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 PROTEINŮ

DIPLOMOVÁ PRÁCE
MASTER'S THESIS

AUTOR PRÁCE Bc. MIROSLAV KADLEC
AUTHOR

BRNO 2015

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 PROTEINŮ

PREDICTION OF PROTEIN STABILITY UPON AMINO ACID MUTATIONS USING EVOLUTION STRATEGY

DIPLOMOVÁ PRÁCE
MASTER'S THESIS

AUTOR PRÁCE — Bc. MIROSLAV KADLEC
AUTHOR

VEDOUCÍ PRÁCE — Ing. JAROSLAV BENDL
SUPERVISOR

BRNO 2015

Abstrakt

Tato práce se zabývá predikcí vlivu aminokyselinových mutací na stabilitu proteinu. Cílem je vytvořit konsenzuální prediktor využívající výstupy vybraných existujících nástrojů za účelem zvýšení úspěšnosti predikce. Optimální konsenzus mezi těmito nástroji byl hledán s pomocí evolučních strategií (ES) ve třech variantách: evoluční strategie s pravidlem 1/5, varianta s autoevolucí typu 2 a metoda CMA-ES. Kvalita nalezených řešení byla následně testována na nezávislé datové sadě. Výsledky všech tří variant dosahovaly podobných přesností predikce, jako nejlepší byl vyhodnocen vektor vah nalezený ES s autoevolucí typu 2. Oproti samostatným prediktorům vykazovala konsenzuální metoda na trénovacích datech zlepšení Pearsonova korelačního koeficientu o 0,057. Na testovací sadě byl její přínos nižší (zlepšení o 0,040). Poměrně malý přínos k přesnosti predikcí na trénovací i testovací datové sadě byl způsoben tím, že pro některé záznamy se nepodařilo získat výsledky všech dílčích nástrojů. V případě vypuštění těchto záznamů přinesla konsenzuální metoda zlepšení Pearsonova korelačního koeficientu o 0,118.

Abstract

This thesis is focused on predicting the impact of amino acid substitution on protein stability. The main goal is to create a consensual predictor that uses the outputs of chosen existing tools in order to improve accuracy of prediction. The optimal consensus of theese tools was designed using evolution strategies in three variants: 1/5 success rule, self-adaptation variant and the CMA-ES method. Then, the quality of calculated weight vectors was tested on the independent dataset. Although the highest prediction performance was attained by self-adaptation method, the differences between all three variants were not significant. Compared to the individual tools, the predictions provided by consensual methods were generally more accurate - the self-adaptation variant imporved the Pearson's corelation coeficient of the predictions by 0,057 on the training dataset. On the testing dataset, the improvement of designed method was smaller (0,040). Relatively low improvement of prediction performance (both on the training and the testing dataset) were caused by the fact, that for some records of testing dataset, some individual tools vere not able to provide their results. When omitting these records, consensual method improved the Pearson's corelations coeficient by 0,118.

Klíčová slova

Aminokyselinové mutace, aminokyselinové substituce, stabilita proteinu, evoluční strategie, proteinové predikce.

Keywords

Amino acid mutations, amino acid substitutions, protein stability, evolution strategy, protein predictions.

Citace

Miroslav Kadlec: Evoluční strategie v úloze
predikce vlivu aminokyselinových mutací
na stabilitu proteinů, diplomová práce, Brno, FIT VUT v Brně, 2015

Evoluční strategie v úloze predikce vlivu aminokyselinových mutací na stabilitu proteinů

Prohlášení

Prohlašuji, že jsem tuto diplomovou práci vypracoval samostatně pod vedením pana Ing. Jaroslava Bendla.

. . . . . .
Miroslav Kadlec
26. května 2015

Poděkování

Děkuji svému vedoucímu panu Ing. Jaroslavu Bendlovi za odbornou pomoc a čas, který mi věnoval při vypracování této práce. Jeho rady a připomínky mi velice pomohly při řešení mnohých problémů.
Děkuji rovněž MetaCentru za poskytnutí přístupu k distribuované výpočetní infrastruktuře nezbytné pro provedení velkého množství výpočtů (projekt LM2010005).

© Miroslav Kadlec, 2015.
Tato práce vznikla jako školní dílo na Vysokém učení technickém v Brně, Fakultě informačních technologií. Práce je chráněna autorským zákonem a její užití bez udělení oprávnění autorem je nezákonné, s výjimkou zákonem definovaných případů.

Obsah

1 Úvod **3**

2 Proteiny **5**
2.1 Dělení proteinů . . . 5
2.2 Aminokyseliny . . . 6
2.3 Proteosyntéza . . . 7
2.4 Struktury proteinů . . . 8
2.4.1 Primární struktura . . . 8
2.4.2 Sekundární struktura . . . 8
2.4.3 Terciární a kvartérní struktura . . . 9

3 Mutace **11**
3.1 Dělení mutací . . . 11
3.2 Vznik mutací . . . 13

4 Stabilita proteinů **15**
4.1 Databáze ProTherm . . . 16
4.2 Predikční nástroje . . . 17
4.2.1 PoPMuSiC2 . . . 17
4.2.2 SDM . . . 18
4.2.3 FoldX 3.0 . . . 18
4.2.4 Eris . . . 18
4.2.5 Auto-mute2 . . . 18
4.2.6 I-Mutant . . . 19
4.2.7 Rosetta . . . 19
4.2.8 iPTREE-STAB . . . 19
4.2.9 mCSM . . . 19
4.2.10 DUET . . . 20

5 Evoluční algoritmy **21**
5.1 Evoluční strategie . . . 22
5.1.1 Mutace . . . 22
5.1.2 Křížení . . . 23
5.1.3 Obnova populace . . . 23
5.1.4 Pravidlo 1/5 . . . 24
5.1.5 Autoevoluce řídících parametrů . . . 25
5.2 CMA-ES . . . 27

6 Návrh a implementace **29**
6.1 Příprava trénovací a testovací datové sady 29
6.1.1 Přidání výsledků dalších nástrojů 29
6.1.2 Záznamy použité pro trénování nástrojů 30
6.1.3 Výběr trénovací a testovací datové sady 31
6.1.4 Statistiky 31
6.2 Implementace evoluční strategie 33

7 Experimenty a zhodnocení výsledků **36**
7.1 Parametry a nastavení 36
7.2 Fáze trénování ES 40
7.3 Fáze testování 43

8 Závěr **48**

A Obsah CD **52**

B Schéma souboru *mutations.txt* **54**

Kapitola 1

Úvod

Proteiny jsou přírodní látky složené z aminokyselin. Představují významnou složku všech živých organismů a v jejich tělech zastávají rozmanité funkce. Existuje velké množství různých proteinů lišících se svými fyzikálními, chemickými a biologickými vlastnostmi. Jednou z hlavních vlastností proteinu je jeho stabilita udávající jeho schopnost zachovat si pevnou trojrozměrnou strukturu při zvyšující se teplotě okolního prostředí. Je určena celkovou silou vazeb mezi jednotlivými atomy proteinu.

Působením určitých vlivů může docházet k substitucím jednotlivých aminokyselin, ze kterých je protein složen. Tyto mutace jsou do jisté míry náhodné a jsou základním jevem umožňujícím biologickou evoluci. Dochází při nich nejen k záměně konkrétní aminokyseliny, ale ke změně vlastností proteinu jako celku. Ovlivněna tedy může být i zmíněná stabilita proteinu. Existují softwarové nástroje pracující na různých principech, které dokáží změnu stability konkrétního proteinu pro zadanou aminokyselinovou substituci predikovat.

Teoretický uvod do problematiky proteinů, včetně popisu jejich vlastností a struktury, je obsažen v druhé kapitole práce. Součástí je i přehled trojrozměrnych struktur, do nichž se proteiny formují a popis vazeb mezi atomy, které tuto strukturu udržují. Ve třetí kapitole jsou popsány aminokyselinové mutace, ke kterým v proteinech může docházet. Jsou uvedeny jejich příčiny, četnosti, s jakými k jednotlivým mutacím dochází, a jejich vliv na vlastnosti proteinů. Čtvrtá kapitola obsahuje stručné vysvětlení pojmu stability proteinu a způsobů její kvantifikace. Dále je popsána databáze ProTherm obsahující experimentálně zjištěné změny stability některých proteinů při konkrétních mutacích. Součástí této kapitoly je i popis metod pro predikci těchto změn a konkrétních nástrojů použitých v této práci pro získání konsenzuálního výsledku. Z nich jsou důkladněji rozepsány programy DUET, Automute2 a PoPMuSiC2. V páté kapitole je vysvětlen obecný princip evolučních algoritmů, operátory mutace, rekombinace a způsoby obnovy populace. Podrobněji je popsána evoluční strategie využívající pravidlo 1/5, varianta s autoevolucí řídících oparametrů a strategie CMA-ES.

Šestá kapitola popisuje praktickou část diplomové práce. Je zde rozebrán princip výběru trénovací a testovací datové sady a uvedeny statistiky popisující tyto množiny záznamů. Druhá část této kapitoly popisuje implementaci evoluční strategie ve třech variantách včetně konfiguračního souboru pro nastavení aplikace a ostatních vstupních a výstupních souborů. Sedmá kapitola je věnována provedeným experimentům a jejich výsledkům. Jsou zde popsány závislosti kvality nalezeného řešení na volbě jednotlivých parametrů a rozepsán postup trénování ES. Následuje ohodnocení nejlepších nalezených řešení na testovací datové sadě pomocí několika metrik včetně grafického i numerického srovnání konsenzuální metody s existujícími nástroji. Výsledky práce jsou pak diskutovány z hlediska přínosu evolučních

strategií k problematice predikce změn stability proteinů v závislosti na provedené mutaci.

V závěru je shrnuta odvedená práce, stručně popsány výsledky testování a zmíněny nejdůležitější závěry z nich vyvozené.

Kapitola 2

Proteiny

Proteiny, neboli bílkoviny, jsou biomakromolekuly se zcela zásadním významem pro život na zemi. Vyskytují se ve všech živých buňkách a podle svých fyzikálních, chemických a biologických vlastností plní rozmanité funkce. V tělech živočichů převládá funkce stavební, proteiny v jejich tkáních tvoří až 80% podíl všech přítomných organických látek. Pletiva rostlin jsou na bílkoviny méně bohatá, protože je tvoří z velké části polysacharidy. Rostliny však dokážou proteiny vytvářet, takzvaně syntetizovat z některých přírodních anorganických látek. Živočichové naproti tomu musejí proteiny přijímat v potravě (v organických látkách), štěpit je na aminokyseliny a z nich opět skládat jiné potřebné bílkoviny.

2.1 Dělení proteinů

Proteiny je možné rozdělit na jednoduché a složené. Zatímco jednoduché proteiny jsou tvořeny pouze aminokyselinami, ve složených proteinech je vázána ještě další neproteinová, obvykle organická látka dodávající komplexu specifické vlastnosti. Podle této neproteinové složky se rozlišuje několik typů bílkovin uvedených s konkrétními příklady v tabulce 2.1.

Typ proteinu	Neproteinová složka	Příklad
jednoduchý protein	není	serumalbumin
glykoprotein	sacharid	imunoglobuliny
nukleoprotein	nukleová kyselina	chromozomy
metaloprotein	iont kovu	hemoglobin
lipoprotein	lipid	chylomikrony (transportní forma tuků)

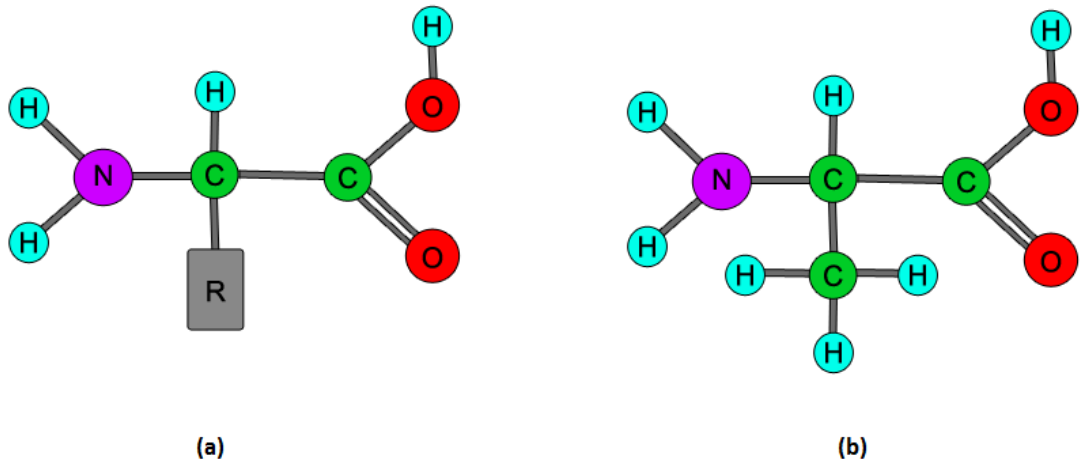
Tabulka 2.1: Příklady složených proteinů. [32]

V tělech organismů plní proteiny různé funkce. Stavební a strukturní proteiny jsou přímo součástí buněčných struktur tvořících tkáně organismu. Příkladem mohou být kolagen a elastin vyskytující se ve vazivu, kůži a cévách, v nichž zajišťují jejich pevnost a pružnost. Enzymy působí jako katalyzátory některých chemických reakcí, například pepsin v žaludku umožňuje štěpení bílkovin přijímaných v potravě. Jako zásobní protein slouží například ovalbumin, který je hojně obsažen ve vaječném bílku. Transportní proteiny na sebe vážou jiné látky a rozvádí je po těle. Příkladem je krevní barvivo hemoglobin které váže kyslík a zajišťuje jeho distribuci krevním řečištěm. Ochrannou funkci plní imunoglobuliny schopné rozeznat a zlikvidovat cizí bakterie a viry v těle. Důležité pro pohybovou soustavu jsou kontraktilní proteiny, např. myozin a aktin podílející se na stazích svalových vláken. Hormony

jsou proteiny vyplavované některými žlázami do krve a spouštějící různé biologické procesy. Někteří živočichové také tvoří toxiny, neboli jedy, které používají k obraně a k lovu [32].

2.2 Aminokyseliny

Stavebními jednotkami proteinů jsou aminokyseliny. Jde o látky obsahující karboxylovou skupinu *(−COOH)*, funkční aminovou skupinu *(-NH$_2$)* a postranní řetězec. Jednotlivé aminokyseliny se od sebe odlišují právě strukturou, velikostí, elektrickým nábojem, rozpustností a dalšími vlastnostmi tohoto postranního řetězce. Aminokyselin existuje velké množství, nicméně v bílkovinách se jich vyskytuje 20. Tyto tzv. proteinogenní aminokyseliny mají karboxylovou skupinu i aminoskupinu navázanou na jednom atomu uhlíku označovaném jako *α-uhlík*, proto se také nazývají *α-aminokyseliny*. Na obrázku 2.1 je znázorněn obecný vzorec a konkrétní příklad aminokyseliny.



Obrázek 2.1: Obecný vzorec aminokyseliny (a) a konkrétní příklad alaninu (b). U konkrétní aminokyseliny je řetězec R nahrazen specifickým postranním řetězcem.

Pro zjednodušení zápisu byly zavedeny jednopísmenné a třípísmenné zkratky pro všechny *α-aminokyseliny*. Podle vlastností postranního řetězce je možné rozdělit aminokyseliny do následujících skupin [22]:

- **Nepolární:** Glycin (Gly, G), Alanin (Ala, A), Prolin (Pro, P), Valin (Val, V), Leucin (Leu, L), Isoleucin (Ile, I) a Methionin (Met, M)
- **Aromatické:** Fenylalanin (Phe, F), Tyrosin (Tyr, Y), Tryptofan (Trp, W)
- **Polární:**
  - **Kladně nabité:** Lysin (Lys, K), Histidin (His, H), Arginin (Arg, R)
  - **Nenabité:** Serin (Ser, S), Threonin (The, T), Cystein (Cys, C), Asparagin (Asn, N), Glutamin (Gin, Q)
  - **Záporně nabité:** Aspartát (Asp, D), Glutamát (Glu, E)

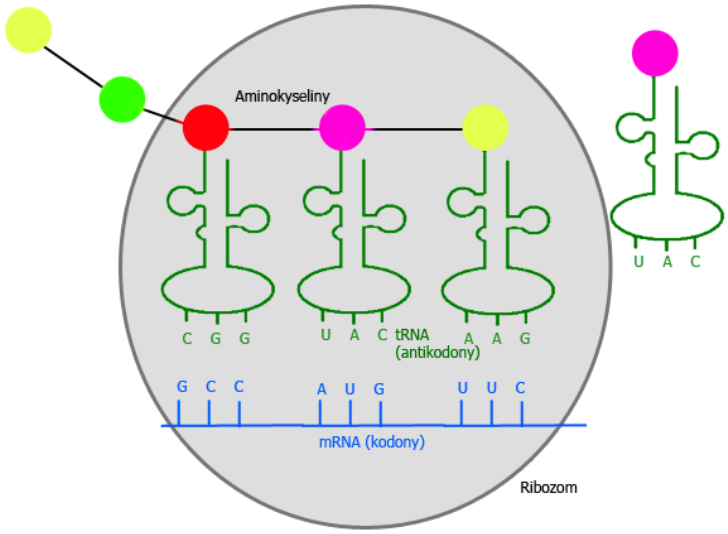
2.3 Proteosyntéza

Proteiny jsou polypetidy, tedy řetězce složené z aminokyselin spojených tzv. peptidovými vazbami. Peptidová vazba vzniká reakcí mezi karboxylovou skupinou jedné aminokyseliny a aminoskupinou druhé. Spojení dvou aminokyselin se nazývá *dipeptid*, krátký peptidový řetězec je *oligopeptid* a dlouhé aminokyselinové sekvence se označují jako *polypeptidy*. Polypeptidy v tělech živočichů jsou výsledkem procesu zvaného proteosyntéza [22]. Při proteosyntéze se výsledný protein spojuje z jednotlivých volných aminokyselin získaných zpravidla štěpením bílkovin přijímaných v potravě. Předpis určující, ze kterých aminokyselin a v jakém pořadí má být výsledný protein složen, je kódován v DNA (deoxyribonukleové kyselině).

První částí proteosyntézy je tzv. transkripce probíhající v buněčném jádře. Při něm se podle předlohy vytvoří řetězec mRNA (mediátorové ribonukleové kyseliny). Vlákno DNA, které má za běžných podmínek tvar dvojšroubovice, se rozplétá a rozdělí se na dva jednoduché řetězce nukleotidů. Každý nukleotid obsahuje jednu ze čtyř dusíkatých bází. Na rozpletené DNA vlákno se váže enzym RNA-polymeráza v místě, kde začíná úsek kódující nějaký protein. RNA-polymeráza se pohybuje po vlákně a k jednotlivým nukleotidům navazuje komplementární nukleotidy mRNA. Dusíkaté báze DNA a jejich komplementy z RNA jsou uvedeny v tabulce 2.2. Mezi připojovanými nukleotidy mRNA působí vzájemné vazby, díky nimž vzniká podobné vlákno, jako předlohový DNA řetězec. Po přepsání celého úseku se mRNA vlákno oddělí a v ribozomu poslouží jako vzor při jevu zvaném translace.

báze DNA	komplementární báze RNA
adenin	uracil
thymin	adenin
cytosin	guanin
guanin	cytosin

Tabulka 2.2 : Rozdělení proteinů podle složení. [32]



Obrázek 2.2: Průběh translace. Antikodony v tRNA (znázorněny zeleně) se párují s kodony mRNA (znázorněny modře). tRNA nese aminokyselinu, která je ke stávajícímu řetězci připojena peptidovou vazbou.

Během translace se podle předpisu kódovaného v mRNA vytváří odpovídající řetězec aminokyselin. Proces probíhá v ribozomu za přispění molekul tRNA (transferové ribonukleové kyseliny), na nichž jsou navázány aminokyseliny. Průběh translace je znázorněn na obrázku 2.2. Trojice za sebou jdoucích bází mRNA se nazývají triplety neboli kodony a každý z nich jednoznačně určuje nějakou aminokyselinu, nebo má speciální význam iniciačního či terminačního kodonu. Podobně se v určité části tRNA nachází trojice bází zvaných antikodony. Detekce iniciačního kodonu na mRNA spouští proces translace. Po přiblížení antikodonu ke komplementárnímu kodonu dochází k vazbám mezi jednotlivými bázemi a tRNA je dočasně připojena na vlákno mRNA. Tím se aminokyselina připojená k této molekule tRNA dostane do blízkosti aminokyseliny připojené před ní a mezi jejich karboxylovou a aminovou skupinou vznikne peptidová vazba. Zpracováním celého úseku mRNA až po terminační kodon vznikne výsledný protein.

