VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV INFORMAČNÍCH SYSTÉMŮ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF INFORMATION SYSTEMS

PREDIKTOR VLIVU AMINOKYSELINOVÝCH SUBSTITUCÍ NA FUNKCI PROTEINŮ

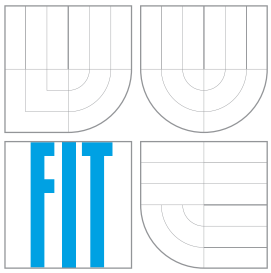
DIPLOMOVÁ PRÁCE
MASTER'S THESIS

AUTOR PRÁCE
AUTHOR

Bc. MILOŠ MUSIL

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

PREDIKTOR VLIVU AMINOKYSELINOVÝCH SUBSTITUCÍ NA FUNKCI PROTEINŮ

PREDICTOR OF THE EFFECT OF AMINO ACID SUBSTITUTIONS ON PROTEIN FUNCTION

DIPLOMOVÁ PRÁCE
MASTER'S THESIS

AUTOR PRÁCE
AUTHOR

Bc. MILOŠ MUSIL

VEDOUCÍ PRÁCE
SUPERVISOR

Ing. JAROSLAV BENDL

BRNO 2015

Abstrakt

Tato práce se zaobírá problematikou predikce škodlivosti aminokyselinových substitucí pomocí metody fylogenetické analýzy, inspirované nástrojem MAPP. Nezanedbatelné množství genetických onemocnění je způsobeno nesynonymními SNPs, projevujícími se jako jednobodové mutace na úrovni proteinů. Schopnost identifikovat tyto škodlivé substituce by mohla být užitečná v oblasti proteinového inženýrství pro testování, zda navržená mutace nepoškodí funkci proteinu a stejně tak k identifikaci choroby způsobujících škodlivých mutací. Experimentální ohodnocení navržených mutací je však nákladné a vyvstala tak potřeba pro predikci vlivu aminokyselinových substitucí počítačovými metodami. Tato práce popisuje návrh a implementaci nového predikčního nástroje, založeného na principech evoluční analýzy a studiu rozdílnosti fyzikálně-chemických vlastností mezi původní a substituovanou aminokyselinou. Vyvinutý algoritmus byl otestován na čtyř datasetech, čítajících celkem 74 192 mutací na 16 256 proteinových sekvencích. Prediktor dosáhl až 72% přesnosti a ve srovnání s většinou v současné době existujících nástrojů je jeho výpočet výrazně méně náročný na počítačový čas. Ve snaze dosáhnout maximální možnou efektivitu nástroje byl optimalizační proces zaměřen na výběr nejvhodnějších (a) nástrojů třetích stran, (b) rozhodovacího prahu a (c) sady fyzikálně-chemických vlastností.

Abstract

This thesis discusses the issue of predicting of the effect of amino acid substitutions on protein funkcion, based on phylogenetic analysis method, inspired by tool MAPP. Significant number of genetic diseases is caused by nonsynonymous SNPs manifested as single point mutations on the protein level. The ability to identify deleterious substitutions could be useful for protein engineering to test whether the proposed mutations do not damage protein function same as for targeting disease causing harmful mutations. However the experimental validation is costly and the need of predictive computation methods has risen. This thesis describes desing and implementation of a new in silico predictor based on the principles of evolutionary analysis and dissimilarity between original and substituting amino acid physico-chemical properties. Developed algorithm was tested on four datasets with 74,192 mutations from 16,256 sequences in total. The predictor yields up to 72% accuracy and in the comparison with the most existing tools, it is substantially less time consuming. In order to achieve the highest possible efficiency, the optimization process was focused on selection of the most suitable (a) third-party software for calculation of a multiple sequence alignment, (b) overall decision threshold and (c) a set of physico-chemical properties.

Klíčová slova

Aminokyselinové substituce, mutace, predikce škodlivosti mutací, fylogenetická analýza, proteinové inženýrství, MAPP.

Keywords

Amino acid substitution, mutations, prediction of the effect of amino acid substitutions, phylogenetic analysis, protein engineering, MAPP.

Citace

Miloš Musil: Prediktor vlivu aminokyselinových
substitucí na funkci proteinů, diplomová práce, Brno, FIT VUT v Brně, 2015

Prediktor vlivu aminokyselinových substitucí na funkci proteinů

Prohlášení

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

.....
Miloš Musil
22. května 2015

Poděkování

Děkuji panu Ing. Jaroslavu Bendlovi za odborné vedení práce a dále děkuji za přístup k výpočetním a úložným zařízením ve vlastnictví skupin a projektů, které přispívají k národní výpočetní síti MetaCentrum, provozované v rámci programu Projects of Large Infrastructure for Research, Development and Innovations.

Obsah

1 Úvod **3**

2 Proteiny **4**
2.1 Proteiny a jejich rozdělení 4
2.2 Aminokyseliny 5
2.3 Překlad DNA kódu do sekvence proteinu 5
2.4 Struktura proteinů 6
2.5 Mutace 7

3 Problematika hledání homologů **10**
3.1 Homologní sekvence 10
3.2 Konzervovanost aminokyselin 10
3.3 Databáze nr90 12
3.4 Nástroje pro vyhledávání homologních sekvencí 12

4 Zarovnání sekvencí **15**
4.1 Zarovnání sekvencí 15
4.2 Vícenásobné zarovnání 17
4.3 Vybrané nástroje pro zarovnání sekvencí 19

5 Fylogenetické stromy **21**
5.1 Fylogenetický strom 21
5.2 Metody založené na vzdálenosti 22
5.3 Metody založené na znacích 23
5.4 Metody založené na pravděpodobnosti 25
5.5 Nástroje pro konstrukci fylogenetického stromu 26

6 Predikce škodlivosti mutací **28**
6.1 Jednonukleotidové polymorfismy 28
6.2 Principy predikčních metod 29
6.3 Přehled existujících predikčních nástrojů 29

7 Výběr rysů **33**
7.1 Rozdělení metod 33
7.2 Extrakce rysů 33
7.3 Výběr rysů 35
7.4 Rozhodovací stromy 36
7.5 WEKA 36

8 Návrh a implementace **37**
8.1 Návrh predikčního nástroje 37
8.2 HotSpot Wizard 41
8.3 Použité datasety 42
8.4 Návrh experimentů 44
8.5 Databáze AAIndex 45

9 Výsledky **46**
9.1 Použité metriky 46
9.2 Základní testování 47
9.3 Výběr nástrojů třetích stran 48
9.4 Vliv zvoleného rozhodovacího prahu 50
9.5 Vliv sloupců bohatých na mezery 51
9.6 Výběr rysů 51
9.7 Srovnání s ostatními nástroji 53

10 Závěr **56**

A Výsledky základního testování **63**

B Vliv rozhodovacího prahu **65**

C Obsah DVD **66**

Kapitola 1

Úvod

Proteiny jsou základní složkou všech živých organismů, a tedy poznání jejich funkce je nezpochybnitelným přínosem pro mnoho oblastí výzkumu, jakým je vývoj nových léků, či určení léčby "na míru"pro konkrétního pacienta.

V sekvencích proteinů často dochází k tzv. mutacím, kdy je pro příklad jedna aminokyselina zaměněna za jinou. Tyto mutace jsou základem evoluce, ale stejně tak nesou zodpovědnost za řadu vážných onemocnění. Zkoumat důsledky všech aminokyselinových substitucí na funkci proteinu by však z hlediska finanční a časové náročnosti bylo experimentálně prakticky nemožné. Z tohoto důvodu vznikla řada nástrojů, snažících se o co nejpřesnější odhad důsledků těchto mutací pomocí počítačových metod.

Cílem této práce je vytvořit, otestovat a vyhodnotit nástroj pro predikci škodlivosti aminokyselinových substitucí, stavícím na principech fylogenetické analýzy, který bude dále označován jako RAPHYD (Rapid PHYlogenetic predictor of Deleteriousness).

V rámci kapitoly 2 poskytnu teoretický úvod o proteinech a aminokyselinách, ze kterých jsou tyto proteinové molekuly složeny. Dále popíši strukturu proteinů (primární, sekundární, terciární i kvartérní), typy mutací a základní dogma molekulární biologie. V kapitole 3 proberu problematiku vyhledávání homologních sekvencí a v kapitolách 4 a 5 pak po řadě techniky a nástroje pro tvorbu vícenásobného zarovnání a fylogenetických stromů. V kapitole 6 uvedu přehled nástrojů pro predikci škodlivosti aminokyselinových substiticí a zaměřím se na nástroj MAPP, který je základem vlastní implementace projektu. Závěrem teoretické části v kapitole 7 popíši několik metod pro výběr a extrakci rysů.

Praktická část práce je rozdělena na dvě kapitoly. V kapitole 8 popíši techniky uplatněné při vývoji nového predikčního nástroje, použité datasety a návrh experimentů. Kapitola 9 tuto práci uzavře přehledem výsledků, dosažených v základním testování, a jednotlivých krocích následujících optimalizací.

Kapitola 2

Proteiny

Proteiny (bílkoviny) jsou základním stavebním kamenem všech živých organismů, utvářejí hmotu buněk a určují její biochemické vlastnosti, stejně jako i většinu buněčných funkcí. Pochopení procesu vzniku proteinů a jejich účelu v buňkách tak nalézá široké uplatnění v medicíně, zemědělství, průmyslu a řadě dalších oborů aplikujících principy tzv. proteinového inženýrství. V rámci této kapitoly se zmíním o základním rozdělení proteinů, jejich struktuře, procesu translace z DNA kódu do proteinové molekuly a o důvodech vzniku a vlivu možných mutací.

2.1 Proteiny a jejich rozdělení

Proteiny jsou biopolymery, tvořené jedním nebo více polypeptidovými řetězci, které po ukončení procesu translace zaujaly energeticky nejvýhodnější prostorovou konformaci. Polypeptidové řetězce lze chápat jako sekvenci polymerů aminokyselin, spojených navzájem peptidovými vazbami a právě pořadím aminokyselin v řetězci a jejich chemickými vlastnostmi je určena trojrozměrná struktura proteinu a jeho funkce. Na základě funkce proteinu pak můžeme rozlišovat [18]:

- **Enzymy:** katalyzují rozpad a tvorby kovalentních vazeb a podílejí se na biochemických procesech v organismu. Příkladem může být pepsin, zodpovědný za trávení bílkovinné potravy v žaludku či RNA polymeráza podílející se na transkripci DNA molekuly do mRNA.
- **Strukturní proteiny:** jsou základní stavební jednotkou buněk a tkání. Příkladem je kolagen, tvořící základní stavební hmotu pojivových tkání, nebo keratin, jenž je základní složkou vlasů a nehtů.
- **Transportní proteiny:** jsou zodpovědné za přenos malých molekul a iontů v organismu. Protein hemoglobin slouží k transportu kyslíku z plic do tkání a v opačném směru k odvodu oxidu uhličitého. Dalším příkladem může být transferin zodpovídající za odvod železa.
- **Zásobní proteiny:** skladují molekuly nebo buňky v organismu. Ferritin je vnitrobuněčný protein, sloužící jako hlavní zásobní forma železa, které je jinak pro buňky samo o sobě toxické.
- **Pohybové proteiny:** vytvářejí sílu potřebnou k aktivnímu směrovanému pohybu v buňce. Příkladem takového proteinu je aktin či myosin.

- **Ochranné a obranné:** jsou součástí "samoopravné"schopnosti organismu. Protein imunoglobin je součástí imunitního systému a jeho účelem je identifikace a eliminace cizích objektů v těle (bakterie, viry, ...). Proteiny fibrin a fibrinogen se pak podílejí na procesu srážlivosti krve.

2.2 Aminokyseliny

Existuje celkem 20 standardních aminokyselin (nad těchto 20 lze zařadit navíc selenocystein, pyrolysin a N-formylmethionin), odvozených od organických kyselin, v nichž je na centrální α uhlík navázána aminová ($-NH_2$) a karboxylová (-COOH) funkční skupina. Jednotlivé aminokyseliny se pak od sebe odlišují postranními řetězci (R), které určují chemické vlastnosti aminokyselin a potažmo i z nich konstruovaných proteinů. Uvnitř proteinové molekuly jsou jednotlivé aminokyseliny vzájemně spojeny peptidovou vazbou, která propojuje vždy karboxylovou skupinu jedné aminokyseliny s amino skupinou druhé. Zřetězením více aminokyselin vzniká peptidový řetězec, zakončený aminovou a karboxylovou skupinou a zbytky aminokyselin vystupují jako postranní řetězce (R) z hlavní osy řetězce peptidového.

