy are investigated and compared: evolution strategy with the 1/5-rule and evolution strategy with the type 2 of control parameters self-adaptation. Two independent datasets of mutations were created for training and evaluating the performance of designed meta-classifier. The performed experiments and obtained results suggest that the evolution strategy could be considered as a beneficial approach for prediction of protein stability changes. However, the special attention must be paid to careful selection of tools for integration and compilation of training and testing datasets.

Klíčová slova

Evoluční strategie, protein, stabilita, predikce, aminokyselina, mutace.

Keywords

Evolution strategy, protein, stability, prediction, aminoacid, mutation.

Citace

David Pavlík: Evoluční strategie v úloze predikce vlivu aminokyselinových mutací na stabilitu proteinu, diplomová práce, Brno, FIT VUT v Brně, 2014

Evoluční strategie v úloze predikce vlivu aminokyselinových mutací na stabilitu proteinu

Prohlášení

Prohlašuji, že jsem tento semestrální projekt vypracoval samostatně pod vedením pana Ing. Jaroslava Bendla. Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

.....
David Pavlík
26. května 2014

Poděkování

Rád bych tímto poděkoval především vedoucímu práce Ing. Jaroslavu Bendlovi za jeho profesionální vedení a aktivní pomoc při řešení problémů. Také bych zde rád poděkoval za možnost využít distribuovanou výpočetní infrastrukturu MetaCentra (projekt LM201005) k ohodnocení datových sad proteinových mutací pomocí testovaných nástrojů.

Obsah

Kapitola 1

Úvod

Tato práce je zaměřena na problematiku predikce vlivu aminokyselinových mutací na stabilitu proteinu. Vzhledem k úzké provázanosti stability proteinů s jejich funkcí je toto téma poměrně zásadní při studiích chorob a návrhu či vývoji nových proteinů. Ačkoliv již existuje velké množství nástrojů pro tuto predikci, každý má své výhody a nevýhody a nelze tedy brát výsledky jednoho nástroje jako spolehlivé. Na základě tohoto faktu je v této práci zkoumán možný přínos technik evolučních algoritmů, konkrétně pak evoluční strategie, právě v úloze zlepšení predikce vlivu aminokyselinových mutací na stabilitu proteinu. Cílem je tedy vytvoření meta-klasifikátoru využívající výsledky predikcí vybrané sady nástrojů a přiřazující jim váhy získané pomocí evoluční strategie.

V druhé kapitole je předmětem studie problematika proteinů, popis jejich vzniku při proteosyntéze, rozbor jejich struktury a se stabilitou související aminokyselinové mutace. Pozornost je věnována především jednobodovému nukleotidovému polymorfismu, jehož vliv na stabilitu proteinu byl předmětem predikce analyzovaných nástrojů, resp. tvořeného meta-klasifikátoru.

Třetí kapitola se zabývá otázkou stability proteinů, zejména její změnou a metodami souvisejícími s její predikcí, také pak rozdělením a výčtem existujících nástrojů používaných pro predikci stability. Podrobnějšímu pohledu jsou následně podrobeny jednotlivé nástroje, které byly vybrány pro konstrukci zmíněného meta-klasifikátoru.

Ve čtvrté kapitole jsou obecně diskutovány evoluční algoritmy, zejména pak evoluční strategie, jejíž dva typy byly použity v rámci zkoumání možného přínosu v otázce predikce vlivu aminokyselinových mutací na stabilitu proteinu. Je popsán postup a princip evoluční strategie, zejména pak související techniky autoevoluce řídících parametrů.

Pátá kapitola popisuje provedený postup při tvorbě meta-klasifikátoru. Jsou zde rozebrány kroky při dolování mutací z dostupné databáze pro trénovací datovou sadu, řešené problémy při dolování a jsou zde zobrazeny a popsány statistiky vytvořené trénovací datové sady. Následně jsou popsány provedené kroky při realizaci dávkových výpočtů predikcí stabilit pro vybrané nástroje a zobrazeny statistiky nástrojů pro jednotlivé datové sady mutací. Poslední částí páté kapitoly je rozbor postupu při tvorbě nezávislé testovací datové sady vytvořené z dostupných patentů. Jsou popsány jednotlivé patenty a následně zobrazeny statistiky, stejně jako u trénovací datové sady.

Zhodnocení provedených experimentů a diskuse nad dosaženými výsledky jsou předmětem šesté kapitoly. Při vyhodnocování úspěšností jednotlivých nástrojů a konsensuálních metod byly použity celkem tři různé metriky, jejichž výstup je ve formě grafů také představen. Kapitola rovněž obsahuje výsledné nastavení parametrů evoluční strategie a rozebírá skutečnosti vyplývající z dosažených výsledků.

Poslední sedmá kapitola je pak závěrečným shrnutím provedené práce, vyzdvihuje přínosy této práce a shrnuje zjištěné skutečnosti a vyplynutá doporučení z dosažených výsledků zmíněných v šesté kapitole.

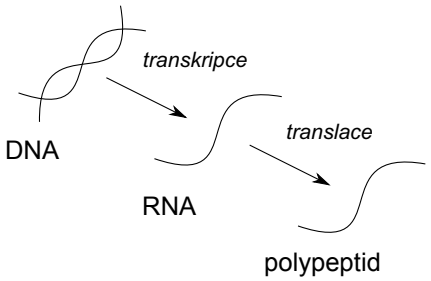
Kapitola 2

Proteiny

Proteiny lze charakterizovat jako hlavní funkční jednotky živých organismů, které se ve všech buňkách podílí na důležitých buněčných procesech. Funkce proteinů úzce souvisí s jejich konkrétním prostorovým uspořádáním neboli konformací. Konformace daného proteinu vychází z jeho primární struktury, kterou lze chápat jako lineární řetězec základních stavebních prvků (v daném pořadí), jež jsou nazývány aminokyselinami [5].

Právě konformace, resp. funkce, jednotlivých proteinů jsou hlavním předmětem jejich zkoumání. Podle funkce lze proteiny rozdělit do několika kategorií [5]. Pro představu následuje výčet jen některých z nich:

- enzymy – podílejí se na katalýze různých reakcí v buňce;
- strukturní proteiny – poskytují mechanickou oporu buňkám a tkáním;
- transportní proteiny – slouží v úloze přenosu malých molekul a iontů;
- pohybové proteiny – jsou základem pro pohyb buněk a tkání;
- signální proteiny – přenáší důležité informační signály mezi buňkami;



Obrázek 2.1: Schéma jednotlivých kroků proteosyntézy.

2.1 Syntéza proteinů

Proteiny vznikají v komplexním procesu zvaném proteosyntéza. Proteosyntéza je popsána v [38]. Při prvním kroku genové exprese, tj. transkripci, dochází k přenosu genetické informace uložené v genech do mediátorové RNA (mRNA), která tuto informaci nese k místům

syntézy polypeptidů. Druhým krokem proteosyntézy je tzv. translace, při níž dochází k přenosu informace z mRNA do sekvencí aminokyselin v polypeptidových genových produktech tj. proteinech.

Při transkripci se jedno vlákno DNA genu použije jako templát pro syntézu komplementárního vlákna RNA, které se označuje jako genový transkript [18].

Samotná translace se řídí podle pravidel genetického kódu zobrazeného v tabulce 2.1. Proces translace spočívá v přepisu kodonů na jednotlivé aminokyseliny. Každá aminokyselina je určena jedním nebo více kodony, tato vlastnost se nazývá degenerace genetického kódu [38]. Některé z kodonů[1] mají speciální funkci. *Iniciační* kodony se podílejí na určení počátku translace, zatímco *terminační* kodony určují konec polypeptidového řetězce.

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

Tabulka 2.1: Tabulka genetického kódu [38].

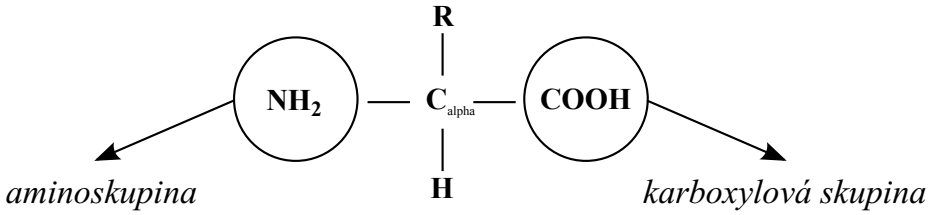
2.2 Aminokyseliny

Existuje celkem 20 základních aminokyselin, které se skládají ze součástí ukázaných na obrázku 2.2: centrální α-uhlík (C_α), atom vodíku (H), aminoskupina (NH_2) a karboxylová skupina (COOH) [18]. Aminokyseliny se od sebe liší postranními skupinami (na obrázku 2.2 značených R jako radikál[2]), které lze rozdělit do čtyř typů [38]:

- hydrofobní (neboli nepolární),
- hydrofilní (neboli polární),
- kyselé a
- bazické.

[1] Kodon je trojice nukleotidů zapsaná v sekvenci (řetězci) mRNA za sebou.
[2] Radikál je vysoce reaktivní částice díky jednomu nebo více volných elektronů v obalu [33].

Tato postranní skupina pak do značné míry ovlivňuje výslednou prostorovou konfiguraci proteinu resp. jeho funkci. Vznik řetězce aminokyselin spočívá ve vytvoření *peptidových vazeb*. Peptidová vazba vzniká mezi aminoskupinou jedné aminokyseliny a karboxylovou skupinou jiné. Atom uhlíku z karboxylové skupiny sdílí elektrony s dusíkovým atomem aminoskupiny. Při této reakci (vytvoření peptidové vazby) se uvolní molekula vody [5]. Řetězec aminokyselin se pak nazývá polypeptid. Proto lze o proteinech mluvit také jako o polypeptidech.



Obrázek 2.2: Struktura aminokyseliny.

Názvy základních 20 aminokyselin s typem jejich postranní skupiny, odpovídající třípísmennou a jednopísmennou zkratkou zobrazuje tabulka 2.2.

Název aminokyseliny	**Třípísmenná zkratka**	**Jednopísmenná zkratka**	**Postranní skupina**
glycin	Gly	G	
alanin	Ala	A	
leucin	Leu	L	
izoleucin	Ile	I	
fenylalanin	Phe	F	Hydrofobní
tryptofan	Trp	W	
prolin	Pro	P	
metionin	Met	M	
valin	Val	V	
serin	Ser	S	
treonin	Thr	T	
asparagin	Asn	N	Hydrofilní
glutamin	Gln	Q	
cystein	Cys	C	
tyrozin	Tyr	Y	
kyselina asparagová	Asp	D	
kyselina glutamová	Glu	E	Kyselé
kyselina asparagová	Asp	D	
lyzin	Lys	K	
arginin	Arg	R	Bazické
histidin	His	H	

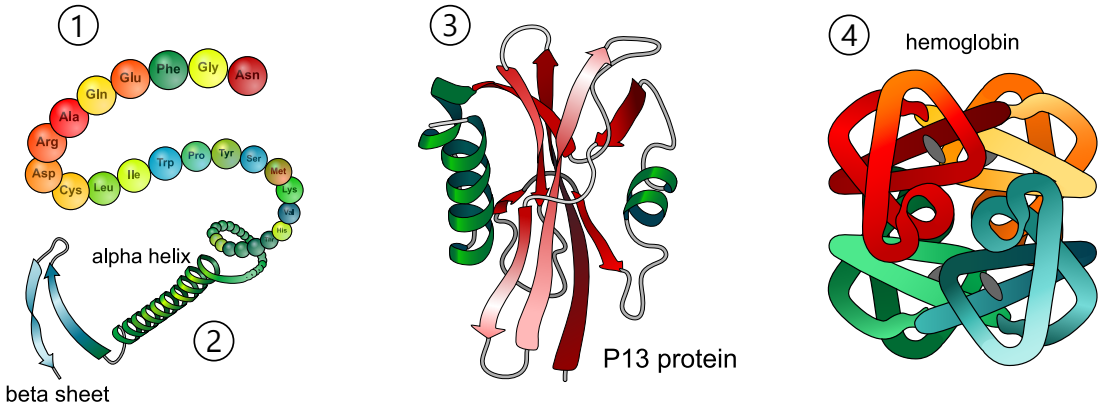
Tabulka 2.2: Tabulka základních 20 aminokyselin, jejich zkratek a postranních skupin [38].

2.3 Struktura proteinu

Složitou trojrozměrnou strukturu proteinů lze dle [18] rozdělit do čtyř úrovní organizace:

- *primární struktura* - sekvence aminokyselin určená nukleotidovou sekvencí genu,
- *sekundární struktura* - vyplývá z prostorových vztahů aminokyselin uvnitř segmentů proteinu,
- *terciární struktura* - způsob složení proteinu do trojrozměrného uspořádání a
- *kvartérní struktura* - spojení dvou nebo více terciárních struktur, tzv. řetězců, pomocí nekovalentních vazeb (kvartérní strukturu má pouze menší část proteinů, jelikož většina proteinů je tvořena pouze jedním řetězcem).

Jak již bylo řečeno, protein lze chápat jako řetězec aminokyselin. Jednotlivé aminokyseliny jsou v řetězci spojeny kovalentní peptidovou vazbou. Opakující se pořadí atomů podél řetězce se nazývá *polypeptidová kostra* (nebo také *proteinová páteř*) [5]. K této kostře jsou pak připojeny tzv. *postranní řetězce* různých aminokyselin, které na základě svého typu, uvedeného v tabulce 2.2, určují strukturu proteinu. Každý typ proteinu je jedinečný svou sekvencí a počtem aminokyselin, ovšem právě pořadí chemicky různých postranních řetězců odlišuje jeden protein od druhého. Aminokyseliny s hydrofobním postranním řetězcem (např. fenylalanin, leucin, valin a tryptofan) mají snahu se shlukovat uvnitř molekuly proteinu, aby se vyhnuly kontaktu s vodným prostředím, které protein uvnitř buňky obklopuje. Na druhou stranu aminokyseliny s hydrofilní postranní skupinou (např. serin, glutamin a cystein) se snaží udržet na povrchu molekuly, kde mohou pak s molekulami vody (a dalšími hydrofilními látkami) vytvářet vodíkové můstky, zatímco hydrofobní aminokyseliny vytvářejí vazby uvnitř proteinu [5].



Obrázek 2.3: Všechny úrovně organizace struktury proteinu dle [18]: (1) Primární struktura, (2) Sekundární struktura, (3) Terciární struktura, (4) Kvatérní struktura. Převzato z [3].

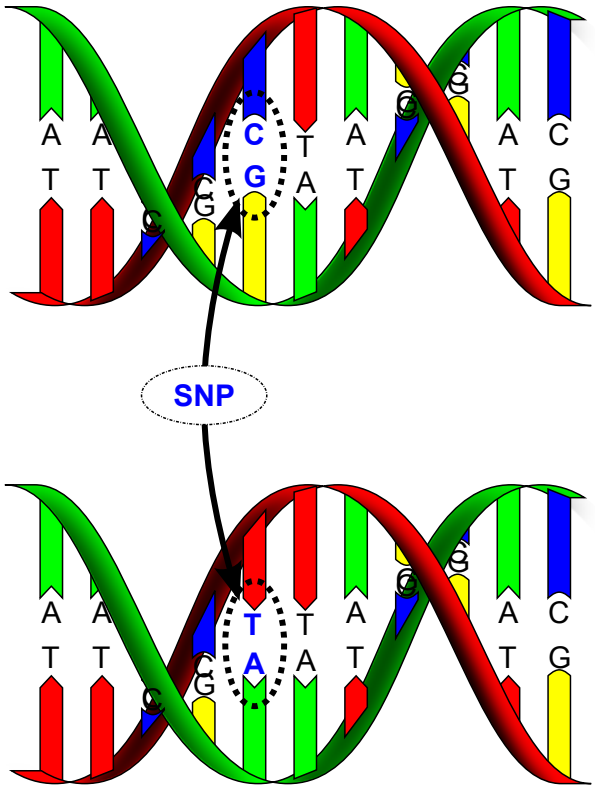
Stejně tak konce polypeptidového řetězce se navzájem liší. Jeden konec nese vždy volnou aminoskupinu (NH_2) a nazývá se aminový konec (nebo N-konec), zatímco druhý konec nese volnou karboxylovou skupinu (COOH) a nazývá se karboxylový konec (nebo C-konec).

Odlišení obou konců slouží k určení počátku a konce proteinu, jelikož polypeptidový řetězec se čte od N-konce k C-konci [5].

Konečná složená struktura proteinu neboli konformace, které každý protein nabývá, je určena energetickými aspekty - obecně snahou dosáhnout stavu s co nejmenším obsahem volné energie [5]. Polypeptidový řetězec je možné rozvinout neboli *denaturovat* s pomocí jistých rozpouštědel, která poruší nekovalentní vazby držící protein v jeho složené konformaci. Jelikož rozpouštědlo neporušuje kovalentní vazby polypeptidový řetězec zůstane pohromadě a stává se tak volně ohebným. Jakmile dojde k odstranění použitého rozpouštědla, protein se spontánně vrací zpět do své složené konformace tj. *renaturuje*. To ukazuje, že veškerá informace potřebná k určení trojrozměrného tvaru proteinu je uložena v jeho sekvenci aminokyselin [5] neboli v jeho *primární struktuře*. V některých případech závisí skládání proteinů do stabilní konformace na proteinech zvaných *chaperony*, které napomáhají vznikajícím polypeptidům zaujmout správnou trojrozměrnou strukturu [19].

2.4 Vznik aminokyselinové mutace

Původ vzniku aminokyselinové mutace neboli záměny jedné aminokyseliny za jinou (v případě jednobodové mutace) v daném místě polypeptidového řetězce je v genomové DNA, tedy ještě před samým začátkem procesu syntézy proteinu. Jiný název pro jednobodovou mutaci v DNA je tzv. jednoduchý nukleotidový polymorfismus SNP (z angl. *single nucleotide polymorphism*).



Obrázek 2.4: Schéma příkladu jednoduchého nukleotidového polymorfismu v DNA. Převzato z [1].

Jeden z nejznámějších SNP vzniká v genu kódující polypeptid známý jako β-globin

[38]. β-globin je součástí proteinu, jehož úkolem je přenos kyslíku v krvi. Pouhá záměna nukleotidového páru A:T za T:A v DNA řetězci daného genu se po transkripci projeví záměnou kodonu v mRNA z GAG na GUG, což způsobí začlenění valinu do polypeptidového řetězce místo glutamové kyseliny. Tato popsaná mutace je zodpovědná za vznik srpkovité anémie (způsobující neefektivní přenos kyslíku červenými krvinkami srpkovitého tvaru).

SNP se obecně objevuje častěji v nekódujících oblastech genomové DNA. Výskyt v kódující oblasti však nutně nemusí znamenat záměnu aminokyseliny. Změna nukleotidu má sice za následek změnu kodonu, ovšem vzhledem k vlastnosti degenerace genetického kódu, může tento kodon kódovat stejnou aminokyselinu a nedojde tak ke změně polypeptidového řetězce. Tento fakt rozděluje SNP v kódujících oblastech na dva typy:

- synonymní nebo také neutrální (neovlivňující polypeptidový řetězec) a
- nesynonymní (mající za následek změnu polypeptidového řetězce).

Nesynonymní SNP se dále dělí na *nesmyslné* (z angl. *nonsense*), které se projevují vznikem tzv. předčasného stop kodonu vedoucím ke zkrácení polypeptidového řetězce a *mylné* (z angl. *missense*) jež se projevují záměnou aminokyseliny. Další dělení SNP je probráno v následující kapitole o stabilitě proteinů.

Kapitola 3

Stabilita proteinů

Stabilita proteinu vychází z jeho tzv. teploty tání Tm. Při této teplotě dochází k přechodu proteinu do nativní (stabilní) konformace (tzv. proces *renaturace*) nebo do denaturovaného (rozbaleného) stavu (tzv. proces *denaturace*). Stabilita proteinů pak úzce souvisí s otázkou jejich funkce, jelikož funkce proteinu je dána jeho prostorovým uspořádáním. Nestabilní protein může změnit svoji konformaci a tím také svoji funkci. Vzájemné interakce mezi atomy proteinu se pak různě podílejí na jeho stabilitě. Dle [18] je stabilní konformace proteinu určena především faktory jako jsou hydrofobní efekt, vodíkové můstky, van der Waalsovy síly a disulfidické vazby, zatímco u proteinů v denaturovaném (rozbaleném) stavu nás zajímají volné energie.

Stabilitu lze měřit jako změnu tzv. Gibbsovy (volné) energie (ΔG) v jednotkách kcal/mol, což udává množství změny energie v 1 molu látky při přechodu proteinu ze stabilní konformace do denaturovaného stavu či naopak. K určení stability proteinu se používá několik různých metod, jako jsou například cirkulární dichroismus (CD), diferenciální skenovací kalorimetrie (DSC), absorpce světla (Abs), fluorescence (Fl) a jaderná magnetická rezonance (NMR) [18].