2.4 Struktury proteinů

Kromě peptidových vazeb spojujících jednotlivé aminokyseliny do řetězce působí mezi atomy proteinu i další síly. Jsou to například kovalentní vazby a Van der Waalsovy síly. Díky nim se proteiny formují do tzv. foldu, tedy unikátního trojrozměrného uspořádání. Rozlišujeme 4 úrovně struktury proteinů.

2.4.1 Primární struktura

Primární struktura je určena pořadím aminokyselin v peptidovém řetězci. Obvykle se zapisuje jako posloupnost znaků odpovídajících přítomným aminokyselinám postupně od *N-konce* k *C-konci* řetězce. Primární struktura je sama o sobě lineární, ale přímo ovlivňuje vyšší úrovně trojrozměrného uspořádání a tím i vlastnosti a funkci proteinu.

2.4.2 Sekundární struktura

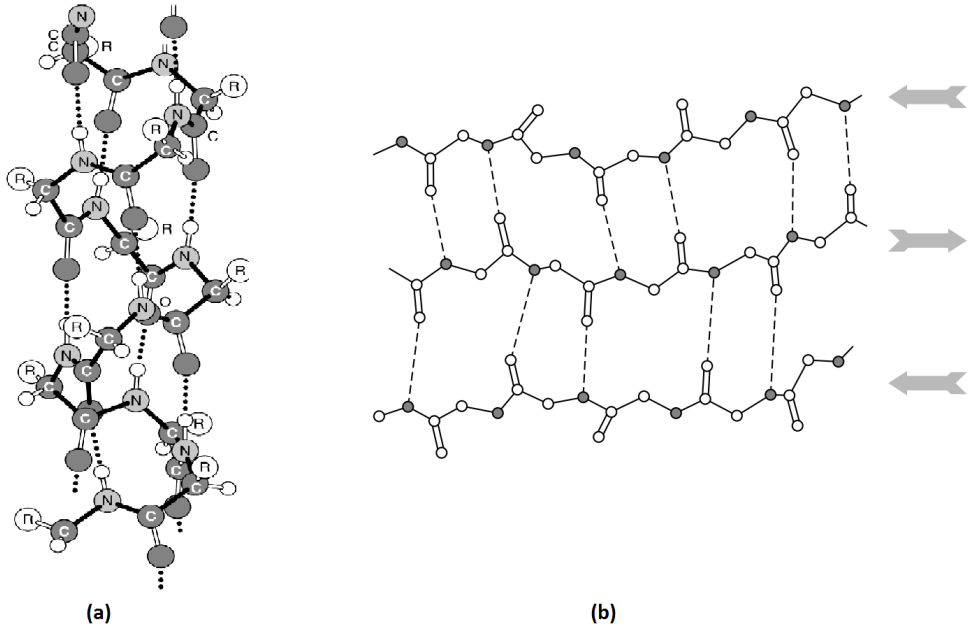
Sekundární struktura je dána výskytem často se opakujících trojrozměrných vzorů ve struktuře proteinu. Prostorové uspořádání proteinu je formováno volným otáčením okolo dvou vazeb a je definováno hodnotami odpovídajících torzních úhlů. Jde o vazbu mezi *α-uhlíkem* a uhlíkem karboxylové skupiny s torzním úhlem ψ a vazbu mezi *α-uhlíkem* a atomem dusíku, kde je torzní úhel označen ϕ. Třetí torzní úhel ω odpovídající rotaci peptidové vazby nabývá pouze hodnoty 180°, ve výjimečných případech 0°. Peptidová vazba je tedy tzv. *planární*. V případě, že v peptidovém řetězci za sebou následuje několik reziduí s podobnými hodnotami torzních úhlů, dochází ke vzniku elementů sekundární struktury. Nejčastějšími dvěma případy jsou *α-helix* a *β-sheet*. Jejich podoba je znázorněna na obrázku 2.3.

Element *α-helix* má tvar pravotočivé šroubovice a formuje se v úsecích proteinu, kde mají torzní úhly hodnoty přibližně $\phi = -57°$ a $\psi = -47°$. Jedna otočka šroubovice má délku průměrně 0,54 nm, což odpovídá počtu 3,6 minokyselin [20]. Délka *α-helixu* se obvykle pohybuje okolo 12 reziduí, existují však případy šroubovic sestávajících z více než 50 aminokyselin [23]. Stabilita *α-helixu* je zajištěna vodíkovým můstkem mezi atomem kyslíku karboxylové skupiny a atomem vodíku navázaném na atomu uhlíku aminokyseliny o 4 místa dále.

Druhým elementem sekundární struktury je *β-sheet*. Tvoří ho dvě nebo více rovnoběžných vláken spojených vodíkovými můstky opět mezi *C=O* skupinou a *N-H* skupinou. K tomu dochází, když torzní úhly nabudou hodnot $\phi = -130°$ a $\psi = -125°$ [20]. Tvar

beta-sheetu se podobá složenému listu. Mají-li vlákna tvořící *β-sheet* stejný směr, hovoříme o tzv. paralelním *β-sheetu*, pokud je směr vláken opačný, jde o antiparalelní *β-sheet*. Vzhledem k tomu, že v případě antiparalelního *β-sheetu* jsou *C=O* a *N-H* skupiny orientovány přímo naproti sobě, jsou vodíkové můstky kratší a celý element je stabilnější [20].

Kromě *α-helixu* a *β-sheetu* existují ještě další elementy sekundární struktury, které se však vyskytují méně často. Úseky peptidového řetězce, na kterých se nenachází žádný element sekundární struktury se označují jako *coil*.



Obrázek 2.3: Schematický nákres α-helixu (a) a β-sheetu (b) [20].

2.4.3 Terciární a kvartérní struktura

Terciární struktura označuje celkové trojrozměrné uspořádání peptidového řetězce včetně elementů sekundární struktury do tzv. proteinových domén. Proteinová doména je část proteinové struktury schopná vykonávat určité chemické nebo fyzikální úkony, jako například vázat jiné ligandy. Malé a jednoduché proteiny se mohou skládat pouze z jedné domény, jiné obsahují domén více. Podle terciální struktury rozlišujeme tři základní proteinové třídy [8]:

- globulární proteiny - molekuly mají kulovitý tvar, většina globulárních proteinů je rozpustná a funguje jako enzymy,
- fibrilární proteiny - molekuly mají protáhlý tvar, většinou plní stavební a strukturální funkci,
- membránové proteiny - obvykle slouží jako receptory nebo tvoří kanály, skrz které mohou polární molekuly procházet buněčnými membránami.

Některé proteiny jsou tvořeny více polypeptidovými řetězci, neboli protomery. Kvartérní struktura proteinu definuje celkovou kompozici a u proteinů s více řetězci udává prostorové vazby mezi jednotlivými podjednotkami. *Monomerické* proteiny se skládají pouze z jednoho

polypeptidového řetězce. *Dimery* obsahují řetězce dva. Jde-li o dvě kopie téhož polypeptidového řetězce, označuje se daný protein jako *homodimer*, jsou-li naopak řetězce odlišné, hovoří se o *heterodimeru* [20].

Každý protein má své unikátní prostorové uspořádání, do kterého se za normálních podmínek formuje. Působením fyzikálních nebo chemických změn však může dojít k rozložení proteinového foldu a ztrátě vlastností proteinu, tzv. denaturaci. Schopnost bílkoviny zachovat si své trojrozměrné uspořádání navzdory působení těchto podmínek se nazývá stabilita proteinu. Této problematice je věnována kapitola 4.

Kapitola 3

Mutace

V oblasti genetiky se mutacemi označují takové změny nukleotidové sekvence genomu organismu, které respektují syntaktická pravidla zápisu genetické informace. Tyto změny jsou nezbytné pro biologickou evoluci a diverzifikaci života na Zemi. Genetický materiál po proběhlé mutaci tedy dále slouží jako funkční předloha k expresi genů a tím může ovlivnit například proteiny vznikající při proteosyntéze. Mutace dělíme na škodlivé, neutrální nebo prospěšné podle toho, jak ovlivňují život jedince. Podle způsobu vzniku pak rozlišujeme 2 druhy mutací. Ke *spontánním mutacím*, které jsou méně časté, dochází například v důsledku chyby při replikaci a reparaci DNA. Běžnější *indukované mutace* vznikají působením tzv. mutagenních faktorů.

3.1 Dělení mutací

Existuje více kritérií, podle kterých je možné mutace třídit. Jedním z nich je úroveň, na které působí:

Genomové mutace působí na nejvyšší úrovni a projevují se změnou počtu chromozomů, případně chromozomových sad. Chromozomy jsou komplexní molekuly, které se nacházejí v jádrech buněk a obsahují genetickou informaci. Jsou složeny z řetězců DNA a z histonů, proteinů udržujících jejich specifické trojrozměrné uspořádání. Organismy mají určitý počet sad chromozomů, které dohromady tvoří genom. Jev, při kterém dojde k odchylce v počtu jednotlivých genů se nazývá aneuploidie a má zásadní vliv na život jedince [21]. Příkladem aneuploidie u člověka je Turnerův nebo Downův syndrom. Znásobení počtu celých chromozomových sad se nazývá polyploidie. Taková mutace není u většiny živočichů slučitelná se životem.

Chromozomové mutace mění tvar a strukturu jednotlivých chromozomů. Jsou následkem tzv. chromozomových zlomů zapřičiněných působením mutagenních faktorů. Struktura chromozomu je porušena, přičemž může dojít k uvolnění fragmentu DNA a následné přestavbě. Chromozomové mutace lze dále rozdělit na balancované (zůstane-li zachováno původní množství genetické informace) a na nebalancované (pokud se množství genetického materiálu změní).

Bodové mutace (též nazývané genové) působí na úrovni konkrétních bází sekvence DNA. Pokud ke změnám jednotlivých nukleotidů dojde v místě genu, mohou mít tyto mutace vliv na výsledky proteosyntézy. Existují 3 způsoby, jak může bodová mutace změnit vlákno DNA. Jsou to:

- inzerce,
- delece,
- substituce.

Inzercí se označují mutace, kdy je do nukleotidové sekvence vložena jedna nebo více nových bází. Míra vlivu na výsledný syntetizovaný protein je určena právě tím, kolik nukleotidů je při mutaci vloženo. Změny v počtu překládaných nukleotidů při translaci totiž způsobí posun čtecího rámce. Jednotlivé aminokyseliny jsou kódovány konkrétními kodony tvořenými trojicemi nukleotidů. Je-li počet míst, o kolik se čtecí rámec posune, dělitelný třemi, výsledek překladu mutované sekvence bude odpovídat původnímu proteinu s několika aminokyselinami navíc. Běžnější jsou však mononukleotidové a dinukleotidové mutace, při nichž je vložen jeden nebo dva nukleotidy. Posun čtecího rámce pak nerespektuje původní kodony a posunutá sekvence nukleotidů je překládána jako posloupnost jiných kodonů do zcela odlišné sekvence aminokyselin. Zároveň může vlivem posunu čtecího rámce dojít k výskytu terminačního kodonu v místě, kde dříve nebyl. Po jeho přečtení je proces translace předčasně ukončen a výsledný protein je zkrácen. Takto drastická změna obvykle zapříčiní nefunkčnost proteinu [10]. Je také zřejmé, že čím blíže k iniciačnímu kodonu k mutaci dojde, tím větší je zasažená část proteinu a také pravděpodobnost výskytu terminačního kodonu.

V případě delece dochází ke ztrátě několika nukleotidů. Stejně jako u inzerce závisí míra vlivu na počtu odstraněných nukleotidů. Nejčastější mononukleotidové a dinukleotidové delece způsobí posun čtecího rámce nerespektující hranice kodonů a výsledkem je zcela jiná aminokyselinová sekvence s rizikem předčasného ukončení translace [10].

Při substituci dochází k nahrazení jednoho nebo více nukleotidů stejným počtem jiných. Může dojít ke změně aminokyselin kódovaných zasaženými kodony. Ostatní aminokyseliny jsou přeloženy stejně, protože k posunu čtecího rámce nedochází. Stále však existuje riziko změny zasaženého kodonu na terminační kodon a v důsledku toho předčasné ukončení procesu translace [10].

V DNA rozlišujeme nukleotidy se dvěma typy bází: purinovými (adenin a thymin) a pyrimidinovými (cytosin a guanin) bázemi. Je-li při substituci nahrazen původní nukleotid nukleotidem s bází jiného typu, označuje se mutace jako *transverze*. V případě nahrazení nukleotidem se stejným typem báze jde o *tranzici*. Rozdíly mezi purinovými a pyrimidinovými bázemi mohou mít vliv i na četnost mutací. V případě chloroplastové DNA bylo například zjištěno, že nukleotidy ohraničené z obou stran nukleotidy s purinovou bází jsou 2,2krát náchylnější k transverzi, než k tranzici. Naopak u nukleotidů, jejichž oba sousedé mají pyrimidinovou bázi, je pravděpodobnost tranzice 1,5krát vyšší, než pravděpodobnost transverze [10].

Jak již bylo zmíněno výše, konkrétní aminokyseliny jsou kódovány jednotlivými kodony. Protože je však počet různých aminokyselin několikanásobně menší než počet různých kodonů, jedna aminokyselina je obvykle kódována několika různými kodony. Díky tomu se nemusí bodová substituce některého nukleotidu projevit, je-li původní kodon nahrazen kodonem determinujícím stejnou aminokyselinu. Taková mutace se pak označuje jako **synonymní mutace (synonymous)**. Z hlediska přírodního výběru jsou tyto mutace neviditelné, tedy neutrální. Pro konkrétní organismus však může mít význam, zda je nový kodon překládán vzácně se vyskytující, nebo naopak běžnou tRNA [10]. **Mutace se změnou smyslu (missense)** mění původní triplet na jiný, který kóduje odlišnou aminokyselinu. Taková změna pak do určité míry ovlivňuje vlastnosti výsledného proteinu. Je-li aminokyselina nahrazena aminokyselinou s podobnými fyzikálními a chemickými vlastnostmi,

hovoříme o *konzervativní záměně*. Konzervativní záměny nemusí příliš měnit trojrozměné uspořádání proteinu a tedy i jeho biologická funkce může být víceméně zachována. Genetický kód je uspořádán takovým způsobem, že většina záměn aminokyselin, ke kterým může dojít v důsledku bodové substituce, jsou právě konzervativní [10]. Dojde-li v důsledku substituce ke změně nějakého kodonu na kodon terminační, označuje se taková mutace za **nesmyslnou (nonsense)**. V místě takové mutace pak dojde k předčasnému ukončení procesu translace a výsledný protein je odpovídajím způsobem zkrácen [10].

3.2 Vznik mutací

Spontánní mutace, tedy mutace bez zjevné vnější příčiny, jsou poměrně vzácné. Někteří biologové dokonce předpokládali, že k veškerým mutacím dochází v důsledku působení mutagenních faktorů. Při replikaci DNA, což je jev, kdy se genetická informace kopíruje a zdvojuje, dochází k tzv. tautomerním přechodům. Tyto jevy jsou pouze dočasné a krátkodobé a dochází při nich k přechodům nukleotidových bází z běžných strukturních forem na formy méně obvyklé, v nichž jsou schopné vytvářet vodíkové můstky s nesprávnými nukleotidy. V důsledku toho může být do vytvářené sekvence DNA začleněn chybný nukleotid. Příkladem je báze thymin v enol-formě, která se páruje s cytosinem namísto adeninu, nebo imino-forma cytosinu vytvářející vazby s adeninem místo guaninu. Většina takto vzniklých chyb bývá opravena buď přímo DNA-polymerázou, nebo v rámci následného působení reparačních enzymů; některé mutace však mohou zůstat neopraveny. Pravděpodobnost těchto mutací je tedy malá, ale vzhledem k tomu, že tautomerní přechody jsou statisticky náhodné, je možné tyto mutace označit za spontánní [10]. Častěji dochází k mutacím v důsledku působení některých fyzikálních jevů nebo chemických látek, které souhrně nazýváme mutagenní faktory.

Jedním z nejčastějších fyzikálních mutagenů jsou různé druhy záření (ionizující i neionizující). Míra mutace struktury DNA je pak úměrná absorbovanému množství záření. Mezi ionizující záření patří Röntgenové a gama paprsky a proudy neutronů, protonů a elektronů o vysoké energii. Jejich působením může docházet ke tvorbě nežádoucích kovalentních vazeb mezi protilehlými nukleotidy, nebo ke vzniku zlomů na sekvencí DNA. Nejčastějším neionizujícím zářením s mutagenním účinkem je ultrafialové (UV) záření. Pokud je molekula DNA vystavena UV záření, může v ní docházet k tzv. *dimeraci* dvou sousedních thyminů, tedy k vytvoření chemicky stabilního komplexu. Pokud nejsou dimery ostraněny, způsobují problémy při replikaci i transkripci. UV záření navíc pravděpodobně podporuje tautomerní přechody zmíněné v odstavci o spontánních mutacích [10].

Chemické mutagenní faktory jsou látky, které mění strukturu DNA svojí chemickou reaktivitou. V dnešní době je známé velké množství chemických mutagenů; mezi nejúčinnější z nich patří:

- Alkylační činidla (např. dymethylsulfát či bojový plyn yperit) - tyto látky obsahují alkylové skupiny, které mohou předávat molekulám DNA a tím je poškozovat
- Oxydační činidla
- Desaminující činidla (např. dusitany)
- Interkalační činidla (např. akridiny) - molekuly těchto látek se mohou vsouvat mezi páry bází v sekvenci DNA [10]

Některé z těchto látek se používají při úpravě potravin, výrobě léků a v zemědělství, takže s nimi lidský organismus přichází běžně do styku.

Kapitola 4

Stabilita proteinů

Proteiny se za normálních podmínek formují do trojrozměrného uspořádání popsaného v kapitole 2, tzv. nativního stavu. Ten je stabilizován různými silami jako jsou hydrofobické interakce, elektrostatické síly, vodíkové a disulfidické vazby a van der Waalsovy síly [24]. Procentuální vyjádření vlivu jednotlivých sil je uvedeno v tabulce 4.1. Působením určitých vlivů však může dojít k denaturaci, tedy k porušení této struktury a rozpletení molekuly do neuspořádané a biologicky nefunkční konformace. Schopnost proteinu zachovat si svoji původní trojrozměrnou strukturu se nazývá stabilita proteinu.

Typ vazby	Podíl na stabilitě
Hydrofobické interakce	50,8
Vodíkové vazby	27,1
Van der Waalsovy síly	14,6
Elektrostatické síly	6,4
Disulfidické vazby	1,1

Tabulka 4.1: Procentuální vyjádření podílu jednotlivých sil na stabilitě proteinu. [11]

Stabilita proteinu může být definována pomocí rozdílu tzv. Gibbsovy volné energie ($\Delta\Delta$G) mezi proteinem v nativní konformaci a jeho denaturovaným stavem. Existují různé metody k určení stability proteinu, například diferenční snímací kalorimetrie (DSC), absorpční metoda, fluorescenční metoda nebo isotermální kalorimetrie. Se stabilitou proteinu úzce souvisí jeho teplota tání T_m, při které k denaturaci dochází. Čím vyšší je pak teplota tání proteinu, tím větší je jeho stabilita.

Správná trojrozměrná struktura je zásadní pro fungování proteinu, některé neurodegenerativní poruchy a jiné nemoci jsou způsobeny nesprávným sbalením molekul proteinů [31]. Vzhledem k tomu, že je protein v nativním stavu udržován silami působícími mezi atomy jednotlivých aminokyselin, je zřejmé, že bodové mutace aminokyselin mají obecně vliv na jeho stabilitu a tím i jeho schopnost plnit svoji funkci. Zároveň existuje možnost cílenou mutací měnit stabilitu průmyslově používaných proteinů a tím rozšířit pole jejich působnosti. Příkladem mohou být enzymy používané v pracích práškách. Zvýšení jejich stability by umožnilo používat takový prášek při vyšší teplotě praní se zachováním funkčnosti enzymů v něm obsažených. Studium vlivu aminokyselinových mutací na stabilitu proteinu je jednou z oblastí, jimiž se bioinformatika zabývá.

4.1 Databáze ProTherm

Pro plné porozumění mechanismům proteinové stability je nutný dostatek termodynamických dat. Jejich experimentálním získáváním se zabývá velké množství vědeckých studií, jejichž výsledkem byly mimo jiné změny volné energie ΔΔG při aminokyselinových mutacích konkrétních proteinů za stanovených podmínek. Data z jednotlivých experimentů zkompletoval a publikoval Dr. Gromiha v roce 1999 jako volně dostupnou databázi ProTherm obsahující 3300 záznamů [12]. V dnešní době tato databáze přes 25 000 záznamů mutací ze 740 proteinů. Z této databáze byla vytvořena i trénovací a testovací datová sada pro experimenty prováděné v rámci této práce.