Na základě struktury aminokyselin a chemických vlastností jejich postranních řetězců lze aminokyseliny rozdělit do šesti základních skupin [25]:

- **Aminokyseliny s alifatickým postranním řetězcem:** glycin (Gly), alanin (Ala), valin (Val), leucin (Leu), isoleucin (Ile)
- **Kyselé skupiny s karboxylovou nebo aminovou skupinou na postranním řetězci**: kyselina asparagová (Asp), asparagin (Asn), kyselina glutamová (Glu), glutamin (Gln)
- **Bazické skupiny s aminovou skupinou na postranním řetězci:** arginin (Arg), lysin (Lys)
- **S aromatickým jádrem nebo hydroxylovou skupinou na postranním řetězci:** histidin (His), fenylalanin (Phe), serin (Ser), threonin (Thr), tyrosin (Tyr), tryptofan (Trp)
- **Se sírou v postranním řetězci:** methionin (Met), cystein (Cys)
- **Obsahující sekundární amin:** prolin (Pro)

2.3 Překlad DNA kódu do sekvence proteinu

Proces vzniku proteinu (tzv. proteosyntéza) z DNA lze rozdělit do dvou samostatných kroků: traskripce a translace. Grafický průběh konstrukce proteinu ze sekvence DNA je pak zachycen na obrázku 2.2.

- **Transkripce:** v průběhu transkripce dochází ke kopírování specifické části DNA kódující gen, do molekuly RNA, přičemž základní úlohu sehrává enzym RNA polymerázy, nasedající na oblast tzv. promotoru (signální oblast před začátkem genu). RNA polymeráza se posouvá po vlákně DNA rychlostí přibližně 100 nukleotidů za sekundu a zastavuje se na sekvenci tzv. terminátoru. Kopie daného úseku DNA se do RNA přenáší na základě komplementarity nukleových bází, kdy se puriny párují s pyrimidiny,

konkrétně guanin s cytosinem a adenin s uracilem, který v RNA nahrazuje thymin. Výsledná molekula je označovaná jako mediátorová RNA (mRNA).

- **Sestřih:** vyskytuje se pouze u eukaryot a představuje doplňující krok mezi transkripcí a translací. V případě eukaryot jsou geny složeny z kódujících segmentů (exony), které mohou být na více místech přerušeny segmenty nekódujícími (introny). V takovém případě dochází při transkripci nejprve k sestavení pre-mRNA, ze které následně sestřihem dojde k vyloučení nekódujících segmentů (vzniká mRNA). V rámci mechanismu sestřihu může být rovněž ze stejného transkriptu genů sestaveno několik různých variant mRNA. Tento efektivní způsob komprese genetické informace je označován jako alternativní sestřih a odhaduje se, že přibližně až 20 % lidských genů se vyznačuje touto vlastností.

- **Translace:** je procesem, kdy na základě matriční RNA dochází k vytvoření polypeptidového řetězce aminokyselin. Hlavní úlohu při translaci sehrávají tzv. ribozomy, které nasedají na vlákno mRNA a postupně čtou trojice nukleotidů (tzv. kodony), přičemž každou trojici překládají vždy do jedné z dvaceti aminokyselin. Jednotlivé aminokyseliny získává ribozom prostřednictvím transferové RNA (tRNA). Trojice nukleotidů umožňuje celkem 64 možných kombinací, z čehož logicky vyplývá, že jedna aminokyselina může být kódována více různými kombinacemi trojic (kód je tzv. degenerovaný). Tabulka překladů kodonů na aminokyseliny je znázorněna na obrázku 2.1. Výsledkem translace je polypeptidový řetězec - protein.

2.4 Struktura proteinů

Při popisu struktury proteinu lze rozlišovat čtyři různé úrovně organizace (obrázek 2.3) [18]:

- **Primární struktura:** je zápisem sekvence aminokyselin v polypeptidovém řetězci, definujícím protein.

- **Sekundární struktura:** zachycuje časté elementy, které na krátkých úsecích sekvence proteinu zaujímají velice podobné konformace. Jedná se především o α-helix a β-list. α-helix je uspořádáním, kdy se řetězec stáčí do šroubovice (přibližně 3,6 aminokyselin na závit) a tato konformace je stabilizována vodíkovými můstky mezi nad sebou ležícími peptidovými vazbami. V případě β-listu oproti tomu probíhají dva úseky řetězce paralelně vedle sebe a jsou stabilizovány vodíkovými můstky mezi sousedícími úseky.

- **Terciální struktura:** je trojrozměrným prostorovým uspořádáním polypeptidového řetězce. Na výsledné podobě terciální struktury mají zásluhu především chemické vlastnosti aminokyselin a jejich pořadí v řetězci. Například hydrofóbní aminokyseliny se vyhýbají kontaktu s vodou, a tudíž u nich vládne tendence shlukovat se uvnitř proteinové molekuly. Oproti tomu hydrofilní aminokyseliny se nacházejí blízko povrchu molekuly a vážou se vodíkovými můstky s molekulami vody a jinými látkami.

- **Kvartérní struktura:** popisuje uspořádání polypeptidových řetězců v proteinu. Kvartérní strukturou popisujeme pouze tzv. oligomerní proteiny, které jsou tvořeny více jak jedním polypeptidovým řetězcem.

	U	C	A	G	
U	UUU Phe UUC UUA Leu UUG	UCU UCC UCA Ser UCG	UAU Tyr UAC UAA Stop UAG Stop	UGU Cys UGC UGA Stop UGG Trp	U C A G
C	CUU CUC CUA Leu CUG	CCU CCC CCA Pro CCG	CAU His CAC CAA Gln CAG	CGU CGC CGA Arg CGG	U C A G
A	AUU AUC AUA Ile AUG Met	ACU ACC ACA Thr ACG	AAU Asn AAC AAA Lys AAG	AGU Ser AGC AGA Arg AGG	U C A G
G	GUU GUC GUA Val GUG	GCU GCC GCA Ala GCG	GAU Asp GAC GAA Glu GAG	GGU GGC GGA Gly GGG	U C A G

Obrázek 2.1: Tabulka popisující genetický kód. Aminokyseliny jsou uvedeny svými třípísmennými zkratkami. STOP značí kodon, signalizující ukončení translace [10].

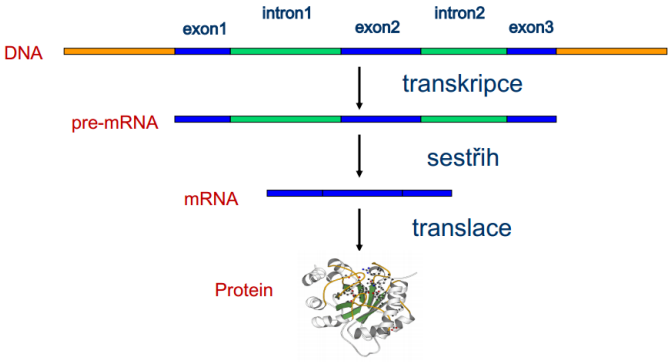
Trojrozměrná struktura proteinu má společně s chemickými vlastnostmi největší vliv na jeho funkci. Na povrchu proteinu, případně uvnitř tunelů dochází ke vzniku tzv. vazebních míst, díky kterým protein získává schopnost vázat se na specifickou malou množinu jiných molekul (ligandy). K vytvoření asociace s ligandem musí vazebné místo odpovídat tvarově i chemickým složením, a proto i malá změna v aminokyselinové sekvenci proteinu může narušit jeho trojrozměrnou strukturu a zamezit tak proteinu schopnost vázat se na ligand. Při predikci škodlivosti aminokyselinových mutací nás tak především zajímá, nakolik daná mutace ovlivní prostorové uspořádání proteinu a tím i jeho funkci.

2.5 Mutace

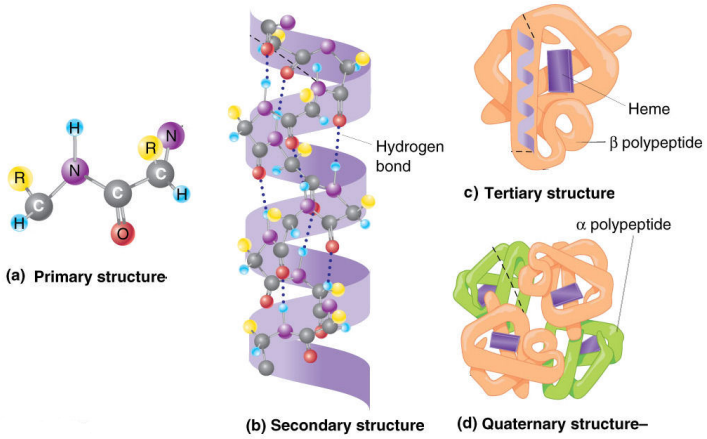
Mutace jsou náhodnými nebo cílenými změnami v DNA. Významným způsobem ovlivňují proces evoluce, která by bez mutací byla omezena na pouhou rekombinaci již stávajících genů, a do níž jsou formou mutací zanášeny nové varianty genů (geny znevýhodňující jedince jsou následně evolucí odstraněny, zatímco neutrální mutace se přenášejí do další generace). Jedná se tedy o veškeré obměny genetické informace, které nevznikly v důsledku segregace, či rekombinace již existujících genotypů.

Dle úrovně, na níž se mutace vyskytla, dále můžeme rozlišovat [38]:

- **Mutace genové:** mění informaci nesenou na úrovni genu (např. záměna nukleotidu v sekvenci DNA). Tento typ mutací je pro nás z hlediska predikce nejzásadnější.



Obrázek 2.2: Centrální dogma molekulární biologie. Převzato z [26].



Obrázek 2.3: Primární, sekundární, terciální a kvartérní struktura proteinu. Převzato z [62].

- **Mutace chromozomové:** změna struktury chromozomu.
- **Mutace genomové:** změna počtu chromozomů.

2.5.1 Typy mutací

Rozlišujeme tři základní druhy genových mutací [38]:

- **Substituce:** jedná se o záměnu jednoho nebo více nukleotidů za jiné, přičemž nedochází ke změně délky proteinu, tudíž ani k posunu čtecího rámce při transkripci/translaci. Substituce proto bývá obecně méně škodlivá, než inzerce či delece.
- **Inzerce:** vložení jednoho nebo více nukleotidů způsobující zvětšení délky původní sekvence. Významnost inzerce určuje především počet vložených nukleotidů. Jak bylo uvedeno v kapitole 2.3, ribozomy se při překladu nukleotidů na aminokyseliny posouvají po mRNA vlákně se čtecím okénkem o velikosti tři. Tři vložené nukleotidy tedy do struktury proteinu pouze přidají jednu novou aminokyselinu, zatímco jeden nebo dva přidané nukleotidy zapříčiní posun celého čtecího rámce.

- **Delece:** mutace analogická s inzercí, kdy je ze sekvence odebrán jeden nebo více nukleotidů a změněna její délka. Analogicky k inzercím, nejmenší změnu aminokyselinové sekvence způsobí odebrání tří nukleotidů (nebo násobků tří).

Vznik mutace však nemusí vždy znamenat ovlivnění funkce proteinu, či životaschopnosti organismu. Pouze malou část genetického kódu (u člověka cca. 1,5 % [31]) tvoří proteiny kódující geny, a tudíž k většině mutací dochází v nekódujících oblastech. I taková mutace však může ovlivnit regulaci tvorby určitého proteinu, případně tvorbu zastavit a zapříčinit vznik proteinů neúplné délky. Dojde-li k mutaci uvnitř kódující oblasti, můžeme dále rozlišovat mutace [38]:

- **Synonymní:** vychází ze skutečnosti, že genetický kód je tzv. degenerovaný. Záměny některých nukleotidů v kodonu mohou vést k translaci do identických aminokyselin a prostorové uspořádání proteinu tedy zůstává zachováno, jako by k mutaci nedošlo.
- **Nesynonymní:** opak výše uvedeného. Změnou nukleotidu v kodonu dochází ke změně aminokyseliny, a tudíž i ke změně konformace proteinu.
- **Nesmyslné:** mutace vytvářející STOP kodon a způsobující tak předčasné ukončení translace proteinu.
- **Posunové:** mění čtecí rámec, v důsledku i změnu aminokyselin a často vedou k předčasné identifikaci STOP kodonu a ukončení překladu proteinu.

2.5.2 Příčiny vzniku mutací

S ohledem na způsob jejich vzniku můžeme mutace rozdělit na mutace spontánní a indukované.