Hlavní oblastí měření stability proteinů, především tedy její změny, je predikce změny stability v důsledku aminokyselinové mutace. Snaha predikovat změnu stability na základě mutace může pomoci v otázce návrhu nových či úpravy již existujících proteinů s požadovanou mírou stability, enzymatickou aktivitou či snahou vázat se na jiné molekuly (proteiny, DNA, léky, atd.) [31]. Jedná se tedy o predikci změny Gibbsovy volné energie ($\Delta\Delta$G) mezi původním proteinem (tzv. wild-type protein) a jeho mutantem. Na základě hodnoty této změny lze rozdělit aminokyselinové mutace na [24]:

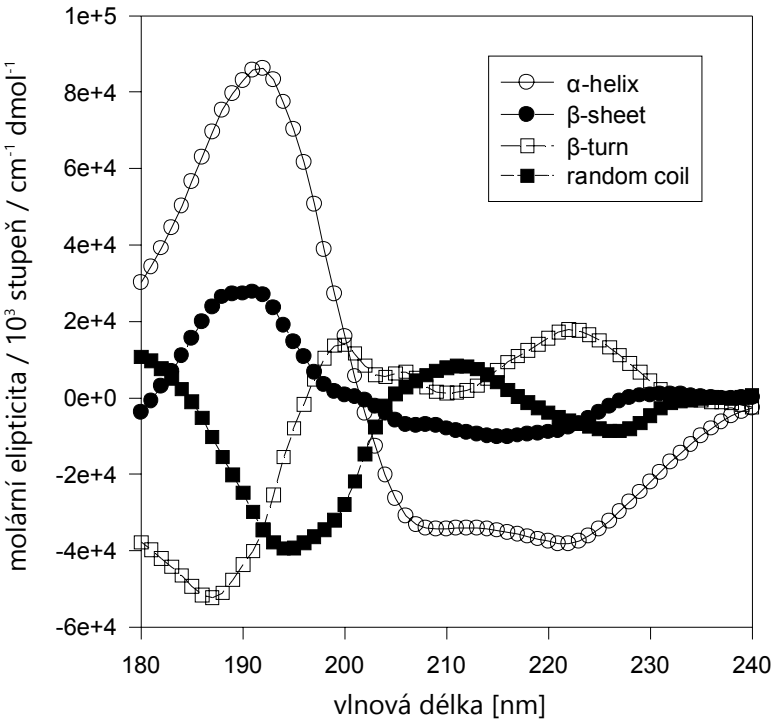
- stabilizující ($\Delta\Delta G \leq -0{,}5$ kcal/mol),
- neutrální ($-0{,}5$ kcal/mol $< \Delta\Delta G < 0{,}5$ kcal/mol) a
- destabilizující ($0{,}5$ kcal/mol $\leq \Delta\Delta G$).

Práh 0,5 kcal/mol je odvozen od průměrné hodnoty maximální chyby experimentálního měření nad několika datesety mutací obsažených v databázi ProTherm [25]. Lze se ovšem setkat také s odlišnými prahy. Studie [11] používá hodnotu prahu 1 kcal/mol pro klasifikaci do všech tří tříd zmíněných výše. Jiná studie [31] provádí binární klasifikaci mutací do tříd (stabilizující/destabilizující) na základě dvou prahů: 0 kcal/mol a 2 kcal/mol. Hodnotu 0 kcal/mol používá pro klasifikaci stabilizující ($\Delta\Delta G < 0$ kcal/mol) a destabilizující ($\Delta\Delta G > 0$ kcal/mol) mutace. Hodnotu 2 kcal/mol volí pro identifikaci tzv. *hot-spotů*, tj. míst,

u kterých je téměř jisté, že mají silný efekt - ať již kladný (stabilizující), nebo záporný (destabilizující). Pro *hot-spot* mutace tedy platí $|\Delta\Delta G| > 2$ kcal/mol.

3.1 Metody

Cirkulární dichroismus je výborným nástrojem pro rychlé určení sekundární struktury a vlastností skládání a vazeb proteinů [17]. Stručně lze tuto techniku charakterizovat jako měření nerovnoměrné absorpce kruhově polarizovaného světla, které lze rozložit na pravotočivou a levotočivou složku. V momentě, kdy různé molekuly interagují s takovýmto světlem, levotočivá složka je absorbována jinak než pravotočivá. Tento rozdíl nám pak dává informaci o struktuře molekul proteinu. Měření základních typů sekundární struktury proteinů pomocí cirkulárního dichroismu je vidět na obrázku 3.1.



Obrázek 3.1: Spektrum molární elipticity měřené cirkulárním dichroismem pro základní sekundární struktury [10].

Kalorimetrie je technika používaná primárně pro měření teplotních vlastností materiálů. Jedním z několika druhů kalorimetrie je právě *diferenciální skenovací kalorimetrie* popsaná například v [16]. Jedná se o aparaturu (obrázek 3.2), pomocí níž je analyzována změna fyzikálních vlastností molekuly spolu se změnou tepla v časovém horizontu. Spočívá v simultánním ohřívání dvou vzorků obsahujících roztok se zkoumanou molekulou a roztok bez ní. Pro každý vzorek je pak potřeba vyvinout různé množství energie pro získání shodné teploty. Rozdíl této energie pak určuje, kolik tepla bylo absorbováno či uvolněno zkoumanou molekulou. V otázce proteinů se využívá pro evaluaci faktorů ovlivňující stabilitu proteinu, zejména pak teploty tání.

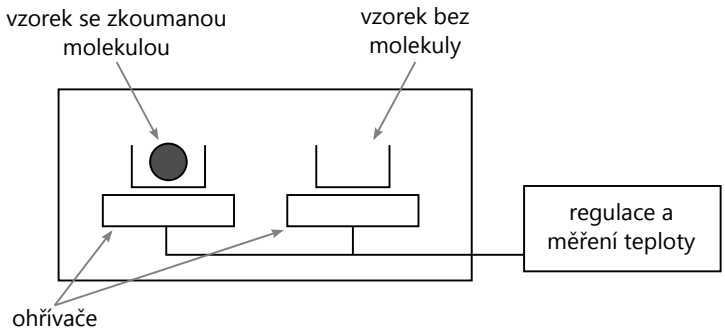
Změny *absorpce světla* proteinů (měřené v jednotkách absorbance) v nativním a dena-

turovaném stavu lze využít pro určení termodynamické stability či kinetických vlastností skládání. Nejčastěji jsou využívány vlnové délky v oblasti UV, kolem 200nm [29]. Příklad měření hodnot absorbance je na obrázku 3.3.

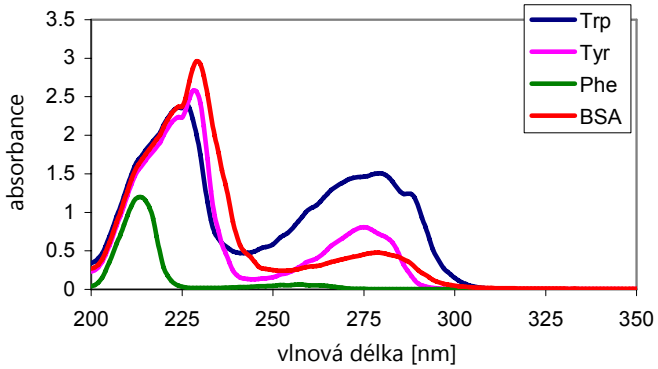
Technika *fluorescence* v úloze analýzy skládání a denaturace proteinů je využívána především v případě analýzy proteinů obsahující jednu z následujících aminokyselin [28]:

- tyrozin,
- tryptofan nebo
- fenylalanin.

Především tyto tři aminokyseliny se podílejí na výsledné (měřitelné) fluorescenční intenzitě proteinů (viz. příklad hodnot excitace zobrazené na obrázku 3.4 pro tryptofan a tyrozin). Vzhledem ke změně stavu proteinu se mění poloha těchto aminokyselin vůči povrchu a tím pádem také intenzita měřeného emitovaného světla. Na fluorescenční intenzitě proteinů se v jisté míře také podílejí některé enzymatické kofaktory a porfyriny [20].



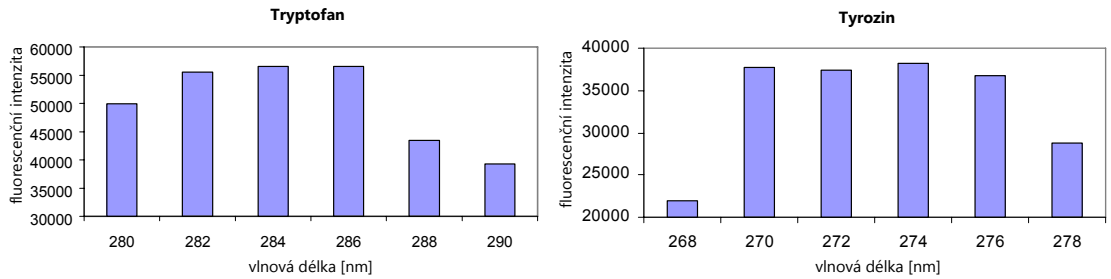
Obrázek 3.2: Schéma aparatury pro diferenciální skenovací kalorimetrii.



Obrázek 3.3: Graf absorbance měřené u aromatických aminokyselin a proteinu BSA (*Bovine serum albumin*). Jsou patrné dvě hlavní oblasti, kde aromatické aminokyseliny absorbují UV záření nejvíce [20].

Jaderná magnetická resonance patří do skupiny spektroskopií a využívá magnetických vlastností jádra atomů. Tyto vlastnosti souvisejí s lokálním uspořádáním molekul a jejich

měřením lze získat informace o vazbách atomů, jejich prostorové vzdálenosti a pohybu vůči sobě [32].



Obrázek 3.4: Grafy s intenzitou fluorescence pro aminokyseliny trpytofan a tyrozin [20].

3.2 Nástroje

Porozumění mechanismu, kterým mutace ovlivňují stabilitu proteinů, je důležité především v otázce vztahu struktury a funkce proteinů, návrhu nových proteinů, charakterizace mechanismů chorob a vývojových dynamik organismů [24]. Na základě tohoto bylo vyvinuto několik metod pro predikci změn volných energií rozkladu proteinů ($\Delta\Delta G$) mezi wild-type proteinem a jeho mutantem.

Přehled a popis hlavních metod a přístupů použitých v nástrojích pro predikci změny stability proteinů lze nalézt ve [24]. Existují metody využívající energetické funkce a metody, které aplikují principy strojového učení. Metody založené na energetických funkcích lze dál rozdělit na metody využívající tzv. fyzikální potenciál, jež se snaží simulovat silová pole atomů ve struktuře proteinu a jsou tím poměrně výpočetně náročné. Další skupinou související s energetickou funkcí jsou metody založené na tzv. statistickém potenciálu. Jejich snahou je získat funkci potenciálu ze statistických analýz náchylnosti proteinu na různá prostředí, frekvence substitucí a korelace sousedících residuí zjištěných experimentálně v proteinové struktuře. Poslední skupinou této kategorie jsou metody založené na tzv. empirickém potenciálu, který je kombinací váhovaných fyzikálních a statistických energetických vlastností a strukturálních deskriptorů. Druhou hlavní kategorií jsou metody založené na principech strojového učení. Takové nástroje jsou nejprve naučeny (natrénovány) na proteinech a jejich mutantech, u kterých byla změna volné Gibbsovy energie experimentálně změřena. Přehled kategorií a některých zástupců nástrojů pro predikci je v tabulce 3.1.

3.2.1 I-Mutant2.0

Prvním z vybraných nástrojů byl I-Mutant2.0, u něhož byla použita jak sekvenční, tak i strukturní verze. Nástroj umožňuje ohodnotit změnu stability proteinu po provedení jednobodové mutace na tomto proteinu. Sekvenční varianta nástroje I-Mutant2.0 využívá pro predikci $\Delta\Delta G$ pouze informace získané z primární struktury proteinu, tedy jeho sekvence, která je hlavním parametrem při spouštění nástroje. Dalšími parametry sekvenční verze jsou zejména pozice mutace, teplota a pH okolí. Výstupem je pak seznam záznamů o predikci, kde každý záznam obsahuje:

- *Position* - nastavená pozice mutace,
- *WT* - aminokyselina přítomná na dané pozici wild-type proteinu,

- *NEW* - aminokyselina, na kterou bylo mutováno,
- *Stability* - efekt mutace (na základě znaménka predikované hodnoty stabilizující při kladné, destabilizující při záporné hodnotě),
- *RI* - tzv. *reliability Index* udávající úroveň věrohodnosti predikce na základě výstupu SVM[1],
- *DDG* - predikovaná hodnota ΔΔG,
- *pH* - nastavená hodnota pH a
- *T* - nastavená hodnota teploty ve stupních Celsia.

Hlavním rozdílem strukturní varianty jsou jiné parametry na vstupu nástroje. Místo souboru se sekvencí potřebuje nástroj PDB soubor se strukturou wild-type proteinu, dále pak navíc odpovídající DSSP[2] soubor získaný z webového serveru [22] a případně i označení řetězce proteinu (v případě, že je protein tvořen více řetězci). Výstup je v podstatě shodný se sekvenční variantou, pouze obsahuje jednu hodnotu navíc, kterou je *RSA* - tzv. *Relative Solvent Accessible Area* udávající, jak velká část plochy aminokyseliny je v kontaktu s okolím.

Metodika přístupu	**Další dělení**	**Příklady zástupců nástrojů**
založené na energetických funkcích	fyzikální potenciál	EGAD, CC/PBSA
	statistický potenciál	Hunter, PoPMuSiC, Dmutant, MultiMutate, SDM
	empirický potenciál	FoldX, CUPSTAT, Rosetta, PEATSA, Eris
využívající metody strojového učení	SVM, Neuronové sítě, rozhodovací stromy	I-Mutant2.0, I-Mutant3.0, AUTO-MUTE, MUpro, iPTREE-STAB

Tabulka 3.1: Přehledová tabulka nástrojů zastupujících jednotlivé metodiky predikce.

3.2.2 FoldX

Druhým vybraným nástrojem je FoldX [36]. Jeho využití spočívá v predikcích důležitosti interakcí probíhajících v proteinech a proteinových komplexech související se stabilitou. Při predikci změny stability proteinu po jednobodové mutaci nástroj principiálně pracuje tak, že z původní struktury wild-type proteinu s využitím zákonů kvantové chemie vypočítá novou proteinovou strukturu obsahující zakomponovanou mutaci. Pak tyto dvě struktury porovnává, vypočítá energie a určí predikovanou změnu volné energie ΔΔG. Výstupem tohoto nástroje je jednak predikovaná hodnota ΔΔG, ale pak její rozklad na několik dalších energetických hodnot jež přispěly k výsledné predikované hodnotě ΔΔG. Spouštění tohoto nástroje je poměrně komplexní a umožňuje nastavení několika různých parametrů. Pro tuto práci byl využit modul *BuildModel*, jehož nastavení je popsané v [2].

[1] Support Vector Machine

[2] *Define Secondary Structure of Proteins* - princip, jakým se tento soubor vytváří a co obsahuje, je popsán v [21].

3.2.3 Rosetta

Již pokročilejší nástroj využívaný zejména v proteinovém inženýrství se nazývá Rosetta. Tento nástroj byl vytvořen pro účely různorodých biomolekulárních modelovacích úloh. Ze základních úloh jsou sestaveny tzv. protokoly (algoritmy), které lze používat jednotlivě i zřetězeně. Jedním z těchto protokolů je - v této práci použitý - *RosettaDDG* [35], který se snaží stanovit vliv změn v sekvenci na stabilitu proteinu. Pracuje tak, že ze vstupního (předzpracovaného) PDB souboru wild-type proteinu generuje strukturní model jeho mutantu. Predikovaná hodnota $\Delta\Delta G$ je získána jako rozdíl energií mezi wild-type strukturou a strukturou jednobodového mutantu. Ve skutečnosti je doporučeno generovat 50 modelů wild-type struktur i struktur mutantu a nejpřesnější hodnotu $\Delta\Delta G$ získat jako rozdíl mezi průměrem nejlepších tří wild-type struktur a nejlepších tří struktur mutantu. Ačkoliv v článku srovnávající nástroje pro predikci $\Delta\Delta G$ [31] je Rosetta hodnocena jako nejhorší ze zvolené sady nástrojů, tak v článku [23] zabývajícím se podobnou sadou nástrojů, tuto skutečnost vyvracejí, jelikož zjistili, že v [31] používali Rosettu pro predikci $\Delta\Delta G$ nekorektně. Především kvůli špatně nastaveným parametrům a použití výchozího nastavení, které je pro tuto predikci nevhodné.

3.2.4 Eris

Posledním zvoleným nástrojem je Eris [42], který využívá výpočetní balík *Medusa* [14] navržený pro molekulární modelování a design proteinů. Tento balík využívá právě pro výpočet změny stability proteinu po jednobodové mutaci. Volnou energii vyjadřuje jako váhovanou sumu van der Wallsových sil, statistických energií rozpustnosti, vodíkových vazeb a statistických energií souvisejících s uhlíkovým skeletem proteinu [41].

Kapitola 4

Evoluční algoritmy

Princip evolučních algoritmů (EA) je popsán v [27]. EA jsou založeny na metafoře evoluce. Řešení nějaké úlohy je převedeno na proces evoluce populace náhodně vygenerovaných řešení. Každé řešení je zakódováno do řetězce symbolů (parametrů) a ohodnoceno tzv. fitness funkcí, která vyjadřuje kvalitu řešení. Čím je hodnota fitness funkce větší, tím je dané řešení perspektivnější a častěji vstupuje do reprodukčního procesu evoluce, během něhož jsou generována nová řešení. Obecný princip algoritmu lze zapsat pomocí následujícího pseudokódu.

```
čas t = 0;
inicializace G(t);
vyhodnocení G(t);
while(not zastavovací_pravidlo)
{
   t = t + 1;
   selekce G(t) z G(t-1);
   změna G(t);
   vyhodnocení G(t);
}
```

Obrázek 4.1: Pseudokód obecného evolučního algoritmu [27].

Z pseudokódu je patrné, že každý EA začíná *inicializací*, která představuje vytvoření počáteční populace jedinců (řešení) G v čase neboli kroku evoluce t. Nejčastěji se populace vytváří náhodně, ale je možné (a často také výhodné) využít heuristik vycházejících ze znalosti řešeného problému.

Následuje *vyhodnocení* jednotlivých jedinců z vytvořené populace. Dochází často k nalezení nejlepších jedinců či vypočtení statistických vlastností populace. Záleží na potřebách a ukončující podmínce, pro kterou chceme EA ukončit.

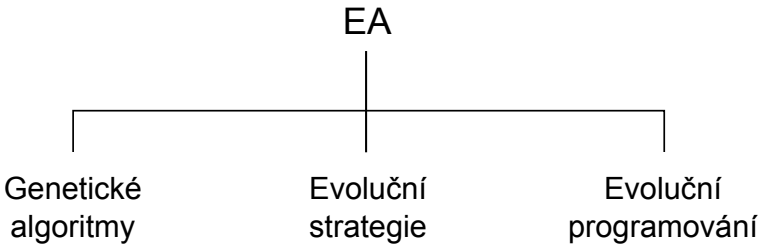
Dalším krokem je pak *selekce*, která spočívá v simulaci procesu přirozeného výběru. Existuje několik technik a mechanismů selekce (tzv. selekčních operátorů). Většinou se využívá již vypočteného ohodnocení, v některých případech je vhodné využít pro větší diversitu nové populace i náhodný výběr. Právě díky stochastickým principům se EA přibližují skutečné evoluci. Nejenom, že se do nové populace dostávají především kvalitní jedinci, ale

i slabší jedinci mají jistou šanci do nové populace vstoupit. Výsledkem selekce je pak nová populace připravená pro další proces změn.

Proces označen v pseudokódu jako *změna* představuje další zásah do vytváření nové populace z již existující. Používá se zde tzv. rekombinačních operátorů, které mohou být dvojího typu [27]:

- mutace (nový jedinec vzniká pozměněním jiného) a
- křížení (nový jedinec vzniká ze dvou a více jedinců jejich kombinací).

Vzhledem k tomu, že celý tento proces evoluce by mohl běžet v podstatě neustále, zavádí se zde tzv. *zastavovací pravidlo*, pomocí kterého se určí moment ukončení běhu evoluce. Ve většině případů se používá podmínka na již vyhovující kvalitu jedince v populaci, nebo počet kroků evoluce. Obecně bývá pravidlem, že více kroků evoluce sice vede k lepším výsledkům, ale za cenu rostoucí doby běhu, a proto je vhodné tento proces omezit v počtu kroků. Je totiž také možné, že proces evoluce se zastaví ve smyslu kvality jedinců a nebude již schopný tuto kvalitu v dalších krocích vylepšit. Evolučních technik využívajících základního principu EA je několik. Základní rozdělení je vidět na obrázku 4.2.



Obrázek 4.2: Základní dělení evolučních algoritmů.

4.1 Evoluční strategie

Pro charakter této práce byl zvolen druh evolučních algoritmů zvaný *evoluční strategie* (dále jen ES). ES se svým charakterem hodí především pro optimalizační úlohy v oblasti reálných funkcí vektorového argumentu [27]. Hlavním specifikem ES je reprezentace problému pomocí vektoru reálných čísel, často doplněný o vektor řídících parametrů. ES tedy přímo pracuje s reálnými čísly a nesnaží se je převést do binární podoby (jako tomu je např. u genetických algoritmů).