Databáze ProTherm je implementována jako 3DinSight relační databázový systém obsahující data o struktuře, funkci a vlastnostech biomolekul. V databázi jsou obsaženy následující experimentálně zjištěné údaje (žádná vypočtená nebo predikovaná data do ní zahrnuta nebyla):

- Informace o proteinech:
  - Název proteinu, jméno organismu
  - Identifikátor proteinu v databázi PIR, SWISS-PROT či RCSB PDB
  - Enzyme Commission číslo
  - Detaily provedené mutace
  - Sekundární struktura v místě mutace
  - Plocha mutovaného rezidua vystavená vnějšímu prostředí
- Experimentální podmínky:
  - Teplota
  - pH (kyselost/zásaditost prostředí)
  - Buffer, Ion
  - Koncentrace proteinu
  - Míra DSC, CD a další
  - Metoda denaturace
- Termodynamická data:
  - Denaturace prostřednictvím denaturantu:
    - Gibbsova (volná) energie ΔG_{H_2O}
    - Změna volné energie $\Delta\Delta G_{H_2O}$
    - Koncentrace denaturace
    - Sklon denaturační křivky
    - Teplota
  - Denaturace teplotou
    - Gibbsova (volná) energie ΔG
    - Změna volné energie $\Delta\Delta G$
    - Teplota tranzice
    - Změna teploty tranzice
    - Změna enthalpie
    - Změna tepelné kapacity

4.2 Predikční nástroje

Existuje celá řada softwarových nástrojů, které slouží pro predikci vlivu mutací na stabilitu proteinu. Vstupem těchto programů je obvykle soubor obsahující původní nemutovanou sekvenci a textově zapsaná aminokyselinová mutace. Jako výsledek je pak obvykle vrácena hodnota ΔΔG, proto byla také tato veličina použita pro porovnávání úspěšnosti metod a výpočet celkového výsledku. Cílem této práce je nalézt mezi dostupnými metodami co nejspolehlivější a nejpřesnější konsenzus, z existujících nástrojů proto byla vybrána poměrně velká sada vhodných programů. Následuje jejich výčet společně se stručným popisem jejich principu a dalších specifik.

Rozlišujeme několik přístupů k řešení problému predikce vlivu aminokyselinových mutací na stabilitu proteinu:

- **Metody založené na strojovém učení** využívají datasety s experimentálně zjištěnými daty k natrénování klasifikačního modelu.
- **Metody založené na energetické funkci** popisující interakce mezi jednotlivámi atomy. Dále se dělí podle způsobu určení energetické funkce na přístupy založené na:
  - fyzikálním potenciálu (simulují rozložení sil mezi atomy),
  - statistickém potenciálu (využívají statistické analýzy známých proteinových struktur),
  - empirickém potenciálu (kombinují váhované fyzikální a statistické výrazy). [16]

4.2.1 PoPMuSiC2

Prvním vybraným nástrojem je program PoPMuSiC2. Jako vstup využívá informace o struktuře mutovaného proteinu. Jádrem výpočtu je lineární kombinace 15 energetických funkcí a jednoho nezávislého členu podle vzorce:

$$\Delta\Delta G \sum_{i=1}^{13} \alpha_i(A)\Delta\Delta W_i + \alpha_{14}(A)\Delta\Delta W_+ + \alpha_{15}(A)\Delta\Delta W_- + \alpha_{16}(A). \tag{4.1}$$

Prvních 13 členů je součet tzv. statistických potenciálů extrahovaných z databáze známých proteinových struktur. Tyto potenciály popisují různé interakce přispívající ke stabilitě proteinu. Podle komplexnosti je možné je rozdělit na lokální potenciály popisující korelaci mezi vzájemně blízkými rezidui v řetězci aminokyselin a nelokální potenciály, které vyjadřují působení reziduí blízkých v trojrozměrném prostoru. Na stabilitu proteinu může mít vliv i změna objemu molekuly v důsledku mutace, což statistické potenciály nemohou pokrýt. PoPMuSiC2 proto zavádí další dva termy ΔV_+ vyjadřující zvětšení a ΔV_- udávající zmenšení objemu molekuly. Poslední člen je tzv. nezávislý term jehož hodnota je konstantní a rovna 1. Pro určení všech 16 koeficientů jednotlivých členů rovnice byla použita jednoduchá neuronová síť [7].

PoPMuSiC2 poskytuje uživatelské rozhraní prostřednictvím webové služby. Ta je dostupná pouze registrovaným uživatelům. Díky tomu nástroj nabízí možnost ukládat vstupy a výstupy dříve proběhlých mutací, což usnadňuje například znovuprovedení některých experimentů. Vstupní sekvenci program přijímá ve formátu PDB [2]; je možné nahrát soubor z disku, nebo zadat pouze PDB identifikátor, podle něhož si PoPMuSiC2 dokáže příslušný soubor dohledat a stáhnout. Mutace je možné zadávat v jednom ze tří režimů:

- Single: slouží pro zadávání jednotlivých bodových substitucí zapsaných jako pozice mutace, wild-type aminokyselina a nová aminokyselina, jíž je původní nahrazena.
- Systematic: provede predikci $\Delta\Delta G$ nad zadanou PDB strukturou systematicky pro všechny možné mutace na všech pozicích řetězce, tedy celkem 19× počet_reziduí.
- File: umožňuje uživateli nadefinovat mutační soubor s libovolným počtem mutací. Tento soubor je navíc automaticky uložen pro případné další použití.

Výsledkem predikce je hodnota $\Delta\Delta G$ udávající změnu stability proteinu. Tato hodnota je základem pro konstrukci trénovacího datasetu používaného v této práci.

4.2.2 SDM

Program SDM (Site Directed Mutator) je nástroj pro predikci vlivu aminokyselinových mutací na stabilitu proteinu pracující na základě statistických potenciálů. Využívá sadu tabulek aminokyselinových substitucí ve specifickém prostředí v homologních proteinech. Tabulky byly odvozeny z databáze zarovnání sekvencí. Z těchto dat je vypočteno skóre stability, které je analogií ke změně volné energie mezi wild-type a mutovanou sekvencí [33]. Vstupem nástroje SDM je struktura proteinu zadána buď prostřednictvím PDB identifikátoru, nebo nahrána jako soubor v PDB formátu. Mutace je specifikována zadáním řetězce proteinu, pozice mutace a nové aminokyseliny. Výstup sestává z celkového zařazení mutace (stabilizující/destabilizující), změny volné energie $\Delta\Delta G$ a dalších informací, které nejsou pro účely této práce využity (např. predikovaná sekundární struktura mutované sekvence).

4.2.3 FoldX 3.0

Nástroj FoldX počítá změnu stability pomocí empiricky odvozeného silového pole simulujícího působení sil mezi jednotlivými atomy. Použité silové pole (force field) bylo odvozeno z výsledků laboratorních experimentů na reálných proteinech, z nichž většina je uložena v databázi ProTherm. Tento přístup je kompromisem mezi časově náročným modelováním složitých fyzikálních funkcí na jedné straně a obtížně aktualizovatelným využitím statistického potenciálu na straně druhé. Volná energie je počítána jako váhovaný součet hodnot vyjadřujících, jakou měrou se na stabilitě proteinu podílí různé meziatomové interakce [13]. Program umožňuje uživateli nastavit velké množství parametrů, jako je teplota, pH nebo molární plynová konstanta.

4.2.4 Eris

Program Eris využívá fyzikálního *force field* přístupu modelujícího atomy společně s rychlými algoritmy pro sbalení postranních řetězců a uvolnění atomů proteinové kostry. Volná energie je vyjádřena jako váhový součet sil působících v proteinu, tedy zejména van der Waalsových sil a vodíkových vazeb [34]. Služba je dostupná prostřednictvím webu a od uživatele vyžaduje registraci. Mutovanou proteinovou strukturu lze vložit jako PDB identifikátor, nebo vlastní soubor ve formátu PDB. Mutace je možné zadávat jednotlivě, nebo spustit predikci dávkově. Výstupem programu je predikovaná změna volné energie $\Delta\Delta$G.

4.2.5 Auto-mute2

Nástroj Auto-mute2 využívá pro predikci změn stability proteinů principy strojového učení. Nabízí dva klasifikační modely *Random Forest (RF)* a *Support Vector Machine (SVM)*,

které rozhodují o vlivu mutace pouze v oboru stabilizující/destabilizující a dva regresní modely *Tree Regression (REPTree)* a *SVM Regression (SVMreg)*, které predikují přímo hodnotu $\Delta\Delta G$ [19]. Auto-mute2 je k dispozici jako sada aplikací implementovaná v jazyce Perl; trénování klasifikátoru je realizováno pomocí softwarového balíku WEKA. Sestává z několika nástrojů, přičemž pro účely tohoto projektu je použit nástroj *stability_changes_ddG*, který predikuje stabilitu proteinu jako změnu volné energie. Pomocí parametrů příkazové řádky je možné nastavit:

- způsob předání PDB struktury (automatické stažení z internetu/lokální úložiště),
- jméno souboru se seznamem mutací,
- použitý klasifikační/regresní model,
- jméno výstupního souboru.

4.2.6 I-Mutant

Dalším programem využívajícím strojové učení je I-Mutant. V rámci této práce byly použity verze I-Mutant2.0 [4] a I-Mutant3.0 [5]. Nová verze se kromě toho, že je k dispozici pouze jako webová služba, odlišuje zejména oborem hodnot prováděné klasifikace. Verze 3.0 totiž rozlišuje mutace *stabilizující*, *destabilizující* a *neutrální*. U obou verzí je však možné požadovat výsledek jako změnu volné energie, což bylo využito v praktické části této práce. Z hlediska vstupu lze rozlišit dva režimy predikce. V sekvenčním režimu je využita pouze primární struktura, jako vstup tedy postačí soubor se zapsanou sekvencí aminokyselin. Ve strukturním módu je třeba předat PDB soubor obsahující strukturu mutovaného proteinu. Dalšími důležitými parametry jsou pozice mutace, wild-type aminokyselina na této pozici, nová aminokyselina, teplota a pH.

4.2.7 Rosetta

Rosetta [29] je označení sady programů sloužících pro funkční design proteinů. Jeho součástí je i samostatný modul pro predikci změny stability po provedení mutace (modul ddg_monomer). Společně s nástrojem Eris jsou jedinými reprezentanty metod využívajících fyzikální silové pole. Výpočetní náročnost tohoto přístupu by však měla být vyvážena vyšší úspěšností predikce. Z důvodu dlouhé doby trvání jedné predikce je nástroj Rosetta k dispozici pouze offline

4.2.8 iPTREE-STAB

Nástroj iPTREE-STAB [15] slouží k predikci efektu bodové mutace na stabilitu proteinu (stabilizující/destabilizující) a míry této změny jako $\Delta\Delta G$. K výpočtu využívá metody strojového učení. Jednoduchý rozhodovací strom realizující klasifikaci byl natrénován na datasetu 1859 mutantů 64 různých proteinů [15]. iPTREE-STAB je k dispozici on-line. Jako vstupní data přijímá wild-type aminokyselinu společně se třemi sousedními rezidui na obě strany, novou aminokyselinu, teplotu a pH.

4.2.9 mCSM

Program mCSM [26] využívá ke studiu a následné predikci vlivu jednobodových aminokyselinových mutací na stabilitu proteinu koncept strukturálních signatur založených na

grafech. Prostředí reziduí je reprezentováno grafy, jejichž uzly odpovídají atomům a hrany fyzikálním a chemickým interakcím mezi nimi. Nástroj je dostupný prostředníctvím webového rozhraní, s jehož pomocí uživatel programu předá PDB soubor a definuje požadované mutace. Ty je možné zadat třemi způsoby:

- Single: slouží pro zadávání jednotlivých bodových substitucí zapsaných jako pozice mutace, wild-type aminokyselina a nová aminokyselina, kterou je původní nahrazena.
- Systematic: provede predikci $\Delta\Delta G$ nad zadanou PDB strukturou systematicky pro všechny možné mutace na všech pozicích řetězce, tedy celkem 19× počet_reziduí.
- File: umožňuje uživateli nadefinovat mutační soubor s libovolným počtem mutací. Tento soubor je navíc automaticky uložen pro případné další použití.

Výsledkem predikce je tradičně rozdíl ve stabilitě proteinu vyjádřený jako $\Delta\Delta G$.

4.2.10 DUET

Predikční server DUET [27] v sobě kombinuje dva nástroje pro predikci změny stability proteinu vlivem mutace, konkrétně mCSM a SDM. Pro nalezení optimálního konsenzu využívá metodu strojového učení SVM. Za tímto účelem byl natrénován na datasetu mutací s dostupnými termodynamickými daty odvozeném z databáze ProTherm. Vstupní protein může uživatel nahrát jako soubor ve formátu PDB, nebo zadat PDB identifikátor. Mutace lze vkládat jednotlivě jako pozici a původní a novou aminokyselinu, nebo provést systematickou mutaci, jako v případě nástroje mCSM. Na stránce s výsledkem predikce jsou vypsány dílčí předpovědi pro změnu volné energie jednotlivých metod a vypočtený konsenzus. Dále stránka obsahuje interaktivní vizualizaci zadané PDB struktury [27].

Kapitola 5

Evoluční algoritmy

Evoluční algoritmy jsou stochastické metody pro řešení složitých problémů inspirované darvinistickým paradigmatem. Jsou založené na principu variace a selekce, který je považován za fundamentální princip Darvinovy teorie evoluce [3]. Evolučních algoritmů existuje více typů, spojují je však některé základní znaky.

- Evoluční algoritmy jsou založené na populaci. To znamená, že algoritmus pracuje s celou množinou kandidátních řešení daného problému, které optimalizuje paralelně. Koncept populace je základním principem klasického evolučního procesu.
- Orientace na kvalitu řešení. Každý jedinec je tvořen tzv. chromozomem, což je vhodným způsobem zakódované řešení problému. Dále je každému jedinci přiřazena hodnota fitness, která vyjadřuje jeho kvalitu. V průběhu evolučního algoritmu jsou pak preferováni kvalitnější jedinci, čímž je zajištěna konvergence algoritmu.
- Variace. Jedinci jsou vystaveni variačním operátorům, které mohou měnit jejich chromozomy. Tento princip je hybnou silou evoluce a umožňuje prohledávání stavového prostoru. [35]

```
begin
    t := 0;
    inicializace G(t);
    vyhodnocení G(t);
    while (not ukončovací podmínka) do
    begin
        t := t + 1;
        selekce G(t) z G(t-1);
        změna G(t);
        vyhodnocení G(t);
    end
end
```

Obrázek 5.1: Princip obecného evolučního algoritmu. [17]

Obecný princip evolučního algoritmu je popsán pseudokódem na obrázku 5.1. Na počátku algoritmu je inicializována celá populace vytvořením odpovídajícího počtu jedinců. Každý jedinec je pak ohodnocen hodnotou fitness. Proměnná t slouží k odlišení jednotlivých cyklů iterativního běhu algoritmu. Po inkrementaci proměnné t dochází k výběru jedinců do nové populace na základě jejich fitness hodnoty. Tato populace je označena *G(t)*, zatímco původní

je již *G(t-1)*. Někteří z jedinců nové populace jsou vystaveni působení genetických operátorů, které z nich vytvoří nová řešení. Dva hlavní typy těchto transformací jsou mutace a křížení. Mutace je unární operátor, který provede náhodné drobné změny v jednom jedinci. Křížení je operátor vyššího řádu, který vytvoří nové řešení kombinací několika jedinců. [17]

Důležitou součástí evolučního algoritmu je ukončovací podmínka. Při jejím splnění dojde k přerušení iterativního výpočtu a jako řešení problému je vrácen jedinec s nejlepší hodnotou fitness v aktuální populaci. Jako ukončovací podmínka se obvykle zadává požadavek na kvalitu výsledného řešení nebo maximální počet iterací, po jehož překročení je algoritmus ukončen.

Mezi evoluční algoritmy se řadí genetické algoritmy, evoluční strategie a genetické programování. Díky svým vlastnostem byly pro účely praktické části této práce použity evoluční strategie, jejichž specifika a jednotlivé varianty jsou podrobněji popsány dále.

5.1 Evoluční strategie

Evoluční strategie (zkráceně ES) jsou zvláštním případem evolučních algoritmů. Používají se pro optimalizaci funkcí, které nejsme schopni řešit analyticky. Od genetických algoritmů, které mezi evoluční algoritmy také patří, se původní koncept evolučních strategií odlišoval zejména v tom, že:

- jedinec v populaci je reprezentován vektorem reálných čísel,
- neobsahovaly operátor křížení a noví jedinci vznikali pouze s použitím operátoru mutace.

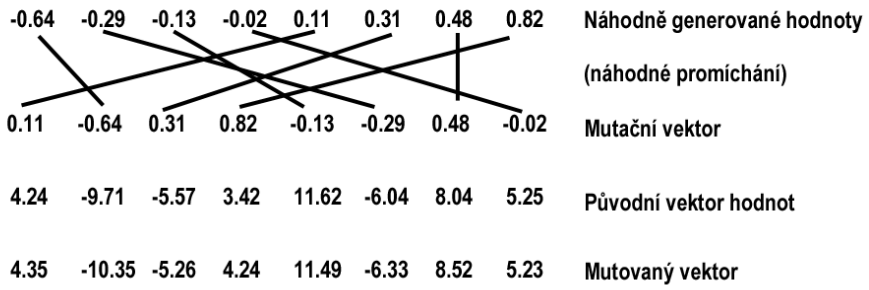
Později však bylo křížení do některých evolučních strategií začleněno.

5.1.1 Mutace

Operátor mutace má v případě evolučních strategií v porovnání s operátorem křížení větší váhu, než u genetických algoritmů. Vzhledem k tomu, že evoluční strategie pracuje s vektorem reálných hodnot, provádí se zpravidla přičtením stejně dlouhého vektoru náhodných malých hodnot ke kandidátnímu řešení. Tento vektor se nazývá vektor mutace a jednotlivé složky jsou generovány s Gaussovým normálním pravděpodobnostním rozdělením. Vznik nového jedince $x_{(i+1)}$ lze zapsat vztahem

$$x_{(i+1)} = x_i + N(0, \sigma), \tag{5.1}$$

kde x je kandidátní řešení, σ vektor standardních odchylek a $N(0, \sigma)$ vektor nezávislých náhodných čísel generovaných s normálním rozdělením, nulovou střední hodnotou a standardními odchylkami danými složkami vektoru σ [17]. Velikost prvků tohoto vektoru ovlivňuje velikost přičítaných odchylek a tedy i sílu mutace. V jednodušších případech může místo vektoru σ existovat pouze jedna hodnota, která pak bude stejná pro všechny složky. Příklad mutace je uveden na obrázku 5.2.

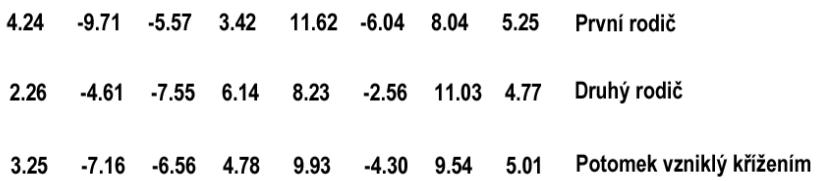


Obrázek 5.2: Příklad aplikace operátoru mutace.

5.1.2 Křížení

V současných evolučních strategiích se kromě mutace používá i operátor křížení, neboli rekombinace. Dva základní typy tohoto mechanismu jsou tzv. pohlavní křížení a křížení panmiktické. Pro vytvoření potomka pomocí pohlavního křížení jsou vybráni jedinci původní populace a všechny složky nového jedince jsou vytvořeny kombinací složek těchto dvou rodičů. Je běžným postupem, že není možné vybrat téhož jedince za oba rodiče. Naproti tomu v případě panmiktického křížení je vybrán jeden z rodičů jako fixní pro celou operaci, ale pro každou složku jeho vektoru je druhý rodič nově náhodně vybrán z populace. Při použití této metody je na populaci nahlíženo jako na zásobárnu genů, z nichž jsou potomci konstruováni. [1]

Konkrétní postupy křížení opět zohledňují fakt, že jedinec je představován vektorem reálných hodnot. Dva tradiční rekombinační operátory používané v evolučních strategiích jsou tzv. křížení diskrétní a křížení průměrem. V případě diskrétního křížení je nový jedinec skládán ze složek rodičovských vektorů a vždy je náhodně rozhodnuto, ze kterého jedince bude hodnota zkopírována. Potomek vzniklý křížením průměrem má každou svoji složku vypočtenou jako průměrnou hodnotu odpovídajících složek obou rodičovských vektorů. Příklad křížení je uveden na obrázku 5.3. Oba zmíněné rekombinační operátory je možné provádět v pohlavní i panmiktické variantě. [1]



Obrázek 5.3: Příklad aplikace operátoru křížení průměrem.