Spontánní mutace jsou takové, které vznikly chybou v replikačním a reparačním mechanismu DNA. Replikace DNA je velice přesná a předpokládá se výskyt přibližně jedné mutace na 10^7 nukleotidů. Replikace navíc podléhá samoopravným mechanizmům snižujícím chybovost na $1 : 10^9$ [7] a spontánní mutace jsou tak v organismu spíše ojedinělým jevem.

Indukované mutace jsou oproti tomu mutace vyvolané uměle nějakým vnějším mutagenem. Tyto mutageny můžeme dále rozlišit na:

- **Fyzikální:** působení ionizujícího (přerušuje kontinuitu vlákna) nebo ultrafialového záření, přičemž stupeň poškození DNA je přímo uměrný absorbované dávce záření.
- **Chemické:** chemické látky, které mohou narušovat DNA (tzv. genotoxiny), například její demetylací. Do této kategorie spadají nejrůznější oxidační, či alkylační činidla.
- **Biologické:** vzniklé v důsledku působení onkogenních virů.

Kapitola 3

Problematika hledání homologů

Řada algoritmů pro predikci škodlivosti aminokyselinových mutací na funkci proteinu využívá technik tzv. fylogenetické analýzy prováděné nad fylogenetickým stromem. Konstrukce stromu vyžaduje sestavení vícenásobného zarovnání nad podobnými (homologními) sekvencemi. V této kapitole se budu zaobírat metodami výběru homologů a zmíním i jejich význam pro fylogenetickou analýzu jako takovou.

3.1 Homologní sekvence

Homologní sekvence jsou takové sekvence, které vycházejí ze stejného zdroje (společného předka), avšak v průběhu evoluce od sebe byly odděleny do dvou nebo více vývojových větví. Lze předpokládat, že každá taková vývojová větev bude zpočátku obsahovat úseky DNA kódující proteiny se stejnou nebo velice podobnou funkcí, a tedy zákonitě i podobnou aminokyselinovou sekvencí, která se vlivem dalších mutací v následujících generacích bude svou rozdílností stále více vzdalovat. Z tohoto důvodu nás v rámci analýzy zajímá rovněž i konstrukce fylogenetického stromu.

Je možné dále rozlišovat tři základní pojmy [22]:

- **Homologní sekvence:** jsou sekvence sdílející stejného předka (ancestrální gen), přestože mohou být rozdílné a nacházet se v genomech různých druhů. Homologní sekvence lze dále dělit na sekvence paralogní a ortologní.
- **Paralogní sekvence:** jsou sekvence, povětšinou se nacházející na dvou rozdílných místech v genomu stejného druhu (vznikají duplikací ancestrálního genu), kde mohou plnit jiné, leč podobné funkce v odlišném kontextu.
- **Ortologní sekvence:** jsou sekvence vzniklé během procesu speciace (dělení druhu), v důsledku čehož se sekvence vyskytuje v genech dvou rozdílných druhů.

3.2 Konzervovanost aminokyselin

Jedním z hlavních důvodů vyhledávání homologních sekvencí je zjišťování tzv. konzervovanosti segmentů a aminokyselin v rámci nalezené skupiny homologů. Ta je základem pro funkci prediktorů, pracujících na bázi fylogenetické analýzy.

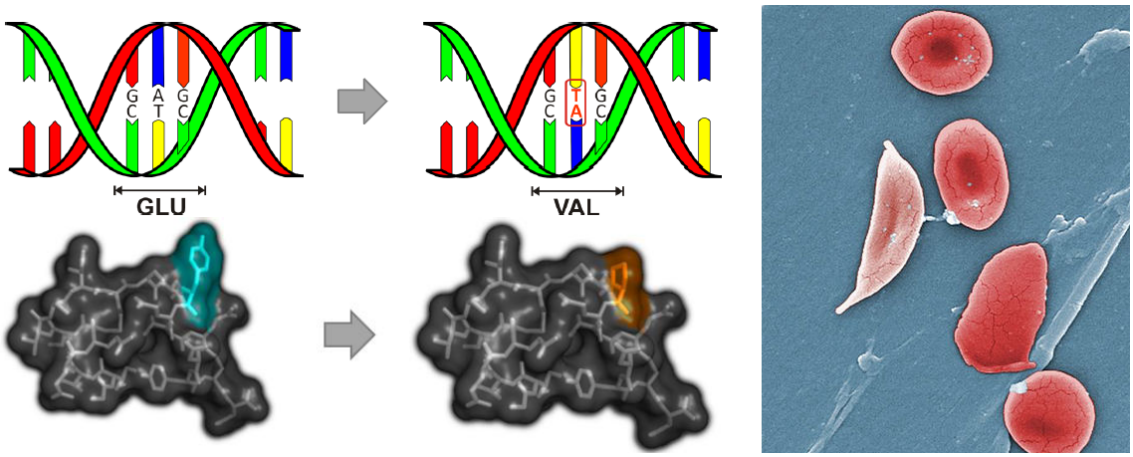
Míru konzervovanosti aminokyselin v rámci sekvence vybraného genu můžeme určit z vícenásobného zarovnání homologních sekvencí a jim příslušejícího fylogenetického stromu.

Zarovnáme-li sekvence pod sebe, můžeme následně snadno vyčíst, na kterých pozicích dochází nejčastěji k obměnám, a kde se naopak aminokyseliny příliš často nemění. Pozice, u kterých dochází k obměnám pouze v malé míře, jsou pak označovány za konzervované.

Ne všem konzervovaným sekvencím napříč zarovnáním však lze přikládat stejnou váhu. U sekvencí (např. u proteinů), které jsou od sebe více evolučně vzdálené, lze očekávat větší výskyty obměn v genetickém kódu, a tedy konzervované segmenty sobě blízkých sekvencí nemají stejnou vypovídající hodnotu jako u těch evolučně vzdálených. Informaci potřebnou k ohodnocení vícenásobného zarovnání lze odvodit z fylogenetického stromu.

Ve vztahu k predikci škodlivosti aminokyselinových substitucí nás konzervovanost segmentů zajímá především s ohledem na evoluční teorii, na základě níž lze předpokládat, že vysoce konzervované sekvence jsou ve své podobě pro přežití organismu významné, a tudíž mutace v této části genetického kódu zpravidla vedly k úmrtí, či znevýhodnění jedince a jeho následnému vyřazení z procesu evoluce (takto mutovaný gen se nepřenesl do další generace).

Posledním zvažovatelným faktorem konzervovanosti aminokyselin je pak jejich záměna v rámci jistých aminokyselinových skupin, kdy záměna jedné aminokyseliny za jí fyzikálně a chemicky podobnou (např. leucin a isoleucin) nebude mít zákonitě tak značný dopad na výsledný protein jako záměna za aminokyselinu skupin odlišných (např. glycin a tryptofan). Jednoduchým příkladem může být onemocnění zvané srpkovitá anémie, kde na šesté pozici β-řetězce hemoglobinu dochází k mutaci hydrofilní glutamové kyseliny (Glu) za hydrofobní valin, snažící se z povrchu skrýt do jádra proteinu a způsobující tak deformaci tvaru červených krvinek. Při deformaci dochází k promáčknutí krvinek do podoby protáhlých srpků, blokujících vlásečnice, čímž omezují průtok krve a mohou tak zapříčinit i závažné poškození orgánů a tkání. Podobné škodlivé mutace jsou z procesu evoluce většinou vyloučeny, zatímco méně závažné mutace (záměna hydrofilní aminokyseliny za hydrofilní) se teoreticky mohou šířit do dalších generací. Konkrétně mutace zapříčiňující srpkovitou anémii činí heterozygota odolného vůči malárii a tedy se i přes znevýhodnění jedince dodnes vyskytuje v subtropických oblastech. Mutace způsobující srpkovitou anémii je podrobněji zachycena na obrázku 3.1. Druhým příkladem může být cystická fibróza, postihující převážně dýchací a trávící soustavu. Toto onemocnění je způsobeno mutací genu, produkujícího protein CFTR, uloženého na dlouhém raménku sedmého chromozomu.



Obrázek 3.1: Mutace, způsobující dědičné onemocnění srkovitou anémií. Na pravé straně jsou znázorněny zdravé (kulaté) a poškozené (protáhlé) buňky červených krvinek. Převzato z [60].

3.3 Databáze nr90

Současné databáze proteinových a DNA sekvencí se vyznačují značnou redundancí a řada proteinů se tak vyskytuje v mnoha instancích, zaevidovaných například různými výzkumnými týmy, či v rámci několika různých experimentů. Mezi těmito záznamy zpravidla neplatí 100% shoda, což může být důsledkem chyb použitých experimentálních metod. Vybrat do seznamu homologů více instancí stejného proteinu by mohlo mít negativní dopad na vypovídající hodnotu nalezených konzervovaných segmentů a tím i na přesnost predikce, postavené na bázi fylogenetické analýzy. Stejně tak lze očekávat vyšší úspěšnost predikce, jestliže před paralogními sekvencemi upřednostníme ty více vzdálené (přesto stále vycházející ze stejného ancestrálního genu) sekvence ortologní. Je však zapotřebí brát v potaz, že takto vybrané ortologní sekvence již nemusejí v organismu plnit stejnou funkci, a tudíž identická mutace ve dvou ortologních sekvencích se u jedné může projevovat jako škodlivá, zatímco u druhé jako neutrální.

Databáze nr90 [59] uchovává filtrovanou podmnožinu reprezentativních sekvencí se vzájemnou shodou maximálně 90 %, díky čemuž řeší výše uvedené problémy a navíc šetří výpočetní čas redukcí vstupní databáze pro vyhledávání homologů.

3.4 Nástroje pro vyhledávání homologních sekvencí

3.4.1 BLAST

BLAST (Basic Local Aligment Search Tool [70]) je jedním ze základních algoritmů pro vyhledávání biologických sekvencí na základě jejich podobnosti. Ke své funkci využívá heuristický přístup, blízký metodě lokálního zarovnání Smith-Waterman [75], který je však pro prohledávání rozsáhlých genomových databází příliš pomalý. Samotný běh algoritmu lze rozdělit do tří hlavních fází:

- **Osévání (Seeding):**
  - Hledaná sekvence je rozdělena do slov o určité velikosti (běžně W = {2, 3} pro proteiny a $W > 6$ pro DNA/RNA). Rozdělení je provedeno skrze posuvné okénko, které je postupně posouváno po jednom znaku zleva doprava.
  - Ke každému slovu je následně vyhledána množina alternativních slov a k těmto alternativním slovům je vypočteno skóre podobnosti vzhledem k původnímu slovu (např. pomocí matice BLOSUM - viz 4.1.1). Slova se dle jejich skóre porovnají s prahovou hodnotou a pouze slova se skóre podobnosti vyšším, než je uživatelem zadaný práh (u proteinů běžně $T > 9$), jsou ponechány v tabulce podobností.
  - Slova z tabulky podobnosti jsou závěrem vyhledávána v druhé sekvenci (z databáze).
- **Rozšiřování (Extension):**
  - Nalezne-li algoritmus některé slovo z tabulky podobností, snaží se nalezený úsek rozšiřovat na obě strany.
  - Jestliže skóre vzroste nad určitou požadovanou hodnotu S, bude předáno na výstup. Naopak klesne-li skóre pod nastavený práh X, rozšiřování bude ukončeno a na výstup se uloží pozice nalezeného maxima.

  - Výsledkem jsou nalezené úseky s nejvyšším dosaženým skóre.

- **Ohodnocení (Evaluation):**
  - Ze seznamu nalezených segmentů jsou odstraněny ty s nízkou statistickou významností (tento rozhodovací práh bude dále označován jako e-value).
  - Vyhledají se skupiny konzistentních úseků, které na sebe navazují a je tudíž možné je spojit do většího celku.

Metodologie BLAST existuje v řadě různých variant pro vyhledávání proteinů vůči proteinům, nukleových kyselin vůči nukleovým kyselinám, proteinů vůči nukleovým kyselinám apod. (BLASTN, BLASTP, BLASTX, TBLASTN, TBLASTX).

3.4.2 PSI-BLAST

Metoda, pracující na rozdíl od nástroje BLAST ve více iteracích, přičemž první iterace je vždy shodná s výstupem nástroje BLAST. Po ukončení první iterace je vytvořena profilová sekvence, zastupující všechny nalezené sekvence. V dalších iteracích se vyhledává na základě této profilové sekvence (namísto původní, uživatelem zadané) a po skončení každé iterace je tato profilová sekvence znovu upravena dle v iteraci nalezených sekvencí. Výhodou iterativní metody PSI-BLAST [70] je skutečnost, že nástroj zvládne nalézt vzdáleněji příbuzné sekvence, než jaké by byly výstupem přímého podobnostního vyhledávání nástroje BLAST.