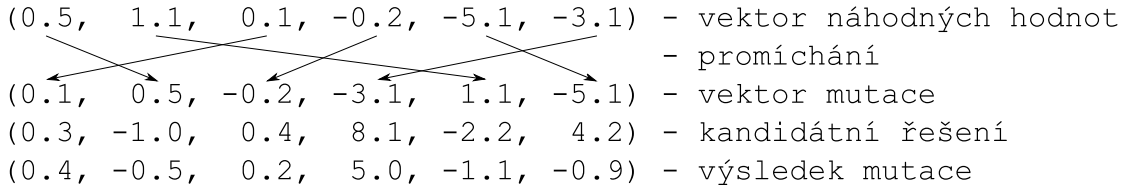
4.1.1 Obnova populace

V současné době jsou (podle typu obnovy populace) nejznámější evoluční strategie těchto dvou typů [27]:

- $(\mu + \lambda)$-ES (tzv. plusová) a
- (μ, λ)-ES (tzv. čárková).

Ve strategii $(\mu + \lambda)$-ES se z aktuální populace sestávající z μ rodičů generuje λ potomků. Dochází k porovnání kvality všech jedinců (dle nastavené fitness funkce) a nová populace

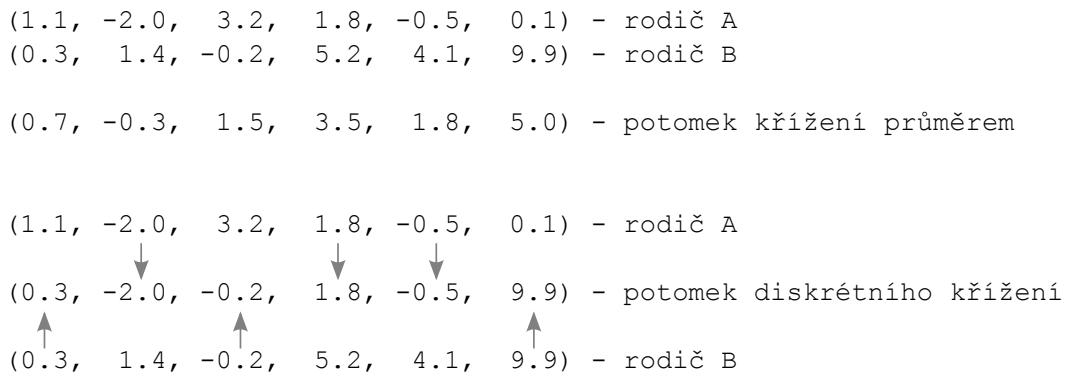
má pak μ nejlepších členů. Tento princip se pak opakuje pro další populace. Strategie (μ, λ)-ES používá mechanismus, kdy dochází k úplnému nahrazení μ (všech) jedinců v původní rodičovské populaci. Vybírá se z jimi vygenerovaných potomků o počtu λ, ze kterých se vybere μ nejlepších. Může tak zde dojít k nahrazení jedinců rodičovské populace horšími potomky, což na druhou stranu v některých případech umožňuje opustit lokální optima. Pro účely diplomové práce byla vybrána varianta $(\mu + \lambda)$-ES konkrétněji $(1 + 1)$-ES, kdy je na počátku vygenerován náhodný rodič a z něho je vždy vygenerován jeden nový potomek, jenž je jeho mutací. Nový potomek se stává rodičem v následující populaci v případě, že je jeho fitness lepší než fitness rodiče. V opačném případě rodič zůstává v populaci a je znovu mutován na nového potomka.



Obrázek 4.3: Princip mutace v ES.

4.1.2 Mutace

Jako u všech evolučních algoritmů, i zde se využívá rekombinačního operátoru mutace. Pro mutaci konkrétního jedince se nejprve vypočítá vektor nezávislých náhodných čísel, lze jej nazývat vektorem mutace. Tato čísla odpovídají Gaussově normálnímu rozdělení s uživatelem definovanou střední směrodatnou odchylkou. Jednotlivé prvky vektoru mutace se pak náhodně mezi sebou vymění, aby došlo k eliminaci případných závislostí při generování náhodných hodnot. Mutace je pak provedena sečtením původního vektoru kandidátního řešení s vektorem mutace z čehož vznikne nový vektor neboli nové kandidátní řešení [27]. Příklad takové mutace je znázorněn na obrázku 4.3.



Obrázek 4.4: Princip dvou druhů křížení v ES: křížení průměrem (nahoře) a diskrétní křížení (dole).

4.1.3 Křížení

Mezi dvě hlavní metody křížení používaných u ES patří *diskrétní křížení* a *středové křížení* (nebo také *křížení průměrem*). Vstupem obou druhů křížení jsou dvě kandidátní řešení (dva vektory), jinak nazývané jako rodiče. Diskrétní křížení generuje nového potomka jako nový vektor, jehož komponenty jsou náhodně po jednom vybírány z jednoho nebo druhého rodiče. Středové křížení spočívá v průměrování jednotlivých hodnot rodičů. Hodnoty vektoru potomka jsou tedy aritmetickým průměrem odpovídajících hodnot jeho rodičů. Oba typy křížení jsou vyobrazeny na následujícím obrázku 4.4. Jelikož je v práci používána (1+1)-ES, není aplikováno křížení.

4.1.4 Pravidlo 1:5

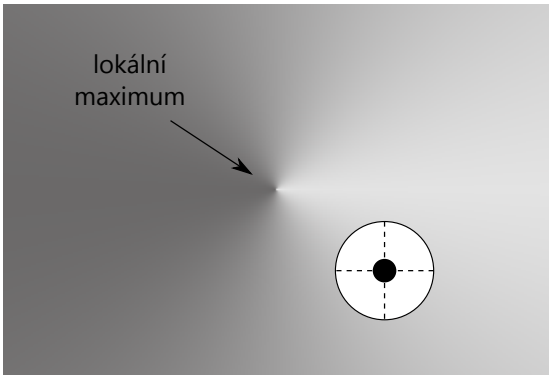
Velice důležitým řídícím parametrem evoluce je standardní odchylka. Pomocí její modifikace na základě úspěšnosti ES lze zvýšit efektivnost prohledávání v prostoru všech řešení. Pokud je algoritmus ES úspěšný, je vhodnější prohledávat ve větších krocích, naopak je-li algoritmus neúspěšný, je vhodné zmenšit krok. Úspěšnost algoritmu $\phi(k)$ lze definovat jako poměr počtu úspěšných mutací (s lepší fitness než rodič) k počtu celkově provedených k mutací. Například zvolíme-li $k = 10$ a úspěšných mutací bude $4z$ 10, pak je úspěšnost rovna 40% a standardní odchylku tedy zvětšíme, abychom zároveň zvětšili krok. Vzorec pro výpočet nové standardní odchylky je pak

$$\sigma_{new} = \begin{cases} c_d\sigma_{old} & \text{pro } \phi(k) < 1/5 \\ c_i\sigma_{old} & \text{pro } \phi(k) > 1/5 \\ \sigma_{old} & \text{jinak} \end{cases} \text{, kde } c_d = 0{,}82 \text{ a } c_i = 1/c_d = 1{,}22. \tag{4.1}$$

Konstanty c_i a c_d jsou nastaveny na základě experimentů provedených v [8].

4.1.5 Autoevoluce řídících parametrů

V případě reprezentace problému pomocí vektoru o dvou a více sekcích, z nichž jedna představuje kandidátní řešení problému, druhá a další představují řídící parametry evoluce, hovoříme o technice zvané autoevoluce (nebo také autoadaptace). Jak prvky řešení, tak řídící parametry podléhají procesu evoluce a mění svoje hodnoty. Existují celkem 3 přístupy jejichž podrobný popis lze nalézt v [15].



Obrázek 4.5: Pohyb kandidátního řešení (černé kolečko) při autoevoluci typu 1 je pro všechny směry se stejnou pravděpodobností (bílý kruh).

Autoevoluce typu 1 se vyznačuje tím, že jako řídící parametr je zvolena pouze jedna proměnná určující standardní odchylku, která ovlivňuje generování vektoru mutace a je tedy pro všechny jeho prvky shodná. Kandidátní řešení je tvaru $(x_1, x_2, ..., x_n, \sigma)$. Novou standardní odchylku lze vypočítat pomocí vzorce

$$\sigma_{new} = \sigma_{old} e^{(\tau N(0,1))}, \tag{4.2}$$

kde τ je tzv. parametr učení, podle [30] je doporučeno volit $\tau = \frac{1}{n^{\frac{1}{2}}}$(kde n je počet prvků vektoru jedince - v této práci odpovídá počtu nástrojů). Nové řešení se pak vypočte pomocí vztahu

$$x_i' = x_i + \sigma_{new} N(0,1). \tag{4.3}$$

Na obrázku 4.5 je vidět, že v případě autoevoluce typu 1, je pohyb[1] kandidátního řešení k lokálnímu maximu ve všech směrech se stejnou pravděpodobností.

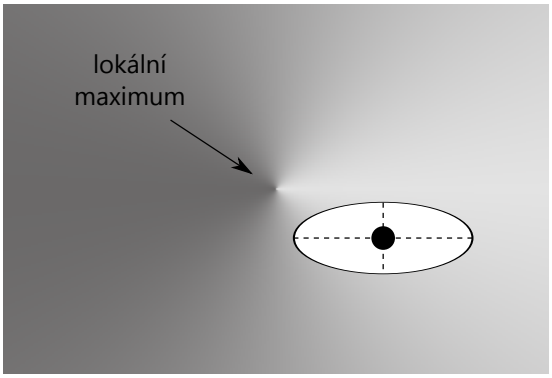
Autoevoluce typu 2 spočívá v zavedení vlastní směrodatné odchylky pro každou hodnotu vektoru kandidátního řešení. Vektor řídících parametrů má tedy stejný počet prvků (směrodatných odchylek), jako je počet prvků vektoru řešení. Kandidátní řešení je pak tvaru $(x_1, x_2, ..., x_n, \sigma_1, \sigma_2, ..., \sigma_n)$. Autoevoluce typu 2 zavádí nový parametr, jímž je specifický parametr učení τ_i, pro každý hledaný parametr cílového řešení x_i. Nová směrodatná odchylka σ_i' se pro daný cílový parametr x_i vypočítá pomocí vztahu

$$\sigma_i' = \sigma_i e^{(\tau N(0,1) + \tau_i N_i(0,1))}, \tag{4.4}$$

kde τ_i je specifický parametr učení pro cílový parametr x_i a τ je společný parametr učení. Podle [30] jsou tyto parametry voleny jako: $\tau_i = \frac{1}{\sqrt{2\sqrt{n}}}$ a $\tau = \frac{1}{\sqrt{2n}}$. Nové řešení se pak vypočte pomocí vztahu

$$x_i' = x_i + \sigma_i' N_i(0,1). \tag{4.5}$$

Toto nastavení pak způsobuje rychlejší migraci potomků v ose x k lokálnímu maximu, jak je vidět na obrázku 4.6.

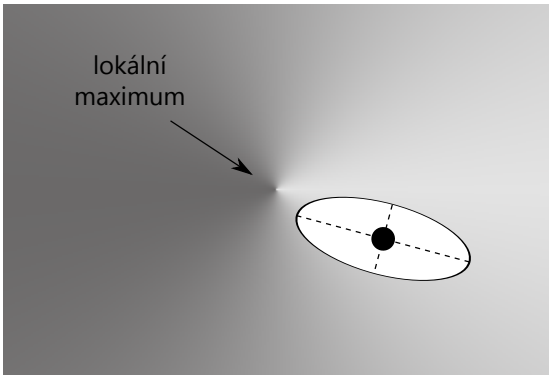


Obrázek 4.6: Pohyb kandidátního řešení (černé kolečko) při autoevoluci typu 2 je ve směru osy x s větší pravděpodobností (bílý kruh).

Autoevoluce typu 3 se také nazývá tzv. *korelovaná* mutace, která přidává třetí vektor pomocí jehož hodnot jsou mutace proměnných korelovány. Tento vektor představuje tzv.

[1] Pohyb je zde míněn ve smyslu změny od pozice předchozího řešení k nově vygenerovanému.

rotační úhly, které jsou zahrnuty při tvorbě kovarianční matice, pomocí které pak dochází ke generování vektoru mutace. Tato metoda umožňuje oproti autoevoluci typu 2 natočení elipsy (pravděpodobnosti generování potomků) ve směru k lokálnímu maximu, jak ukazuje obrázek 4.7. Ovšem je nutné vzít v potaz, že dochází nejenom k možnosti rychlejší konvergence k lokálnímu maximu, ale také možnosti konvergence opačným směrem. Příslušné vztahy pro autoevoluci typu 3 lze najít v [30].



Obrázek 4.7: Pohyb kandidátního řešení (černé kolečko) při autoevoluci typu 3 je nakloněn ve směru k lokálnímu maximu.

Kapitola 5

Implementace

Implementační část této práce spočívala nejprve ve vytvoření vlastní relační MySQL databáze *Stability*, jejíž schéma je součástí přílohy A, pak vydolování ohodnocených mutací pro trénovací datovou sadu za využití dolovacího skriptu. Následovalo pak vytvoření skriptů pro řízení dávkových výpočtů predikcí změn stabilit proteinů pro vybrané existující nástroje. Na výsledky těchto výpočtů byla aplikována evoluční strategie dvou typů za účelem vytvoření rozhodovacího modelu trénovaného na vybudované trénovací datové sadě. Pro závěrečné ověření úspěšnosti tohoto modelu byla také vytvořena nezávislá testovací datová sada.

5.1 Trénovací datová sada

Jako zdroj pro trénovací datovou sadu byla zvolena volně dostupná databáze ProTherm obsahující experimentálně zjištěná data k aminokyselinovým mutacím proteinů. Tato databáze byla kompletně převedena do již zmiňované vlastní relační MySQL databáze *Stability*.

5.1.1 Dolování

Proces dolování byl proveden pomocí automatického skriptu napsaného v jazyce Perl. Při dolování se vyskytlo několik problémů, které byly řešeny pro zachování dostatečně velkého počtu vydolovaných mutací. Databáze ProTherm v době dolování obsahovala přes 25000 záznamů a jelikož data pochází od různých autorů, kteří jsou pak sami částečně zodpovědní za případnou korekci, bylo nutné provádět dodatečné opravy, či případně neopravitelné záznamy zcela přeskočit. Pro účely této práce byla nejdůležitější například uvedená naměřená hodnota $\Delta\Delta G$ a zda se jedná o jednobodovou mutaci, či nikoliv.

Mezi dílčí úkoly dolovacího skriptu stojící za zmínku patří například automatické stahování PDB souborů (z databáze Protein Data Bank) ke všem uvedeným proteinům, extrakce aminokyselinové sekvence ze záznamů SEQRES a převedení z 3-písmenných aminokyselinových zkratek na 1-písmenné. Pomocí .pdb souborů pak skript kontroluje korektnost mutací a to jak pozice, tak správného wild-type residua na dané pozici v sekvenci proteinu. V případě nesrovnalostí se tyto mutace skript snaží přepočítat a opravit. K tomu využívá záznamů ATOM. Může se totiž stát, že daná experimentální metoda zjišťování struktury není schopna s dostatečnou přesností určit, jaké aminokyseliny se na dané pozici vyskytují (typicky jde o pozice na začátcích či koncích řetězce). Tato skutečnost je pak patrná v indexaci aminokyselin uvedené v záznamech s atomovými koordináty ATOM a lze pak

pomocí nich pozice odpovídajících mutací přepočítat. Díky tomuto postupu byla přepočítána pozice cca 1000 záznamů z databáze ProTherm, které by byly jinak zahozeny jako chybné.

Při následné konstrukci trénovací datové sady byly vybrány takové mutace, které

- bylo možné ohodnotit celou sadou zvolených nástrojů pro predikci stabilit,
- mají v databázi ProTherm definovanou hodnotu $\Delta\Delta G$,
- byly experimentálně změřeny v rozsahu $pH \in \langle 3, 9 \rangle$ a teplotě pod 50°C.

Pro poslední podmínku navíc platí, že existuje-li více záznamů stejné mutace změřené při rozdílné hodnotě pH, pak se použije jen jediný záznam nejbližší fyziologickému pH = 7. Mají-li však mutace totožné hodnoty pH, tak bude do datové sady vložen záznam se zprůměrovanými hodnotami $\Delta\Delta G$.

5.1.2 Statistiky

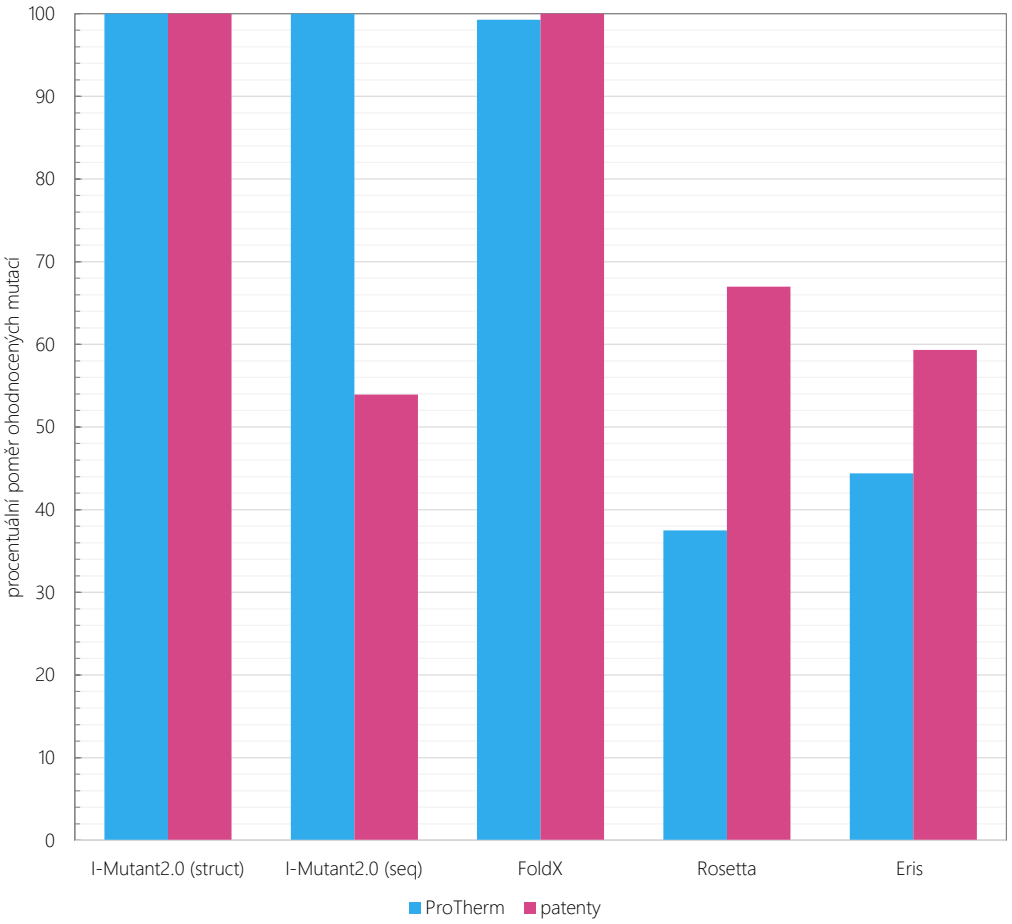
Pomocí výše popsaného dolovacího skriptu bylo získáno a uloženo do databáze 11910 záznamů, z nichž 9642 bylo jednobodových mutací. Po následné aplikaci pravidel specifikující oblast mutací trénovací datové sady, se počet záznamů zúžil na 892. I když se může na první pohled zdát, že to je nízké číslo, datová sada pokrývá poměrně velkou část stavového prostoru mutací, jak ukazuje procentuální distribuce mutací v tabulce 5.1, případně další zobrazení distribuce mutací pomocí grafů je pak obsahem přílohy B.