5.1.3 Obnova populace

Nejjednodušší evoluční strategie pracují pouze s jedním jedincem. Označují se jako dvoučlenné ES, nebo též zápisem $(1 + 1)$-ES, který vyjadřuje, že populace má velikost jedna a v každé iteraci je generován jeden potomek. Z principu těchto strategií je patrné, že operátor křížení postrádá význam, poněvadž je k dispozici pouze jeden rodič. Tato evoluční strategie obsahuje následující čtyři kroky [17]:

1. Volba počátečního řešení x a odchylek σ.
2. Vznik mutovaného jedince podle vzorce $x_{(i+1)} = x_i + N(0, \sigma)$.
3. Pokud $f(y) < f(x)$, jedinec x je nahrazen jedincem y.
4. Pokud je splněna ukončovací podmínka, výpočet je zastaven, jinak se pokračuje bodem 2. [6]

Pokročilejší evoluční strategie zavádějí pojem populace tak, jak je chápán z biologického hlediska, tedy jako množinu jedinců, kteří mezi sebou soutěží. Tyto ES jsou označovány jako vícečlenné, čemuž odpovídá zápis $(\mu + \lambda)$-ES nebo (μ, λ)-ES. Hodnota μ označuje velikost populace a λ značí počet potomků generovaných v každé iteraci. Celý $(\mu + \lambda)$-ES postup sestává z následujících čtyř kroků.

1. Volba počáteční populace P sestávající z μ kandidátních řešení $x^{(i)}, i = 1, 2, ..., \mu$. Je také zvolena hodnota σ vyjadřující sílu mutace.
2. Tvorba λ mutovaných jedinců $y^{(j)}$ podle vzorce $y^{(j)} = x^{(i)} + N(0, \sigma)$.
3. Úprava popupace P jako sjednocení rodičovských řešení $x^{(i)}$ a potomků $y^{(j)}$. Z takto upravené populace je pro další iteraci vybráno μ nejlepších řešení a zbytek je smazán.
4. Pokud je splněna ukončovací podmínka, výpočet je zastaven, jinak se pokračuje bodem 2 [6].

Strategie (μ, λ)-ES se od popsaného postupu liší výběrem nové populace. Ta je pro následující iteraci vybírána pouze z množiny potomků, žádní jedinec tedy nepřežije do další iterace. Tento mechanizmus se též nazývá elitářský a je vhodné jej použít v případě, kdy je obtížné nalézt úspěšné mutace. V těchto případech by totiž strategie $(\mu + \lambda)$-ES mohla vést ke stagnaci kvůli uváznutí v lokálním extrému [6].

Vícečlenné evoluční strategie také umožňují zavedení operátoru křížení. Ve výše uvedeném postupu by se tato změna projevila v bodě 2), kde by byla nová řešení generována pomocí mechanizmů popsaných v podkapitole Křížení a takto vzniklí jedinci by byli teprve vystaveni mutaci.

5.1.4 Pravidlo 1/5

Nejdůležitějším řídicím parametrem jednoduchých evolučních strategií je síla mutace σ. Jeho flexibilními úpravami během výpočtu je možné zlepšit konvergenci algoritmu k hledanému maximu. Prof. Rechenberg [28] vypočítal pro ideální velikost parametru σ pravděpodobnosti úspěšné mutace v populaci. Při ideální velikosti mutačního kroku jsou tyto pravděpodobnosti $p_1^{opt} = 0,184$ a $p_2^{opt} = 0,270$. Na základě toho prof. Rechenberg vyslovil pravidlo 1/5, podle kterého je optimální podíl úspěšných mutací z celkového počtu 1/5. Je-li úspěšných více mutací, měla by se zvýšit síla mutace, v opačném případě naopak snížit. K úpravě dochází jednou za n iterací, n je tedy i celkový počet, z něhož se počítá podíl úspěšných mutací. Míra úprav parametru σ je dána vzorcem:

$$\sigma^{t+n} = \begin{cases} c_d \times \sigma^t \text{ , pokud } p_s^t < 1/5 \\ c_i \times \sigma^t \text{ , pokud } p_s^t > 1/5 \\ \sigma^t \text{ , pokud } p_s^t = 1/5 \end{cases} \tag{5.2}$$

kde p_s^t je frekvence úspěšných mutací, c_d snižující faktor a c_i zvyšující faktor. Hodnoty faktorů by měly být podle prof. Schwefela $c_d = 0,82$ a $c_i = 1/c_d = 1/82 = 1,22$. [30]

Ačkoliv řešené problémy mají obecně rozdílné charakteristiky, umožňuje tato heuristika vhodně dynamicky upravovat sílu mutace. Na druhou stranu může mít její použití za následek snížení efektivity nebo robustnosti evoluční strategie a například její předčasné ukončení.

5.1.5 Autoevoluce řídících parametrů

Mimo pravidlo 1/5 existují pokročilejší způsoby pro adaptaci řízení výpočtu. Při tzv. autoevoluci řídících parametrů je každý vektor jedince řešení rozšířen o sekci obsahující proměnné, které ovlivňují jeho mutace. V každé iteraci evoluční strategie pak podléhají evolučnímu procesu nejen parametry tvořící kandidátní řešení problému, ale i jeho řídící parametry. Existují tři hlavní způsoby autoevoluce řídících parametrů. [9]

Nejjednodušší typem autoevoluce je nekorelovaná mutace s jednou hodnotou mutačního kroku, též nazývána jako autoevoluce typu 1. Vektor každého jedince je rozšířen o jedinou položku σ určující sílu mutace pro všechny ostatní parametry. Samotný prvek σ je pak v každé iteraci násoben Eulerovým číslem umocněným na náhodné číslo z intervalu $(0, 1)$, celá mutace tedy probíhá podle vzorců:

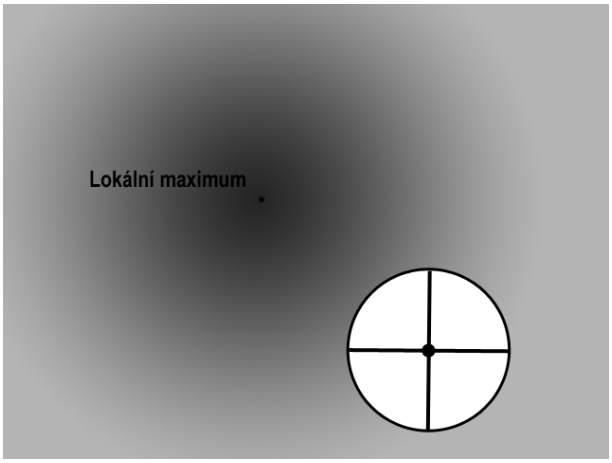
$$\sigma' = \sigma \times e^{\tau \times N(0,1)}, \tag{5.3}$$

$$x_i' = x_i + \sigma' \times N_i(0,1)', \tag{5.4}$$

kde $N_i(0, 1)$ označuje Gaussovo rozložení pravděpodobnosti a konstanta τ je označována jako parametr učení a měla by mít hodnotu $\tau = 1/\sqrt{n}$. Aby nedocházelo ke snížení parametru σ na hodnoty velmi blízké nule, je interval přípustných hodnot tohoto parametru zdola omezen konstantním prahem ϵ_0 pravidlem

$$pokud \quad \sigma' < \epsilon_0, \quad pak \quad \sigma' = \epsilon_0. \tag{5.5}$$

Efekt takové mutace na jedince popisujícího řešení problému ve dvourozměrném prostoru je ilustrován na obrázku 5.4. Protože velikost mutačního kroku je popsána pouze jednou hodnotou, je síla mutace stejná ve všech směrech. Všichni přípustní jedinci získatelní mutováním původního jedince kolem něj tvoří kruh.



Obrázek 5.4: Efekt autoevoluce typu 1. Kandidátní řešení je znázorněno černým kolečkem. Potomci jsou vybíráni z jeho okolí naznačeného bílým kruhem. [9]

Rozšířením výše popsané metody je autoevoluce typu 2, která zavádí vlastní velikost mutačního kroku každému prvku kandidátního řešení. Jedinec má pak tvar $\langle x_1, ..., x_n, \sigma_1, ..., \sigma_n \rangle$. Globální parametr τ' zůstává zachován, je však přidán tzv. specifický parametr učení. Mutace je pak popsána vzorcem

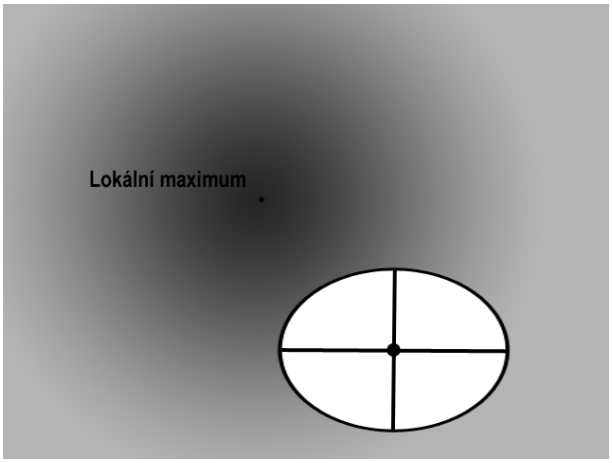
$$\sigma_i' = \sigma_i \times e^{\tau' \times N(0,1) + \tau \times N(0,1)}, \tag{5.6}$$

$$x_i' = x_i + \sigma' \times N_i(0,1)', \tag{5.7}$$

kde $N_i(0,1)$ označuje Gaussovo rozložení pravděpodobnosti a $\tau = 1/\sqrt{2n}$ je specifický parametr učení. Doporučená hodnota společného parametru učení je narozdíl od autoevoluce typu 1 rovna $\tau' = 1/\sqrt{2\sqrt{n}}$. Zavedené omezení rozsahu mutačního kroku je rozšířeno na všechny prvky řídící části vektoru jedince, tedy:

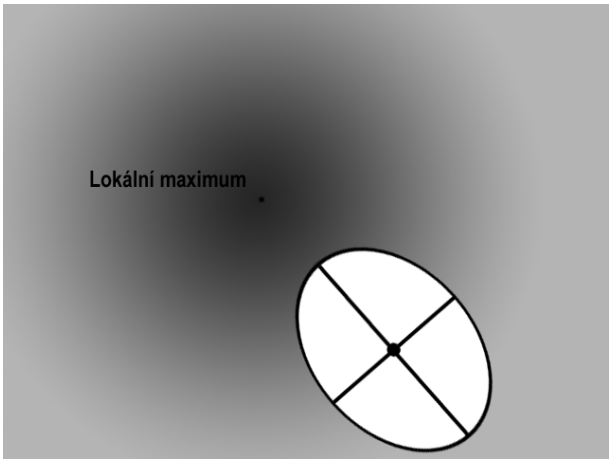
$$pokud \quad \sigma' < \epsilon_0, \quad pak \quad \sigma' = \epsilon_0. \tag{5.8}$$

Efekt autoevoluce typu 2 na jedince popisujícího řešení problému ve dvourozměrném prostoru je ilustrován na obrázku 5.5. Podprostor, v něm se mohou nacházet mutovaní jedinci má tvar elipsy okolo jedince původního, mutační krok může být ve směrech jednotlivých os různý. Díky tomu se bod představovaný jedincem rychleji přesouvá do blízkosti hledaného maxima, osy elipsy jsou však rovnoběžné se soustavou souřadnic.



Obrázek 5.5: Efekt autoevoluce typu 2. Kandidátní řešení je znázorněno černým kolečkem. Potomci jsou vybíráni z jeho okolí naznačeného bílou elipsou. [9]

Třetím popsaným typem autoevoluce je tzv. korelovaná mutace. Ta umožňuje, aby elipsa okolo kandidátního řešení, z níž se vybírají potomci, byla v prostoru ideálně natočena (ilustrováno na obrázku 5.6.). Toho je dosaženo použitím tzv. kovarianční matice, která se používá ke generování mutovaných jedinců. Ta má rozměry $n \times n$, kde n je počet hledaných parametrů a jednotlivé její položky se počítají ze síly mutace odpovídajících prvků kandidátního řešení a tzv. rotačního úhlu. Rotační úhly jsou udané pro každou dvojici hledaných parametrů, jedinec má pak tvar $\langle x_1, ..., x_n, \sigma_1, ..., \sigma_n, \alpha_1, ..., \alpha_k \rangle$, kde $k = n \times (n-1)/2$.



Obrázek 5.6: Efekt autoevoluce typu 1. Kandidátní řešení je znázorněno černým kolečkem. Potomci jsou vybíráni z jeho okolí naznačeného bílou elipsou, která je svojí delší poloosou natočená k lokálnímu maximu. [9]

5.2 CMA-ES

Koncept generování potomků s použitím kovarianční matice využvá i metoda CMA-ES (anglicky Covariance Matrix Adaptation Evolution Strategy). Jako jeden z parametrů pro mutaci je použit též průměr z aktuálních hodnot kandidátních řešení. Samotná tvorba nového potomka probíhá podle vzorce

$$x_k^{(g+1)}\ m^{(g)} + \sigma^{(g)} N(0, C^{(g)}) \quad pro \quad k\ = 1, ..., \lambda, \tag{5.9}$$

kde $m^{(g)}$ je průměrná hodnota hledaných parametrů, $\sigma^{(g)}$ je globální mutační krok v generaci g, $C^{(g)}$ je kovarianční matice v dané generaci a λ definuje velikost populace a zároveň počet generovaných potomků. Pro novou populaci se vypočítá hodnota m podle vzorce

$$m^{(g+1)} = \sum_{i=1}^{\mu} w_i x_{i:\lambda}^{(g+1)}, \tag{5.10}$$

kde $\mu \leq \lambda$ je velikost rodičovské populace, w_i jsou váhové koeficienty pro rekombinaci a $x_{i:\lambda}^{(g+1)}$ je i-tý nejlepší jedinec ze seznamu kandidátních řešení seřazených podle fitness funkce. Dalším krokem je výpočet tzv. evolučních cest, tedy hodnot pro určení velikosti mutačního kroku σ a kovarianční matice. Vstupy výpočtu jsou známé hodnoty evolučních cest a mutačního kroku z předchozí generace, průměr hledaných parametrů v aktuální a předchozí generaci a předem definované konstanty. Výpočet evolučních cest se anglicky nazývá *rank-one-update*, dále se provádí tzv. *rank-μ-update* opět na základě stávajícího mutačního kroku σ, stávajícího průměru m, dále seznamu kandidátních řešení seřazeného podle fitness hodnoty a dalších předem stanovených parametrů. Provedením *rank-one-update* a *rank-μ-update* se aktualizuje kovarianční matice, následně se upraví velikost mutačního kroku σ a přepočítají se další pomocné proměnné pro výpočty v následující generaci. Konkrétní vzorce je možné nalézt v [14].

CMA-ES představuje pokročilou metodu pro nelineární optimalizaci, která se dokáže vyrovnat s některými případy, kdy jiné metody selhávají v důsledku nekonvexního nebo příliš členitého prohledávaného prostoru. [14]

Kapitola 6

Návrh a implementace

Praktická část této práce zahrnovala rozšíření a úpravu trénovací a testovací datové sady pro evoluční strategie, kterou vytvořil pro svoji diplomovou práci Ing. David Pavlík [25]. Jednalo se zejména o přidání dat získaných ohodnocováním dalšími nástroji a o doplnění příznaku, že daný záznam byl použit pro trénování konkrétního nástroje do tabulek s výsledky jednotlivých nástrojů. Dále implementační část spočívala ve vytvoření aplikace pro hledání konsenzu mezi zvolenými nástroji s použitím evoluční strategie, experimentování s ní a testování nalezených řešení.

6.1 Příprava trénovací a testovací datové sady

Část dat pro trénování a testování evoluční strategie byla převzata z existující relační MySQL databáze *Stability*, kterou pro svoji diplomovou práci vytvořil Ing. Pavlík [25]. Její součástí je dvojice tabulek *protherm* a *protherm_mutations*, které obsahují vybrané záznamy vydolované z databáze ProTherm. V tabulce *protherm* jsou uloženy informace o zkoumaném proteinu a podmínkách, za jakých byly experimenty provedeny včetně změřené hodnoty $\Delta\Delta G$ vyjádřenou v kcal/mol. Tabulka *protherm_mutations* pak obsahuje data o konkrétních mutacích, např. pozice substituce a původní a mutovaná aminokyselina. Obě tabulky lze propojit skrz atribut *uid* a získat tak kompletní informace potřebné pro další práci. Databáze *Stability* dále obsahuje tabulky s hodnotami $\Delta\Delta G$ predikovanými jednotlivými nástroji pro dané mutace. Ty jsou stejně jako referenční hodnoty vyjádřené v kcal/mol, lze je tedy přímo porovnávat.

6.1.1 Přidání výsledků dalších nástrojů

V rámci své práce zpracoval Ing. Pavlík nástroje *Eris*, *FoldX*, *Rosetta* a *I-Mutant2.0*, z nich byly převzaty první tři jmenované. Pro účely této práce byla zároveň použita i data z dalších zdrojů. Z práce Ing. Malinky [18] byly převzaty zpracované výsledky čtyř nástrojů (*SDM*, *I-Mutant3.0* ve strukturní verzi, *iPTREE-STAB* a *mCSM*).

V rámci implementační části této práce jsem tuto sadu dat doplnil o výsledky nástrojů *Auto-mute2*, *PoPMuSiC2* v nové verzi a *DUET*. Pro získání potřebných dat byl vytvořen skript *dp.py*, který spolupracuje se zmíněnými nástroji a dotazuje se na predikce pro konkrétní mutace. Vstupem programu je soubor ve formátu XML, který umožňuje nastavit, pro které nástroje se predikce budou provádět a dále zadat seznam elementů obsahujících parametry jednotlivých substitucí. Požadovaná struktura je popsána XSD schématem v příloze B a příklad souboru *mutations.xml* je uveden na obrázku 6.1. Tento vstupní soubor

byl zpravidla vytvářen automaticky pomocným skriptem, který pracoval přímo s databází *Stability* a uložené mutace vypsal v požadovaném formátu.

```
<mutations duet="0" popmusic="1" automute="0">  <mutation uid="14" pdbId="1IOB" temp="25.0" ph="6.5" wild="T" mut="A" pos="9" chain="-" />
  <mutation uid="15" pdbId="1IOB" temp="25.0" ph="6.5" wild="T" mut="L" pos="9" chain="-" />
  <mutation uid="16" pdbId="1IOB" temp="25.0" ph="6.5" wild="T" mut="Q" pos="9" chain="-" />
  <mutation uid="17" pdbId="1IOB" temp="25.0" ph="6.5" wild="T" mut="G" pos="9" chain="-" />
  <mutation uid="19" pdbId="1IOB" temp="25.0" ph="6.5" wild="K" mut="G" pos="97" chain="-" />
  <mutation uid="20" pdbId="1IOB" temp="25.0" ph="6.5" wild="K" mut="V" pos="97" chain="-" />
  <mutation uid="21" pdbId="1VQB" temp="25.0" ph="7.0" wild="E" mut="F" pos="30" chain="-" />
  <mutation uid="23" pdbId="1VQB" temp="25.0" ph="7.0" wild="C" mut="S" pos="33" chain="-" />
```

Obrázek 6.1: Ukázka souboru mutations.txt.

Nástroj Auto-mute2 byl používán jako samostatná aplikace spouštěná na lokálním počítači. Skript *dp.py* tedy vždy připravil vstupní soubor pro tento nástroj a zavolal skript *stability_changes_ddG.pl* s parametry *-t 0 -f input.txt -m 2 -o output.txt*. Výsledný soubor byl pak zpracován a byly přečteny výsledky predikce včetně hodnoty předpovězené $\Delta\Delta G$.

Nástroj *DUET* běží jako webová služba a uživateli je dostupný prostřednictvím webového rozhraní. Dílčím úkolem procedury získávající predikce tohoto nástroje bylo stažení souboru ve formátu PDB pro daný protein a sestavení a vyslání http dotazu ve správném formátu. Z odpovědi na tento dotaz byly pak získány výsledky predikce.