3.4.3 FASTA

Podobně jako v předchozí sekci popsaný přístup BLAST pracuje také i FASTA [88] nad algoritmem vyhledávání Smith-Waterman a podobně lze algoritmus rozdělit do čtyř kroků:

- Nalezení identických segmentů
- Výpočet jejich skóre např. pomocí matice BLOSUM (viz 4.1.1)
- Spojení segmentů do delších úseků
- Výpočet optimálního skóre

Rozdíl mezi algoritmy FASTA a BLAST je především ten, že FASTA v počáteční fázi toleruje pouze zcela identické segmenty, zatímco BLAST uvažuje i segmenty obsahující záměny znaků. Dále rozdílem od algoritmu BLAST, snažícího se rozšiřovat nalezené úseky slov, FASTA propojuje nalezené identické úseky. Díky tomu je FASTA citlivější a poskytuje přesnější výsledky, ale pochopitelně se vyznačuje vyšší výpočetní náročností.

3.4.4 HMMER

Algoritmus balíku HHMER [65] využívá k analýze a vyhledávání metodu založenou na porovnávání profilů skrytých Markovových modelů, konstruovaných na základě zarovnání sekvencí. Podobně jako nástroj PSI-BLAST patří do skupiny sequence-profile algoritmů. Novější verze využívají pro zvýšení výkonu heuristické filtry k hledání vysoce hodnocených shod bez mezer v sekvenčních databázích. Rychlost metody HMMER je srovnatelná s metodou BLAST, bez výrazných negativních vlivů na přesnost vyhledávání.

3.4.5 BLAT

BLAT [85] je relativně nový algoritmus, podobný metodě BLAST, vyvinutý původně pro rychlé zarovnání milionů myších genomů vůči sekvenci genomu lidského. Stejně jako v případě metody BLAST algoritmus nejprve vyhledává krátké segmenty s vysokou mírou podobnosti, které se následně snaží rozšiřovat do obou stran. Oproti BLASTu ale používá nový přístup k indexování, kdy si v paměti uchovává hash tabulku (index list) celé cílové databáze. Indexy jsou sestaveny ze souřadnic všech nepřekrývajících se k-tic (k-mer) v cílové databázi s vyjímkou těch s vysokým počtem opakování. BLAT následně vytvoří seznam všech překrývajících se k-tic ve zpracovávané sekvenci a vyhledá je v cílové databázi. Současně si buduje seznam "zásahů", kde se vyskytly shody mezi srovnávanými sekvencemi. BLAT má oproti metodě BLAST několik výhod, z nichž lze vyzdvihnout například přímé spojení s vyhledávačem UCSC a především pak jeho rychlost. Ve srovnání se staršími nástroji BLAT dosahoval až pěti set násobného zrychlení při zarovnání DNA/mRNA sekvencí a až padesáti násobného zrychlení při zarovnání proteinů.

Kapitola 4

Zarovnání sekvencí

V předcházející kapitole 3 jsem se zaobíral významem homologů pro prediktory škodlivosti aminokyselinových substitucí. Vyhledávání homologů však není možné bez zarovnání dvou zkoumaných sekvencí a prediktory, založené na principu fylogenetické analýzy, zpravidla vyžadují vícenásobné zarovnání homologních sekvencí jako svůj vstup. V této kapitole se tedy zaměřím na problematiku jednoduchého i vícenásobného zarovnání. Ve stručnosti rozeberu rozdíly progresivních a iterativních metod a zmíním některé z nejvýznamnějších nástrojů pro vícenásobné zarovnání sekvencí.

4.1 Zarovnání sekvencí

Zarovnání sekvencí je jednou ze základních úloh bioinformatiky. Jako nejjednodušší situaci si můžeme představit dva řetězce znaků (nukleotidů nebo aminokyselin), ve kterých v průběhu evoluce došlo k neznámému počtu změn. Tyto změny mohou být dvojího charakteru:

- Substituce, tedy záměna jedné aminokyseliny za jinou (v rámci evolučního procesu poměrně častá). V případě substituce se délka sekvence nemění.
- Vložení / odstranění znaku (dochází k němu s nižší pravděpodobností). Mění se délka sekvence, je třeba vložit mezeru.

Úkolem zarovnání je (prostřednictvím mezer) přiřadit k sobě znaky v sekvencích tak, aby podávaly maximální míru podobnosti. Konstrukce zarovnání umožňuje vyhledávat sekvence v databázích (stejné, či podobné - homologní), stanovit míru konzervovanosti jednotlivých nukleotidů / aminokyselin nebo kalkulovat evoluční vzdálenost porovnávaných sekvencí. Na základě penalizace mezer jsou rozlišovány tři typy zarovnání:

- **Globální zarovnání:** předpokládá sekvence stejné (podobné) délky. Sekvence jsou v případě globálního zarovnání posuzovány jako celek a výskyt mezery je vždy penalizován. V případě nestejně dlouhých sekvencí, na jejichž začátku a konci se často vyskytují mezery, jsou tyto mezery rovněž penalizovány.
- **Semi-globální zarovnání:** stejně jako v případě globálního jsou sekvence posuzovány jako celek. Nedochází však k penalizaci mezer na začátku a konci sekvencí.
- **Lokální zarovnání:** slouží k zarovnávání sekvencí, jejichž délka je výrazně odlišná a naším požadavkem je tedy nalézt všechny podřetězce (kratší sekvence) v sekvenci

delší. Rozdílem od globálního a semi-globálního zarovnání jsou kromě mezer penalizovány i neshody znaků. Mezery na začátku a konci sekvence jsou ignorovány.

4.1.1 Základní metody zarovnání

Skórovací matice

Jak již bylo zmíněno výše, v některých případech je kromě výskytu mezer zapotřebí penalizovat i záměny mezi různými znaky. Ne každá záměna má však ekvivalentní váhu. Pro příklad na úrovni nukleotidů je záměna mezi puriny (adenin za guanin) nebo pyrimidiny (cytosin za thymin) daleko pravděpodobnější než záměny mezi purinem a pyrymidinem. Na úrovni kodonů jsou jednobodové mutace (SNP) častější než mutace dvou, či tříbodové. A podobně u aminokyselin jsou obvyklejší mutace fyzikálně a chemicky podobných aminokyselin (pro příklad hydrofilní arginin je častěji zaměněn za glutamin, než hydrofobní valin). Tento jev je dán skutečností, že hlavním smyslem zarovnání sekvencí je hledání jejich evolučního vztahu. Předpokládáme-li tedy, že srovnávané sekvence jsou evolučně příbuzné, pak lze očekávat, že substituce za podobné aminokyseliny (nukleotidy) častěji způsobila přežití organismu a přenos mutace do další generace.

S vědomím různých pravděpodobností pro záměny jednotlivých aminokyselin (nukleotidů) byly vytvořeny tzv. skórovací matice, které těmto pravděpodobnostem přiřazují číselné ohodnocení. Výsledné skóre celého zarovnání je spočteno jako součet ohodnocení každé pozice v zarovnání, posuzované dle skórovací matice. Od skóre jsou navíc odečteny mezery, které mají své vlastní ohodnocení. Dle konkrétní použité metody můžou být všechny mezery penalizovány stejně nebo v závislosti na svém pořadí v sekvenci, případně mohou být mezery na začátku a konci sekvence zcela ignorovány.

Touto metodou je možno nad sekvencemi spočíst každé možné zarovnání a následně vybrat to s nejvyšším ohodnocením. Zarovnání s nejvyšším skóre je pravděpodobně evolučně nejbližší. Příklad skórovací matice BLOSUM, která společně s maticemi PAM [17] patří k nejznámějším, je znázorněn na obrázku 4.1.

Naivní přístup, kdy jsou kalkulovány všechny přípustné kombinace zarovnání spadá do třídy exponenciální složitosti a u delších sekvencí je v konečném důsledku nerealizovatelný (zarovnání sekvencí o délce pouhých 100 a 95 znaků vede na 55 milionů kombinací). V praxi se z důvodu urychlení užívá různých heuristik.

Dynamické programování

Hlavním problémem naivního přístupu je velké množství generovaných zarovnání. Dynamické programování řeší tento problém jeho dělením na menší podproblémy. Poprvé bylo aplikováno v roce 1970 Needlemanem a Wunchem [68] a rychle si vybudovalo cestu do popředí všech bioinformatických algoritmů.

Pro příklad uvažme dvě sekvence ACAGTAG a ACTCG. V prvním kroku nám vyvstávají tři možnosti: 1) zarovnáme první dva znaky, 2) vložíme mezeru do první sekvence, 3) vložíme mezeru do druhé sekvence. První krok nám tedy dává tři rozpracované stavy, přičemž celkové skóre závisí na prvním kroku a zbytku sekvence - iterativně rozkládáme na podproblémy (konstrukce stromu). Skóre jednotlivých kroků lze zaznamenávat do tabulky, v níž řádky tvoří znaky jedné a sloupce znaky druhé sekvence. Výpočet tabulky probíhá následovně:

- Každá vnitřní buňka se vypočte jako maximum ze tří možností:

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

Obrázek 4.1: Symetrická matice BLOSUM-62, sloužící k ohodnocení zarovnávaných pozic v sekvencích proteinů. Číslice v názvu značí procentuální míru podobnosti sekvencí. Převzato z [78].

  - Převzetí levé hodnoty s přičtením penalizace za vložení mezery.
  - Převzetí hodnoty shora s přičtením penalizace za vložení mezery.
  - Převzetí hodnoty z levého horního rohu s přičtením skóre za shodu nebo s penalizací za záměnu znaku.
- Hodnota v pravém spodním rohu tabulky reprezentuje skóre optimálního zarovnání.
- Zpětným průchodem tabulky z pravého spodního rohu směrem k levému hornímu rohu získáme zpětně podobu optimálního zarovnání. Zde se opět zvažují tři možnosti (vybere se vždy ta, ze které byla spočtena hodnota aktuální buňky):
  - Posun zpět ve směru diagonály.
  - Posun nahoru - vložení mezery do horizontálního řetězce.
  - Posun doleva - vložení mezery do vertikálního řetězce.

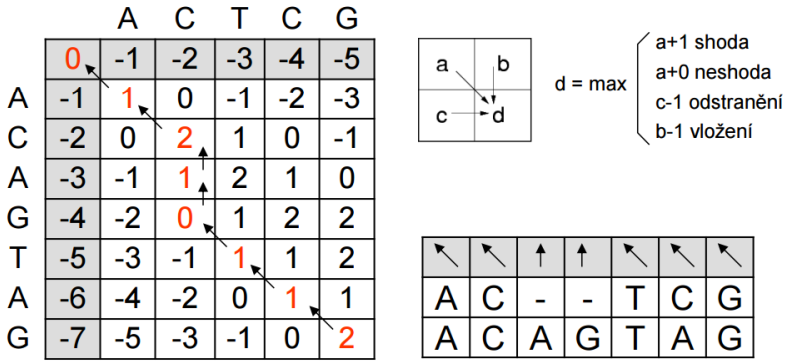
V případě dynamického programování může existovat více ekvivalentních možností. Příklad použití je znázorněn na obrázku 4.2.

4.2 Vícenásobné zarovnání

4.2.1 Progresivní metody

Průběh progresivních metod vícenásobného zarovnání lze rozdělit do tří základních kroků:

- Sestaví se pomocný strom (guided tree) na základě podobnosti.



Obrázek 4.2: Příklad tabulky dynamického programování s vyznačeným zpětným průchodem. Převzato z [78].

- Z množiny k sekvencí se zarovnají dvě sekvence a získá se tak množina o k - *1* sekvencích.
- Předcházející krok je opakován, dokud nejsou zarovnány všechny sekvence.

Ve druhém z uvedených kroků dochází k výběru vždy dvou sekvencí pro zarovnání, přičemž pořadí výběru těchto dvojic výrazně ovlivňuje kvalitu výsledného zarovnání. Mezivýsledek v n-tém kroku navíc již v následujících krocích nelze upravovat. Pro určení pořadí výběru sekvencí se běžně používá tzv. guided tree. Tento pomocný strom je vytvořen na základě podobnosti pomocí algoritmu UPGMA (viz 5.2.2). Zarovnávány jsou pak k sobě vždy nejpodobnější sekvence od listů ke kořeni stromu.