	A	C	D	E	F	G	H	I	K	L	M	N	P	Q	R	S	T	V	W	Y
A	0	0,11	0,22	0,22	0,11	1,35	0,22	0,11	0,22	0,34	0,11	0,11	0,45	0,22	0,11	0,56	0,45	0,78	0,11	0,11
C	0,56	0	0	0	0	0,11	0	0,11	0	0,11	0,11	0	0	0	0	0,78	0,11	0,45	0	0
D	2,58	0,11	0	0,67	0,11	0,67	0,67	0,11	0,67	0,11	0,11	1,23	0,11	0,11	0,11	0,56	0,11	0,11	0	0
E	1,91	0,22	0,56	0	0,22	0,45	0,11	0,11	1,12	0,22	0,22	0,45	0,11	1,12	0	0,34	0,34	0,11	0,11	0,11
F	1,12	0	0	0	0	0,22	0	0	0,11	0,45	0	0	0	0	0	0,11	0,11	0,34	0,45	0,22
G	1,12	0,11	0,22	0,11	0	0	0,11	0	0,11	0,11	0	0,11	0,11	0	0,22	0,56	0,11	0,34	0,11	0
H	0,56	0,11	0,11	0,11	0	0,22	0	0	0,11	0,22	0	0,11	0,11	0,34	0,11	0,22	0,11	0	0,11	0,56
I	2,24	0,22	0	0	0,22	0,78	0	0	0	0,45	0,22	0,11	0	0	0	0	0,56	2,8	0,11	0
K	1,79	0	0,11	0,9	0,22	0,67	0,11	0	0	0	0,45	0,45	0,11	0,45	0,67	0	0	0,22	0	0,11
L	2,24	0,34	0	0,11	0,22	0,56	0,22	0,34	0	0	0	0,11	0,22	0,11	0,34	0,11	0,56	1,46	0,11	0,11
M	0,34	0	0	0	0,11	0	0	0,11	0,22	0,45	0	0	0	0	0	0	0,11	0,34	0	0
N	1,68	0	0,67	0	0	0	0,11	0,11	0	0,11	0,11	0	0	0,11	0	0,22	0,11	0,11	0	0
P	1,35	0	0	0	0	0,11	0	0	0	0,11	0	0	0	0	0	0,11	0	0,11	0	0
Q	1,23	0	0	0	0	0,56	0	0,11	0,22	0,11	0	0	0,11	0	0,11	0,11	0	0	0	0
R	0,9	0,11	0	0,11	0	0,11	0,22	0	0,22	0	0,11	0	0	0,22	0	0,11	0	0	0	0
S	1,68	0,11	0,11	0,11	0	0,34	0	0	0	0,11	0	0,11	0	0,11	0	0	0,34	0,22	0	0
T	1,57	0,22	0,34	0,22	0,11	0,78	0,22	0,34	0	0,11	0	0,22	0,34	0,34	0,11	0,78	0	1,79	0	0,11
V	4,04	1,01	0	0,11	0,22	1,46	0,34	1,91	0,11	1,12	0,11	0,22	0,34	0	0,11	0,22	2,35	0	0	0,22
W	0	0	0	0	0,9	0	0,22	0	0	0,22	0	0	0	0	0	0	0	0	0	0,56
Y	0,9	0,22	0,22	0	3,25	0,34	0,11	0	0	0,11	0	0,22	0,11	0,22	0,11	0,22	0	0,11	0,34	0

Tabulka 5.1: Procentuální zastoupení jednotlivých mutací v trénovací datové sadě. Řádky odpovídají zdrojovým (wild-type) aminokyselinám a sloupce odpovídají aminokyselinám, na které bylo mutováno.

5.2 Dávkové výpočty

Druhým krokem implementační části bylo vytvoření skriptů pro řízení dávkových výpočtů predikcí stabilit pro vybrané existující nástroje. Pro každý nástroj byl vytvořen samostatný skript. Vstupem všech skriptů je CSV[1] soubor obsahující sloupce se všemi parametry zkoumané mutace, které daný nástroj umožňuje při predikci nastavit. Každý záznam tedy odpovídá jedné konkrétní mutaci. Skripty pracují podle principiálně stejného schématu, kdy prochází jednotlivé záznamy vstupního souboru resp. jednotlivé mutace a spouští výpočet na daném nástroji. V případě, že některé nástroje počítají mutaci na dané pozici na všechny ostatní základní aminokyseliny, skript si tento výpočet uchovává a pro výpočet mutací na stejném proteinu, stejné pozici a za stejných podmínek nástroj znovu nespouští a používá již vypočtené výsledky uložené v souboru. Příkladem takového nástroje je I-Mutant2.0.



Obrázek 5.1: Procentuální poměr ohodnocených mutací pro jednotlivé nástroje na zdrojových množinách mutací pro obě datové sady.

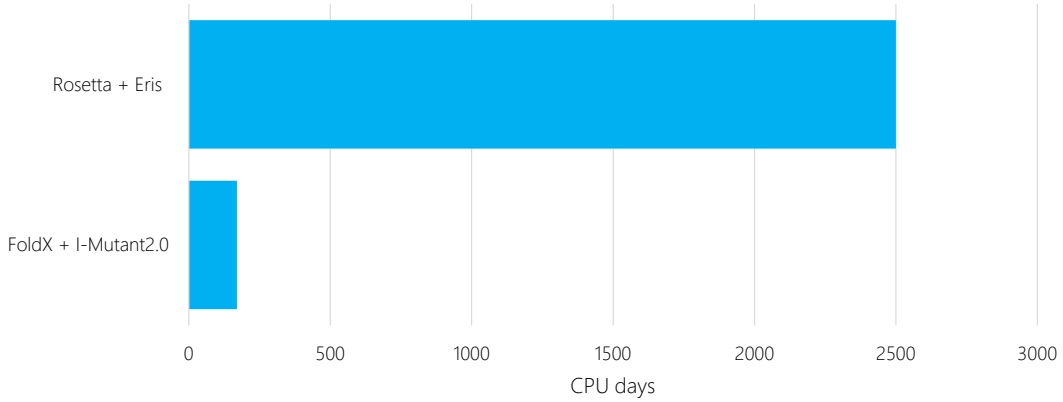
[1] V praxi se používají různé varianty formátování oproti RFC standardu [37], zde je jako oddělovač použit středník a hodnoty nejsou uvozeny v uvozovkách.

5.2.1 Statistiky

Jedním z faktorů, který ovlivňoval velikost trénovací či testovací datové sady, byla schopnost nástrojů ohodnotit jakékoliv mutace. Na obrázku 5.1 je vidět, jak si jednotlivé nástroje vedli při ohodnocování zdrojové množiny mutací pro trénovací datovou sadu (ProTherm) a pro testovací datovou sadu (patenty). Nejméně úspěšný nástroj (pro danou zdrojovou množinu mutací) v otázce schopnosti ohodnotit mutace pak určil počet záznamů (mutací) dané datové sady. Jak je z grafu patrné, Rosetta i Eris rapidně zredukovaly velikosti trénovací i testovací datové sady. Nejčastějšími důvody byla nemožnost přečíst PDB soubor z důvodu chybějících záznamů pro proteinovou páteř, přítomnost neobvyklých aminokyselin, nebo chybová návratová hodnota. S podobnými problémy při ohodnocování mutací se potýkali například v [40].

Jedním z kritérií při výběru skupiny nástrojů byla možnost jejich instalace na lokálním výpočetním systému pro možnost využití výpočetního střediska MetaCentrum. Toto umožnilo následnou paralelizaci a urychlení výpočtů, kdy mohl jeden nástroj běžet v několika instancích a zpracovávat jenom část z trénovací nebo testovací datové sady. Při prvotních zkouškách paralelizace běhu nástroje I-Mutant2.0 (strukturní verze) bylo zjištěno, že není možné spustit souběžně dvě instance tohoto nástroje, které pracují na rozdílných mutacích, ale stejném proteinu. Tento nástroj si totiž vytváří pomocný soubor pro daný protein, u kterého docházelo ke kolizím po-té, co jedna instance soubor smazala a druhá z něj teprve potřebovala číst apod.

Pro představu výpočetní náročnosti jednotlivých nástrojů slouží graf 5.2, který ukazuje souhrnnou výpočetní náročnost pro skupiny nástrojů v jednotkách procesorových dnů. Jak je patrné, Rosetta a Eris, jakožto zástupce *state of the art*[2] nástrojů, jsou sice vysoce výpočetně náročné, ale jak vyplynulo z provedených testů, podávají téměř konstantní úspěšnost predikce. Na rozdíl od nástroje I-Mutant2.0 tak u nich nenastane situace, kdy na jednu sadu mutací podávají dobré výsledky a na jinou u nich dochází k velikému propadu úspěšnosti predikce, jako tomu nastalo právě u I-Mutant2.0. Uživatel si tedy může být relativně jist, jaké výsledky od nich může očekávat, nezávisle na ohodnocované mutaci.



Obrázek 5.2: Rozdíly vytížení výpočetních zdrojů MetaCentra mezi zvolenou sadou nástrojů.

[2]Nástroje na úrovni doby (dobově vyspělé).

5.3 Aplikace evoluční strategie

Úloha evoluční strategie v tomto problému spočívala v nalezení vah k jednotlivým nástrojům a jejich následné aplikaci ve formě násobících koeficientů při kalkulaci konsensuálního výsledku tvořeného kombinací výstupů jednotlivých nástrojů. Cílem bylo, aby se výsledná hodnota blížila co nejvíce realitě. Ideální stav by byl takový, že po aplikaci vah se získá přesnější hodnota predikce, než hodnota predikce nejlepšího nástroje.

Zkoumaná hodnota výstupu nástrojů byla predikce změny Gibbsovy volné energie $\Delta\Delta G$. Mějme tedy množinu n nástrojů $T = \{t_1, t_2, ..., t_n\}$. Od každého z nich získáme jeho hodnotu predikce $\Delta\Delta G$, což nám tvoří množinu predikcí, nazvěme ji P, kde $P = \{p_{t_1}, p_{t_2}, ..., p_{t_n}\}$. Nalezené hodnoty vah pomocí evoluční strategie pro jednotlivé nástroje tvoří množinu $W = \{w_1, w_2, ..., w_n\}$. Pak výsledná hodnota predikce $\Delta\Delta G$ s využitím kombinace n nástrojů pomocí evoluční strategie (nazvěme ji $meta_{ddg}$) je rovna následujícímu vztahu:

$$meta_{ddg} = \frac{\sum_{i=1}^{n} p_{t_i} w_i}{\sum_{i=1}^{n} w_i} \tag{5.1}$$

V této práci byla implementována jednak základní evoluční strategie s pravidlem 1/5 a také pokročilejší evoluční strategie s autoevolucí řídících parametrů (autoevoluce typu 2), obě popsané v kapitolách 4.1.4 a 4.1.5. Důvodem implementace obou variant byla snaha zjistit, zda v tomto případě autoevoluce řídících parametrů pozitivně ovlivní výsledek, nebo je pro tento řešený problém zbytečná a vyplatí se použít základní pravidlo 1/5.

Pro tento účel byl vytvořen skript *es.pl*, který implementuje oboje zmiňované varianty. Parametry tohoto skriptu je CSV soubor s trénovací datovou sadou (jeho formát specifikuje tabulka D.1 v příloze D) a případně i varianta evoluční strategie (implicitně skript spouští evoluční strategii s autoevolucí typu 2).

ES TYPE: AE type 2			EPOCHS = 100		ITER = 100
FOLDX	I2SEQ	I2STR	ROSETTA	ERIS	KK
0.505	3.835	0.381	0.424	0.079	0.5765293793

Tabulka 5.2: Ukázka výstupu skriptu *es.pl*.

Prvním krokem je tedy inicializace prvního jedince (chromozomu), který je zároveň rodičem. Dochází k výpočtu jeho tzv. fitness funkce, kterou zde představuje metrika zvaná Pearsonův korelační koeficient. Metrika udává míru podobnosti mezi dvěma množinami jako hodnotu z intervalu $\langle -1, 1 \rangle$, kde -1 značí zcela nepřímou závislost a 1 značí zcela přímou závislost. Hodnoty okolo 0 pak značí, že množiny nemají žádnou závislost. Dvě množiny, mezi kterými je fitness funkce počítána, jsou jednak množina X představující reálné hodnoty predikce $\Delta\Delta G$ získané z trénovací resp. testovací datové sady a množina Y reprezentující odpovídající hodnoty $meta_{ddg}$ vypočtené pomocí vzorce 5.1. Vzorec pro výpočet Pearsonova korelačního koeficientu je pak

$$P_{kk} = \frac{AVG(XY) - AVG(X)AVG(Y)}{\sqrt{AVG(X^2) - AVG^2(X)}\sqrt{AVG(Y^2)AVG^2(Y)}}, \tag{5.2}$$

kde AVG značí aritmetický průměr.

Skript následně pouští cyklus, který končí po dosažení nastaveného počtu epoch, kde každá epocha odpovídá zvolené variantě evoluční strategie. Výstupem skriptu je formátovaná tabulka s nejlepšími řešeními tak, jak jsou postupně nalézány. Příklad takového výstupu je vidět v tabulce 5.2. Nejlepší nalezené řešení je tedy vždy v posledním řádku tabulky. Tabulka obsahuje záhlaví s nastavenými parametry, dále pak záhlaví názvů sloupců s řešením a pak už jednotlivá nejlepší řešení v každé epoše. Sloupce s názvy nástrojů udávají nalezenou váhu pro daný nástroj, sloupec *KK* pak obsahuje hodnotu Pearsonova korelačního koeficientu pro dané váhy.

5.4 Testovací datová sada

Zdrojem pro testovací datovou sadu byly vybrány volně dostupné patenty: [4], [13], [12] a [6] nalezené pomocí služby Google Patents[3] s využitím klíčových slov: enzyme, protease, variants, improved stability, improved activity, improved affinity. Patenty byly jako zdroj testovací datové sady vybrány kvůli tomu, že splňovaly požadavek nezávislosti testovací datové sady vůči trénovací datově sadě. Tato nezávislost byla potvrzena neúspěšným hledáním sekvencí popsaných v patentech v trénovací datové sadě z databáze ProTherm. Celkem byly dolovány tyto čtyři patenty:

1. Z patentu číslo US2010/0192985 [4] byly dolovány mutace související s experimentem zkoumajícím efektivitu patentovanou varianty serinové proteázy při odstraňování skvrn tvořených krví, mlékem a inkoustem - cílem je zlepšit vlastnosti pracích prášků či tablet. Pro zmíněný experiment byla serinová proteáza získána z bakterie *Bacillus subtilis*. Proteázy [26] jsou skupinou enzymů, které štěpí proteiny. Proteázy dále patří do třídy hydroláz [26], které nesou svůj název díky tomu, že katalyzují hydrolytické štěpení peptidové vazby aminokyselin. Jedním z typů proteáz, kam se právě řadí serinové proteázy, jsou endoproteázy, které štěpí proteiny uvnitř polypeptidového řetězce a narušují jeho terciální strukturu. Serinové proteázy se vyznačují tím, že obsahují katalytickou -OH skupinu (serin) v aktivním místě (místo, kterým proteáza na štěpený protein působí) [26]. Více informací o serinových proteázách lze nalézt v [34].

2. Druhým dolovaným patentem byl patent číslo US2009/0314286 [13] zkoumající pozměněné vlastnosti po mutaci variant patentované α-amylázy, získané z bakterie *Bacillus stearothermophilus*. Mutovaná α-amyláza pak může sloužit při přeměně škrobů, produkci etanolu, praní prádla, umývání nádobí, čištění pevných ploch, či při produkci sladidel. Amyláza je enzym zajišťující štěpení škrobu na jednodušší sacharidy. Amylázy patří, stejně jako proteázy, také do třídy hydroláz a katalyzují tedy hydrolitické štěpení peptidových vazeb. Existují tři typy amylázy: α-amyláza, β-amyláza a γ-amyláza. α-amyláza získaná z výše jmenované bakterie se například používá jako aditivum (potravinářská přídatná látka) v kombinaci s moukou a jelikož štěpí škrob v mouce na jednoduché sacharidy, urychluje tak proces kvašení droždí [39].

3. Dalším patentem zvoleným pro dolování byl patent číslo US8236542 [12], jehož předmětem bylo zkoumání zmutovaných variant celulázy, především těch, které se v menší míře oproti nezmutované variantě váží na materiály nerostlinného původu (netvořené z celulózy) [12]. Celulázy jsou soubor enzymů katalyzující štěpení celulzoy, což je polysacharid tvořený řetězením molekul glukózy. Celulóza se vyskytuje především v rost-

[3] www.google.com/patents

linách, a tedy již zmíněný enzym celulázu mají hlavně býložravci z důvodu trávení rostlin [7].

4. Posledním z dolovaných patentů byl patent číslo US2011/0262999 [6], který se zabývá návrhem proteáz, které jsou použitelné v jistých podmínkách a potřebách. Konkrétně se zabývá subtilysin proteázou (patřící do skupiny serinových proteáz), získanou z bakterie *Bacillus amyloliquefaciens*. Tato specifická proteáza je v dnešní době zkoumána jako modelový případ pro změnu vlastností enzymů na základě jednobodových mutací [9]. Tyto mutace pak vedou k zvýšené aktivitě, změně účelu použití enzymu nebo změně pH aktivity [9].

5.4.1 Dolování

Dolování záznamů mutací pro testovací datovou sadu spočívalo ve využití nástrojů pro automatické rozpoznávání textu z obrázků. Rozpoznávání textu bylo nutností z důvodu dostupnosti zmíněných patentů pouze ve formě naskenovaných dokumentů. Prvním použitým nástrojem byl Adobe Acrobat se zabudovanou schopností rozpoznávat text v PDF dokumentech tvořených z obrázků. Postupem času se ukázal jako nedostačující pro dolování textu ve formě tabulkových dat. Po aplikaci rozpoznávání textu bylo zkopírování tabulkových dat nutné doplnit následným dodatečným zpracováním, které bylo časově náročné. Funkce rozpoznávání textu je v aplikaci Adobe Acrobat spíše vhodná pro souvislé texty. Z tohoto důvodu byl použit nástroj ABBY Fine Reader, který se specializuje právě na převod naskenovaných dokumentů na editovatelné dokumenty. Hlavní výhodou byla pak možnost nastavit rozvržení dolované tabulky a následná možnost exportu dané tabulky do formátu XLS pro další zpracování v programu Microsoft Excel.

Popis proteinu v patentu	**Nalezený protein (řetězec)**	**Statistika zarovnání programu BLAST**
serinová proteáza (Bacillus subtilis)	1NDQ (A)	identities 100%, positives 100%, gaps 0%
α-amyláza (Bacillus stearothermophilus)	1HVX (A)	identities 99% (508/515), positives 98% (509/515), gaps 0%
celuláza	1CB2 (A)	identities 100%, positives 100%, gaps 0%
subtilysin proteáza (Bacillus amyloliquefaciens)	2SIC (E)	identities 100%, positives 100%, gaps 0%

Tabulka 5.3: Výsledek hledání referenčních proteinů pro sekvence vydolované z patentů. Druhý sloupec obsahuje PDB identifikátor vybraných (nejpodobnějších) proteinů z programu BLAST. Třetí sloupec obsahuje hodnoty: *identities* - poměr identických residuí vůči jejich celkovému počtu, *positives* - poměr strukturně podobných residuí vůči jejich celkovému počtu a *gaps* - poměr vložených mezer při zarovnání vůči celkové délce zarovnávaných sekvencí.

Microsoft Excel byl využit pro závěrečnou korekci případných chyb při rozpoznávání textu. Jednalo se především o chyby typu záměny čísel za písmena a naopak (0 za O, 1 za l, 5 za S, apod.). Pro tyto účely bylo vytvořeno schéma v aplikaci Microsoft Excel, které

kontrolovalo korektnost zápisu mutace. Každá mutace je v patentech značena jako XNY, kde:

- X je jednopísmenný kód zdrojové aminokyseliny,
- N je číslo (ne vždy o stejném počtu číslic) značící pozici mutace v řetězci a
- Y je aminokyselina, na kterou bylo mutováno.

Dolovaná sekvence mutovaného proteinu sloužila pak jako referenční pro zjištění, zda zdrojová aminokyselina X se opravdu vyskytuje na pozici N v řetězci.

Pro účely ohodnocení mutací z testovací datové sady byly nalezeny nejpodobnější proteiny (z dostupných databází) k vydolované sekvenci z patentu. Daná sekvence byla vložena do webového serveru poskytující program BLAST. Pro tyto účely byla využita jeho varianta *blastp* pro zarovnávání sekvencí aminokyselin. Jako tzv. *query* (neboli dotazovaná) sekvence byla vložena vydolovaná sekvence z patentu. BLAST pak pomocí zarovnávání sekvencí oproti dotazované našel ve zvolené proteinové databázi odpovídající proteiny a ty pak vypsal seřazené podle podobnosti. Jako prohledávaná databáze proteinů byla zvolena databáze PDB (*Protein Data Bank*). Tento postup bylo nutné provést pro získání PDB souborů k proteinům a pro následnou možnost ohodnocení testovací datové sady všemi nástroji. Jako odpovídající proteiny byly vybrány ty, jenž jsou uvedené v tabulce 5.3.