Prostřednictvím webového rozhraní je přístupný také nástroj *PoPMuSiC2*. Ten však vyžaduje od uživatele přihlášení. Odpovídající procedura tedy musí kontrolovat, zda odpověď serveru neobsahuje přesměrování na stránku s přihlášením a pokud ano, tak přihlášení provést. Skript zároveň pracuje se soubory cookie, díky čemuž je možné zadávat dotazy jako přihlášený uživatel. Výsledky predikce nástroj *PoPMuSiC2* neposílá formou odpovědi na http dotaz, ale umístí je na server, odkud si jej uživatel může stáhnout. Procedura proto opakovaně v intervalu 8 sekund kontroluje, zda je již výsledek připraven a případně jej stáhne a zpracuje.

Výstupy skriptu *dp.py* jsou soubory *automute2.sql*, *popmusic.sql* a *duet.sql*. Jejich obsahem je sada SQL příkazů, které zajistí vložení informací o provedených predikcích do databáze. Soubory lze nahrát a spustit pomocí uživatelského rozhraní phpMyAdmin, v databázi *Stability* je vyžadována existence tabulek *protherm_automute2*, *protherm_duet* a *protherm_popmusic*. Skript navíc při zpracování vstupního souboru s mutacemi zjistí, zda jsou již výsledkové soubory přítomny a pokud ano, přeskakuje mutace, jejichž výsledky již byly konkrétním nástrojem predikovány.

6.1.2 Záznamy použité pro trénování nástrojů

Některé z nástrojů, mezi nimiž se hledá optimální konsenzus, jsou založené na strojovém učení a existuje překryv mezi jejich trénovací datovou sadou a databází *ProTherm*. Pro získání relevantnějších výsledků práce byly identifikovány a v dalších fázích vynechány záznamy, které sloužily k trénování použitých nástrojů. Za tímto účelem byl do tabulek s výsledky predikcí přidán boolovský atribut *usedForTraining*. Tento příznak je nastaven na hodnotu *True*, jestliže byl daný záznam v tabulce použit k trénování odpovídajícího nástroje. Pro automatizaci postupu byla celá procedura napsána do skriptu *trainSet.py*. K většině nástrojů (ne však ke všem) byla datová sada dostupná formou výčtu mutací, označování trénovacích záznamů probíhalo tedy následovně.

Nástroje DUET a mCSM mají stejnou trénovací sadu, která je kompletně dostupná a přiložena v souboru *duet_mcsm_train.txt* ve složce *dataSets/trainRecords/inputTrainSets/*. Stačilo tedy odpovídající záznamy v databázi označit příznakem *usedForTraining*.

Dostupnou trénovací sadu mají také nástroje *iPTREE-STAB*, *PoPMuSiC2*, *Auto-mute2* a *iMutant3.0*. Je uvedena vždy v odpovídajích souborech ve složce *dataSets/trainRecords/-inputTrainSets/*.

Nástroj *Eris* pracuje na principu výpočtu energií podle silového pole, které bylo trénováno na 34 proteinech uvedených v souboru *eris_train.txt*. Jako trénovací byly proto označeny všechny mutace těchto proteinů.

Nástroje *Rosetta* a *FoldX* nebyly nijak trénovány, žádné záznamy tedy nebylo třeba vyčleňovat.

6.1.3 Výběr trénovací a testovací datové sady

Po kompletaci dat bylo třeba vybrat vhodné záznamy z databáze *ProTherm* jako trénovací a testovací sadu dat pro evoluční strategii. V prvé řadě šlo o vyloučení mutací způsobujících změnu volné energie menší než 0,5 kcal/mol. Tento krok byl učiněn kvůli omezené přesnosti experimentální metody, s jejíž pomocí byla databáze ProTherm vytvořena. Hodnoty změn volné energie velmi blízké k nule proto mohou být chybně označeny jako *stabilizující* respektive *destabilizující*. Protože záznamů, pro něž by se podařilo získat predikce všech nástrojů, nebylo dostateční množství, byly vybrány všechny záznamy, pro které je výsledek dokázala predikovat změnu stability alespoň polovina nástrojů. Tato množina dat pak byla rozdělena v poměru 2:1 na trénovací a testovací datovou sadu takovým způsobem, že v obou byl stejný podíl stabilizujících a destabilizujících substitucí.

K usnadnění tvorby datových sad byl vytvořen skript *createTD.py*. Nejprve proběhne komunikace s databází *Stability*, ze které jsou získány všechny záznamy ze spojení všech tabulek nástrojů a tabulek *protherm* a *protherm_mutations* obsahujících referenční údaje. Z nich jsou pak vybírány ty, které mají definovanou hodnotu $\Delta\Delta G$ pro alespoň polovinu nástrojů a zároveň nemají u žádného nástroje s existujícím výsledkem predikce nastaven příznak *usedForTraining*. Vznikne sada použitelných záznamů, která je dále dozdělena podle znaménka u referenční hodnoty $\Delta\Delta G$ na seznam mutací **stabilizujících** ($\Delta\Delta G$ je záporná) a **destabilizujících** ($\Delta\Delta G$ je kladná). Oba seznamy jsou dále náhodně promíchány a z obou je každý třetí vybrán do testovací datové sady a zbylé dva do trénovací. Obě sady jsou nakonec ještě náhodně promíchány a vypsány do souborů *testSet.txt*, respektive *trainSet.txt*. Ty jsou tvořeny hodnotami $\Delta\Delta G$ oddělenými tabulátory v pořadí: *referenční hodnota, Eris, Rosetta, FoldX, iPTREE-STAB, mCSM, iMutant3.0, SDM, Auto-mute2, DUET*.

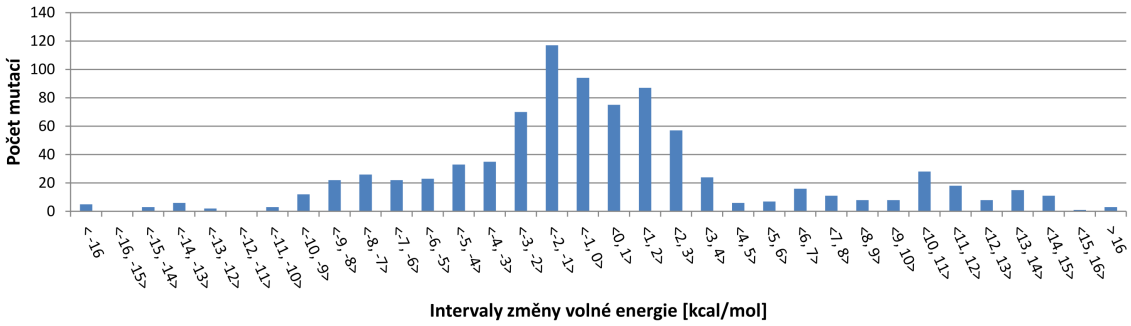
6.1.4 Statistiky

Výše popsaným způsobem bylo vybráno celkem 956 záznamů, z nich 642 připadlo do trénovací datové sady a 314 do testovací. Jednotlivé mutace pokrývají poměrně velkou část stavového prostoru a jsou v něm rovnoměrně rozloženy. Procentuální zastoupení konkrétních mutací v množině, ze kterých byly datové sady vybírány, je graficky shrnuto na obrázku 6.2. Z grafu je patrné, že velké množství mutací spočívalo v náhradě *kyseliny asparagové (D)*, *threoninu (T)* a *valinu (V)*, častou cílovou aminokyselinou pak byl *alanin (A)*.

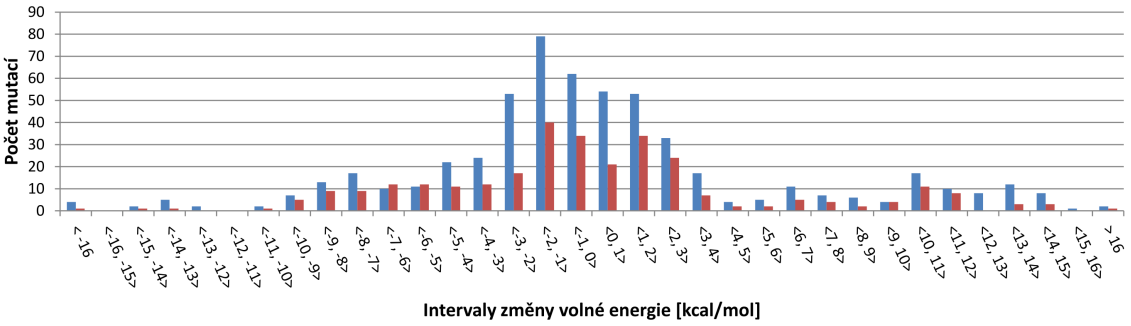
	A	C	D	E	F	G	H	I	K	L	M	N	P	Q	R	S	T	V	W	Y
A	0,00	0,32	0,11	0,11	0,11	0,54	0,11	0,00	0,11	0,11	0,00	0,00	0,11	0,00	0,64	0,11	0,21	0,96	0,00	0,00
C	1,18	0,00	0,00	0,00	0,00	0,11	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	1,18	1,71	0,11	0,00	0,00
D	0,86	0,00	0,00	0,11	0,32	0,11	0,43	0,21	2,46	0,21	0,00	1,61	0,11	0,00	0,64	0,21	0,43	0,00	0,21	0,11
E	2,03	0,00	0,00	0,00	0,11	0,21	0,00	0,00	6,00	0,00	0,00	0,00	0,11	5,14	0,00	0,21	0,00	0,21	0,00	0,00
F	0,11	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,11	1,61
G	0,64	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,11	0,11	0,00	0,21	0,21	0,00	0,00	0,00	0,00
H	0,21	0,00	0,00	0,11	0,00	0,43	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,11	0,00	0,00	0,21	0,00	0,00	2,78
I	0,43	0,00	0,00	0,64	0,75	0,32	0,00	0,00	0,64	0,11	0,00	0,00	0,00	0,00	0,21	0,11	0,86	0,86	0,00	0,00
K	0,00	0,11	0,00	3,85	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,11	0,00	3,43	0,32	0,00	0,00	0,00	0,11	0,43
L	0,32	0,21	0,00	0,00	1,50	0,21	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,96	0,00	0,43
M	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,11	0,00	0,00	0,00	0,00	0,00	0,43	0,00	0,00
N	1,18	0,11	0,86	0,00	0,00	0,11	0,11	0,96	1,07	0,11	0,21	0,00	0,00	0,00	0,00	0,54	0,43	0,21	0,00	0,00
P	1,07	0,00	0,00	0,00	0,00	0,11	0,00	0,00	0,00	0,32	0,00	0,00	0,00	0,00	0,00	0,21	0,00	0,00	0,00	0,00
Q	0,11	0,11	0,00	0,00	0,00	0,11	0,00	0,00	0,54	0,00	0,00	0,00	0,00	0,00	0,11	0,00	0,00	0,00	0,00	0,11
R	0,21	0,00	0,00	0,43	0,00	0,11	0,43	0,00	0,11	0,00	0,00	0,00	0,00	0,11	0,00	0,11	0,00	0,00	0,00	0,00
S	0,75	0,11	0,00	0,00	0,00	0,00	0,00	0,00	1,50	0,11	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
T	2,03	0,21	0,00	2,57	0,75	0,11	0,21	3,53	0,00	1,07	0,11	0,00	0,21	0,21	0,86	0,75	0,00	2,14	0,00	1,18
V	0,64	0,32	0,00	1,28	0,96	0,43	0,00	0,64	2,78	0,86	0,00	0,11	0,11	2,57	0,00	0,11	0,54	0,00	0,43	1,07
W	0,00	0,00	0,00	0,00	0,11	0,00	0,11	0,00	0,00	0,11	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,75
Y	0,00	0,11	0,00	0,00	1,61	0,21	0,11	0,00	0,00	0,00	0,00	0,00	0,11	0,43	0,00	0,00	0,00	0,00	0,43	0,00

Obrázek 6.2: Distribuce konkrétních mutací v datové sadě. Písmena u řádků definují původní aminokyseliny, sloupce pak odpovídají aminokyselinám, jimiž byly zdrojové nahrazeny.

Pohled na distribuci změn volné energie napříč sjednocením testovací a trénovací datové sady nabízí graf na obrázku 6.3. Rozložení má tvar podobný Gaussově křivce, z čehož lze vyčíst, že mutace způsobující malou změnu jsou v této množině záznamů zastoupeny častěji. Zároveň lze z distribuce hodnot odhadnout, že počty *stabilizujících* a *destabilizujících* mutací jsou přibližně vyrovnané, *stabilizující* však mírně převažují. Nižší sloupce grafu v intervalech $\langle -1; 0\rangle$ a $\langle 0; 1\rangle$ jsou způsobeny odstraněním mutací se změnou $\Delta\Delta G$ nižší, než 0,5 kcal/mol. Po náhodném rozdělení této množiny záznamů do trénovací a testovací datové sady zůstalo rozložení změn volné energie v obou podmnožinách v podstatě zachováno, jak ukazuje graf na obrázku 6.4.



Obrázek 6.3: Distribuce změn volné energie v datové sadě.



Obrázek 6.4: Distribuce změn volné energie po rozdělení na trénovací (modré sloupce) a testovací (červené sloupce) datovou sadu.

6.2 Implementace evoluční strategie

V rámci této práce byla evoluční strategie implementována v úloze hledání optimální konsenzu mezi zvolenými nástroji za účelem predikce změn stability proteinů. Ten je představován vektorem vah přiřazeným jednotlivým nástrojům. Změna stability je vyjádřena jako změna volné energie $\Delta\Delta G$. Výsledná hodnota konsenzuální metody je pak vypočítána jako kombinace predikcí těchto nástrojů podle vzorce:

$$\Delta\Delta G_{cons} = \frac{\sum_{i=1}^{n} p_i w_i}{\sum_{i=1}^{n} w_i}, \tag{6.1}$$

kde n je počet nástrojů, p_i je hodnota $\Delta\Delta G$ predikovaná nástrojem i pro danou mutaci a w_i je váha přiřazená nástroji i. Vypočtená změna stability by v ideálním případě měla být bližší k experimentálně získané hodnotě, než predikce provedená nejúspěšnějším samostatným nástrojem.

Evoluční strategie byla implementována ve třech variantách: základní verze s pravidlem 1/5, složitější evoluční strategie s autoevolucí řídících parametrů (autoevoluce typu 2) a nejpokročilejší strategie založenoá na adaptaci kovarianční matice (CMA-ES). Jedním z cílů práce je také ověřit, zda použití pokročilejší evoluční strategie povede k nalezení lepšího řešení.

Hledání vektoru vah i testování nalezeného řešení je implementováno ve skriptu *es.py*. Konkrétní parametry výpočtu lze nastavovat pomocí konfiguračního souboru *settings.conf*. Jde o tyto údaje:

- METHOD - varianta evoluční strategie (0 = pravidlo 1/5, 1 = autoevoluce typu 2 a 2 = CMA-ES),
- POP_SIZE - velikost populace,
- TOOLS_COUNT - počet použitých nástrojů,
- T_MAX - počet úseků evoluční stregie, po nichž dojde k úpravě řídících parametrů
- I_MAX - počet generací evoluční strategie v jednom takovém úseku (celkový počet iterací je pak roven $T_MAX \times I_MAX$),
- STD_DEV_MODIF - konstanta pro zmenšení mutačního kroku při autoevoluci typu 2
- STD_DEV_INC - konstanta pro zvětšení mutačního kroku při autoevoluci typu 2
- EPS - minimální hodnota mutačního kroku
- TRAINING - příznak, zda se skript spustí v režimu trénování (training = 1) nebo testování (training = 0),
- X_OVER - příznak, zda se při trénování evoluční strategie použije křížení (X_OVER = 1) či nikoliv (X_OVER = 0).

Všechny evoluční strategie byly implementovány podle principů popsaných v kapitole Evoluční strategie. Jako hodnota fitness k porovnání kvality jednotlivých kandidátních řešení byl použit Pearsonův korelační koeficient mezi vektorem referenčních hodnot $\Delta\Delta G$, které jsou součástí trénovací datové sady a vektoru odpovídajích výsledků $\Delta\Delta G$ vypočtených podle daného vektoru vah. Pearsonův korelační koeficient určuje korelaci neboli vzájemný vztah mezi dvěma veličinami. Může nabývat hodnot z intervalu $\langle -1, 1 \rangle$, přičemž hodnota -1 značí tzv. zcela nepřímou závislost, 1 naopak závislost zcela přímou. Čím více se hodnota Pearsonova korelačního koeficientu blíží k 1, tím jsou vektory více přímo závislé a tím lepší je i dané kandidátní řešení.

Výstupem trénování je vektor vah, který měl z celé populace v poslední generaci nejlepší hodnotu fitness. Pro snadnější další zpracování při testování nalezených řešení je připsán na jeden řádek bez popisků do jednoho ze tří výsledkových souborů *candidate_1_5.txt*, *candidate_autoevo_2.txt* nebo *candidate_cmaes.txt* podle toho, která evoluční strategie byla použita k jeho nalezení. Součástí výstupu skriptu je také soubor *generations_XY.txt*, kde XY je aktuální datum a čas. Do tohoto souboru jsou v každé generaci připsány na jeden řádek pořadí generace a hodnota fitness nejlepšího řešení v aktuální populaci. Z těchto dat lze vyčíst, jakým způsobem a jakou rychlostí se při evoluci zlepšuje nalezené řešení a je možné je použít k porovnání efektivity různých evolučních strategií.

Podstatou testovací části skriptu je ověřit úspěšnost nalezeného řešení na nezávislé datové sadě. Vstupem pro testování je soubor *testSet.txt* obsahující testovací záznamy a dále pak výsledky trénovací fáze, tedy soubory *candidate_1_5.txt*, *candidate_autoevo_2.txt* nebo *candidate_cmaes.txt*, v nichž jsou uložena nejkvalitnější kandidátní řešení. Je-li jich v některém souboru přítomno více, vybere testovací procedura podle hodnoty fitness to nejkvalitnější. Pro porovnání výkonnosti konsenzuálního řešení s úspěšností ostatních nástrojů jsou jako vstupy načteny také soubory se záznamy trénovací datové sady jednotlivých nástrojů. Ty jsou získány pomocí jednoduchého skriptu pracujícího s databází *Stability* a uloženy pod jménem *název nástroje_train.txt*. Úspěšnost každého nástroje je pak otestována postupně na jeho trénovací datové sadě (pokud byl nástroj trénován) na trénovací a testovací datové sadě vytvořené pro evoluční strategii a na sjednocení těchto dvou sad. Vzhledem k tomu, že z vytvořených datových sad jsou vyjmuty trénovací záznamy zvolených nástrojů, je očekávatelné, že jejich úspěšnost na těchto množinách bude o něco horší než na trénovací datové sadě. Zároveň lze předpokládat, že výsledky nástrojů budou velmi podobné na trénovací, testovací i sjednocené datové sadě evolučních strategií. Kromě Pearsonova korelačního koeficientu je v rámci hodnocení jednotlivých nástrojů počítán i počet tzv. *skutečně pozitivních* (anglicky true positive, TP), *skutečně negativních* (true negative, TN), *falešně pozitivních* (false positive, FP) a *falešně negativních* (false negative, FN) výsledků. Tyto metriky ohodnocují pouze schopnost nástroje určit správně efekt mutace, tedy zda je **stabilizující** nebo **destabilizující**. Jako pozitivní byl stanoven stabilizující efekt (tedy záporná hodnota změny volné energie $\Delta\Delta G$), jednotlivé metriky jsou pak definovány takto:

- skutečně pozitivní - počet záznamů, kde predikovaná i referenční hodnota $\Delta\Delta G$ jsou záporné,
- falešně pozitivní - počet záznamů, kde predikovaná hodnota $\Delta\Delta G$ je záporná, zatímco hodnota referenční je kladná,
- skutečně negativní - počet záznamů, kde predikovaná i referenční hodnota $\Delta\Delta G$ jsou kladné,

- falešně negativní - počet záznamů, kde predikovaná hodnota $\Delta\Delta G$ je kladná, zatímco hodnota referenční je záporná.

Veškeré metriky vypočítané v rámci testování jsou vypsány do souboru *metrics.txt*. Každý řádek odpovídá jednomu druhu metriky, hodnoty na řádcích jsou odděleny tabulátory. Po zkopírování obsahu souboru do tabulkového editoru MS Excel jsou v každém sloupci uvedeny všechy hodnoty pro konkrétní kombinaci nástroje a datové sady.