Mezi problémy progresivních metod patří jejich rychlost, kdy nejnáročnější částí je porovnání všech dvojic sekvencí a sestavení tabulky podobnosti. Pro příklad máme-li 100 sekvencí délky N, je třeba provést celkem 4950 srovnání dvojic (pro každou dvojici výpočet N^2 položek v tabulce v případě použití dynamického programování) - pro urychlení lze užít heuristik. Jednou z metod je počítání k-tic (k-mer counting), při níž se obě srovnávané sekvence X a Y rozdělí do k-tic s využitím posuvného okénka. Pro každou k-tici se následně zaznamená frekvence výskytu v sekvencích X a Y a na základě počtu výskytů se vypočte podobnost obou sekvencí například pomocí euklidovské vzdálenosti. Druhým problémem je skutečnost, že sestavené zarovnání již nelze měnit a tedy je nezbytné výsledek iterativně zpřesňovat.

Nejznámnějším zástupcem progresivních metod je rodina nástrojů CLUSTAL (viz 4.3.1). Jako další zde můžeme zařadit nástroj Kalign [77], či T-Coffee [12].

4.2.2 Iterativní metody

Jak již bylo zmíněno dříve, jedním z hlavních problémů progresivních metod je, že v průběhu algoritmu nelze zarovnání měnit. Zvolíme-li tedy na začátku běhu algoritmu nevhodné sekvence pro zarovnání, je tato chyba šířena celým výpočtem. Řešením jsou iterativní metody. V jádru pracují podobně jako progresivní metody, ale při přidávání dalších sekvencí do zarovnání se pokouší optimalizovat již zarovnané sekvence. Jeden z možných průběhů lze popsat následovně:

- Předpokládejme pomocný strom (guided tree) a výsledek vícenásobného zarovnání na vstupu algoritmu.
- Strom se rozstřihne na dvě části a pro každou část se sestaví vícenásobné zarovnání.
- Z obou zarovnání se sestaví výsledné zarovnání a ohodnotí se jeho skóre.
- Jestliže bylo dosaženo lepšího skóre, původní strom se nahradí novým a proces zpřesňování je opakován dokud jsou nalézána lepší skóre nebo dokud ještě nebyly vyzkoušeny všechny možnosti dělení stromu. Při volbě větve k rozpůlení se postupuje od kořene k listům.

Nejznámějším zástupcem iterativní metody vícenásobného zarovnání je nástroj MUSCLE (viz 4.3).

4.3 Vybrané nástroje pro zarovnání sekvencí

4.3.1 CLUSTAL

CLUSTAL je jednou z nejrozšířenějších rodin algoritmů pro vícenásobné zarovnání. Pracuje na bázi progresivních metod, tedy se drží kroků uvedených v 4.2.1: 1) porovná všechny dvojice mezi sebou a sestaví tabulku podobnosti; 2) sestaví pomocný strom (guided tree); 3) dle pomocného stromu provede postupné zarovnání dvojic.

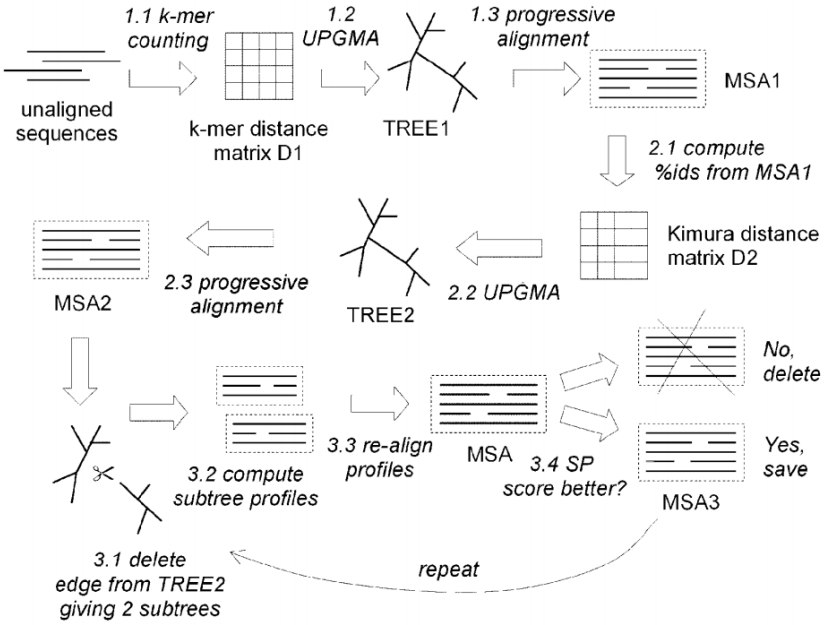
Existuje ve velkém množství variací:

- **CLUSTAL [14]:** Využívá BLAST/FASTA pro sestavení tabulky podobnosti, UPGMA pro konstrukci guided tree a skórovací matice PAM a BLOSUM.
- **CLUSTALV [13]:** Přepracování originálního programu, umožňující ukládat a znovu využívat dříve spočtená zarovnání. Navíc umožňuje zkonstruovat fylogenetický strom po sestavení zarovnání.
- **CLUSTALW [35]:** Oproti původní metodě používá pro sestavení tabulky podobnosti dynamické programování a pro konstrukci stromu algoritmus Neighbour-joining (viz 5.2.1). Skórovací matice se volí dynamicky na základě míry identity zarovnávaných sekvencí. Při výpočtu zarovnání je navíc každá sekvence váhována na základě své významnosti.
- **CLUSTALX [36]:** Oproti CLUSTALW, který je konzolovou aplikací, je navíc doplněn o grafické rozhraní.
- **CLUSTALΩ [23]:** Nejnovější verze, vykazující znatelný nárust výkonu oproti předcházejícím (pro srovnání složitost CLUSTALW je $O(N^2)$ a u CLUSTALΩ je $O(NlogN)$). CLUSTALΩ nahrazuje sekvence vektory n-dimenzionálního prostoru, kde každý vektor zastupuje vzdálenost od referenční sekvence. Vektory mohou být rychle shlukovány pomocí k-means nebo UPGMA. Vícenásobné zarovnání je následně vytvářeno pomocí skrytých markovových modelů.

4.3.2 MUSCLE

MUSCLE [64] je přesnou iterativní metodou pro vícenásobné zarovnání, poprvé publikovanou Robertem C. Edgarem v roce 2004. Jedná se o iterativní metodu, jíž lze rozdělit do tří následujících kroků (grafické znázornění na obrázku 4.3):

- **Výpočet předběžného zarovnání:** pomocí metody k-mer counting se sestaví matice podobnosti. Z této se vytvoří první pomocný strom (TREE1) metodou UPGMA a sestaví se první vícenásobné zarovnání sekvencí (MSA1).
- **Přepočet zarovnané sekvence:** z MSA1 se spočte nová matice podobnosti (přesnější výpočet pomocí Kimurovy vzdálenosti) a sestaví se druhý pomocný strom TREE2 a nové zarovnání MSA2.
- **Aplikace iterativního zpřesňování nad TREE2 a MSA2:** uplatněním algoritmu, popsaném v 4.2.2.



Obrázek 4.3: Grafické znázornění algoritmu MUSCLE. Převzato z [64].

4.3.3 MAFFT

MAFFT [47] je balíkem nástrojů, zahrnující progresivní i iterativní metody. Lze zde nalézt:

- Metody podobné nástroji CLUSTALW s urychlením prostřednictvím určení podobnosti sekvencí pomocí Fourierovy transformace. Nejrychlejší varianta, ale méně přesná.
- Iterativní metoda s urychlením pomocí Fourierovy transformace. Vyšší přesnost za cenu zpomalení algoritmu.
- Kromě předešlého se navíc používá výpočet skóre konzistence zarovnání mezi sekvencemi. Nejpřesnější, určeno pro jednotky až desítky sekvencí.

Kapitola 5

Fylogenetické stromy

Pro predikci pomocí fylogenetické analýzy je zapotřebí bližšího poznání vztahů mezi jednotlivými sekvencemi ve vícenásobném zarovnání. K tomuto účelu jsou na vstup prediktoru vkládány tzv. fylogenetické stromy. V této kapitole uvedu základní metody pro konstrukci fylogenetických stromů a stručný výčet existujících nástrojů.

5.1 Fylogenetický strom

Fylogenetické stromy [44] umožňují zachycovat evoluční vztahy mezi geny, proteiny nebo organismy. Kořen stromu představuje společného předka všech ostatních organismů (genů, proteinů) ve fylogenetickém stromu, od kterého byly tyto v průběhu evoluce prostřednictvím mutací odvozeny. Tento společný předek však nemusí existovat. V takovém případě hovoříme o nekořenových stromech. Nekořenové stromy je možné transformovat do kořenové podoby například vložením sekvence vzdálené od všech ostatních sekvencí stromu (tzv. outgroup rooting). Kromě kořeně lze ve fylogenetickém stromu dále identifikovat:

- **Listové uzly:** představují prvky (organismy, proteiny, ...), obsažené v analyzované datové sadě.
- **Vnitřní (nelistové) uzly:** reprezentují společného předka dvojic listových a vnitřních uzlů stromu. Tento předchůdce nemusel v procesu evoluce reálně existovat (jedná se pouze o předpokládanou podobu sekvence, ze které potomci vycházejí).
- **Hrany:** určují evoluční vzdálenost organizmů, proteinů, genů, atd. Pod evoluční vzdáleností si lze představit dobu, která uplynula od rozdělení vývojové větve nebo počet změn, který od jedné sekvence k druhé proběhl.

Základním textovým formátem pro zápis fylogenetických stromů je formát Newick. Uzly stromu, které mají předpokládaného stejného nejbližšího předka, jsou vždy společně uzávorkovány. Příklady zápisu jsou uvedeny na obrázku 5.1.

```
(A,B,(C,D));                          pojmenování listových uzlů + vzdáleností
(A,B,(C,D)E)F;                        listové i vnitřní uzly jsou pojmenovány
(A:0.1,B:0.2,(C:0.3,D:0.4):0.5);      pojmenování listových uzlů + vzdálenosti
(A:0.1,B:0.2,(C:0.3,D:0.4)E:0.5)F;    pojmenování všech uzlů + vzdálenosti
```

Obrázek 5.1: Příklady možných zápisů fylogenetického stromu ve formátu Newick.

5.2 Metody založené na vzdálenosti

Metody založené na vzdálenosti se obecně snaží o konstrukci fylogenetického stromu za využití matice vzájemných evolučních vzdáleností nukleotidových, či aminokyselinových sekvencí. Těchto metod lze spolehlivě využít pouze tehdy, je-li matice vzdáleností aditivní, a tedy jsou splněny následující podmínky:

- $D_{a,b} + D_{c,d} \leq D_{a,c} + D_{b,d}$
- $D_{a,b} + D_{c,d} \leq D_{a,d} + D_{b,c}$

Přičemž $D_{a,b}+D_{c,d}$ vyjadřuje součet délek všech hran kromě středové propojovací hrany, $D_{a,c} + D_{b,d}$, $D_{a,d} + D_{b,c}$ vyjadřují součet délek všech hran včetně dvounásobku středové hrany a a, b, c, d jsou uzly stromu. Není-li matice aditivní, pak lze najít pouze přibližné řešení, např. aproximací pomocí metody nejmenších čtverců.

5.2.1 Neighbour-Joining

Cílem metody neighbour-joining [58] je nalézt takové dva uzly, které k sobě mají dle matice evolučních vzdáleností nejblíže a současně jsou nejvíce vzdálené od ostatních. Algoritmus lze rozdělit do pěti kroků:

- Spočte se matice vzdáleností $D(i, j)$ a sestaví se hvězdicový strom s jediným společným středem a s uzly odpovídajícími vstupním sekvencím.
- Zvolí se dva uzly A a B, reprezentující výsledek funkce $min[D(a, b) - u(a) - u(b)]$, kde $D(a, b)$ značí vzdálenost uzlu A od B (lze získat z matice vzdáleností) a $u(a)$, respektive $u(b)$ představuje vzdálenost uzlu A, potažmo B od všech ostatních. Hodnotu $u(a)$ lze spočíst jako $u(a) = 1/(N - 2) \sum_{i \in I} D(a, i)$, kde I je množinou všech uzlů a N jejich počet. Vybrané uzly A a B jsou posléze spojeny do jednoho uzlu U.
- Spočtou se příslušné délky nově vzniklých hran jako $L(a, u) = (D(a, b)+u(a)-u(b))/2$, respektive $L(b, u) = (D(a, b) + u(b) - u(a))/2$.
- Vzdálenost nově vytvořeného uzlu U od všech ostatních uzlů se vypočítá dle vzorce $D(u, i) = D(a, i) + D(b, i) - D(a, b)/2$.
- Není-li uzel U posledním zbývajícím uzlem, algoritmus se vrací k druhému bodu.