	A	C	D	E	F	G	H	I	K	L	M	N	P	Q	R	S	T	V	W	Y
A	0	1,05	0,59	0,67	0,59	1,09	0,5	0,55	0,42	0,67	0,46	0,5	0,42	0,5	0,42	1,13	1,01	0,88	0,5	0,55
C	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
D	0,13	0,13	0	0,04	0,13	0,08	0,08	0,04	0,08	0,08	0,13	0,17	0	0,08	0,13	0,13	0,08	0,04	0,08	0,08
E	0,13	0,17	0,08	0	0,17	0,17	0,13	0,13	0,08	0,13	0,08	0,17	0,13	0,13	0,04	0,17	0,17	0,21	0,17	0,17
F	0	0,08	0	0	0	0,04	0,08	0,04	0,04	0,08	0,04	0,08	0,04	0,04	0,04	0	0,08	0,08	0	0,08
G	0,59	0,42	0,29	0,34	0,55	0	0,34	0,55	0,42	0,46	0,59	0,17	0,5	0,34	0,46	0,71	0,42	0,5	0,46	0,38
H	0,17	0,04	0,04	0,08	0,13	0,17	0	0,13	0,08	0,13	0,13	0,17	0	0,08	0,13	0,21	0,21	0,13	0,13	0,17
I	0,17	0,17	0,08	0,04	0,17	0,17	0,08	0	0,08	0,21	0,34	0,17	0,08	0,25	0,13	0,29	0,17	0,25	0,25	0,13
K	0,13	0,08	0,04	0,04	0,08	0,17	0,13	0,13	0	0,17	0,17	0,08	0,13	0,08	0,13	0,17	0,13	0,17	0,13	0,17
L	0,55	0,42	0,17	0,29	0,42	0,38	0,21	0,38	0,25	0	0,46	0,25	0,17	0,25	0,08	0,42	0,5	0,38	0,29	0,38
M	0,13	0,13	0	0,04	0,04	0,08	0,08	0,04	0,04	0,08	0	0,08	0	0,08	0	0,08	0,08	0,08	0,04	0,04
N	0,42	0,59	0,59	0,5	0,59	0,63	0,29	0,42	0,46	0,42	0,42	0	0,21	0,34	0,38	0,63	0,55	0,5	0,38	0,5
P	0,38	0,38	0,29	0,34	0,25	0,42	0,17	0,42	0,17	0,29	0,34	0,21	0	0,25	0,25	0,34	0,29	0,34	0,29	0,29
Q	0,34	0,25	0,17	0,29	0,25	0,34	0,21	0,17	0,29	0,34	0,25	0,25	0,25	0	0,34	0,29	0,29	0,29	0,34	0,34
R	0,29	0,21	0,17	0,17	0,17	0,25	0,21	0,13	0,29	0,21	0,25	0,17	0,08	0,21	0	0,25	0,25	0,17	0,25	0,25
S	0,97	0,8	0,5	0,76	0,71	1,05	0,71	0,8	0,5	0,76	0,76	0,67	0,5	0,67	0,84	0	0,97	0,67	0,71	0,67
T	0,5	0,46	0,29	0,29	0,29	0,42	0,25	0,5	0,34	0,46	0,34	0,5	0,25	0,34	0,25	0,63	0	0,42	0,34	0,34
V	0,67	0,5	0,29	0,46	0,5	0,67	0,38	0,46	0,17	0,71	0,59	0,42	0,46	0,46	0,25	0,76	0,88	0	0,34	0,29
W	0,08	0,04	0,08	0,08	0	0	0	0,04	0,04	0,04	0,08	0,04	0	0,04	0,04	0,04	0,04	0,04	0	0,04
Y	0,21	0,21	0,21	0,13	0,21	0,21	0,13	0,17	0,13	0,13	0,25	0,25	0,13	0,13	0,13	0,08	0,25	0,21	0,25	0

Tabulka 5.4: Procentuální zastoupení jednotlivých mutací v testovací datové sadě. Řádky odpovídají zdrojovým (wild-type) aminokyselinám a sloupce odpovídají aminokyselinám, na které bylo mutováno.

5.4.2 Statistiky

Pomocí popsaného postupu dolování mutací z patentů pro testovací datovou sadu bylo získáno celkem 14612 záznamů mutací, z nichž jednobodových bylo celkem 6839. Jelikož

tento postup nebyl tolik automatizovaný a byla nutná manuální kontrola nástrojů pro rozpoznávání textu a korektnosti mutací, většina chybných záznamů tak byla opravena a došlo k vynechání pouze 32 chybných záznamů. Z prvního patentu [4] bylo získáno 3965 záznamů jednobodových mutací, z druhého patentu [13] pak 2631 jednobodových mutací. Třetí patent [12] byl zdrojem 171 jednobodových mutací a poslední čtvrtý patent [6] přispěl do celkového počtu 40 mutacemi.

Jelikož tato datová sada byla dolována z jednotlivých experimentů, nebylo zapotřebí dále specifikovat pravidla pro případy existence vícero záznamů stejné mutace, jako v případě dolování mutací z databáze ProTherm. Ovšem i u testovací datové sady došlo poměrně k velké redukci počtu mutací, které bylo možné ohodnotit všemi nástroji, jak ukazuje graf 5.1. Z celkového počtu 6839 mutací bylo možné ohodnotit pouze 2382 záznamů. Avšak i za těchto okolností bylo výsledné rozložení mutací takové, že pokrývalo téměř celý stavový prostor. Jediným nedostatkem by zde mohl být fakt, že v testovací datové sadě chybí mutace aminokyseliny cystein, jak je vidět v tabulce rozložení aminokyselinových mutací 5.4. Obsahem přílohy C jsou pak detailní grafy rozložení mutací pro jednotlivé aminokyseliny, rovněž jako u trénovací datové sady.

Kapitola 6

Experimenty a výsledky

Tato kapitola se zabývá shrnutím výsledků dosažených z provedených experimentů a diskusí nad zjištěnými skutečnostmi. Nejprve je popsáno nastavení evoluční strategie a po-té jsou diskutovány výsledky na jednotlivých datových sadách. Pro reprezentaci výsledků a ověření úspěšnosti jednotlivých nástrojů a konsensuálních metod, byly použity metriky:

- *Pearsonův korelační koeficient*,
- *Spearmanův koeficient pořadové korelace* (Spearmanův korelační koeficient) a
- *normalizovaná přesnost*.

6.1 Nastavení ES

Při experimentování s nastavením parametrů evoluční strategie bylo postupováno tak, že byly nejprve nastaveny hodnoty doporučené literaturou (pokud byly dostupné) a poté byly tyto hodnoty upravovány a sledovány, zda mají pozitivní vliv na rychlejší konvergenci k hledanému řešení. Hledaným řešením zde byla nejvyšší možná hodnota Pearsonova korelačního koeficientu, snaha byla o získání koeficientu co nejblíže číslu 1. Konstanty ovlivňující počty iterací byly nejprve nastaveny na vyšší čísla a poté snižovány pro získání dostatečného počtu iterací pro zatím nejlepší nalezené řešení. Výsledné parametry jsou vypsány v tabulce 6.1.

Parametr	**Hodnota**
počáteční σ	0,5
počáteční váhy jednotlivých nástrojů	1
počet iterací (potomků) v jedné epoše	35
počet epoch (generací)	82
konstanta c_i pro zvětšení kroku pro ES 1/5	1,22
konstanta c_d pro zmenšení kroku pro ES 1/5	0,82
minimální σ pro ES AE2	0,2
parametr učení τ pro ES AE2	0,3162
specifický parametr učení τ_i pro ES AE2	0,4728

Tabulka 6.1: Nastavení parametrů ES při získání nejlepšího řešení a zároveň použité ve skriptu *es.pl*.

Hodnoty parametrů specifických pro evoluční strategii s pravidlem 1/5 byly nastaveny podle [8]. Hodnoty parametrů specifických pro evoluční strategii s autoevolucí typu 2 byly nastaveny dle vzorců uvedených v kapitole 4.1.5 podle [30]. I přes vysoký počet experimentů (řádově stovek pro dané nastavení), různá nastavení ostatních parametrů (σ, počtu iterací a epoch, počáteční hodnoty vah, apod.) nijak rapidně neovlivňovala kvalitu dosažených výsledků (docházelo zejména k brzkým uváznutí algoritmu na špatném řešení). Toto mohlo být způsobeno povahou řešeného problému, kde na základě informací poskytnutých evoluční strategii nebylo možné nalézt lepší řešení, a to ani při různé změně parametrů. Nejlepšího výsledku na trénovací datové sadě bylo dosaženo s parametry uvedenými v tabulce 6.1. Konkrétní hodnoty vah pro jednotlivé nástroje ukazuje tabulka 6.2.

	FoldX	I-Mutant2.0 (seq)	I-Mutant2.0 (stuct)	Rosetta	Eris
ES (AE 2)	-2.364	-35.97	0.7197	-3.228	-0.578
ES (1/5)	0.578	8.793	1.948	0.808	0.151

Tabulka 6.2: Nejlepší váhy pro jednotlivé nástroje nalezené pomocí evoluční strategie používající pravidlo 1/5 a autoevoluci typu 2. Je zde v obou případech vidět vysoké nadhodnocení nástroje I-Mutant2.0 (sekvenční verze) oproti ostatním nástrojům.

	FoldX	I-Mutant2.0 (seq)	I-Mutant2.0 (stuct)	Rosetta	Eris
ES (AE 2)	4.168	-	31,949	5.78	1.146
ES (1/5)	0.549	-	4.196	0.766	0.151

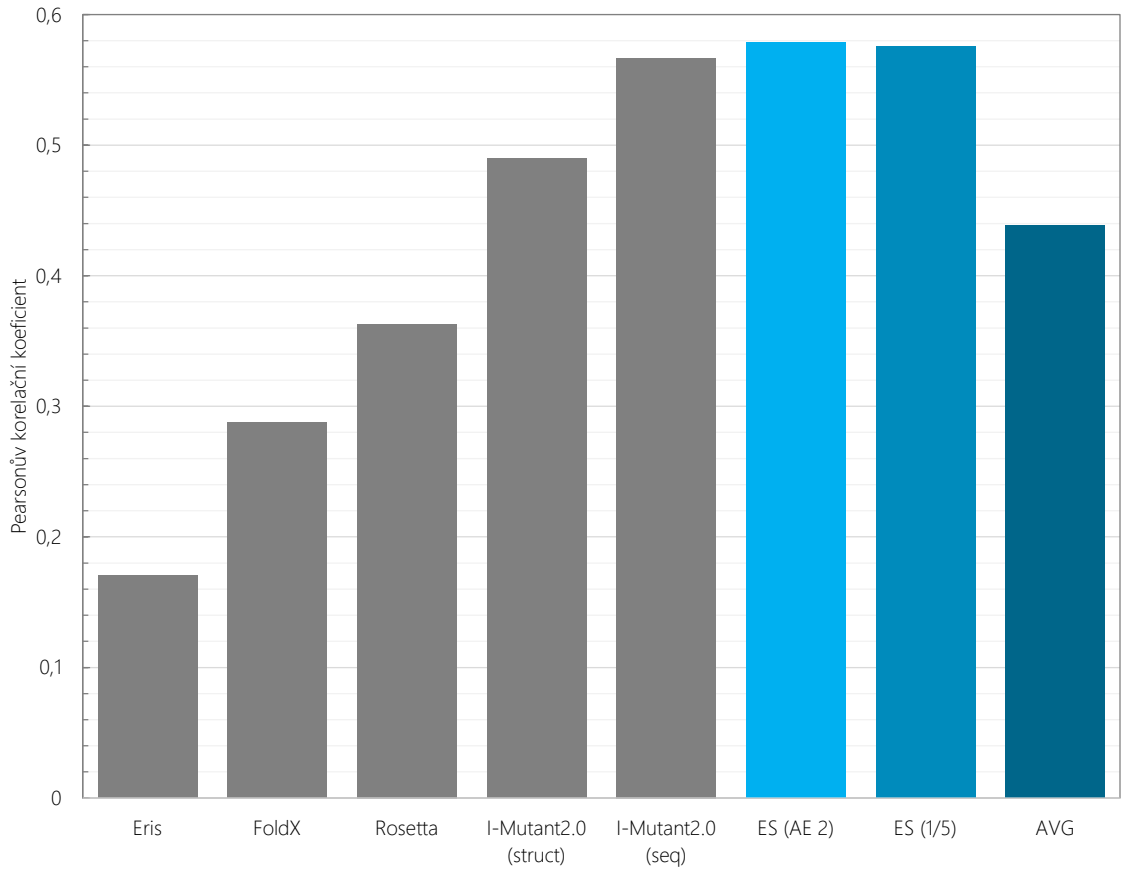
Tabulka 6.3: Nejlepší váhy pro jednotlivé nástroje (bez nejlepšího z nich) nalezené pomocí evoluční strategie používající pravidlo 1/5 a autoevoluci typu 2. Je zde vidět (oproti vahám v tabulce 6.2) zvýšení vlivu ostatních nástrojů.

6.2 Výsledky trénování

Jak již bylo uvedeno v kapitole o implementaci, pro zjištění úspěšnosti vytvořeného metaklasifikátoru s využitím evoluční strategie, bylo použito metriky Pearsonova korelačního koeficientu. Byla zjišťována korelace jednak výsledků jednotlivých nástrojů (samostatně) a jednotlivých konsensuálních metod (evoluční strategie dvou typů a prostého neváhovaného konsensu) s hodnotami experimentálně zjištěných změn stabilit. Výsledky jsou vyneseny v grafu 6.1.

Při prvním pohledu na to, jak si jednotlivé nástroje vedly, je vidět poměrně veliká převaha nástroje I-Mutant2.0 (v obou jeho verzích). To může na první pohled připadat neobvyklé, z důvodu rozdílů složitosti a náročnosti použitých metod predikce stability u jednotlivých nástrojů. Oproti ostatním nástrojům totiž I-Mutant2.0 nepoužívá relativně složité kvantově-chemické výpočty, ale pouze SVM model, získaný strojovým učením nad datovou sadou mutací, která byla tvořena převážně mutacemi z databáze ProTherm - je tedy zřejmé, že trénovací datová sada vytvořená pro účely této práce má značný překryv s trénovací datovou sadou tohoto nástroje. Toto vedlo k jeho zkresleným (nadhodnoceným) výsledkům. Skutečné predikční schopnosti tohoto nástroje (především na jemu neznámých mutacích) jsou výrazně slabší, jak bude ukázáno v kapitole 6.3.

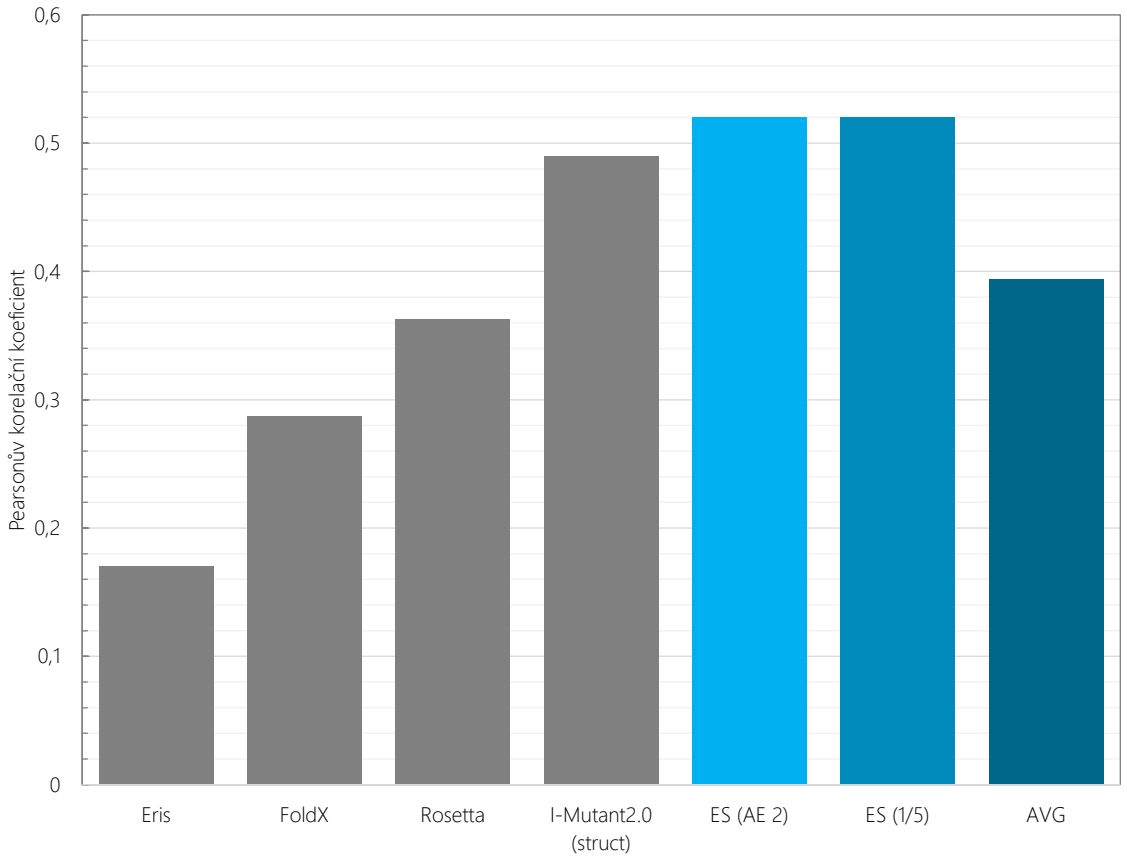
Při pohledu na výsledky jednotlivých konsensuálních metod je patrný markantní rozdíl mezi váhovaným a neváhovaným konsensem. Je zřejmé, že neváhovaný konsensus nepřináší oproti nejlepšímu nástroji žádné zlepšení. Naopak zhoršuje nejlepší nástroj o cca 12%. Naproti tomu váhovaný konsensus, jehož váhy byly nalezeny pomocí evoluční strategie s pravidlem 1/5 zlepšily výsledek nejlepšího nástroje o necelé jedno procento. Pomocí váhovaného konsensu jehož váhy byly nalezeny pomocí evoluční strategie s autoevolucí typu 2, bylo možné výsledek zlepšit o cca 1,2%. Lze tedy říci, že se potvrdily prvotní předpoklady o zlepšení nejlepšího nástroje za využití evoluční strategie a také o větším zlepšení při využití autoevoluce řídících parametrů oproti pravidlu 1/5.



Obrázek 6.1: Úspěšnost predikce jednotlivých nástrojů a konsensuálních metod na trénovacím datasetu.

Přínos evoluční strategie není z dosažených výsledků na trénovací datové sadě bohužel nijak markantní. Toto zjištění vedlo k pokusu, kdy byl ze sady kombinovaných nástrojů odstraněn nadhodnocený I-Mutant2.0 (sekvenční verze), aby došlo ke zvýšení vah ostatních nástrojů a současně tak možnému většímu ovlivnění výsledků predikce meta-klasifikátoru ostatními nástroji. Při kompletní sadě všech pěti nástrojů byly přibližné příspěvky (vypočtené z vah tabulky 6.2) nástroje I-Mutant2.0 (sekvenční verze) 72%, I-Mutant2.0 (strukturní verze) 15%, Rosetty 6%, FoldX 4% a Eris 1%. Po odstranění I-Mutant2.0 (sekvenční verze) došlo k následujícímu rozdělení (vypočtené z vah tabulky 6.3): I-Mutant2.0 (strukturní verze) 74%, Rosetta 13%, FoldX 9% a Eris 2%. Je tedy vidět dvojnásobné ovlivnění výsledků ostatními nástroji oproti prvotnímu rozdělení. Následně bylo možné pomocí evo-

luční strategie (obou typů) získat téměř trojnásobného zlepšení oproti původnímu měření se všemi nástroji. Konkrétně došlo ke zlepšení nejlepšího nástroje o více jak 3%. Tento fakt je patrný v grafu 6.2, je také vidět vyrovnání úspěšností obou typů evolučních strategií. Jak je vidět, oba typy dosahují stejného zlepšení nejlepšího nástroje a již lze považovat přínos evoluční strategie za výraznější.



Obrázek 6.2: Úspěšnost predikce jednotlivých nástrojů a konsensuálních metod (bez nejlepšího nástroje) na trénovacím datasetu.

Jelikož metrika *Pearsonova korelačního koeficientu* udává především to, jak jsou hodnoty predikce $\Delta\Delta G$ jednotlivých nástrojů a konsensuálních metod přesné, byla použita druhá metrika a to tzv. *normalizovaná přesnost* pro jiný úhel pohledu a možnost dalšího hodnocení vytvořeného meta-klasifikátoru. Tato metrika spočívá ve specifikaci úspěšnosti nástrojů na základě korektní klasifikace mutací do dvou tříd:

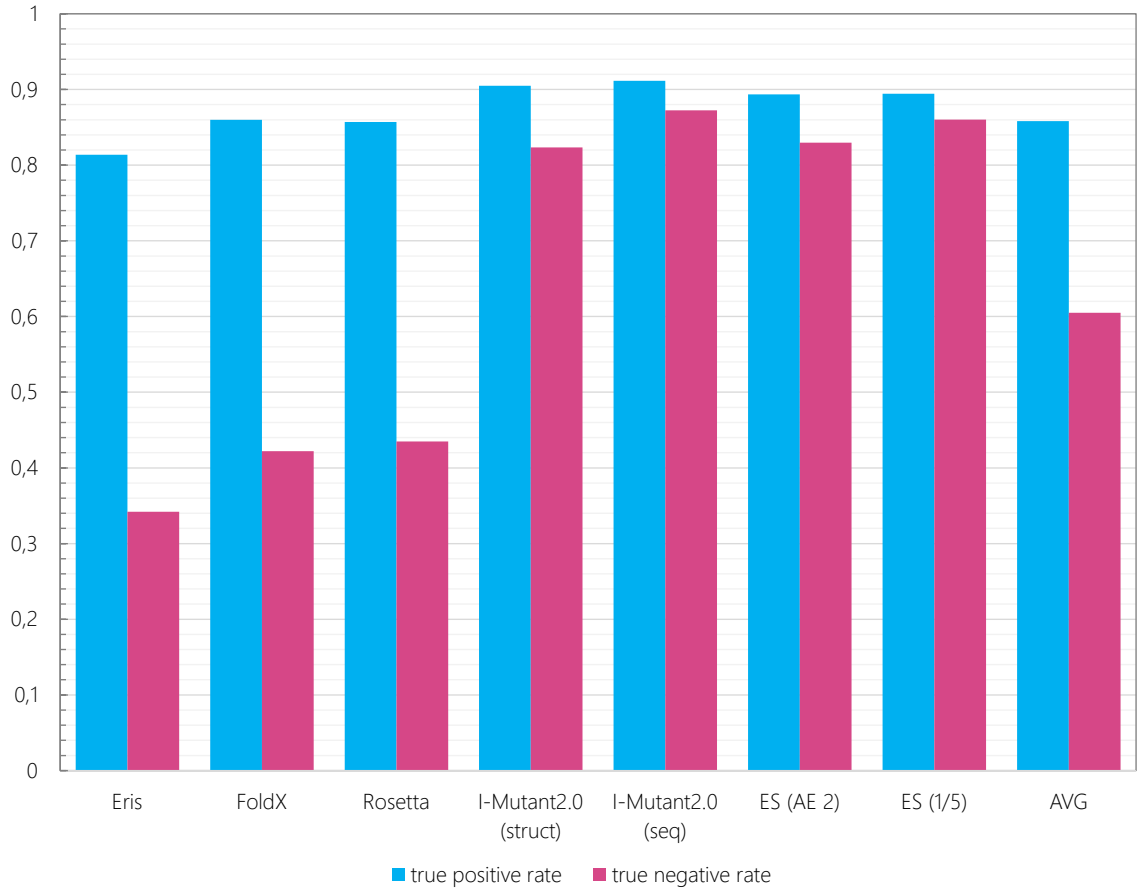
- *DELETERIOUS* - značící škodlivé mutace (při $\Delta\Delta G < 0$) a
- *BENIGN* - značící neutrální mutace (při $\Delta\Delta G \geq 0$).