Kapitola 7

Experimenty a zhodnocení výsledků

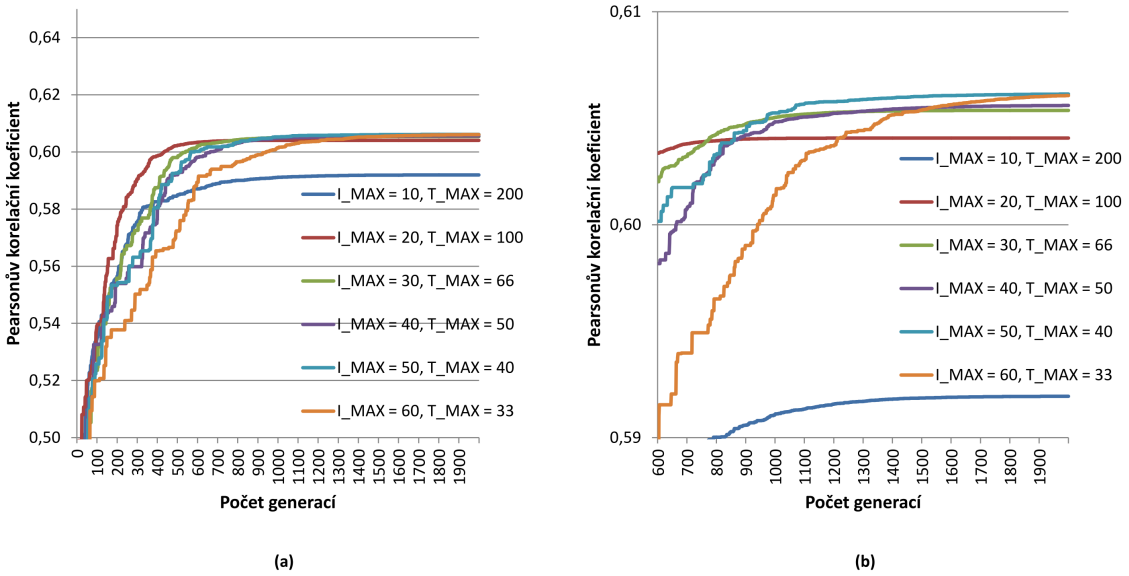
V následující kapitole jsou rozepsány experimenty provedené s evolučními strategiemi a následně rozebrány a zhodnoceny jejich výsledky. První část kapitoly je věnována srovnání jednotlivých parametrů, jimiž je možné ovlivňovat běh evolučních strategií a výběru optimálního nastavení algoritmu. Dále je popsán samotný průběh trénování evoluční strategie, rozebrána nalezená kandidátní řešení a diskutovány jejich výsledky na různých datových sadách v porovnání s jednotlivými nástroji.

7.1 Parametry a nastavení

První spuštění evolučních experimentů proběhla s cílem zjistit, jakým způsobem vstupní parametry strategie ovlivňují její průběh a rychlost konvergence k maximu, a na základě toho vybrat optimální nastavení, které povede k nalezení nejlepšího kandidátního řešení. Pro většinu proměnných lze v literatuře dohledat obecné optimální hodnoty. V těchto případech byla v támci experimentu evoluční strategie spuštěna pro srovnání i s hodnotami mírně pozměněnými pro ověření, zda budou referenční hodnoty nejlepší volbou i v této konkrétní aplikace ES.

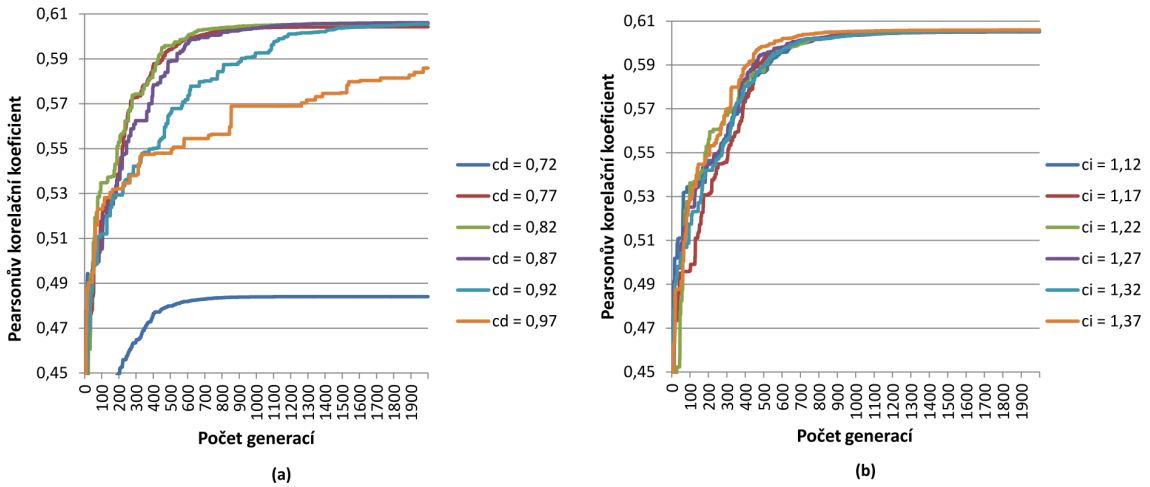
V případě evoluce s použitím pravidla 1/5 byly hledanými parametry pouze *I_MAX* počet generací v úsecích mezi jednotlivými adaptacemi mutačního kroku, *T_MAX* počet těchto úseků a konstanty c_d pro zmenšení a c_i pro zvětšení mutačního kroku. První dva parametry ovlivňují délku experimentu, celkový počet generací je určen součinem $T_MAX \times I_MAX$. Pro získání relevantního srovnání byla evoluční strategie spuštěna vždy s celkovým počtem 2 000 generací a měněn byl jen poměr mezi parametry *I_MAX* a *T_MAX*. Trénování každého poměru proběhlo pětkrát a výsledné hodnoty Pearsonova korelačního koeficientu byly zprůměrovány. Jako počáteční kandidátní řešení byl kvůli zachování stejných podmínek zvolen vektor se všemi prvky rovnými 0,5. Průběh růstu kvality řešení je znázorněn na obrázku 7.1 (a). V první části trénování se nejrychleji zvyšovala kvalita řešení při nastavení častějších adaptací mutačního kroku, tedy *I_MAX = 20, T_MAX = 100* (červená křivka) a *I_MAX = 10, T_MAX = 200* (tmavě modrá křivka). Lze to vysvětlit faktem, že v tomto stavu docházelo k častým adaptacím mutačního kroku po malém počtu iterací. Při detailnějším pohledu na druhou polovinu trénování (obrázek 7.1 (b)) je však patrné, že tato nastavení sice dosáhla poměrně rychle kvalitního řešení, ale přibližně od generace 700 se jejich kvalita zlepšovala pouze nepatrně. To bylo pravděpodobně způsobeno uváznutím

v lokálním maximu fitness funkce. V celkovém měřítku si podle grafu nejlépe počínala evoluční strategie s nastavením *I_MAX = 50, T_MAX = 40* a *I_MAX = 60, T_MAX = 33*. První jmenované nastavení navíc vedlo k nalezení vůbec nejkvalitnějšího řešení ze všech pokusů k těmto parametrům, hodnoty *I_MAX = 50, T_MAX = 40* byly proto použity dále jako nejlepší nastavení s tím, že při vyšším počtu iterací by bylo vhodnější zvýšit parametr *I_MAX* na úkor *T_MAX*.



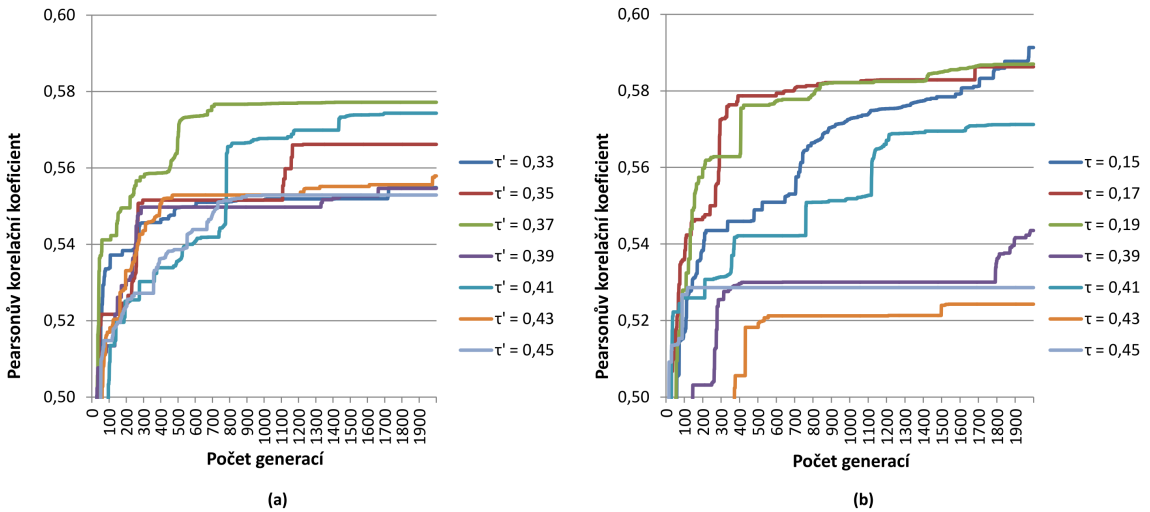
Obrázek 7.1: Závislost kvality nalezeného řešení na volbě parametrů T_MAX a I_MAX. Celkový pohled (a). Detailní pohled na výsledky trénování (b).

Obdobně proběhlo hledání parametrů c_i a c_d. Byly otestovány referenční hodnoty uvedené v literatuře ($c_i = 1,22$ a $c_d = 0,82$) [30] a pak několik hodnot menších a větších o násobky *0,05*. Průběh učení je zobrazen na obrázku 7.2 (a). Rozdíly mezi výsledky evoluční strategie s různým nastavením nebyly příliš velké. Experiment potvrdil, že s nastavením blízkým referenčním hodnotám evoluční strategie konverguje rychle a nalézá kvalitní řešení (zelená křivka), při vyšším počtu generací však vedou mírně vyšší hodnoty $c_d = 0,87$ a $c_d = 0,92$ (fialová a světle modrá křivka) k ještě lepšímu řešení. V případě parametru c_i (obrázek 7.2 (b)) měly všechny křivky velmi podobný průběh a výslednou kvalitu řešení i při velkém zvětšení, byla proto použita hodnota uvedená v literatuře (fialová křivka). Použité hodnoty tedy byly $c_i = 1,22$ a $c_d = 0,88$.



Obrázek 7.2: Závislost kvality nalezeného řešení na volbě parametrů pro modifikaci mutačního kroku. Volba parametru c_d (a). Volba parametru c_i (b).

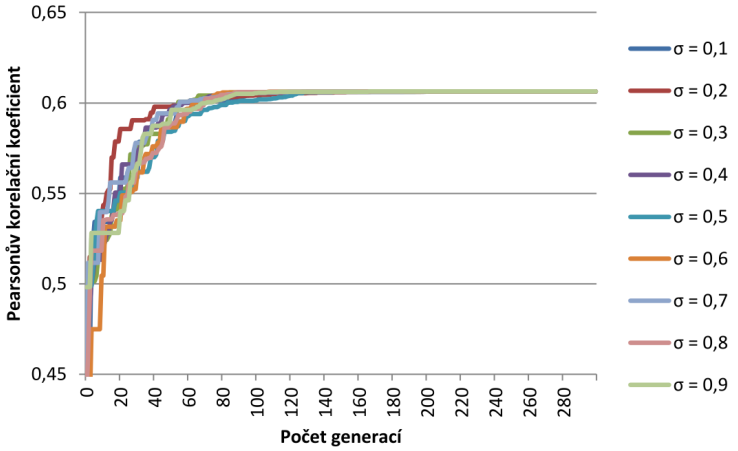
Evoluční strategie s autoevolucí typu 2 provádí narozdíl od pravidla 1/5 úpravu vektoru standardních odchylek v každé generaci před mutací samotného kandidátního řešení. Z toho důvodu nemá význam hledat optimální poměr mezi parametry I_MAX a T_MAX, kvalitu řešení ovlivňuje pouze celkový počet generací $gen_{max} = T_MAX \times I_MAX$. Experimentu však byly podrobeny společný parametr učení τ' a specifický parametr učení τ. Podle literatury by se jejich hodnota měla vypočítat podle počtu hledaných parametrů kandidátního řešení (rovnice pro výpočet jsou uvedeny v kapitole 5 v sekci Autoevoluce řídících parametrů). Pro deset hledaných vah jsou tyto hodnoty $\tau' = 0,39$ a $\tau = 0,21$. Zda jsou opravdu optimální bylo ověřeno nastavením parametrů na hodnoty blízké hodnotám vypočteným a následným porovnáním dosažené úspěšnosti. Grafy průběhu trénování pro různé hodnoty obou parametrů jsou na obrázku 7.3. Z křivek je patrné, že s referenčními hodnotami evoluční strategie kvalita predikce nejrychleji stoupá a zároveň dosahuje nejkvalitnějších řešení. Tyto hodnoty parametrů proto byly použity při dalších srovnávacích experimentech.



Obrázek 7.3: Závislost kvality nalezeného řešení na volbě společného a specifického parametru učení. Společný parametr učení (a). Specifický parametr učení (b).

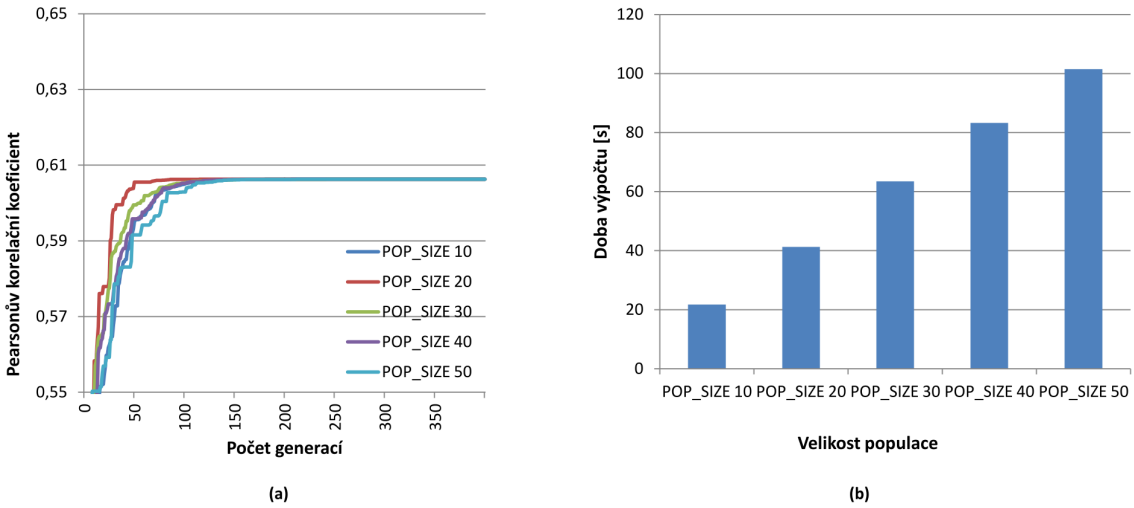
Strategie CMA-ES obsahuje velké množství parametrů používaných pro selekci, rekombinaci, adaptaci kovarianční matice a další výpočty. Většinu z nich není podle [14] vhodné měnit. V tomto článku jsou rovněž poskytnuty vzorce, jak určit jejich optimální hodnoty. Doporučeno je naopak přizpůsobit velikost mutačního kroku σ a počet jedinců v populaci λ. Podle velikosti populace se počítají hodnoty některých dalších parametrů a její zvětšení obvykle přináší obvykle přináší značné zvýšení kvality řešení za cenu snížené rychlosti konvergence. [14]

Nejprve byla strategie CMA-ES otestována s různou hodnotou parametru σ. Pro celkové srovnání byly otestovány hodnoty od *0,1* do *0,9* s přírůskem vždy *0,1*. Experiment ukázal, že počáteční hodnota parametru σ má na kvalitu nalezeného řešení a rychlost konvergence jen velmi malý vliv. I při detailním pohledu (obrázek 7.4) mají všechny křivky podobný tvar a po necelých 200 iteracích dospějí do stejného řešení s Pearsonovým korelačním koeficientem *0,606*. Při dalším testování byl tedy parametr σ nastaven na hodnotu *0,5*.



Obrázek 7.4: Závislost kvality nalezeného řešení na volbě parametru σ.

Nakonec byla strategie CMA-ES spouštěna s různou velikostí populace s cílem zjistit, zda a do jaké míry větší populace zlepší kvalitu nejlepšího řešení, jakým způsobem to ovlivní rychlost konvergence a jak se zvýší doba výpočtu. Z grafu na obrázku 7.5 (a) je patrné, že s menšími populacemi evoluční strategie skutečně konvergovala rychleji. K nalezení kvalitnějšího řešení však při zvětšení populace nedošlo. Zdá se tedy pravděpodobné, že získané výsledky jsou již velmi blízko maximu, kterého lze na této datové sadě dosáhnout. Vzhledem k tomu, že zvětšování populace způsobí značný nárůst doby výpočtu, byl tento parametr pro srovnání s ostatními nástroji nastaven na hodnotu 20, při které není doba výpočtu příliš dlouhá, evoluční strategie rychle konverguje a zároveň existuje dostatečný prostor pro diverzifikaci jedinců. Z obrázku 7.5 (b) je patrné, že doba výpočtu roste se zvětšující se populací přibližně lineárně.



Obrázek 7.5: Závislost kvality nalezeného řešení na volbě velikosti populace (a). Doba běhu evoluční strategie při různě velkých populacích (b).

7.2 Fáze trénování ES

V rámci trénování byla každá varianta evoluční strategie opakovaně spuštěna se zvolenými optimálními hodnotami parametrů. Z množiny výsledných kandidátních řešení nalezených v jednotlivých bězích bylo pro každou ES vybráno nejlepší na základě Pearsonova korelačního koeficientu, normalizované přesnosti a metriky MCC. Normalizovaná přesnost vychází z počtu *true positive*, *true negative*, *false positive* a *false negative* výsledků predikce vysvětlených v kapitole Implementace. Z těchto údajů je pak tzv. senzitivita a specificita podle vzorců

$$sensitivity = \frac{true_positive}{true_positive + false_negative}, \tag{7.1}$$

$$specificity = \frac{true_negative}{true_negative + false_positive}. \tag{7.2}$$

Normalizovaná přesnost je pak dána vztahem

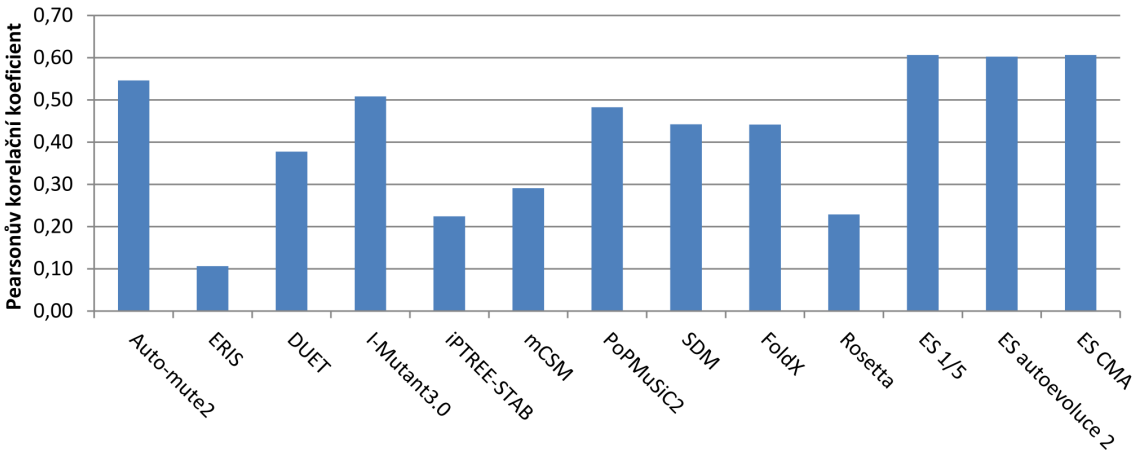
$$acc_norm = \frac{true_positive}{true_positive + true_negative}. \tag{7.3}$$

Posledním kritériem hodnocení kvality predikce byla metrika MCC (Matthew's correlation coeficient). Ta podobně jako normalizovaná přesnost vychází z počtů TN, TP, FN a FP mutací a lze ji vypočítat podle vzorce:

$$MCC = \frac{(TP \times TN) - (FN \times FP)}{\sqrt{(TP + FN) \times (TP + FP) \times (FN + TN) \times (TN + FP)}}. \tag{7.4}$$

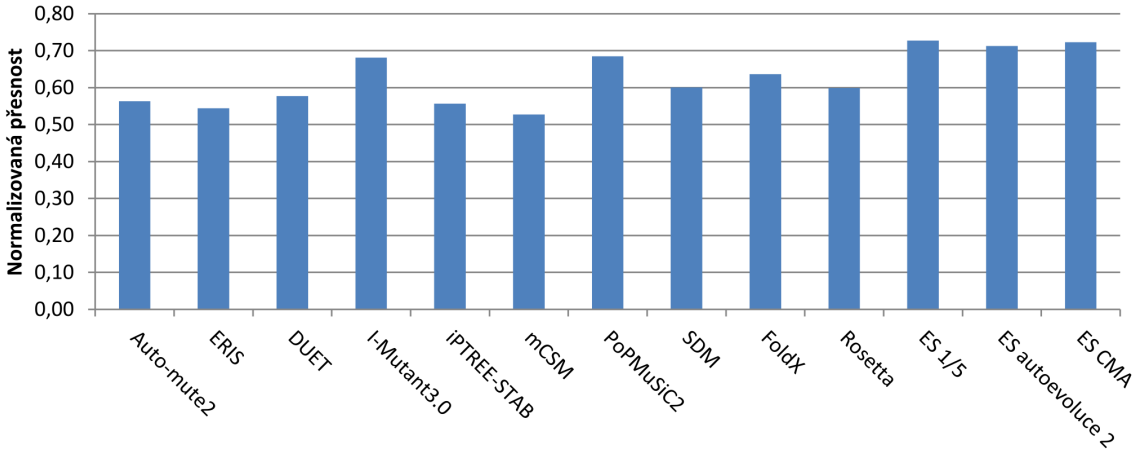
	eris	rosetta	foldX	iptree	mcsm	imutant3	sdm	automute	duet	popmusic
Pravidlo 1/5	-0,0046	0,0310	0,0159	0,0748	-0,0263	0,8519	0,2056	0,8175	-0,1317	-0,2832
Autoevoluce typu 2	-0,0099	0,1171	0,6644	2,1868	-2,3157	42,2336	10,7113	33,8586	-6,0239	-7,7748
CMA-ES	-0,2535	1,5671	1,3579	3,9588	-4,3502	45,3296	11,3158	42,5762	-2,9283	-17,6956
Pravidlo 1/5 (norm.)	-0,0030	0,0200	0,0102	0,0483	-0,0170	0,5493	0,1326	0,5271	-0,0849	-0,1826
Autoevoluce typu 2 (norm.)	-0,0001	0,0016	0,0090	0,0297	-0,0314	0,5735	0,1454	0,4597	-0,0818	-0,1056
CMA-ES (norm.)	-0,0031	0,0194	0,0168	0,0489	-0,0538	0,5605	0,1399	0,5264	-0,0362	-0,2188

Obrázek 7.6: Nejlepší nalezené řešení s pomocí jednotlivých variant evoluční strategie. Horní polovina tabulky uvádí vektory vah tak, jak byly vráceny evoluční strategií, v dolní polovině jsou pak hodnoty normalizovány tak, aby jejich součet v každém vektoru byl roven jedné.