Metoda obecně produkuje nekořenové stromy. Pro převod do kořenového tvaru je možné využít Midpoint nebo Outgroup algoritmu.

5.2.2 UPGMA

UPGMA [27] je velice jednoduchou a rychlou metodou, která i v současné době nalézá široké uplanění. Její hlavní nevýhodou je skutečnost, že pro konstrukci stromu předpokládá konstantní evoluční rychlost ve všech větvích stromu, což může mít negativní dopad na přesnost algoritmu, není-li tato podmínka vstupními daty naplněna. Algoritmus lze opět rozdělit do čtyř jednoduchých kroků:

- Z tabulky evolučních vzdáleností jsou vybrány dvě sekvence A a B s nejvyšší mírou podobnosti.

- Dvojice sekvencí je spojena do jedné sekvence AB a přepočítají se vzdálenosti v tabulce dle vztahu $D_{AB,C} = (D_{A,C} + D_{B,C})/2$.
- Spojení sekvencí je zakresleno do konstruovaného fylogenetického stromu.
- Algoritmus se vrací k prvnímu kroku, dokud nejsou spojeny všechny dvojice.

Kromě již uvedeného problému s předpokladem konstantní rychlosti evolučního vývoje dále metoda UPGMA produkuje tzv. ultrametrické stromy. V důsledku toho je vzdálenost od kořeně ke všem uzlům stejná.

5.3 Metody založené na znacích

Výše popsané metody, založené na vzdálenosti, pracují s maticí vzdálenosti, která musí být před samotnou konstrukcí stromu sestavena z vícenásobného zarovnání sekvencí. Tímto krokem však dochází ke ztrátě části informací. Metody založené na znacích se tento problém snaží obejít konstrukcí fylogenetického stromu přímo na základě vícenásobného zarovnání. Oproti metodám, založeným na vzdálenosti, je tento druh metod navíc schopen ze sekvencí na listech stromu dopočítat sekvence předků v uzlech nelistových.

Jestliže ohodnotíme hrany mezi uzly stromu, pak součet všech těchto ohodnocení nám udává tzv. parsinomy skóre. Cílem je sestavit strom s minimálním parsinomy skóre, přičemž vyvstávají dva základní problémy:

- **Malý parsinomy problém:** Jak nalézt ohodnocení vnitřních úzlů?
- **Velký parsinomy problém:** Jak nalézt optimální tvar fylogenetického stromu?

5.3.1 Malý parsinomy problém

V rámci této třídy problémů se snažíme ze sekvencí na listech fylogenetického stromu reprodukovat sekvence předků na nelistových uzlech stromu (tyto rodičovské sekvence se nemusely dochovat). Sekvence vnitřních uzlů stromu jsou dopočítány tak, aby výsledkem stromu bylo minimální parsionomy skóre. Jednotlivé znaky sekvence jsou v tomto případě na sobě nezávislé a lze je tedy posuzovat odděleně. Vstupem malého parsionomy problému je tedy již zkonstruovaný fylogenetický strom.

Fitchův algoritmus

Jedním z řešení malého neváhovaného parsinomy problému je Fitchův algoritmus [87]. Algoritmus využívá myšlenku zpětného průchodu u dynamického programování, tedy rekurzivní přístup, kdy je skóre rodiče spočteno na základě skóre obou jeho potomků.

V prvním kroku výpočet probíhá od listů ke kořeni stromu. Pro každý nelistový uzel stromu je vypočtena množina přípustných znaků dle následujícího pravidla: překrývají-li se množiny přípustných množin potomků, vyber do rodiče jejich průnik, jinak vyber jejich sjednocení. Ve druhém kroku je proveden zpětný průchod od kořene k listovým uzlům. Pro kořen lze zvolit libovolný znak z množiny přípustných, pro ostatní vnitřní uzly je pak uplatněno pravidlo: obsahuje-li množina stejný znak, jako jeho předek, pak vyber tento znak i pro aktuální uzel, v opačném případě vyber libovolný znak z množiny přípustných. Pro ohodnocení se používá Hammingova vzdálenost, tedy nula, jsou-li znaky rodiče a potomka identické, jinak jedna.

Sankoffův algoritmus

Využití Hammingovy vzdálenosti pro ohodnocení stromu není přesné, jelikož záměny mezi různými znaky (nukleotidy, aminokyseliny) nemusejí mít stejnou pravděpodobnost. Pro získání lepších výsledků je tudíž nezbytné využít skórovacích matic (viz 4.1.1) - váhovaný malý parsinomy problém. Sankoffův algoritmus [80] je zobecněnou verzí Fitchova algoritmu pro neváhovaný malý parsinomy problém.

Ke každému uzlu je vytvořena tabulka skóre pro všechny možné znaky, která se následně doplňuje od listových uzlů ke kořeni dle následujících pravidel:

- **Listový uzel:** jestliže se znak na dané pozici skórovací tabulky shoduje se znakem v sekvenci listu, je do tabulky vložena nula, jinak nekonečno.
- **Vnitřní uzel:** hodnota pro každý znak ve skórovací tabulce je vypočítána jako $s_t(rodic) = min_i\{s_i(levyPotomek) + \delta_{i,t}\} + min_j\{s_j(pravyPotomek) + \delta_{j,t}\}$, kde s_i, s_j, s_t značí položku ve vektoru uzlu a $\delta_{i,t}$ hodnotu pro danou kombinaci i, t ve skórovací matici.

Ve druhém kroku je proveden zpětný průchod od kořene k listům, přičemž nejmenší skóre v kořenu stromu odpovídá nejmenšímu parsinomy skóre celého stromu. Algoritmus tedy pokračuje směrem k listům a pro každý vnitřní uzel vybírá takový znak, který odpovídá vybranému skóre v rodičovském uzlu.

5.3.2 Velký parsinomy problém

Velký parsinomy problém se snaží nalézt optimální strukturu fylogenetického stromu. Vstupem je tedy matice vícenásobného zarovnání, přičemž počet listových uzlů stromu odpovídá počtu řádků v zarovnání. Hlavním problémem je rozsah stavového prostoru stromů, které mohou být ze sekvencí zkonstruovány - jedná se o NP těžký problém s nezbytností využití heuristik.

Nearest-neighbour interchange

Jedna z mnoha využívaných heuristik pro řešení velkého parsinomy problému. Vychází z předpokladu, že pro každou hranu stromu lze sestrojit nekořenový strom, obsahující čtyři podstromy. Tyto čtyři podstromy mohou být dále kombinovány třemi možnými, vzájemně transformovatelnými způsoby. Uvážíme-li všchny přípustné konstrukce, pak lze sestrojit graf, kde vrcholy označují strom a hrany spojují takové stromy, které jsou sousedy. Postup algoritmu je následující:

- Algoritmus začíná v libovolném uzlu grafu sousedů.
- Pro všechny své sousedy spočte parsinomy skóre.
- Přemístí se do toho uzlu, kde vykazoval nejnižší skóre.

Hlavním problémem tohoto přístupu je, že algoritmus obvykle končí v lokálním minimu a je tedy nezbytné algoritmus spouštět opakovaně nebo využít další techniky. Alternativou jsou algoritmy, používající metod prořezávání a přeuspořádávání stromu.

5.4 Metody založené na pravděpodobnosti

Metody založené na znacích neuvažovaly různé délky větví fylogenetického stromu a snažily se tedy sestavit strom s nejnižším počtem změn na úrovni znaků. Tento nedostatek řeší metody založené na pravděpodobnosti (maximum likelihood). Tato třída metod se snaží o nalezení takového modelu stromu, který s největší pravděpodobností odpovídá vstupní množině dat a jsou navíc schopny dopočítat také ohodnocení hran. Maximum likelihood pracuje vždy s kořenovým stromem a lze jej rozdělit na tři podprobémy: maličký, malý a velký likelihood problém. Oproti metodám, založeným na znacích, jsou maximum likelihood metody přesnější, ale výpočetně náročnější.

5.4.1 Maličký likelihood problém

Nejnižší úroveň problému, která se z vícenásobného zarovnání sekvencí, tvaru fylogenetického stromu s ohodnocením listových uzlů a délek hran snaží vypočítat ohodnocení vnitřních uzlů stromu a jeho celkové skóre.

Felsensteinův algoritmus

Algoritmus [37], určený k řešení maličkého likelihoood problému. Je založen na principu dynamického programování a podobně jako Sankoffův algoritmus jej lze rozdělit do dvou kroků.

Každému uzlu stromu (vnitřnímu i listovému) je přidělen vektor hodnot, jehož délka je rovna počtu přípustných znaků (např. 4 pro nukleotidy). Hodnoty tohoto vektoru budou dále označovány jako likelihood. V prvním kroku prochází algoritmus stromem od listů ke kořeni a vyhodnocuje likelihood hodnoty pro každý uzel dle následujících pravidel:

- **Listový uzel:** odpovídá-li znak na určité pozici vektoru znaku v listu, vloží se na tuto pozici vektoru jednička jinak nula.
- **Vnitřní uzel:** likelihood hodnota k-té položky z vektoru vnitřního uzlu S_k je spočtena jako $L_{S_k}(k) = [\sum_{S_i} P_{S_k S_i}(t_i) * L_{S_i}(i)] * [\sum_{S_j} P_{S_k S_j}(t_j) * L_{S_j}(j)]$, kde $P_{S_k S_i}(t_i)$ značí pravděpodobnost přechodu mezi znakem k-té položky rodiče a i-tým znakem potomka, a kde $L_{S_i}(i)$ je aktuální hodnotou pro daný znak ve vektoru potomka. Indexy i, j značí levého a pravého potomka.
- **Ohodnocení stromu:** likelihood hodnota stromu je dána jako $L = \sum_{S_0} \pi_{S_0} * L_{S_0}(0)$, kde π_{S_0} značí pravěpodobnost, že na začátku stromu byl určitý znak a $L_{S_0}(0)$ znaku příslušnou likelihood hodnotu ve vektoru kořene stromu.

Druhým krokem je opět zpětný průchod stromem a ohodnocení vnitřních uzlů. Zde stačí pro každý vnitřní uzel S_k dosadit do výše uvedeného vzorce pro výpočet $L_{S_k}(k)$ a následně vybrat maximum.

5.4.2 Malý likelihood problém

Vstupem malého likelihood problému je vícenásobné zarovnání sekvencí a tvar stromu s ohodnocením listových uzlů. Výsledkem je pak ohodnocení vnitřních uzlů a délek hran, které odpovídají maximální likelihood hodnotě pro zadaný strom.

V roce 1981 Felsenstein [37] odvodil iterační vzorec, kterým je možno délku hrany postupně upravovat tak, aby docházelo pouze ke zvyšování maximální likelihood hodnoty

stromu - využití hill climbing algoritmů (problém lokálních maxim). Průběh iteračního algoritmu je následující:

- Vyber počáteční délky hran $t_1, t_2, ..., t_n$
- Pro každou hranu t_i stromu aplikuj iterační vzorec, dokud je nalézána nová délka hrany, způsobující vyšší likelihood stromu.
- Opakuj předcházející krok dokud jsou nalézány hrany zvyšující likelihood hodnotu celého stromu.

5.4.3 Velký likelihood problém

Nejvyšší úroveň maximum likelihood algoritmů. Vstupem je vícenásobné zarovnání sekvencí, výstupem pak struktura stromu s ohodnocením listových i vnitřních uzlů a určené délky hran, odpovídající maximální likelihood hodnotě pro daný strom.

Možným řešením je postupné přidávání nových větví do konstruovaného stromu, přičemž má-li strom $n - 1$ listů, pak existuje $2n - 5$ míst, kam lze novou větev připojit. Po přidání každé nové větve je vždy proveden přepočet délek hran a po vložení několika nových uzlů dojde k prostříhání a reorganizaci stromu.

5.5 Nástroje pro konstrukci fylogenetického stromu

5.5.1 PhyML

PhyML [71] je hojně používaný fylogenetický software založený na maximum likelihood principu. První verze z roku 2003 využívala pro navýšení rychlosti jednoduchou přeskupovací metodu Nearest Neighbour Interchanges (NNIs), která mění spojení vždy čtyř podstromů s hlavním stromem. Novější algoritmus umožňuje prohledávání prostoru s uživatelsky definovanou intenzitou s využitím Subtree Prunning and Regrafting (SPR). Tato metoda odebere vždy jeden podstrom z hlavního stromu a následně jej vrátí na jiné místo stromu. Každá nově vytvořená topologie je ohodnocena pomocí parsinomy skóre a se zohledněním likelihood funkce jsou odfiltrovány nejméně slibné topologie.