Metrika *normalizované přesnosti* rozlišuje čtyři třídy, do kterých klasifikuje jednotlivá vyhodnocení mutací daným nástrojem. Těmito třídami jsou:

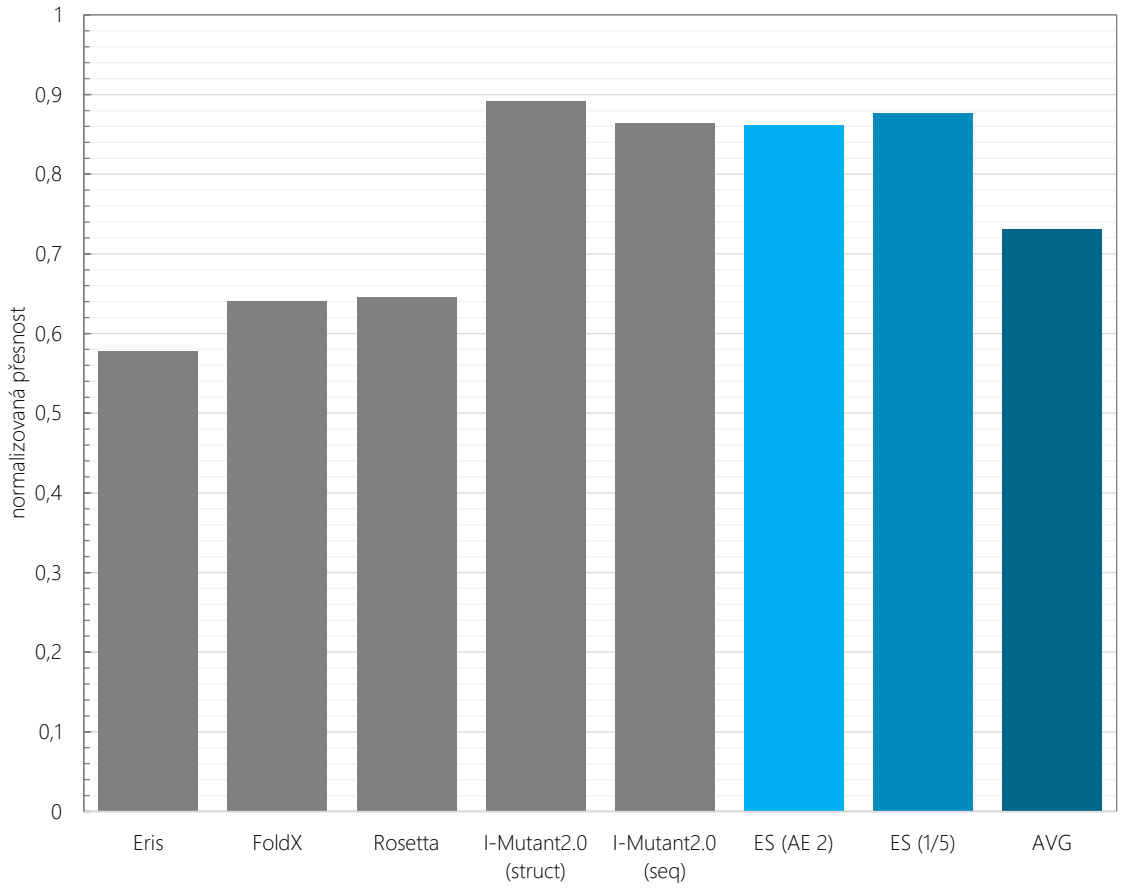
- *true positive* - pokud nástroj klasifikoval mutaci správně jako škodlivou,
- *true negative* - pokud nástroj klasifikoval mutaci správně jako neutrální,

- *false positive* - pokud nástroj klasifikoval mutaci nesprávně jako škodlivou a
- *false negative* - pokud nástroj klasifikoval mutaci nesprávně jako neutrální.

Na základě těchto tříd (statistik) pak vypočte aritmetický průměr z tzv. *true positive rate* (poměr správně vyhodnocených mutací z množiny všech mutací, které byly nástrojem vyhodnoceny jako škodlivé) a tzv. *true negative rate* (poměr správně vyhodnocených mutací z množiny všech mutací, které byly nástrojem vyhodnoceny jako neutrální). Tyto dvě hodnoty jsou zobrazeny v grafu 6.3. Jak si jednotlivé nástroje a konsensuální metody vedli na této metrice, zobrazuje 6.4. Je potřeba říci, že zatímco *Pearsonův korelační koeficient* je zaměřen především na přesnost predikce hodnot $\Delta\Delta G$ a i lehké odchylky od reálných hodnot ovlivňují výsledný koeficient, metrika *normalizované přesnosti* považuje za správnou klasifikaci například tuto situaci: Nástroj predikuje $\Delta\Delta G = -0{,}01$ kcal/mol a reálná hodnota zjištěná v experimentu je $\Delta\Delta G = -5{,}8$ kcal/mol - obě tyto hodnoty spadají do třídy *DELETERIOUS*, nástroj tedy predikoval správně.



Obrázek 6.3: Výsledky dílčích metrik *normalizované přesnosti* všech nástrojů a konsensuálních metod na trénovacím datasetu.



Obrázek 6.4: Výsledky metriky *normalizované přesnosti* všech nástrojů a konsensuálních metod na trénovacím datasetu.

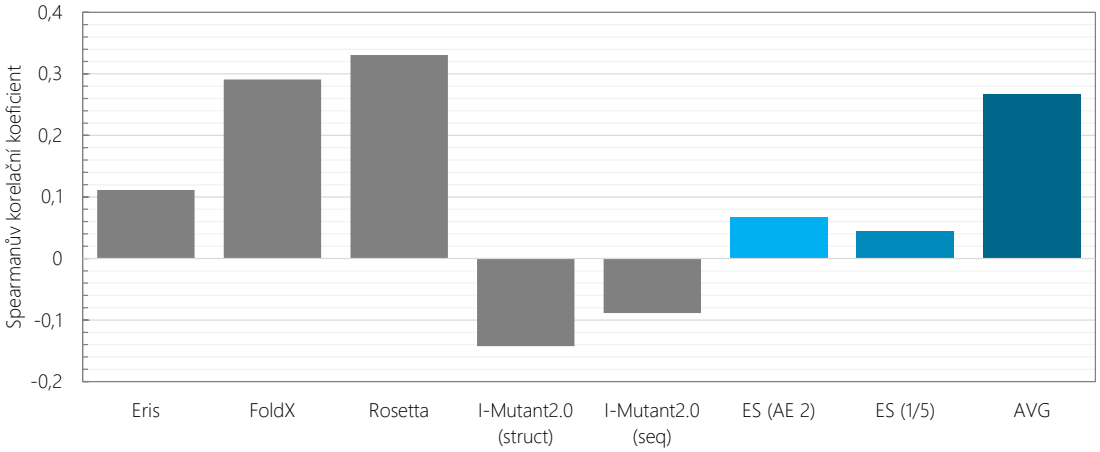
6.3 Výsledky testování

Aby bylo možné ověřit reálnou úspěšnost vytvořeného meta-klasifikátoru, bylo jej potřeba otestovat na nezávislé datové sadě mutací (vzhledem k trénovací datové sadě). Pro tento účel byla vytvořena testovací datová sada mutací z dostupných patentů a meta-klasifikátor otestován, nikoliv pomocí *Pearsonova korelačního koeficientu*, ale pomocí metriky *Spearmanova korelačního koeficientu*.

Důvodem použití jiné metriky byl rozdílný význam jednotlivých hodnot predikce změny stability proteinu v dolovaných patentech. Zatímco v trénovací datové sadě je měřítkem pro klasifikaci mutací přímo hodnota $\Delta\Delta$G, v dolovaných patentech je použit tzv. *performance index* (PI), který lze interpretovat jako hodnotu poměru stability původního (wild-type) a mutovaného proteinu. PI nabývá hodnot od 0,05 (jež jsou přiřazovány škodlivým mutacím) do kladných čísel větších než jedna. Pro hodnoty $0{,}5 \geq PI > 0{,}05$ jsou mutace klasifikovány jako neškodlivé, pro hodnoty $1 \geq PI > 0{,}5$ jako neutrální mutace a pro hodnoty $PI \geq 1$ jako tzv. *up mutations* (lze považovat za stabilizující mutace). Dalším zřejmým důvodem pro obtížné použití obyčejné korelace je nesourodost intervalů, kdy PI nabývá hodnot z intervalu $\langle 0{,}05; \propto)$, zatímco $\Delta\Delta$G nabývá obecně hodnot z intervalu $(-\propto; \propto)$.

Hodnoty $\Delta\Delta$G a PI tedy vzájemně korelovat nelze a z tohoto důvodu byla použita

pořadová korelace, konkrétně *Spearmanův korelační koeficient*. Výsledek této metriky pro všechny nástroje a konsensuální metody je zobrazen v grafu 6.5.



Obrázek 6.5: Výsledky metriky *Spearmanova korelačního koeficientu* všech nástrojů a konsensuálních metod na testovacím datasetu.

Hodnoty *Spearmanova korelačního koeficientu* se pohybují ve stejném intervalu jako hodnoty *Pearsonova korelačního koeficientu*. Základem je uspořádání obou množin, v tomto případě se jedná o množiny hodnot predikcí změn stability proteinu (nazvěme ji množinou $X'_{\Delta\Delta G}$) a množiny reálných hodnot poměru stability mezi wild-type proteinem a jeho mutantem (nazvěme ji množinou Y'_{PI}). Dalším krokem při výpočtu *Spearmanova korelačního koeficientu* je přiřazení hodnot pořadí jednotlivým veličinám obou množin a na těchto nových množinách (nechť $X_{\Delta\Delta G}$ resp. Y_{PI} je množinou hodnot pořadí přiřazených k seřazeným veličinám z množiny $X'_{\Delta\Delta G}$ resp. Y'_{PI}) vypočítat *Pearsonův korelační koeficient*. Vzorec pak vypadá následovně:

$$S_{kk} = \frac{AVG(X_{\Delta\Delta G}Y_{PI}) - AVG(X_{\Delta\Delta G})AVG(Y_{PI})}{\sqrt{AVG(X^2_{\Delta\Delta G}) - AVG^2(X_{\Delta\Delta G})}\sqrt{AVG(Y^2_{PI})AVG^2(Y_{PI})}}, \tag{6.1}$$

kde AVG je aritmetický průměr. V případě, kdy S_{kk} nabývá kladných hodnot z intervalu $\langle -1; 1\rangle$, značí tím stoupající tendenci hodnot množiny Y_{PI} pokud stoupají i hodnoty množiny $X_{\Delta\Delta G}$. Pokud nabývá záporných hodnot z intervalu $\langle -1; 1\rangle$, značí tím klesající tendenci hodnot množiny Y_{PI} se současnou stoupající tendencí množiny $X_{\Delta\Delta G}$. Koeficient $S_{kk} = 0$ znamená, že hodnoty množiny Y_{PI} nemají žádnou tendenci stoupat ani klesat za stoupající tendence hodnot množiny $X_{\Delta\Delta G}$.

Hodnoty v grafu 6.5 potvrzují platnost stanoviska z kapitoly 6.2, že prvotní úspěch nástroje I-Mutant2.0 (obě verze) na trénovací datové sadě byl opravdu způsoben překryvem datových sad mutací pro trénovaní meta-klasifikátoru a pro trénování metod strojového učení tohoto nástroje.

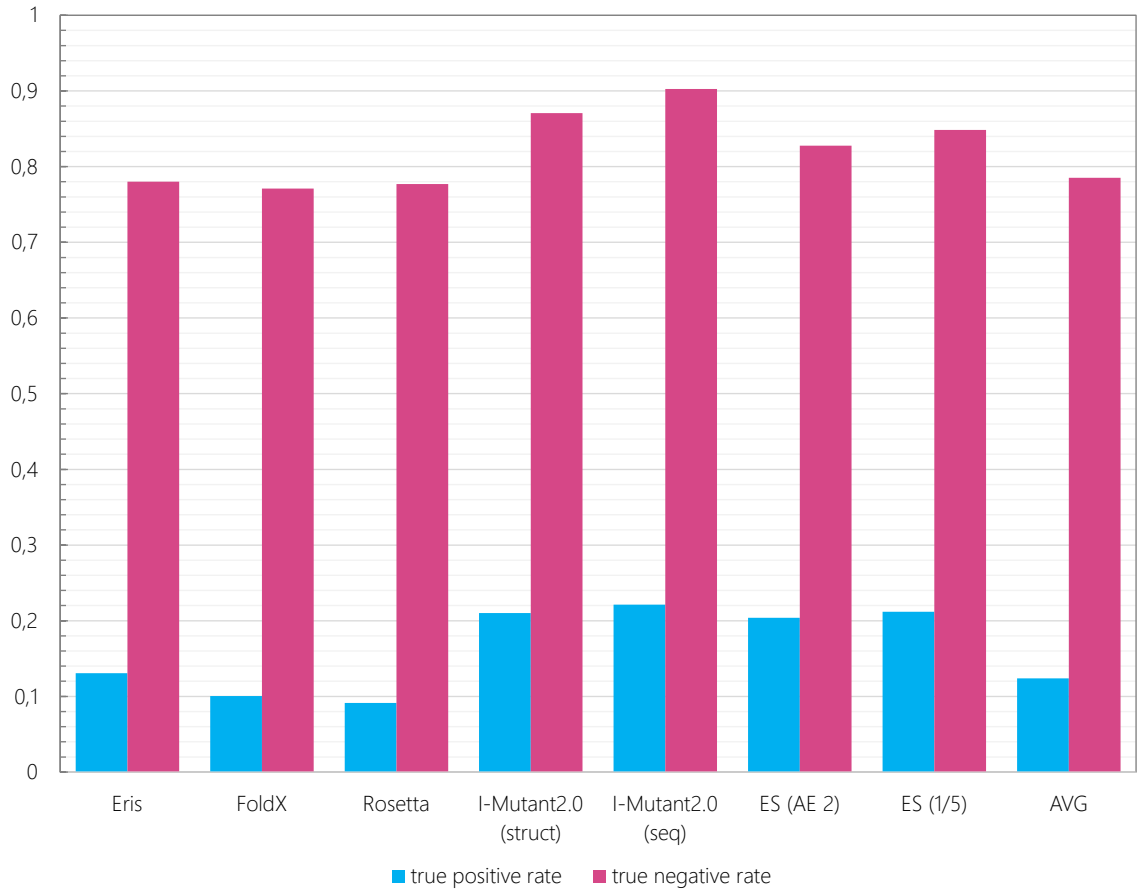
Jak je vidět na této nezávislé datové sadě mutací, nástroj I-Mutant2.0 je zde nejhorším v otázce úspěšnosti predikce změny stability proteinu. Na druhou stranu graf 6.5 potvrzuje, že nástroje s pokročilými technikami (Eris, Rosetta a FoldX) si svoji úspěšnost predikce udržují téměř konstantní, a to jak v porovnání mezi trénovací a testovací datovou sadou, tak v porovnání mezi sebou. Z tohoto faktu lze tvrdit, že navzdory výpočetním náročnostem těchto *state of the art* nástrojů, je žádoucí tyto nástroje používat. Pro tuto sadu zvolených

nástrojů dosahuje evoluční strategie na testovací datové sadě mutací poměrně špatných výsledků, což je způsobeno přiřazením vysokých vah právě nástroji I-Mutant2.0 na trénovací datové sadě, jak je vidět v tabulce 6.2.

Stejně jako u trénovací datové sady, i zde bylo použito metriky *normalizované přesnosti*, pro podrobnější vyhodnocení úspěšnosti nástrojů a konsensuálních metod. Jelikož metrika vyhodnocuje úspěšnost klasifikace mutací do dvou kategorií (*DELETERIOUS* a *BENIGN*), specifikovaných v kapitole 6.2, byly hodnoty PI (používané v patentech) převedeny takto: pro $PI = 0{,}05$ jsou mutace klasifikovány jako *DELETERIOUS* a pro $PI > 0{,}05$ jako *BENIGN*. Nejprve jsou zobrazeny dílčí metriky *normalizované přesnosti* v grafu 6.6 a poté i celkové vyhodnocení této metriky v grafu 6.7.

Zajímavá je zde u všech nástrojů nízká hodnota *true positive rate* udávající poměr správně vyhodnocených škodlivých mutací z množiny všech mutací, které byly nástrojem vyhodnoceny jako škodlivé. Všechny nástroje a tím pádem i konsensuální metody selhávají na testovací datové sadě při klasifikaci škodlivých mutací (pro $\Delta\Delta G < 0$).

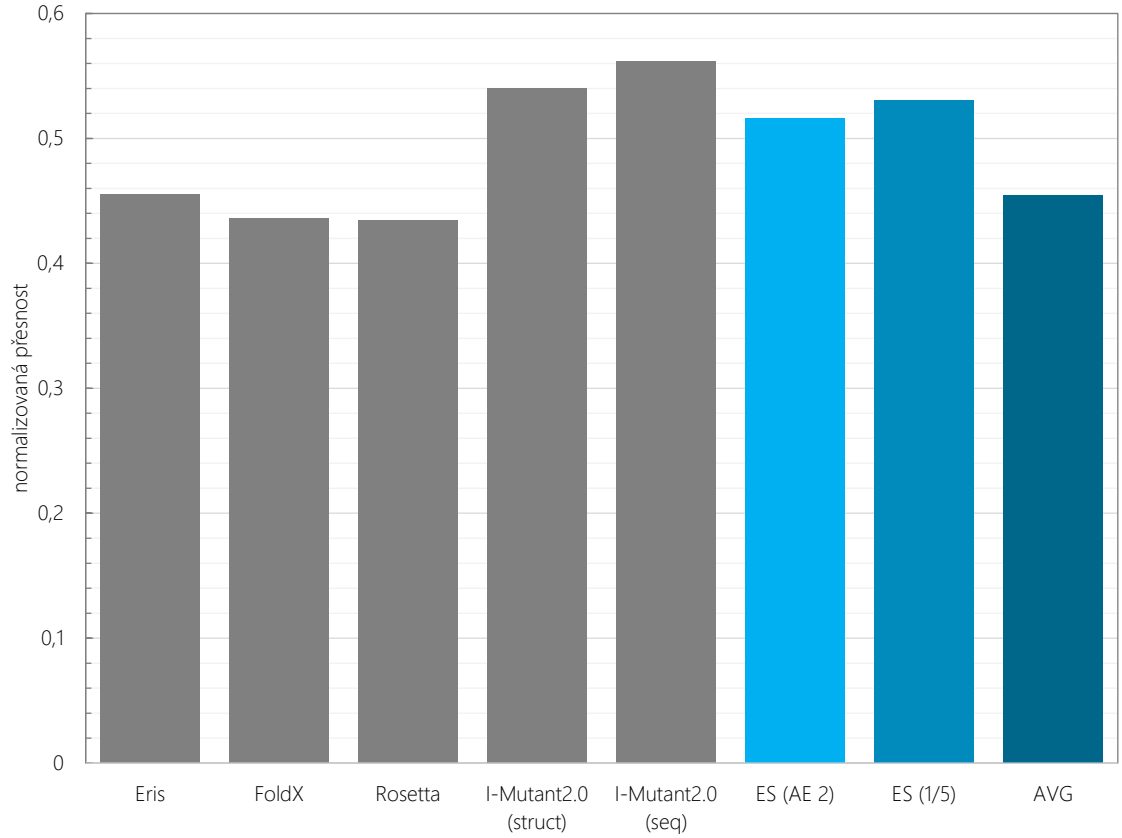
Graf 6.7 opět poukazuje také na úspěšnost nástroje I-Mutant2.0 správně klasifikovat mutace do tříd (škodlivé a neutrální) na základě změny stability proteinu. Důležité je, že i *state of the art* nástroje mají vysokou úspěšnost.



Obrázek 6.6: Výsledky dílčích metrik *normalizované přesnosti* všech nástrojů a konsensuálních metod na testovacím datasetu.