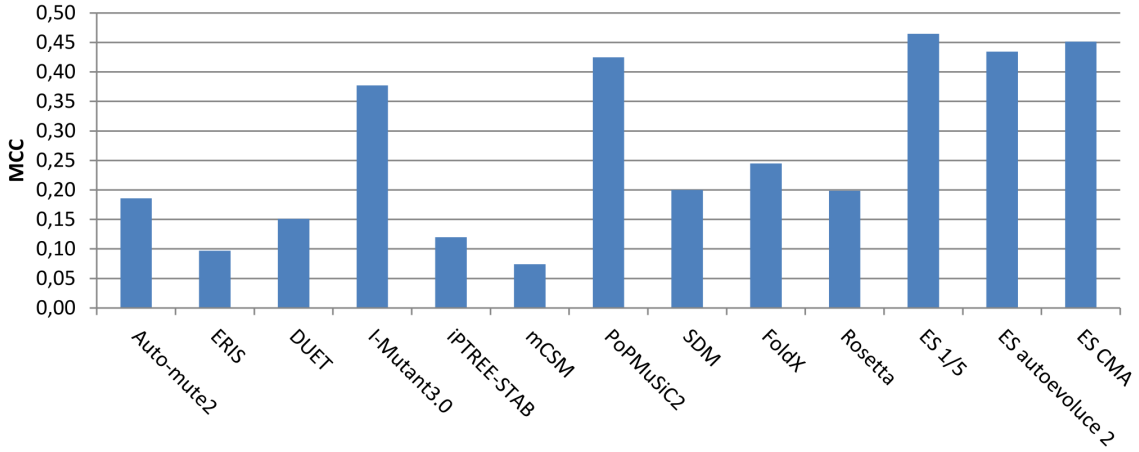


Obrázek 7.7: Srovnání přesnosti predikce jednotlivých nástrojů a metod s pomocí Pearsonova korelačního koeficientu.

Nejlepší nalezená řešení během trénování jednotlivých evolučních strategií jsou uvedena v tabulce na obrázku 7.6. Vysoká váha je zde přiřazena nástrojům Auto-mute2 a iMutant3.0, naopak překvapivě nízkou váhu má nástroj PoPMuSiC2, který sice dosahuje poměrně dobrých výsledků, ale trénovací datová sada obsahuje velké množství záznamů, které nemají výsledek predikce pro PoPMuSiC2 definován (nástroj nebyl schopen predikce provést). Evoluční strategie přirozeně tíhne k výběru nástrojů, které jsou vždy schopné výsledek vypočítat, protože v případě časté absence predikcí některého z nástrojů je narušena optimální rovnováha vah.. Srovnání predikčních nástrojů na základě Pearsonova korelačního koeficientu je na obrázku 7.7. Přesnost nástrojů byla vyhodnocena na sjednocení testovací a trénovací datové sady pro evoluční strategii. Z obou těchto sad jsou vyloučeny záznamy použité pro jejich trénování a obě mají stejné poměry stabilizujících a destabilizujících mutací; mají tedy z pohledu zmíněných prediktorů stejné vlastnosti. Z existujících nástrojů vykazují nejlepší výsledky Auto-mute2 a PoPMuSiC2. Oproti nim přináší konsezuální metoda očekávané zpřesnění predikcí. Při použití pravidla 1/5 jde o zvýšení Pearsonova korelačního koeficientu o 0,060, pro strategii s autoevolucí typu 2 je nárůst 0,057 a CMA-ES vedla ke zlepšení o 0,060 oproti nejlepšímu nástroji Auto-mute2.



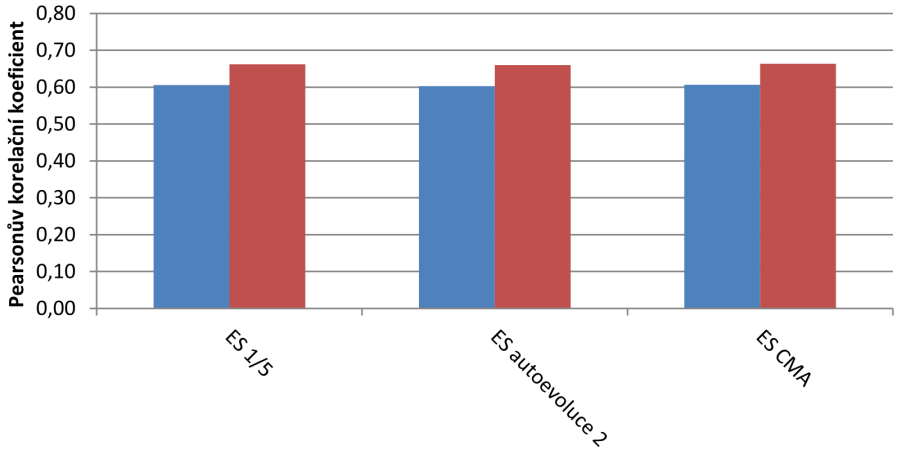
Obrázek 7.8: Srovnání přesnosti predikce jednotlivých nástrojů a metod evoluční strategie s pomocí normalizované přesnosti.



Obrázek 7.9: Srovnání přesnosti predikce jednotlivých nástrojů a metod evoluční strategie s pomocí metriky MCC.

Druhé srovnání proběhlo podle normalizované přesnosti, tři varianty evolučních strategií byly o 0,042 (pravidlo 1/5), 0,028 % (autoevoluce typu 2) a 0,038 % (CMA-ES) přesnější, než nejlepší PoPMuSiC2. Graficky je srovnání znázorněno na obrázku 7.8. Ve srovnání predikčních nástrojů podle metriky MCC (obrázek 7.9) zaznamenaly evoluční strategie menší nárůst úspěšnosti predikce, konkrétně varianta s autoevolucí typu 2 vykazovala zlepšení pouze o 0,01 oproti nástroji PoPMuSiC2. Tyto výsledky jsou pravděpodobně způsobené skutečností, že do trénovací a testovací datové sady jsou zahrnuty i záznamy, které nemají výsledky pro všechny použité nástroje. Při počítání úspěšnosti jejich predikcí na trénovací datové sadě evoluční strategie jsou záznamy bez hodnoty přeskakovány. Naproti tomu v případě trénování evoluční strategie jsou použity všechny záznamy, které obsahují výsledek predikce alespoň pro polovinu nástrojů (při použití pouze kompletních záznamů by byla trénovací a testovací datová sada příliš malá). Pokud chybí hodnoty nástrojů s vysokými přiřazenými váhami, změna stability je počítána z výsledků nástrojů dosahujících nižších přesností, čímž se značně snižuje hodnocení kvality predikcí evoluční strategie. Například nástroj Auto-mute2 vykazuje přesnost srovnatelnou s evolučními strategiemi, ale z datové sady 624 mutací vrátil výsledek pouze pro 196 záznamů, zatímco konsenzuální metodou

byla ohodnocena tato sada celá. Pro srovnání bylo provedeno vyhodnocení přesnosti evolučních strategií na podmnožině trénovací datové sady obsahující pouze mutace, pro které se podařilo získat predikce od všech nástrojů. Těchto záznamů bylo 173, což je nízký počet pro trénování evoluční strategie, bylo však možné na této podmnožině ohodnotit přesnost evolučních strategií. Pearsonův korelační koeficient byl v tomto případě vyšší oproti vyhodnocení na celé trénovací datové sadě, graficky je rozdíl znázorněn na obrázku 7.10.



Obrázek 7.10: Přesnost konsenzuální metody při vyhodnocení na celé trénovací datové sadě (modré sloupce) a na podmnožině záznamů pro něž existují predikce všech nástrojů (červené sloupce).

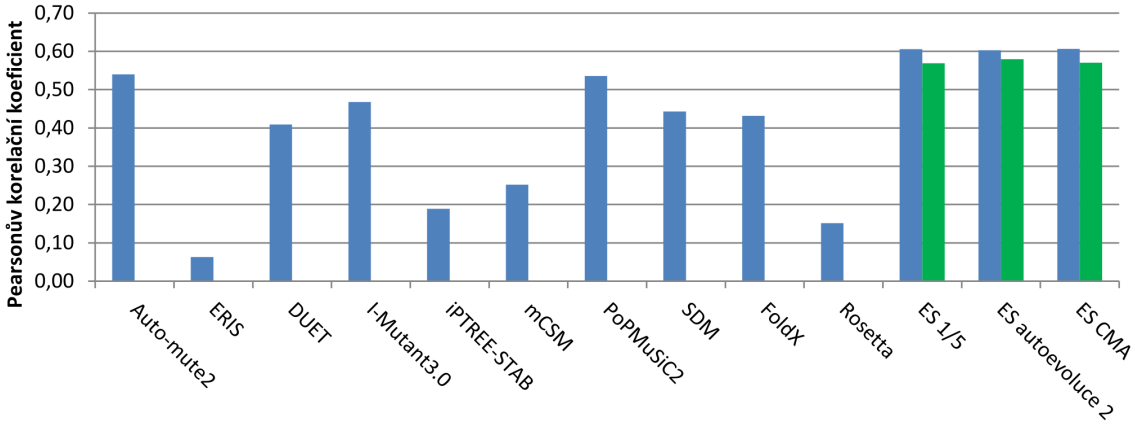
7.3 Fáze testování

Pro co nejrelevatnější výsledky proběhlo testování nalezených řešení na testovací datové sadě, která byla disjunktní s trénovací sadou a neměla také žádný překryv s trénovacími sadami jednotlivých nástrojů. Zároveň byly zachovány stejné poměry stabilizujících a destabilizujících mutací. Detaily výběru těchto sad shrnuje odpovídající podsekce kapitoly Implementace. Teprve otestování na takto nezávislé sadě poskytne skutečné srovnání s ostatními nástroji. Existující nástroje byly za účelem získání většího počtu záznamů opět testovány na sjednocení trénovací a testovací sady evoluční strategie (obě tyto množiny mají z pohledu použitých prediktorů totožné vlastnosti). Lze očekávat, že přínos konsenzuální metody oproti existujícím nástrojům bude nižší, než v případě trénovací datové sady. Ve fázi trénování totiž měla evoluční strategie vyšší přesnost částečně proto, že na dané datové sadě byla natrénována, zatímco trénovací datové sady použitých nástrojů s ní byly disjunktní.

Dále je třeba podotknout, že ačkoliv jsou pro srovnání přesnosti jednotlivých predikčních nástrojů a metod použity tři různé metriky (Pearsonův korelační koeficient, normalizovaná přesnost a Matthewův korelační koeficient), jako fitness hodnota pro trénování byla použita pouze první jmenovaná. Zbývající dvě metriky tedy mohou poskytnout odlišný pohled na přesnost konsenzuální metody.

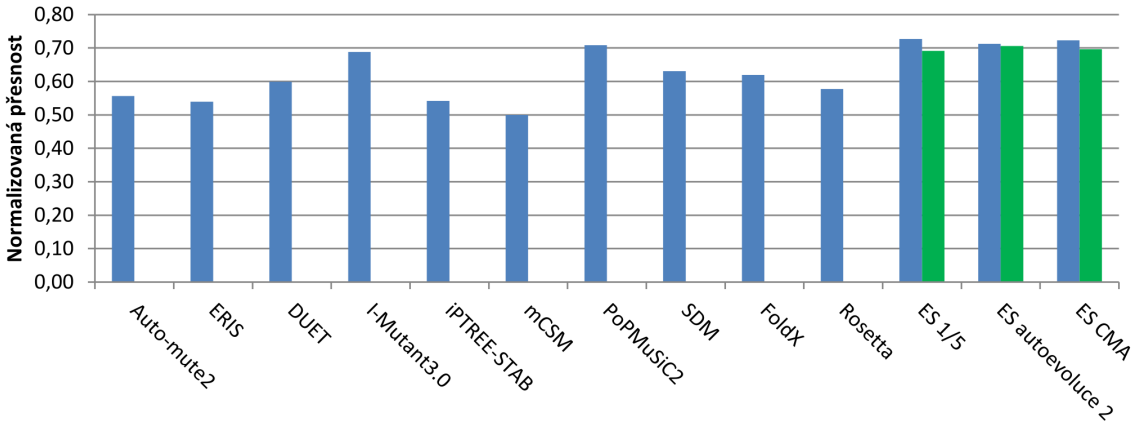
Při vyhodnocení přesnosti podle Pearsonova korelačního koeficientu vykazovaly evoluční strategie určitý přínos, rozdíl oproti existujícím nástrojům byl však podle očekávání nižší. Výsledky experimentu jsou znázorněny na obrázku 7.11. Narozdíl od fáze trénování dosahovala nyní nejvyšší přesnosti varianta s autoevolucí typu 2. Tato situace je vysvětlitelná; optimální vektory vah pro predikci nad trénovací a testovací datovou sadou jsou představovány dvěma body x_{tren} a x_{test} v n-rozměrném prostoru. Tyto body jsou blízko sebe, nikoliv

však ve stejném bodě. Evoluční strategie zpočátku přibližuje svá kandidátní řešení k bodu x_{tren} a tím pádem i k blízkému bodu x_{test}. Velmi blízko těmto bodům však může ES dospět do situace, kdy přiblížení k x_{tren} znamená vzdálení se od bodu x_{test}. V konečném důsledku pak může řešení, které je méně úspěšné na trénovací množině, vykazovat vyšší přesnost na množině testovací než řešení, která se na trénovací datové sadě jeví jako lepší.



Obrázek 7.11: Srovnání přesnosti predikce jednotlivých nástrojů a metod s pomocí Pearsonova korelačního koeficientu. Zelené sloupce označují přesnost konsenzuální metody na testovací datové sadě.

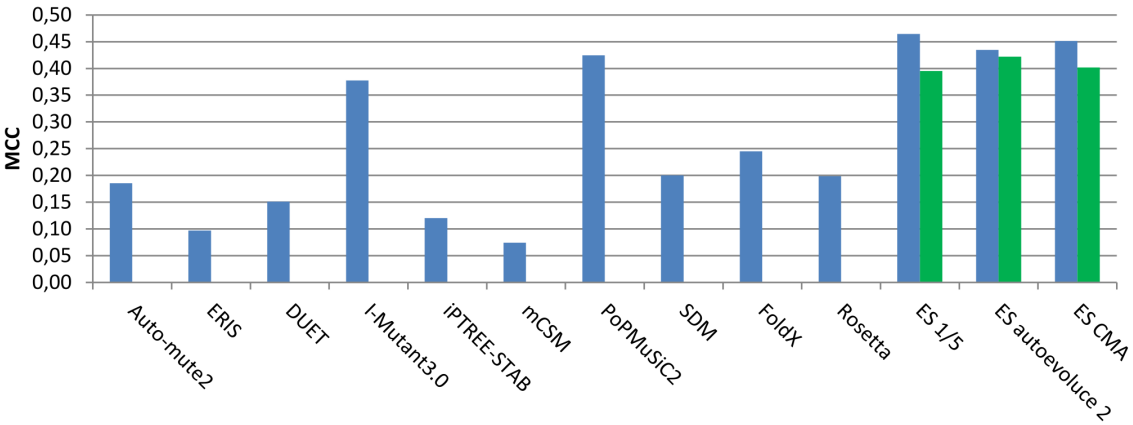
Srovnání predikčních nástrojů a variant evoluční strategie podle dosažené normalizované přesnosti je uvedeno na obrázku 7.12. Nejlepší výsledky má opět řešení nalezené ES s autoevolucí typu 2 ($acc_{norm} = 0,706$), jeho přesnost je vyšší, než nejpřesnější nástroj *PoPMuSiC2* ($acc_{norm} = 0,685$). Zároveň toto řešení vykazuje velmi malý pokles normalizované přesnosti oproti ohodnocení na trénovací datové sadě.



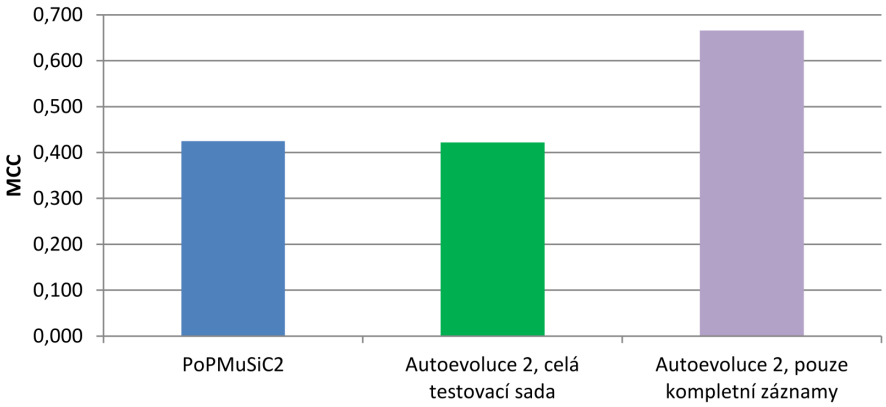
Obrázek 7.12: Srovnání přesnosti predikce jednotlivých nástrojů a metod s pomocí normalizované přesnosti. Zelené sloupce označují přesnost konsenzuální metody na testovací datové sadě.

Varianta evoluční strategie s autoevolucí typu 2 byla ze všech ES na testovací datové sadě nejúspěšnější i podle Matthewova korelačního koeficientu ($MCC = 0,422$), nedosáhla však již tak vysohé hodnoty, jako *PoPMuSiC2* ($MCC = 0,425$). Graf s výsledky tohoto experimentu je na obrázku 7.13. Opět platí, že vyhodnocená přesnost konsenzuální metody je oproti prediktorům degradována tím, že provádí predikce na celé testovací sadě, zatímco existující nástroje jsou testovány pouze na záznamech, které byly schopny ohodnotit. Zatímco konsenzuální prediktor ohodnotil všech 314 testovacích záznamů, zmíněný

PoPMuSiC2 provedl na této množině 155 predikcí. Srovnání, jak by se zvýši přínos ES, pokud by ohodnocovala pouze záznamy s existující predikcí $\Delta\Delta G$ od všech nástrojů, ukazuje graf na obrázku 7.14.