5.5.2 RAxML

Nástroj navržený přednostně pro analýzu obsáhlých datasetů metodou maximum likelihood. Využívá nové instrukční sady pro nízkoúrovňovou optimalizaci, masivní paralelizaci a aproximovaný substituční model. RAxML [6] dosahuje až trojnásobného zrychlení oproti PhyML.

5.5.3 PHYLIP

PHYLIP [16] je multiplatformní, volně dostupný balíček nástrojů pro konstrukci fylogenetických stromů. Balíček obsahuje algoritmy pro parsinomy, vzdáleností matice a likelihood metody. Součástí je i bootstrapping a consensus tree. Algoritmus lze uplatnit pro molekulární sekvence, restrikční místa, geny, distanční matice a další.

5.5.4 SEMPHY

SEMPHY [56] je nástrojem pro konstrukci datově náročných fylogenetických stromů. Základem je maximum likelihood algoritmus pro určení nejpravděpodobnější topologie stromu a optimálních délek větví. Dále využívá algoritmické paradigma strukturálního Expectation Maximization algoritmu, díky kterému je schopné se vypořádat s velkými datovými sadami s vysokou přesností a přijatelnými časovými nároky.

5.5.5 FastTree

První verze FastTree využívala podobně jako PhyML metodu NNIs, doplněnou o kritérium minimum evolution. Tato metoda předpokládá, že topologie stromu s nejnižším součtem vzdáleností všech větví je nejsprávnější. FastTree 2 [53] přidává minimum evolution SPR a maximum likelihood NNIs. FastTree 2 je obecně přesnější, než standardní implementace maximum likelihood NNIs metod, ale dosahuje horších výsledků než maximum likelihood SPR. FastTree 2 je také řádově rychlejší než ostatní z uvedených algoritmů pro konstrukci fylogenetických stromů.

Kapitola 6

Predikce škodlivosti mutací

V průběhu posledních let bylo vyvinuto velké množství nástrojů pro predikci škodlivosti aminokyselinových substitucí, pracujících na principech fylogenetické analýzy, případně technik strojového učení (neuronové sítě, rozhodovací stromy, bayesovský klasifikátor), které s sebou i přes vysokou přesnost předpovědí přináší řadu neduhů.

Problémem strojového učení může být například nedostatek spolehlivých trénovacích / testovacích dat, či výběr dostatečně velké a rozmanité sady k učení. Konkrétně neuronové sítě při nevhodně zvolené sadě dat, případně dlouhém učícím procesu tíhnou k uváznutí ve stavu tzv. přeučení, kdy je síť sice schopna dokonale klasifikovat vzorky trénovací množiny, ale pozbývá schopnosti zobecňovat a pro vzorky mimo svou trénovací množinu podává výsledky s nízkou predikční úspěšností. Druhou obtíží pak bývá vysoká časová náročnost, potřebná pro naučení.

Druhou často využívanou možností jsou nástroje používající principů fylogenetické analýzy a vyhledávání konzervovaných segmentů ve vícenásobném zarovnání. Tyto nástroje bývají zpravidla implementačně průhlednější (jejich funkce není dána například nastavením vah neuronové sítě) a vyžadují nižší čas pro výpočet predikce.

V rámci této kapitoly stručně zmíním funkci některých existujících predikčních nástrojů, podrobněji pak nástroj MAPP, pracující na základě druhého ze zmíněných přístupů, a kterým je tato práce inspirována.

6.1 Jednonukleotidové polymorfismy

Jednobodové polymorfizmy (SNP) jsou substitucí jednoho nukleotidu v DNA sekvenci jednoho druhu za jiný (např. dochází k záměně adeninu za cytosin). V důsledku degenerovanosti genetického kódu, jak bylo popsáno v 2.3, se však SNP při překladu do aminokyselinové sekvence proteinu nemusí projevit - tzv. synonymní mutace. Dále pouze malá část genetického kódu je do proteinů skutečně překládána. SNP vyskytující se v kódujících oblastech a měnící podobu proteinu jsou označována jako nsSNP (non-synomous SNP). Právě tyto nesynonymní polymorfismy lze považovat za příčinu vzniku genetických onemocnění. Pokud bychom znali dopady všech nsSNP na organismus (u člověka se mezi dvěma heterozygotními jedinci odhaduje až 40 000 nsSNP a až 200 000 nsSNP v populaci [51]), mohli bychom stanovovat cílenou, efektivní léčbu individuálně pro každého pacienta na základě jeho genetických dispozic. Experimentální ověření všech nsSNP by však bylo časově i finančně vysoce nákladné (cena jednoho takového experimentu se pohybuje v řádech tisíců až desetitisíců korun), a tedy vzrůstá potřeba predikčních nástrojů, schopných co nejbližší

aproximace reálným účinkům jednonukleotidových polymorfismů. Vybrané nástroje budou popsány v rámci této kapitoly níže.

6.2 Principy predikčních metod

Metody predikce lze obecně rozdělit do dvou stěžejních proudů - na metody, využívající za svůj základ příbuznost sekvencí a na prediktory, vycházející z podobnosti proteinových struktur [34].

- **Metody založené na příbuznosti sekvencí:** tato oblast metod vychází z poznatků o konzervovanosti, popsaných již v 3.2. Obecně je tedy nejprve vyhledána určitá množina (nejlépe) homologních sekvencí a na jejich základě je vypočtena konzervovanost segmentů podle různorodosti aminokyselin, vyskytujících se na dané pozici v sekvenci. Při výpočtu konzervovanosti sekvencí se dále často využívá poznatků o evoluční vzdálenosti jednotlivých sekvencí, plynoucích z ohodnocení fylogenetického stromu, a některé prediktory navíc berou v úvahu i fyzikálně-chemické vlastnosti aminokyselin. Lze předpokládat, že záměny aminokyselin na konzervovaných pozicích budou pravděpodobněji škodlivé a na nekonzervovaných neutrální. Stejně tak záměny aminokyselin podobných chemických tříd mívají nižší dopad na funkci proteinu. Kritickým bodem těchto metod je zřejmě výběr vhodných homologních sekvencí a jejich zarovnání. Evolučně vzdálenější sekvence (upřednostňují se ortology nad paralogy) mají z hlediska konzervovanosti větší vypovídající hodnotu a teoreticky tak mohou zvýšit úspěšnost predikce. Jak již bylo uvedeno v 3.3, geny těchto evolučně vzdálenějších ortologních sekvencí však už nemusejí v organismu plnit stejnou funkci a identická mutace, která se v genomu jednoho projevila jako škodlivá může být u druhého mutací neutrální (dochází ke zkreslení výsledků).
- **Metody založené na podobnosti struktur:** tato oblast metod vychází z předpokladu, že proteiny s podobnou prostorovou strukturou budou mít i podobnou funkci. V přípravné fázi těchto metod tedy nedochází k vyhledávání homologních sekvencí, ale homologních struktur. Vyhledají se tedy proteinové struktury, nejvíce se podobající struktuře vstupní sekvence a k predikci vlivu mutací na dané pozici dochází na základě rozlišení vlastností těchto struktur. Při predikci se berou v potaz například rozdíly ve volné energii proteinových struktur, kontakt s okolím prostřednictvím tvaru a chemických vlastností vazebních míst, tunely uvnitř proteinů, proteinový fold, apod. Obecně vzato jsou metody založené na podobnosti struktur méně přesné (menší krytí záznamů v databázích) a výpočetně náročnější. Pro zpřesnění analýzy se často využívá anotací.

6.3 Přehled existujících predikčních nástrojů

V tomto oddílu textu stručně uvedu několik existujících nástrojů pro predikci škodlivosti aminokyselinových substitucí. Zvláštní pozornost pak bude věnována nástroji MAPP, kterým je tato práce inspirována.

6.3.1 SNAP

SNAP [3] je nástrojem pracujícím na principu strojového učení, využívajícím znalosti a anotace z databáze SwissProt [2]. Pro predikci používá informace o sekundární struktuře

proteinu, pozici dané aminokyseliny vůči jádru proteinu, míře vystavení aminokyseliny na dané pozici okolnímu prostředí, konzervovaných segmentech v proteinu apod. K natrénování neuronové sítě byla použita databáze pro proteinové inženýrství PMD (doplněné o neutrální mutace ze SwissProt) a při testech dosáhl v základním nastavení přesnosti přibližně 77 %. K predikci je využita již natrénovaná neurová síť, ke které nástroj automatizovaně dopočítává potřebné atributy ze vstupní sekvence pomocí volání integrovaných metod. Predikce je v důsledku výpočetně náročná až do řádu desítek minut. Rozhodnutí o škodlivosti je navíc doplněno mírou důvěry v hodnotách 1 - 9.

6.3.2 SIFT

Vstupem metody SIFT (Sorting Intolerant from Tolerant [11]) je podobně jako u předešlé metody pouze sekvence a požadovaná mutace. Na rozdíl od nástroje SNAP je ale SIFT založen na porovnání konzervovaných oblastí v homologních sekvencích. Proces predikce nejprve nalezne množinu podobných sekvencí v databázi prostřednictvím nástroje PSI-BLAST a po jejich zarovnání dojde k výpočtu pravděpodobnosti, založeném na četnosti jednotlivých aminokyselin na určené pozici v zarovnaných sekvencích.

Pokud nějaká pozice v zarovnání obsahuje výhradně například aminokyselinu isoleucin, pak metoda předpokládá, že isoleucin je na dané pozici nezbytným pro správnou funkci proteinu a jakákoliv záměna bude ohodnocena jako škodlivá. V jiném případě, pokud by se na určité pozici v zarovnání vyskytovaly aminokyseliny isoleucinu, leucinu a valinu (hydrofóbní aminokyseliny), bude metoda předpokládat, že záměny v rámci hydrofóbních aminokyselin mohou být v proteinu tolerovány. Výstupem metody je navíc statistika o různorodosti homologních sekvencí, počtu sekvencí v zarovnání a skóre v intervalu ⟨0, 1⟩ určující míru důvěryhodnosti predikce.

6.3.3 PolyPhen-2

Metoda predikující na základě struktury proteinu, sekvence i anotace, získané prostřednictvím databáze SwissProt. Zatímco PolyPhen-1, který využíval klasifikátoru, založeném na empiricky sestavené tabulce pravidel, PolyPhen-2 [82] je postavený na základě strojového učení a naivního bayesovského klasifikátoru. K vytvoření bayesovského klasifikátoru byly jako trénovací data použity datasety HumDiv a HumVar, přičemž prostřednictvím greedy algoritmu bylo vybráno osm sekvenčních a tři strukturní atributy. Při výpočtu jsou pak porovnávány vlastnosti alely zkoumaného řetězce a odpovídajících alel mutantních řetězců a pomocí atributů je porovnáno, nakolik se dvě alely shodují po nahrazení jedné aminokyseliny jinou. Algoritmus PolyPhen-2 se sám stará o výběr zarovnaných sekvencí pro shlukovou analýzu a provádí nad nimi vlastní zarovnání.

Nástroj jako výsledek predikce udává pravděpodobnost, zda je mutace škodlivá a odhad správnosti svého výpočtu. Pro klasifikaci je možno volit mezi bayesovským klasifikátorem naučeným na datasetu HumDiv, zahrnujícím neutrální mutace a mutace způsobující Mendelovské choroby na homologních sekvencích člověku blízkých savčích genů a na datasetu HumVar, obsahujícím výhradně mutace vyskytující se u lidí.

6.3.4 PhD-SNP

PhD-SNP [20] je založen na analýze konzervovanosti sekvencí a technice strojového učení, kdy do predikce vlivu substituce aminokyseliny na dané pozici zahrnuje i okolní residua. Jako atribut dále využívá i údaje o změně stability proteinu po provedení mutace.

6.3.5 MuD

MuD [24] je postaven na algoritmu random forest. Pro predikci využívá jak strukturní, tak sekvenční informace a celkově 14 atributů, povětšinou získaných z databáze SwissProt. Většina atributů je shodná s výše uvedenými nástroji a přidává i některé nové. Přestože úspěšnost metody nepřekonává ostatní a výkonem je srovnatelná s nástrojem SNAP, zajímavé na ní je, že uživateli umožňuje vkládat dodatečné informace za běhu výpočtu a zvyšovat tak přesnost predikce.

6.3.6 PANTHER

PANTHER [61] je dalším z mnoha nástrojů, využívajících k predikci analýzu konzervovanosti proteinového řetězce, k čemuž však používá vlastní databáze s rodinami proteinů. Každá rodina je reprezentována skrytým markovovým modelem. Při analýze vstupní sekvence je pomocí Markovova modelu zjištěno, do které rodiny proteinů tato sekvence patří a významnost mutace je posuzována odlišně dle rodiny, do níž byl protein zařazen. Výhodou je pokračující aktualizace databází rodin. Pokud se nástroji nepodaří vstupní sekvenci zařadit do některé z rodin, není schopen mutaci klasifikovat.