Přínos či úspěšnost evoluční strategie, ať už s použitím pravidla 1/5, nebo s autoevolucí

typu 2, nelze na základě této metriky hodnotit. To především z důvodu, že evoluční strategie byla trénována pro přiřazení vah jednotlivým nástrojům tak, aby byla výsledná hodnota predikce meta-klasifikátoru $meta_{ddg}$ co nejblíže reálné hodnotě. Tato metrika slouží tedy pouze pro jiný úhel pohledu na charakteristiku výsledných hodnot predikce z vytvořeného meta-klasifikátoru a ukazuje jeho úspěšnost správně klasifikovat mutace do třídy škodlivých nebo neutrálních mutací.



Obrázek 6.7: Výsledky metriky *normalizované přesnosti* všech nástrojů a konsensuálních metod na testovacím datasetu.

Kapitola 7

Závěr

Hlavním předmětem této práce bylo vytvoření výpočetního systému simulující algoritmus evoluční strategie (ES) v úloze predikce vlivu aminokyselinových mutací na stabilitu proteinu. Vytvořený meta-klasifikátor využívá výsledků predikcí z celkem čtyř nástrojů, z toho u jednoho nástroje byla použita sekvenční i strukturní verze. Meta-klasifikátor byl trénován na vytvořené trénovací datové sadě z dostupné databáze experimentálně ověřených mutací ProTherm. Dále byl zkoumán přínos dvou typů ES a to ES s pravidlem 1/5 a ES s autoevolucí typu 2. K otestování úspěšnosti meta-klasifikátoru byla pak vytvořena nezávislá trénovací datová sada, jejímž zdrojem byly mutace dolované z dostupných patentů.

Výsledky experimentování a testování ukázaly možný přínos ES v úloze predikce vlivu aminokyselinových mutací, avšak za jistých podmínek. Hlavní podmínkou je pečlivý výběr sady nástrojů pro vytvoření meta-klasifikátoru. Lze doporučit volbu zástupců *state of the art* nástrojů, které svoji úspěšnost predikce změny stability proteinu udržují téměř konstantní a nedojde tak k situaci, kdy na trénovací datové sadě meta-klasifikátoru budou nadhodnoceny některé nástroje a pro jiné datové sady pak meta-klasifikátor bude selhávat, kvůli nízké úspěšnosti nadhodnocených nástrojů. Druhou podmínkou pro možnost využití ES, která také souvisí s nadhodnocováním nástrojů při trénování, je nezávislost trénovací datové sady mutací meta-klasifikátoru s případnými datovými sadami, na kterých byly vybrané nástroje pro tvorbu meta-klasifikátoru trénovány. Za těchto podmínek lze očekávat přínos ES v otázce úspěšnosti predikce. Jedná se o zlepšení úspěšnosti predikce nejlepšího nástroje řádově o jednotky procent (konkrétně 3% v případě této práce).

Ohledně volby typu evoluční strategie nelze z výsledků této práce jednoznačně říci, že ES s autoevolucí typu 2 je výhodnější než základní verze s pravidlem 1/5. ES s autoevolucí typu 2 dosahovala minimálně stejných nebo lepších výsledků než ES s pravidlem 1/5, ale rozdíl nebyl nikterak markantní, spíše v řádu desetinách procenta, maximálně v řádu jednotek procent (konkrétně 0,9%, resp. 2,2%, v případě trénovací, resp. testovací, datové sady).

Vzhledem k dostupným zdrojovým datům aminokyselinových mutací nebylo možné meta-klasifikátor natrénovat na zcela nezávislé trénovací datové sadě. Meta-klasifikátor pak neměl zcela optimální rozložení vah a nedosahoval takové úspěšnosti, která by výrazně přesahovala nejlepší nástroj z vybrané sady. V případě vytvoření nezávislé trénovací datové sady lze předpokládat dosahování lepších výsledků meta-klasifikátoru a lze tak ES pro přínos úspěšnosti predikce vlivu aminokyselinových mutací na stabilitu proteinu doporučit.

Literatura

[1] Single-nucleotide polymorphism In: Wikipedia: the free encyclopedia [online]. 2007, [cit. 2014-04-13].
URL http://en.wikipedia.org/wiki/Single-nucleotide_polymorphism

[2] *FoldX: A force field for energy calculations and protein design [online] - Technický manuál*. 2008.

[3] Protein structure In: Wikipedia: the free encyclopedia [online]. 2008, [cit. 2014-04-12].
URL http://en.wikipedia.org/wiki/Protein_structure

[4] Aehle, W.; Cascao-Pereira, L.; Estell, D.; aj.: Compositions and methods comprising serine protease variants. 5. 8. 2010, US Patent App. 12/616,097.

[5] Alberts, B.: *Základy buněčné biologie: úvod do molekulární biologie buňky*. 1998, ISBN 80-902-9062-0.

[6] Basler, J.; Cascão-Pereira, L.; Estell, D.; aj.: Compositions and Methods Comprising Protease Variants. 27. 10. 2011, US Patent App. 12/963,930.

[7] Bayer, E. A.; Chanzy, H.; Lamed, R.; aj.: Cellulose, cellulases and cellulosomes. *Current Opinion in Structural Biology*, ročník 8, č. 5, 1998: s. 548–557.

[8] Bäck, T.; Hoffmeister, F.; Schwefel, H.-P.: A Survey of Evolution Strategies. In *Proceedings of the Fourth International Conference on Genetic Algorithms*, Morgan Kaufmann, 1991, s. 2–9.

[9] Bott, R.; Ultsch, M.; anthony Kossiakoff; aj.: The three-dimensional structure of Bacillus amyloliquefaciens subtilisin at 1.8 A and an analysis of the structural consequences of peroxide inactivation. *The Journal of biological chemistry*, ročník 263, č. 16, 1988: s. 7895–7906.

[10] Brahms, S.; Brahms, J.: Determination of protein secondary structure in solution by vacuum ultraviolet circular dichroism. *Journal of Molecular Biology*, ročník 138, č. 2, 1980: s. 149–178, doi:10.1016/0022-2836(80)90282-X.

[11] Capriotti, E.; Fariselli, P.; Rossi, I.; aj.: A three-state prediction of single point mutations on protein stability changes. *BMC Bioinformatics*, ročník 9, 2008: str. S6, doi:10.1186/1471-2105-9-S2-S6.

[12] Cascao-Pereira, L.; Kaper, T.; Kelemen, B.; aj.: Compositions and methods comprising cellulase variants with reduced affinity to non-cellulosic materials. 7. 8. 2012, US Patent 8,236,542.

[13] Cuevas, W.; Lee, S.; Ramer, S.; aj.: Geobacillus Stearothermophilus Alpha-Amylase (AmyS) Variants with Improved Properties. 24. 12. 2009, US Patent App. 12/477,028.

[14] Ding, F.; Dokholyan, N. V.: Emergence of Protein Fold Families through Rational Design. *PLoS Computational Biology*, ročník 2, č. 7, 2006: str. e85, doi:10.1371/journal.pcbi.0020085.

[15] Eiben, A.: *Introduction to evolutionary computing*. Natural computing series, 2003, ISBN 3-540-40184-9.

[16] Gill, P.; Moghadam, T. T.; Ranjbar, B.: Differential Scanning Calorimetry Techniques: Applications in Biology and Nanoscience. *Journal of Biomolecular Techniques*, ročník 21, 2010: s. 167–193.

[17] Greenfield, N. J.: Using circular dichroism spectra to estimate protein secondary structure. *Nature Protocols*, ročník 1, 2007: s. 2876–2890, doi:10.1038/nprot.2006.202.

[18] Gromiha, M.: *Protein bioinformatics: from sequence to function*. 2010, 320 s., ISBN 978-813-1222-973.

[19] Hartl, F. U.: Chaperone-assisted protein folding: the path to discovery from a personal perspective. *Nature Medicine*, ročník 17, 2011: s. 1206–1210, doi:10.1038/nm.2467.

[20] Held, P.: Quantitation of Peptides and Amino Acids with a Synergy(TM) HT using UV Fluorescence. 2003.
URL http://www.biotek.com/resources/docs/Synergy_HT_Quantitation_of_Peptides_and_Amino_Acids.pdf

[21] Kabsch, W.; Sander, C.: Dictionary of protein secondary structure: Pattern recognition of hydrogen-bonded and geometrical features. *Biopolymers*, ročník 22, č. 12, 1983: s. 2577–2637, doi:10.1002/bip.360221211.

[22] Kabsch, W.; Sander, C.: DSSP [online]. Centre for Molecular and Biomolecular Informatics, 2011.
URL http://www.cmbi.ru.nl/dssp.html

[23] Kellog, E. H.; Leaver-Fay, A.; Baker, D.: Role of conformational sampling in computing mutation-induced changes in protein structure and stability. *Proteins: Structure, Function, and Bioinformatics*, ročník 79, č. 3, 2011: s. 830–838, doi:10.1002/prot.22921.

[24] Khan, S.; Vihinen, M.: Performance of protein stability predictors. *Human Mutation*, ročník 31, 2010: s. 675–684, doi:10.1002/humu.21242.

[25] Khatun, J.; Khare, S. D.; Dokholyan, N. V.: Can Contact Potentials Reliably Predict Stability of Proteins? *Journal of Molecular Biology*, ročník 336, 2004: s. 1223–1238, doi:10.1016/j.jmb.2004.01.002.

[26] Kodíček, M.: *Biochemické pojmy: výkladový slovník*, ročník 1. 2004.

[27] Lažanský, J.: *Umělá inteligence (3)*. 2001, 117-160 s., ISBN 80-200-0472-6.

[28] Ladokhin, A. S.: Fluorescence Spectroscopy in Peptide and Protein Analysis. *Encyclopedia of Analytical Chemistry*, 2006: s. 5762–5779, doi:10.1002/9780470027318.a1611.

[29] Liu, P.-F.; Avramova, L. V.; Park, C.: Revisiting absorbance at 230nm as a protein unfolding probe. *Analytical Biochemistry*, ročník 398, 2009: s. 165–170, doi:10.1016/j.ab.2009.03.028.

[30] Meyer-Nieberg, S.; Beyer, H.-G.: Self-Adaptation in Evolutionary Algorithms. *Parameter Setting in Evolutionary Algorithm*, 2006: s. 47–76.

[31] Potapov, V.; Cohen, M.; Schreiber, G.: Assessing computational methods for predicting protein stability upon mutation: good on average but not in the details. *Protein Engineering, Design & Selection*, ročník 22, 2009: s. 553–560, doi:10.1093/protein/gzp030.

[32] Prosser, V.: *Experimentální metody biofyziky*. 1989.

[33] Racek, J.; Holeček, V.: Enzymy a Volné Radikály. *Chemické listy*, ročník 93, 1999: s. 774–780.

[34] Rawlings, N. D.; Barrett, A. J.: *Handbook of proteolytic enzymes*, ročník 3. 2013, ISBN 978-0-12-382219-2.

[35] Rohl, C. A.; Strauss, C. E.; Misura, K. M.; aj.: Protein Structure Prediction Using Rosetta. *Methods in Enzymology*, ročník 383, 2004: s. 66–93, doi:10.1016/S0076-6879(04)83004-0.

[36] Schymkowitz, J.; Borg, J.; Stricher, F.; aj.: The FoldX web server: an online force field. *Nucleic Acids Research*, ročník 33, 2005: s. W382–W388, doi:10.1093/nar/gki387.

[37] Shafranovich, Y.: Common Format and MIME Type for Comma-Separated Values (CSV) Files. RFC 4180 (Informational), 2005, updated by RFC 7111.

[38] Snustad, D.; Simmons, M. J.: *Genetika*. Masarykova univerzita, 2009, 871 s., ISBN 978-802-1048-522.

[39] Tenbergen, K.: Dough and Bread Conditioners [online]. 1999 [cit. 2014-05-02]. URL http://www.foodproductdesign.com/articles/1999/11/dough-and-bread-conditioners.aspx

[40] Thiltgen, G.; Goldstein, R. A.; Deane, C. M.: Assessing Predictors of Changes in Protein Stability upon Mutation Using Self-Consistency. *PLoS ONE*, ročník 7, č. 10, 2012: str. e46084.

[41] Yin, S.; Ding, F.; Dokholyan, N. V.: Eris: an automated estimator of protein stability. *Nature Methods*, ročník 4, č. 6, 2007: s. 466–467, doi:10.1038/nmeth0607-466.

[42] Yin, S.; Ding, F.; Dokholyan, N. V.: Modeling Backbone Flexibility Improves Protein Stability Estimation. *Structure*, ročník 15, č. 12, 2007: s. 1567–1576, doi:10.1016/j.str.2007.09.024.

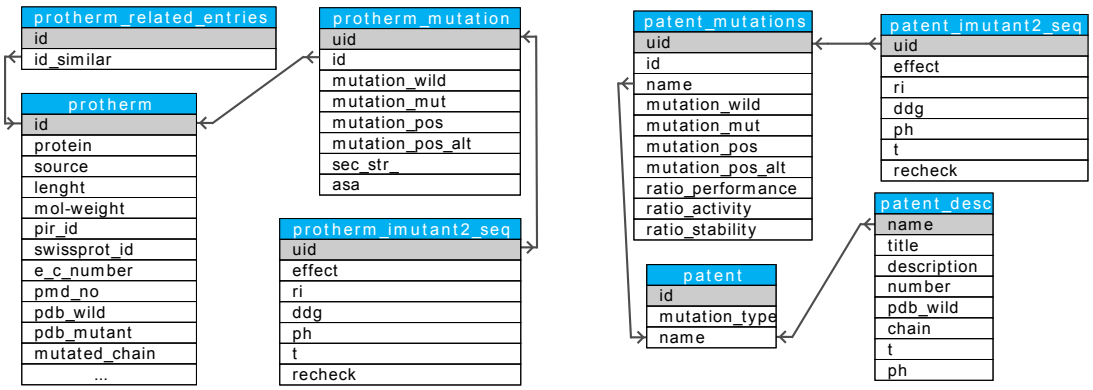
Příloha A

Databázové schéma pro databázi Stability

Tato databáze byla vytvořena pro účely diplomových prací, jejichž cílem je vytvoření meta-klasifikátoru pro predikci vlivu aminokyselinových mutací na stabilitu proteinu. Databáze obsahuje data pro trénování meta-klasifikátoru, zároveň i data pro jeho testování. v neposlední řadě pak výstupy jednotlivých nástrojů po ohodnocení trénovací i testovací datové sady.

Databáze byla rozdělena na dvě hlavní části. v první části jsou záznamy dolované z databáze ProTherm obsahující experimentálně zjištěná data k aminokyselinovým mutacím. Hlavní tabulka *protherm* je pak doplněna tabulkou *protherm_mutation* samotných mutací s úzce souvisejícími informacemi. Tyto záznamy jsou zdrojem mutací pro trénovací datovou sadu.

Druhou částí jsou pak záznamy dolované z patentových vzorů. z různých zdrojů byly extrahovány informace o vlivu aminokyselinových mutací na stabilitu proteinů. Tato část je zdrojem mutací pro testovací datovou sadu.



Obrázek A.1: Schéma hlavních částí databáze *Stability*, z tabulek pro nástroje je uveden jako příklad I-Mutant2.0 (sekvenční verze), tabulky ostatních nástrojů mají prakticky stejné schéma (pro všechny hodnoty výstupu daného nástroje je vytvořen odpovídající sloupec v tabulce).

patent
(obsahuje identifikátor mutací z patentů, jejich typ a název části patentu, ze které byl dolován)

id	int(11)	identifikátor mutace
mutation_type	int(11)	typ mutace (jednobodová = 1, dvoubodová = 2, ...)
name	varchar(40)	název části patentu, ze které byla mutace dolována

patent_desc
(obsahuje podrobný popis jednotlivých částí z dolovaných patentů)

name	varchar(40)	textový unikátní identifikátor části patentu, ze které byly dolovány mutace
title	varchar(104)	název patentu
description	varchar(319)	popis části dolovaného patentu
number	varchar(14)	textový identifikátor patentu v patentové databázi
pdb_wild	varchar(4)	PDB identifikátor proteinu
chain	varchar(1)	označení řetězce v proteinu
t	float	teplota použitá při experimentu
ph	float	hodnota pH

patent_mutations
(obsahuje jednotlivé mutace vydolované z patentů)

uid	int(11)	unikátní identifikátor mutace
id	int(11)	identifikátor mutace (každá jednobodová i vícebodová mutace má vlastní)
name	varchar(52)	odkaz na identifikátor části patentu z tabulky patent
mutation_wild	varchar(1)	zkratka původní aminokyseliny před mutací
mutation_mut	varchar(1)	zkratka nové aminokyseliny po mutaci
mutation_pos	int(3)	pozice mutace v řetězci
mutation_pos_alt	int(3)	přepočtená pozice mutace v řetězci tak, aby seděla na záznamy SEQRES v PDB souboru proteinu
ratio_performance	float	změna efektivity enzymu z hlediska sledované vlastnosti (poměr efektivity wild-type vůči mutované variantě)
ratio_activity	float	změna proteinové aktivity po provedení mutace (poměr aktivity wild-type vůči mutované variantě)
ratio_stability	float	změna proteinové stability po provedení mutace (poměr stability wild-type vůči mutované variantě)

protherm

(obsahuje experimentálně zjištěná termodynamická data k proteinům a k jejich mutacím)

id	int(11)	unikátní identifikátor jednotlivých záznamů
protein	varchar(128)	název proteinu
source	varchar(128)	původ proteinu
lenght	int(11)	celkový počet reziduí v proteinu
mol-weight	float	molekulová hmotnost
pir_id	varchar(32)	PIR identifikátor
swissprot_id	varchar(32)	Swissprot identifikátor
e_c_number	varchar(128)	enzyme commision number
pmd_no	varchar(32)	Protein Mutant Database accession number
pdb_wild	varchar(32)	PDB identifikátor pro proteiny před mutací
pdb_mutant	varchar(32)	PDB identifikátor pro mutované proteiny
mutated_chain	varchar(128)	řetězec obsahující mutaci
no_molecule	int(11)	počet molekul (1 = monomer, 2 = dimer, ...)
sequence_swissprot	text	sekvence aminokyselin z databáze Swissprot
swissprot_id_alias	varchar(128)	Swissprot alias identifikátor
sequence_pdb	text	sekvence aminokyselin z databáze PDB
mutation_type	int(11)	násobnost mutace
t	float	teplota použitá při experimentu
ph	float	hodnota pH
buffer_name	varchar(128)	název použitého bufferu
buffer_conc	varchar(128)	koncentrace bufferu
ion_name_1	varchar(128)	název přidaného iontu
ion_conc_1	varchar(128)	koncentrace přidaného iontu
ion_name_2	varchar(128)	název přidaného iontu
ion_conc_2	varchar(128)	koncentrace přidaného iontu
ion_name_3	varchar(128)	název přidaného iontu
ion_conc_3	varchar(128)	koncentrace přidaného iontu
protein_conc	varchar(128)	koncentrace proteinu při experimentu
measure	varchar(128)	typ měření (fluorescenční spektroskopie, diferenční skenovaní kalorimetr, ...)
method	varchar(128)	metody denaturace (Thermal, Urea, ...)
dg_h2o	varchar(128)	Gibbsova volná energie bez odečtení vlivu denaturantu
ddg_h2o	varchar(128)	změna Gibbsovy volné energie bez odečtení vlivu denaturantu
dg	float	Gibbsova volná energie
ddg	float	změna Gibbsovy volné energie
tmv	float	thermostatic mixing valve
dtm	float	Tm(mutant) - Tm(wild) [°C]
dhvh	float	van't Hoffova entalpická změna
dhcal	float	kalorimetrická změna entalpie
m	float	závislost dg na molární koncentraci denaturantu dg_h2o
cm	float	koncentrace denaturátu
dcp	varchar(128)	změna tepelné kapacity denaturace

state	varchar(128)	počet přechodových stavů
reversibility	varchar(128)	reversibilní denaturace (yes, no, unknown)
activity	varchar(128)	specifická aktivita pro každou mutaci
activity_km	varchar(128)	Machaelis-Mentenova konstanta [mM]
activity_kcat	varchar(128)	Machaelis-Mentenova konstanta [1/s]
activity_kd	varchar(128)	disociační konstanta
key_words	text	klíčová slova
reference	text	odkaz na články v NCBI databázi
author	varchar(128)	jména autorů
remarks	text	komentáře
related_entries	text	seznam odkazů na jiné záznamy vztahující se k aktuálnímu proteinu
db_version	datetime	datum vložení záznamu

protherm_mutation

(obsahuje informace o mutacích pro jednotlivé záznamy z ProTherm databáze)

uid	int(11)	unikátní identifikátor mutace
id	int(11)	identifikátor ProTherm záznamu
mutation_wild	varchar(32)	jednopísmenná zkratka původního rezidua
mutation_mut	varchar(32)	jednopísmenná zkratka nového rezidua
mutation_pos	int(11)	celočíselná pozice mutace
mutation_pos_alt	int(11)	přepočtená pozice mutace v řetězci tak, aby seděla na záznamy SEQRES v PDB souboru proteinu
sec_str_	enum	sekundární struktura mutace (helix, strand, turn, coil)
asa	float	accessible surface area

protherm_related_entries

(obsahuje cizí klíče pro záznamy (experimenty) vztahující se ke konkrétnímu proteinu)