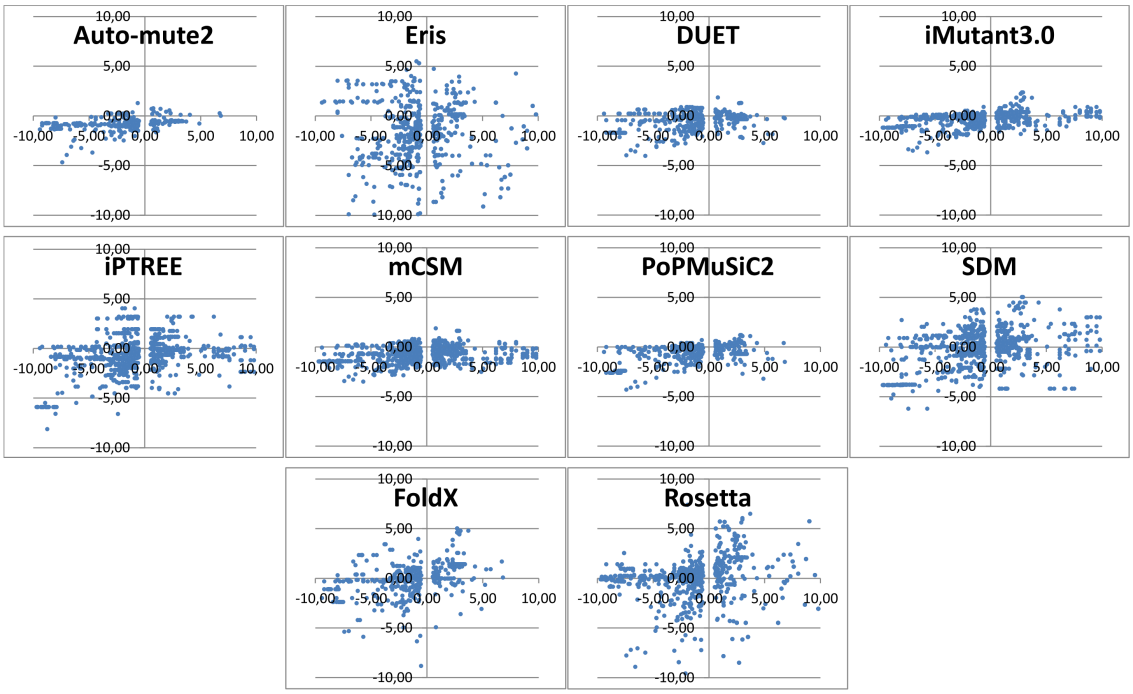


Obrázek 7.13: Srovnání přesnosti predikce jednotlivých nástrojů a metod s pomocí Matthewova korelačního koeficientu. Zelené sloupce označují přesnost konsenzuální metody na testovací datové sadě.



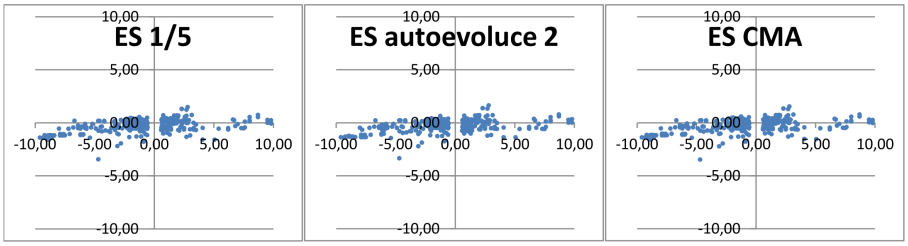
Obrázek 7.14: Srovnání přesnosti nástroje PoPMuSiC2 (modrý sloupec) a konsenzuální metody při vyhodnocení na celé testovací datové sadě (zelený sloupec) a na podmnožině záznamů pro něž existují predikce všech nástrojů (fialový sloupec).

Jiný pohled na úspěšnost predikcí jednotlivých nástrojů poskytují grafy na obrázku 7.15. Na vodorovnou osu jsou vynášeny hodnoty $\Delta\Delta G$ predikované daným nástrojem, na svislou osu pak referenční hodnoty získané z databáze ProTherm. Body v tomto prostoru odpovídají jednotlivým mutacím a z jejich polohy lze vyčíst úspěšnost predikce. Čím blíže jsou body k přímce $y = x$, tím je predikce nástroje přesnější. Dále lze z grafů zjistit informaci o tom, pro jaký typ mutací je který nástroj schopný provádět predikce a jak kvalitně. Nástroj *mCSM* například poměrně správně predikuje stabilizující mutace a přiřazuje jim záporné hodnoty $\Delta\Delta G$ (velké množství bodů ve třetím kvadrantu, vysoký počet *true positive*), zároveň však jako stabilizující často označuje i mutace destabilitující (body ve čtvrtém kvadrantu, *false positive*). O nástrojích *Auto-mute2* a *DUET* je možné říci, že pro destabilizující mutace častěji nejsou schopné predikci provést.



Obrázek 7.15: Srovnání jednotlivých nástrojů pomocí bodových grafů. Každý bod odpovídá mutaci, souřadnice X udává referenční hodnotu $\Delta\Delta G$ a souřadnice Y výsledek predikce.

Stejné grafy pro jednotlivé varianty evolučních strategií jsou na obrázku 7.16. tvar všech tří shluků bodů je podobný a je na nich patrná korelace. Pro skutečně přesné predikce by však bylo potřeba, aby tyto body více přiléhaly k ose prvního a třetího kvadrantu. Konsenzuální metoda do velké míry přejímá problém většiny nástrojů, že absolutní hodnoty predikovaných změn $\Delta\Delta G$ jsou obvykle mnohem nižší, než je tomu ve skutečnosti. Tuto skutečnost nelze z hodnoty Pearsonova korelačního koeficientu detekovat. Navzdory těmto nedostatkům však podává konsenzuální metoda obecně lepší výsledky, než samostatné nástroje. Ze své podstaty je závislá na schopnosti ostatních nástrojů provést svoji predikci pro danou mutaci, nicméně i v případě některých chybějících hodnot jsou její výsledky srovnatelné s již existujícími nástroji.



Obrázek 7.16: Srovnání jednotlivých nástrojů pomocí bodových grafů. Každý bod odpovídá mutaci, souřadnice X udává referenční hodnotu $\Delta\Delta G$ a souřadnice Y výsledek predikce.

Celkové numerické srovnání úspěšnosti evolučních strategií je v tabulce na obrázku 7.17. Lze z ní vyčíst, že řešení nalezená evoluční strategií vykazovala vysokou přesnost. Jejich predikce na podmnožině testovací datové sady, pro kterou se podařilo získat výsledky všech nástrojů, byly přesnější než predikce nejlepších stávajících nástrojů. Zároveň je však konsenzuální metoda velmi citlivá na chybějící výsledky jednotlivých využívaných prediktorů.

V případě celé testovací datové sady byla její přesnost značně nižší a podle normalizované přesnosti i horší, než přesnost nejlepšího nástroje.

	Nejlepší nástroj	ES 1/5 celá testovací sada	ES 1/5 pouze kompl. záznamy	ES Autoevol. 2 celá testovací sada	ES Autoevol. 2 pouze kompl. záznamy	ES CMA celá testovací sada	ES CMA pouze kompl. Záznamy
Pearsonův korel. koef.	0,540 (Auto-mute2)	0,569	0,651	0,580	**0,658**	0,570	0,649
Norm. přesnost	0,708 (PoPMuSiC2)	0,692	0,761	0,706	**0,801**	0,696	0,761
MCC	0,424 (Auto-mute2)	0,395	0,603	0,422	**0,666**	0,402	0,603

Obrázek 7.17: Celkové srovnání přesnosti konsenzuální metody na testovacím datasetu. Řádky odpovídají jednotlivým metrikám. V prvním sloupci je přesnost nejlepšího nástroje, v dalších dvojicích sloupců pak přesnost variant evoluční strategie vždy na celé trénovací sadě a na podmnožině záznamů, pro něž dokázaly predikci provést všechny nástroje.

Kapitola 8

Závěr

Hlavním cílem této práce byl návrh konsenzuálního nástroje pro predikci vlivu aminokyselinových mutací na stabilitu proteinů. Navrhovaný prediktor využívá výstupy existujících nástrojů, které kombinuje podle předem určeného vektoru vah přiřazeným jednotlivým nástrojům. Hledání optimálního váhového vektoru bylo provedeno s využitím evolučních strategií.

Prvním krokem byla volba použitých nástrojů. S cílem zajistit co největší univerzálnost navrhovaného prediktoru byl při jejich výběru kladen důraz na velkou různorodost technik a principů, na nichž jednotlivé nástroje pracují. Celkem bylo vybráno deset různých existujících prediktorů; jejich výčet se stručným popisem je uveden v kapitole Stabilita proteinů.

Jako druhý krok proběhla tvorba trénovací a testovací datové sady pro evoluční strategii. Základem byla databáze ProTherm. Pro záznamy pak byly získány predikce jednotlivých využívaných nástrojů. Část těchto dat byla převzata z diplomových prací Ing. Malinky [18] (převzaty výsledky 3 nástrojů) a Ing. Pavlíka [25] (převzaty výsledky 4 nástrojů). Anotace dalších 3 prediktorů byly doplněny v rámci této práce. Samotný výběr trénovací a testovací datové sady pak kvůli zachování dostatečné velikosti proběhl z podmnožiny záznamů, pro něž se podařilo získat predikce většiny nástrojů.

Evoluční strategie pro hledání optimálního vektoru vah byla implementována ve třech variantách: evoluční strategie s pravidlem 1/5, metoda s autoevolucí typu 2 a varianta CMA-ES. Pro ohodnocení přesnosti navržené konsenzuální metody byl použit Pearsonův korelační koeficient, normalizovaná přesnost a Matthewův korelační koeficient. Na testovací datové sadě se jako nejpřesnější ukázalo řešení nalezené s pomocí autoevoluce typu 2. Na množině záznamů, pro něž dokázaly všechny použité nástroje predikci provést, bylo dosaženo Pearsonova korelačního koeficientu 0,658, což je o 0,118 lepší výsledek, než přesnost dosažená nejlepším nástrojem (*Auto-mute2*).

Navržený prediktor je však velmi citlivý na absenci hodnot $\Delta\Delta G$ predikovaných jednotlivými nástroji. Na celé testovací datové sadě, ve které byly 2/3 záznamů z hlediska získaných predikcí nekompletních, dosáhla konsenzuální metoda Pearsonova korelačního koeficientu 0,580, což je jen o 0,040 lepší výsledek než jakého dosáhl nástroj *Auto-mute2*.

Literatura

[1] Back, T.: *Evolutionary algorithms in theory and practice.* Oxford: Oxford University Press, vyd. 1. vydání, 1996.

[2] Berman, H. M.; Battistuz, T.; Bhat, T. N.; aj.: The protein data bank. *Acta Crystallographica Section D Biological Crystallography*, ročník 58, č. 6, 2000: s. 899–907.

[3] Beyer, H.-G.: *The theory of evolution strategies.* Berlin: Springer, 2001.

[4] Capriotti, E.; Fariselli, P.; Casadio, R.; aj.: I-Mutant2.0: predicting stability changes upon mutation from the protein sequence or structure. *Nucleic Acids Research*, ročník 33, č. Web Server, 2005: s. 306–310.

[5] Capriotti, E.; Fariselli, P.; Rossi, I.; aj.: A three-state prediction of single point mutations on protein stability changes. *BMC Bioinformatics*, ročník 9, č. Suppl 2, 2008: s. W306–W310.

[6] Deb, K.: *Multi-objective optimization using evolutionary algorithms.* Chichester: John Wiley, 2001.

[7] Dehouck, Y.; Kwasigroch, J.; Gilis, D.; aj.: PoPMuSiC 2.1. *BMC Bioinformatics*, ročník 12, č. 1, 2011: s. 151–2096.

[8] Dostál, J.: *Biochemie pro bakaláře.* Brno: Masarykova univerzita, první vydání, 2003.

[9] Eiben, A. E.: *Introduction to evolutionary computing.* Berlin, Heidelberg: Springer Berlin Heidelberg, 2003.

[10] Flegr, J.: *Úvod do evoluční biologie.* Praha: Academia, první vydání, 2007.

[11] Gromiha, M.: *Protein bioinformatics.* New Delhi: Academic Press, 2010.

[12] Gromiha, M. M.; An, J.; Kono, H.; aj.: ProTherm: thermodynamic database for proteins and mutants. *Nucleic Acids Research*, ročník 27, č. 1, 1999: s. 286–288.

[13] Guerois, R.; Nielsen, J. E.; Serrano, L.; aj.: Predicting changes in the stability of proteins and protein complexes. *Journal of Molecular Biology*, ročník 320, č. 2, 2002: s. 369–387.

[14] Hansen, N.: The CMA evolution strategy: A tutorial. *Vu le*, ročník 29, 2005.

[15] Huang, L.-T.; Gromiha, M. M.; Ho, S.-Y.; aj.: IPTREE-STAB: interpretable decision tree based method for predicting protein stability changes upon mutations. *Bioinformatics*, ročník 23, č. 10, 2007: s. 1292–1293.

[16] Khan, S.; Vihinen, M.; Fryar, K. L.; aj.: Performance of protein stability predictors. *Human Mutation*, ročník 31, č. 6, 2010: s. 675–684.

[17] Mařík, V.; Štěpánková, O.; Lažanský, J.: *Umělá inteligence*. Praha: Academia, první vydání, 1993-2013.

[18] Malinka, F.: *Strojové učení v úloze predikce vlivu aminokyselinových mutací na stabilitu proteinu*. Diplomová práce, FIT VUT v Brně, 2014.

[19] Masso, M.; Vaisman, I. I.; Nielsen, J. E.; aj.: AUTO-MUTE 2.0. *Advances in Bioinformatics*, ročník 2014, č. 4, 2014: s. 1–7.

[20] Murray, R. K.: *Harper's illustrated biochemistry*. New York: Lange Medical Books/McGraw-Hill, 26 vydání, 2003, ISBN 00-713-8901-6.

[21] Nečas, O.: *Obecná biologie pro lékařské fakulty*. Praha: H&H, první vydání, 2000.

[22] Nelson, D. L.; Cox, M. M.: *Lehninger Principles of Biochemistry*. New York: W.H. Freeman, Čtvrté vydání, 2004.

[23] Nordlund, T. M.: *Quantitative understanding of biosystems*. Boca Raton: Taylor and Francis, 2011.

[24] Pace, C. N.; Fu, H.; Fryar, K. L.; aj.: Contribution of hydrophobic interactions to protein stability. *Journal of Molecular Biology*, ročník 408, č. 3, 2011: s. 514–528.

[25] Pavlík, D.: *Evoluční strategie v úloze predikce vlivu aminokyselinových mutací na stabilitu proteinu*. Diplomová práce, FIT VUT v Brně, 2014.

[26] Pires, D. E. V.; Ascher, D. B.; Blundell, T. L.: MCSM: predicting the effects of mutations in proteins using graph-based signatures. *Bioinformatics*, ročník 30, č. 3, 2013: s. 335–342.

[27] Pires, D. E. V.; Ascher, D. B.; Blundell, T. L.: DUET: a server for predicting effects of mutations on protein stability using an integrated computational approach. *Nucleic Acids Research*, ročník 42, č. 1, 2014-07-02: s. 314–319.

[28] Rechenberg, I.: *Evolutionsstrategie '94*. Stuttgart: Frommann-Holzboog, 1970.

[29] Rohl, C. A.; Strauss, C. E. M.; Misura, K. M. S.; aj.: Protein structure prediction using Rosetta. *Methods in enzymology*, ročník 383, č. 1, 2004: s. 66–93.

[30] Schwefel, H.-P.: *Numerical optimization of computer models*. New York: Wiley, 1981.

[31] Selkoe, D. J.: Folding proteins in fatal ways. *Nature*, ročník 426, č. 6968, 2003-12-18: s. 900–904.

[32] Vacík, J.: *Přehled středoškolské chemie*. Praha: SPN, první vydání, 1995.

[33] Worth, C. L.; Preissner, R.; Blundell, T. L.: SDM–a server for predicting effects of mutations on protein stability and malfunction. *Nucleic Acids Research*, ročník 39, č. suppl, 2011-06-29: s. W215–W222.

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

[35] Yu, X.; Gen, M.: *Introduction to evolutionary algorithms*. New York: Springer, 2010.

Příloha A

Obsah CD

Přiložené CD má následující strukturu:

/dataSets/*
složka obsahující všechny skripty a soubory spojené s kompletací databáze *Stability*
/dataSets/newRecords/*
složka obsahující skripty a soubory spojené s doplňováním predikcí nových nástrojů
/dataSets/newRecords/AutoMute2/*
složka, do které je třeba nainstalovat program Auto-mute2
/dataSets/newRecords/getMutations.py
skript pro stažení mutací z databáze *Stability* k ohodnocení predikčními nástroji
/dataSets/newRecords/mutations.xml
výstupní soubor skriptu *getMutations.py* obsahující záznamy se staženými mutacemi
/dataSets/newRecords/dp.py
skript pro získávání predikcí nástrojů *Auto-mute2*, *PoPMuSiC2* a *DUET* pro mutace uložené v souboru *mutations.xml*
/dataSets/newRecords/automute2.sql
ukázkový výstupní soubor skriptu *dp.py* s predikcemi nástroje *auto-mute2*
/dataSets/newRecords/automute2.sql
ukázkový výstupní soubor skriptu *dp.py* s predikcemi nástroje *auto-mute2*
/dataSets/newRecords/automute2.sql
ukázkový výstupní soubor skriptu *dp.py* s predikcemi nástroje *auto-mute2*
/dataSets/trainRecords/*
složka obsahující skripty a soubory spojené s označováním záznamů použitých pro trénování jednotlivých nástrojů
/dataSets/trainRecords/trainSet.py
skript pro označení trénovacích záznamů
/dataSets/trainRecords/inputTrainSets/*
složka obsahující soubory s dostupnými trénovacími sadami jednotlivých nástrojů
/dataSets/trainRecords/outputTrainSets/*
složka obsahující SQL soubory pro označení trénovacích záznamů v databázi *Stability*
/evolutionStrategy/*
složka obsahující skripty a soubory spojené s trénováním a testováním evoluční strategie
/evolutionStrategy/createTD.py

skript pro vytvoření souborů s trénovací a testovací datovou sadou

/evolutionStrategy/trainSet.txt
soubor s trénovací datovou sadou

/evolutionStrategy/testSet.txt
soubor s testovací datovou sadou

/evolutionStrategy/es.py
skript pro trénování evoluční strategie a testování nalezených řešení

/evolutionStrategy/settings.cfg
konfigurační soubor skriptu *es.py*

/evolutionStrategy/foundSolutions/*
složka s nalezenými kandidátními kandidátními řešeními

/evolutionStrategy/trainRecords/*
složka se soubory s trénovacími záznamy jednotlivých nástrojů (použití pro testování)

/evolutionStrategy/getTrainSets.py
skript pro stažení trénovacích záznamů z databáze

/evolutionStrategy/metrics.txt
soubor s výstupy testování

/text/dp.pdf
text diplomové práce

/text/src/*
složka se zdrojovými soubory textu diplomové práce

Příloha B

Schéma souboru *mutations.txt*

Skript *dp.py* očekává vstupní soubor *mutations.txt* v XML formátu s následující XSD strukturou.

```
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
    <xs:element name="mutations">
    <xs:complexType>
      <xs:all>
        <xs:element name="mutation"/>
        <xs:attribute name="uid"type="xs:integer"/>
        <xs:attribute name="pdbId"type="xs:string"/>
        <xs:attribute name="temp"type="xs:float"/>
        <xs:attribute name="ph"type="xs:float"/>
        <xs:attribute name="wild"type="xs:string"/>
        <xs:attribute name="mut"type="xs:string"/>
        <xs:attribute name="pos"type="xs:integer"/>
        <xs:attribute name="chain"type="xs:string"/>
      </xs:all>
      <xs:attribute name="duet"type="xs:boolean"/>
      <xs:attribute name="popmusic"type="xs:boolean"/>
      <xs:attribute name="automute"type="xs:boolean"/>
    </xs:complexType>
  </xs:element>
</xs:schema>
```