id	int(11)	unikátní identifikátor odkazu
id_similar	int(11)	identifikátor ProTherm záznamu

protherm_imutant2_seq

(obsahuje záznamy provedených ohodnocení mutací z protherm_mutation od nástroje I-Mutant2.0 (v sekvenční verzi), stejné schéma je i pro záznamy provedených ohodnocení mutací z patent_mutations)

uid	int(11)	unikátní identifikátor mutace
id	int(5)	identifikátor ProTherm záznamu
effect	varchar(8)	efekt mutace: stabilizující nebo destabilizující
ri	int(1)	index věrohodnosti výsledku ohodnocení
ddg	float	rozdíl změn Gibbsovy volné energie
ph	float	hodnota pH
t	float	teplota
recheck	tinyint(1)	stav záznamu (0 - proběhlo OK, 1 - znovu ohodnotit)

protherm_imutant2_struct

(obsahuje záznamy provedených ohodnocení mutací z protherm_mutation od nástroje I-Mutant2.0 (ve strukturní verzi), stejné schéma je i pro záznamy provedených ohodnocení mutací z patent_mutations)

uid	int(11)	unikátní identifikátor mutace
id	int(5)	identifikátor ProTherm záznamu
effect	varchar(8)	efekt mutace: stabilizující nebo destabilizující
ri	int(1)	index věrohodnosti výsledku ohodnocení
ddg	float	rozdíl změn Gibbsovy volné energie
ph	float	hodnota pH
t	float	teplota
rsa	float	relative solvent accsessible area
recheck	tinyint(1)	stav záznamu (0 - proběhlo OK, 1 - znovu ohodnotit)

protherm_foldx

(obsahuje záznamy provedených ohodnocení mutací z protherm_mutation od nástroje FoldX, stejné schéma je i pro záznamy provedených ohodnocení mutací z patent_mutations)

uid	int(11)	unikátní identifikátor mutace
id	int(5)	identifikátor ProTherm záznamu
total_energy	float	rozdíl změn Gibbsovy volné energie
backbone_hbond	float	energie vodíkových vazeb na páteři proteinu
sidechain_hbond	float	energie vodíkových vazeb na postranním řetězci proteinu
van_der_waals	float	suma Van der Waalsových sil
electrostatics	float	podíl elektrostatických energií
solvation_polar	float	rozdíl energií solvatace polárních vazeb
solvation_hydrophobic	float	rozdíl energií solvatace hydrofóbních vazeb
van_der_waals_clashes	float	kolize Van der Waalsových sil
entropy_sidechain	float	vliv na entropii vedlejšího řetězce
entropy_mainchain	float	vliv na entropii hlavního řetězce
sloop_entropy	float	vliv na entropii sLoop struktur
mloop_entropy	float	vliv na entropii mLoop struktur
cis_bond	float	energie cysteinových můstků
torsional_clash	float	torzní kolize
backbone_clash	float	kolize na páteři proteinu
helix_dipole	float	dipól alfa-helixu
water_bridge	float	změna energie pocházející z molekul vody
disulfide	float	energie disulfidických můstků
electrostatic_kon	float	efekt elektrostatických interakcí na asociační konstantu
partial_covalent_bonds	float	částečně kovalentní vazby
energy_ionisation	float	energie způsobené ionizací
entropy_complex	float	složená entropie
recheck	tinyint(1)	stav záznamu (0 - proběhlo OK, 1 - znovu ohodnotit)

protherm_rosetta

(obsahuje záznamy provedených ohodnocení mutací z protherm_mutation od nástroje Rosetta, stejné schéma je i pro záznamy provedených ohodnocení mutací z patent_mutations)

uid	int(11)	unikátní identifikátor mutace
id	int(5)	identifikátor ProTherm záznamu
effect	enum	predikce efektu mutace (INCREASING, DECREASING, NEUTRAL)
ddg	float	rozdíl změn Gibbsovy volné energie
fa_atr	float	lennard-jones přitažlivá energie
fa_rep	float	lennard-jones odpudivá energie
fa_sol	float	lazaridis-karplus solvatační energie
fa_intra_rep	float	lennard-jones repulsive mezi atomy stejného residua
pro_close	float	energie uzavření kruhu prolinu
fa_pair	float	statisticky založené
hbond_sr_bb	float	energie vodíkových vazeb mezi páteřními atomy blízko primární sekvence
hbond_lr_bb	float	energie vodíkových vazeb mezi páteřními atomy vzdálené primární sekvenci
hbond_bb_sc	float	energie vodíkových vazeb mezi páteřními atomy a atomy postranního řetězce
hbond_sc	float	energie vodíkových vazeb mezi atomy postranního řetězce
dslf_ss_dst	float	vzdálenostní skóre disulfidických můstků
dslf_cs_ang	float	skóre cysteinových můstků
dslf_ss_dih	float	skóre torzního úhlu
dslf_ca_dih	float	c-alfa skóre torzního úhlu
rama	float	ramachandranovy preference
omega	float	úhly omega
fa_dun	float	vnitřní energie rotamerů postranního řetězce
p_aa_pp	float	pravděpodobnost výskytu aminokyseliny
ref	float	referenční energie pro každou aminokyselinu
recheck	tinyint(1)	stav záznamu (0 - proběhlo OK, 1 - znovu ohodnotit)

protherm_eris

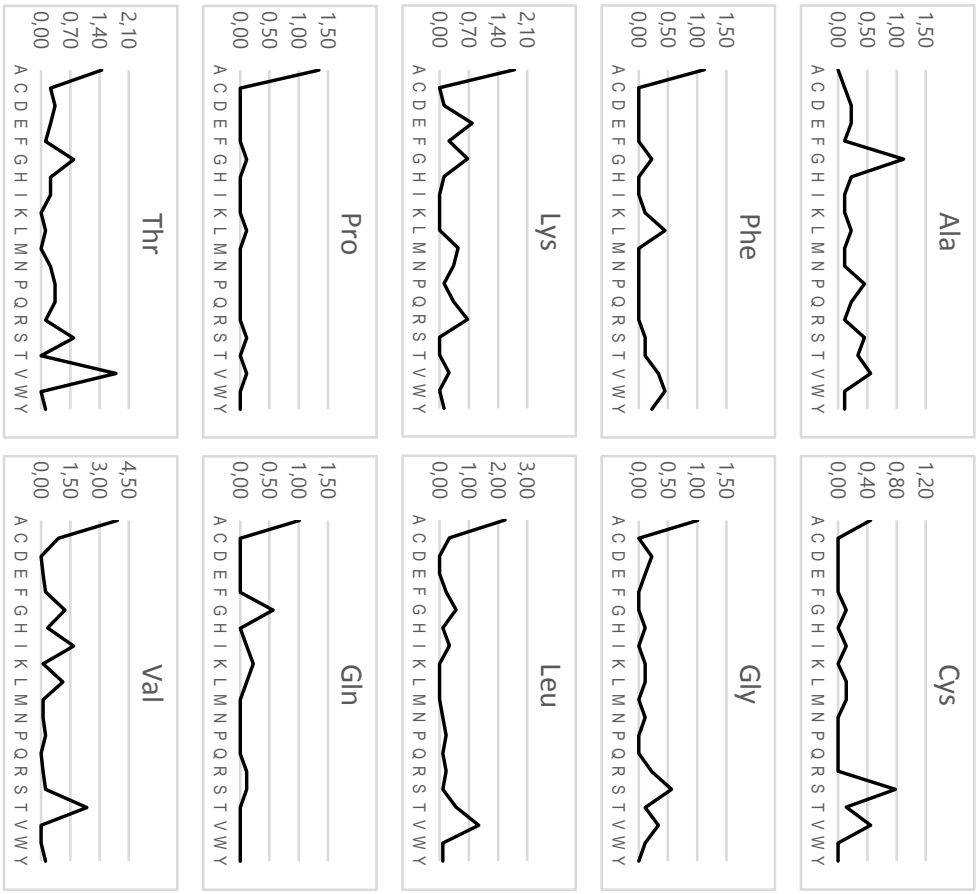
(obsahuje záznamy provedených ohodnocení mutací z protherm_mutation od nástroje Eris, stejné schéma je i pro záznamy provedených ohodnocení mutací z patent_mutations)

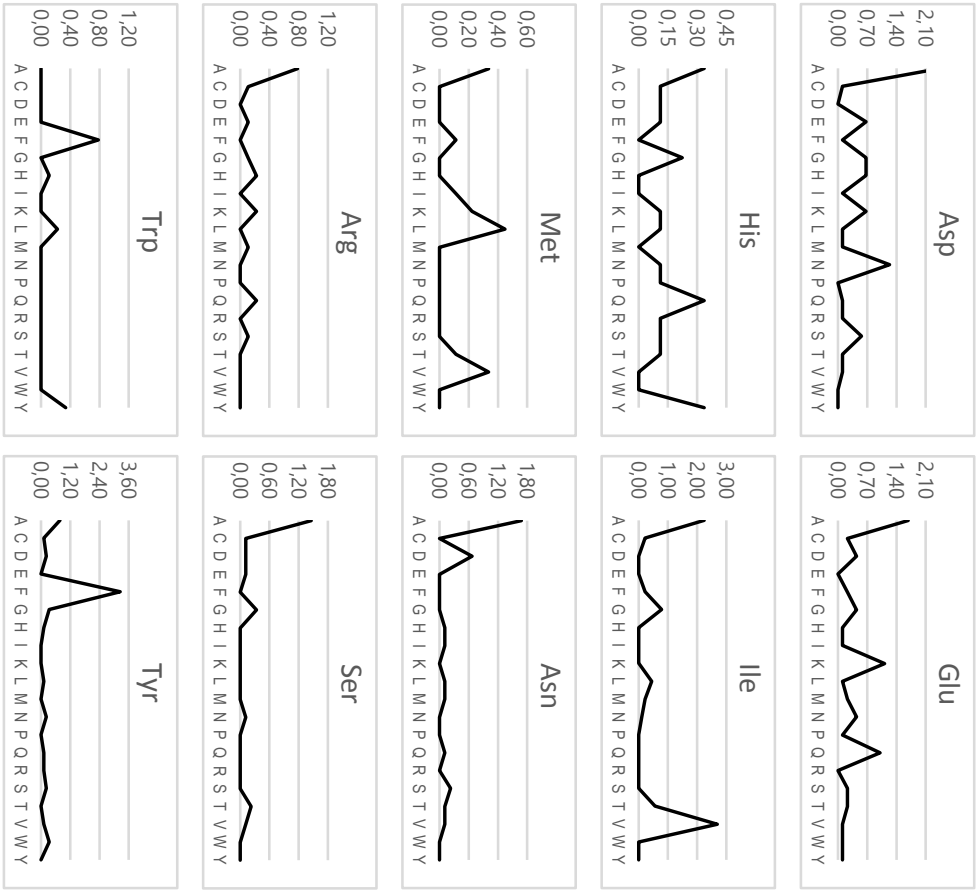
uid	int(11)	unikátní identifikátor mutace
id	int(5)	identifikátor ProTherm záznamu
effect	enum	predikce efektu mutace (INCREASING, DECREASING, NEUTRAL)
ddg	float	rozdíl změn Gibbsovy volné energie
calculated_energy	float	suma vypočtených energií
calculated_energy_stdev	float	standardní odchylka sumy vypočtených energií
recheck	tinyint(1)	stav záznamu (0 - proběhlo OK, 1 - znovu ohodnotit)

Příloha B

Rozložení mutací v trénovací datové sadě

Každý graf odpovídá jedné aminokyselině (v názvu grafu), pro níž ukazuje procentuální zastoupení mutace této aminokyseliny na všechny ostatní v trénovací datové sadě.



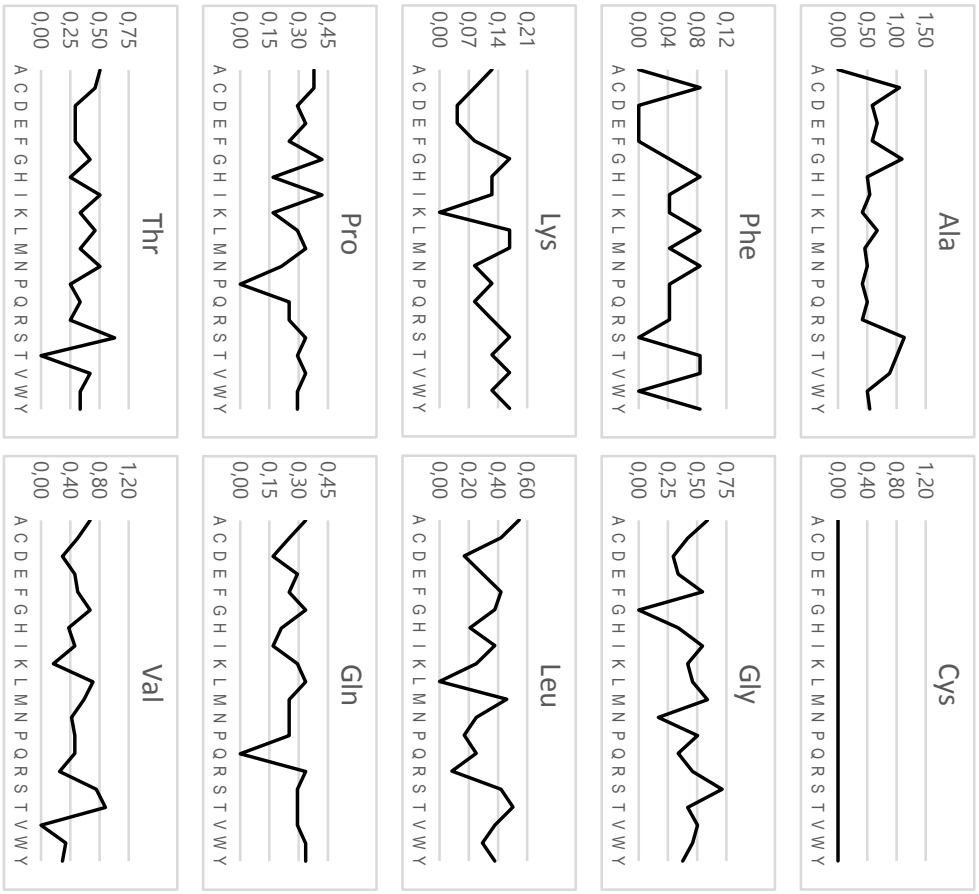


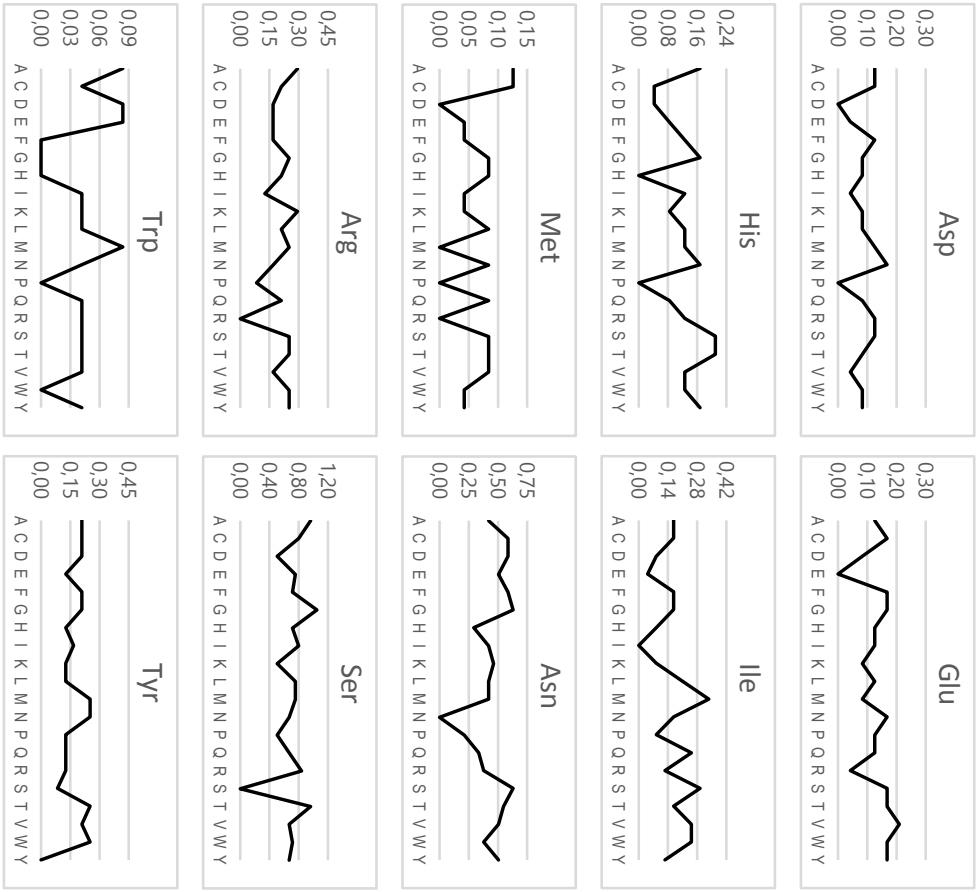
Obrázek B.1: Grafy pro všech 20 základních aminokyselin, které udávají jakým procentem (osa y) je zastoupena mutace dané aminokyseliny (z názvu grafu) na všechny ostatní (osa x) v trénovací datové sadě.

Příloha C

Rozložení mutací v testovací datové sadě

Každý graf odpovídá jedné aminokyselině (v názvu grafu), pro níž ukazuje procentuální zastoupení mutace této aminokyseliny na všechny ostatní v testovací datové sadě.





Obrázek C.1: Grafy pro všech 20 základních aminokyselin, které udávají jakým procentem (osa y) je zastoupena mutace dané aminokyseliny (z názvu grafu) na všechny ostatní (osa x) v testovací datové sadě.

Příloha D

Schéma CSV souborů pro jednotlivé skripty

V případě, že některý ze skriptů pracuje s CSV soubory (ať už na vstupu nebo na výstupu), je zde pro něj uvedena tabulka s tím, jak má struktura (jednotlivé sloupce) CSV souboru vypadat. Pro všechny CSV soubory platí, že jejich hodnoty jsou odděleny středníkem a nejsou uvozeny v uvozovkách.

Název sloupce	**Popis sloupce**
`real_ddg`	reálná hodnota $\Delta\Delta$G
`FoldX_ddg`	hodnota $\Delta\Delta$G nástroje FoldX
`I-M2(seq)_ddg`	hodnota $\Delta\Delta$G nástroje I-Mutant2.0 (sekvenční verze)
`I-M2(struct)_ddg`	hodnota $\Delta\Delta$G nástroje I-Mutant2.0 (strukturní verze)
`Rosetta_ddg`	hodnota $\Delta\Delta$G nástroje Rosetta
`Eris_ddg`	hodnota $\Delta\Delta$G nástroje Eris

Tabulka D.1: Sloupce vstupního CSV souboru skriptu *es.pl* a jejich význam.

Název sloupce	**Popis sloupce**
`uid`	unikátní identifikátor záznamu mutace
`PDB_ID`	unikátní identifikátor proteinu podle Protein Data Bank
`t`	hodnota teploty
`ph`	hodnota pH
`mutation_pos`	pozice mutace
`mutation_wild`	jednopísmenná zkratka původní aminokyseliny
`mutation_mut`	jednopísmenná zkratka aminokyseliny na kterou bylo mutováno
`mutation_chain`	značení řetězce v proteinu, na kterém proběhla mutace

Tabulka D.2: Sloupce vstupního CSV souboru a jejich význam. Shodné pro skripty: *runIm2seq.pl*, *runIm2struct.pl* a *runFoldx.pl*.

Název sloupce	Popis sloupce
`uid`	unikátní identifikátor záznamu mutace
`PDB_ID`	unikátní identifikátor proteinu podle Protein Data Bank
`mutation_pos_alt`	pozice mutace pro záznamy SEQRES v PDB souboru
`mutation_pos`	pozice mutace (původní)
`sequence_pdb`	sekvence jednopísmenných aminokyselin proteinu

Tabulka D.3: Sloupce vstupního CSV souboru skriptu *mpaRecalc.pl* a jejich význam.

Název sloupce	Popis sloupce
`uid`	unikátní identifikátor záznamu mutace
`mutation_pos_alt`	pozice mutace pro záznamy SEQRES v PDB souboru

Tabulka D.4: Sloupce vstupního CSV souboru skriptu *dbMpaC.pl* a jejich význam.

Příloha E

Obsah CD

/dolovani

Tato složka obsahuje skripty použité při dolování především trénovací datové sady mutací a skripty pro dodatečnou úpravu záznamů v databázi *Stability*.

/datasety

Složka obsahuje trénovací datovou sadu i testovací datovou sadu. Obě datové sady jsou uloženy zvlášť ve formě CSV souboru. Jejich struktura je specifikována v přítomném README.

/es

Toto je složka obsahující skript pro trénování meta-klasifikátoru pomocí evoluční strategie (obou typů), jehož výstupem jsou váhy přiřazené jednotlivým nástrojům.

/latex

Složka obsahující veškeré zdrojové soubory pro vytvoření tohoto dokumentu, včetně použitých obrázků a grafů (všechny ve vektorové formě).

/nastroje

V této složce jsou skripty se jmény nástrojů použité pro řízení dávkových výpočtů predikcí stabilit na jednotlivých nástrojích.