6.3.7 PASE

PASE [89] je jednoduchý predikční nástroj, využívající ke své činnosti analýzu konzervovanosti, společně s fyzikálně-chemickými vlastnostmi aminokyselin. Prostřednictvím nástrojů BLAST a CLUSTALW provede v prvním kroku PASE výběr a zarovnání ortologních sekvencí a spočítá míru konzervovanosti mutací požadované pozice a spočítá rozdíl ve fyzikálně-chemických vlastnostech mezi originální a substituovanou aminokyselinou. Na základě míry konzervovanosti a rozdílu mezi aminokyselinami ve druhém kroku následně spočítá predikční skóre jako prosté vynásobení obou koeficientů. Využívá celkem sedm fyzikálně-chemických atributů, a ačkoliv nedosahuje výsledků srovnatelných např. se SIFT, jedná se o velice rychlý, implementačně jednoduchý algoritmus.

6.3.8 SNPdryad

SNPdryad [46] je predikčním algoritmem, užívajícím podobně jako řada předešlých vícenásobné zarovnání (použit nástroj MUSCLE a PhyML pro konstrukci stromu) výhradně ortologních sekvencí, Shannonovu entropii k výpočtu skóre a metodu random forest ke klasifikaci. Jako výsledek produkuje pravděpodobnostní hodnotu pro všechny možné aminokyselinové substituce na dané pozici. Metoda k ohodnocení využívá i osm fyzikálně-chemických vlastností.

6.3.9 MAPP

Metoda MAPP (Multivariate Analysis of Protein Polymorphism [19]), publikovaná v roce 2005 E. A. Stonem a A. Sidowem je podobně jako řada předešlých metod založena na principech fylogenetické analýzy a fyzikálně-chemických vlastnostech aminokyselin a vychází tak ze dvou základních předpokladů:

- rozdíly ve standardních fyzikálně-chemických vlastnotech mezi aminokyselinami a jejich záměny jsou hlavní příčinou zhoršení funkce proteinů

- evoluční obměny aminokyselin u ortologů na dané pozici značí míru konzervovanosti, tedy do jaké míry jsou změny aminokyselin na určité pozici v sekvenci proteinu tolerovány

Za využití těchto dvou předpokladů počítá metoda MAPP fyzikálně-chemické variace v každém sloupci vícenásobného zarovnání a na základě této variace počítá odchylky možných kandidátů, kteří by mohli aminokyselinu na dané pozici nahradit. Lze očekávat, že čím větší je vypočtená odchylka, tím větší je pravděpodobnost poškození funkce proteinu.

Algoritmus nástroje MAPP si na vstupu žádá dva soubory: 1) vícenásobné zarovnání sekvencí ve formátu FASTA, 2) fylogenetický strom ve formátu Newick. Fylogenetický strom musí být rozšířen o informace o vzdálenostech, ale pouze jména listových uzlů mohou být vyznačeny v jeho textové podobě: *(A:0.1,B:0.2,(C:0.3,D:0.4):0.5);* je korektním zápisem, *(A:0.1,B:0.2,(C:0.3,D:0.4)E:0.5)F;* nikoliv. Generování těchto vstupů není do nástroje přímo integrováno, a tedy vstupní soubory musí být dodány uživatelem. Výstupem je seznam neutrálních a škodlivých mutací pro každou z pozic vícenásobného zarovnání, společně s jejich pravděpodobnostním ohodnocením.

K testování predikční metody MAPP byly použity čtyři (silně) experimentálně ověřené studie mutagenezí. Konkrétně se jedná o: 1) Laktózový represor (organismus Escherichia coli), 2) HIV proteázu, 3) HIV reverzní traskriptázu, 4) T4 Lysozym. Na základě vypočítaného pravděpodobnostního skóre byly substituce zařazeny do jedné ze tří tříd: positive (neměnící funkci), itermediate (slabě škodlivé) a negative (silně škodlivé).

V procesu testování bylo využito nástroje CLUSTALW pro sestavení vícenásobného zarovnání sekvencí a SEMPHY pro konstrukci fylogenetického stromu. Pro fyzikálně-chemickou analýzu bylo zvoleno následujících šest expertně vybraných parametrů: hydropatie [39], polarita [42], náboj [42], objem postranního řetězce [1], volná energie α-šroubovicové konformace [81] a volná energie β-listové konformace [81]. Srovnání přesnosti algoritmu MAPP s nástrojem SIFT je zachyceno v tabulce 6.1.

Tabulka 6.1: Srovnání přesnosti predikce metody MAPP s nástrojem SIFT. Převzato z [19].

	Positive vs. deleterious			**Intermediate vs. negative**
	MAPP	**SIFT**	**Increase**	**MAPP**
HIV protease	80,4%	78,6%	1,8%	76,7%
LacI	69,2%	67,9%	1,3%	74,5%
HIV RT	64,1%	55,0%	9,1%	72,9%
T4 lysozyme	73,0%	68,3%	4,7%	62,6%

Kapitola 7

Výběr rysů

Součástí optimalizace nově vyvíjeného nástroje pro predikci efektu aminokyselinových substitucí je také výběr podmnožiny fyzikálně-chemických vlastností z databáze AAIndex. V této kapitole se tedy zaměřím na problematiku výběru rysů. Uvedu rozdíly mezi dvojicí přístupů (výběr a extrakce rysů) a ke každému z přístupů stručně popíši několik vybraných metod. Závěrem se zmíním o balíčku nástrojů pro strojové učení, označovaném jako WEKA.

7.1 Rozdělení metod

Techniky výběru rysů lze na nejvyšší úrovni rozdělit na [28]:

- **Založené na informační hodnotě:** výběr rysů je proveden na základě jejich entropické informace. Snahou je vybrat takovou podmnožinu, která vede k co nejnižší ztrátě informace oproti výchozí datové sadě, a tedy zpravidla dosahující maximální vzájemné ortogonality. Výhodou metod založených na informacích je, že k výběru rysů nepotřebují testovací datovou sadu, ale obecně jsou méně přesné.
- **Založené na experimentech:** výběr rysů je proveden prostřednictvím experimentování na testovací datové sadě. Jsou vhodné k výběru rysů pro specifický model.

Jak bude rozvedeno níže, metody lze dále rozdělit na [45]:

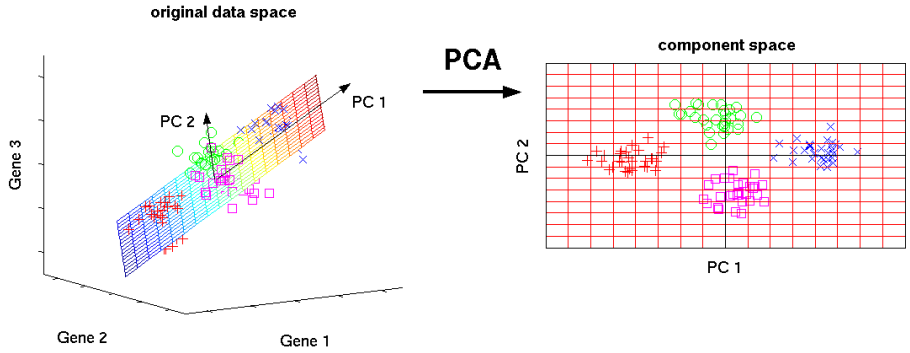
- Metody extrakce rysů (feature extraction)
- Metody výběru rysů (feature selection)

7.2 Extrakce rysů

Feature extraction je vhodným přístupem pro rozsáhlé množiny, u nichž se předpokládá vysoká míra redundance. Z originálního setu vlastností jsou v procesu feature extraction derivovány takové rysy, které nejlépe reprezentují zdrojová data s ohledem na cíl jejich zpracování. Na rozdíl od feature selection přístupů se tedy obvykle jedná o transformaci do nového prostoru rysů (zkonstruovaná množina rysů nemusí být podmnožinou originální datové sady). Z algoritmů pro feature extraction lze uvést např.: Principal component analysis, Semidefinite embedding, Multifactor dimensionality reduction, Partial least squares regression a další.

7.2.1 Principal Component Analysis (PCA)

PCA [29] patří obecně k nejrozšířenějším metodám pro řešení problému redukce dimenzionality. Jedná se o statistickou proceduru, využívající ortogonální transformaci pro konverzi potenciálně korelované datové sady na množinu hodnot lineárně nekorelovaných proměnných, tzv. principal components. Počet takových komponent je vždy nižší nebo v nejhorším případě stejný jako v původní datové sadě. První konstruovaná komponenta je komponentou s nejvyšší možnou odchylkou a je následována dalšími komponentami s maximálními odchylkami a současně splňujícími podmínku vzájemné ortogonality s komponentami předcházejícími. Pro výpočet komponent se využívá kovarianční matice. Z výsledné sady se vybírá zpravidla jen několik prvních komponent, aby se dosáhlo snížení dimenzionality při minimální ztrátě informací.



Obrázek 7.1: Redukce dimenzionality prostřednictvím Principal Component Analysis. Převzato z [55].

7.2.2 Semidefinite embedding (SDE)

Na SDE [48] může být nahlíženo jako na nelineární generalizaci algoritmu PCA. Vysoce dimenzionální data jsou nelineární redukcí mapovány do Euklidovského vektorového prostoru o méně dimenzích následujícím způsobem:

- Je vytvořen graf, kde každý vstup je spojen s k nejbližšími vektory a všechna tato k-sousedství jsou vzájemně propojena.
- Graf je rozbalen pomocí semidefinitního programování, které se snaží maximalizovat vzdálenosti mezi všemi nepropojenými dvojicemi, ale současně zachovávat vzdálenosti mezi prvky v k-sousedství.
- Nízko-dimenzionálního prostoru je dosaženo uplatněním mnohorozměrného škálování.

7.2.3 Principal Component Regression (PCR)

Technika založená na metodě PCA, používaná nejčastěji při řešení problému multikolinearity, kdy dvě a více proměnných v modelu je vysoce korelováno (mohou být tedy vzájemně s vysokou přesností dopočítány z ostatních). Princip PCR [30] lze rozdělit do tří kroků:

- Je provedeno PCA a vybrána podmnožina principálních komponent.

- Provede se regrese vektoru pozorovaných výsledků na vybrané principální komponenty jako proměnné s využitím metody nejmenších čtverců. Tímto způsobem je získán vektor regresních koeficientů s počtem dimenzí shodným s počtem vybraných principálních komponent.
- Vektor je zpětně transformován do rozsahu skutečných proměnných z PCA. Takto je získán odhad regresních koeficinetů, charakterizujících originální model, přičemž počet dimenzí je roven počtu vstupních proměnných (principálních komponent).

7.3 Výběr rysů

Feature selection je druhým přístupem, který redukce dimenzionality dosahuje prostřednictvím hledání nejlepší podmnožiny rysů ze zdrojových dat s ohledem na cíl dalšího zpracování. Snažíme se tedy nalézt minimální podmnožinu rysů takovou, že rozdělení pravděpodobnosti různých tříd pro zadané hodnoty těchto rysů je co nejbližší původnímu rozdělení za využití všech rysů (snížení počtu rysů nevede k výraznému poklesu obsažené informace). Oproti metodám feature extraction se je obecně schopen lépe vypořádat s irelevantními informacemi, které by se mohly vyskytnout ve vstupní sadě dat.

Feature selection spojuje algoritmy pro vyhledávání společně s evaluačními metodami. Naivním přístupem je ohodnotit všechny možné kombinace a vybrat z nich nejlepší. Z důvodů vysoké výpočetní náročnosti se však zavádí některé z metaheuristik. Metaheuristické algoritmy lze rozdělit do tří kategorií [28]:

- **Wrapers:** k ohodnocení podmnožin rysů používají prediktivní model. Každá nová podmnožina je použita k natrénování modelu a ten je otestován. Jako vzdálenostní funkce je použita úspěšnost prediktivního modelu na testovacím datasetu. Tento přístup je výpočetně náročný, ale zpravidla poskytuje nejlepší výsledky pro konkréntní model.
- **Filters:** tato třída metod používá jinou měřící veličinu, než je přesnost natrénovaného modelu. Běžně používanými metrikami jsou míra vzájemné informace, Pearsonův korelační koeficient, mimo a vnitro třídní vzdálenost. Bývají rychlé, výpočetně nenáročné, ale výsledek není specifický pro žádný prediktivní model, a tedy ve většině případů poskytuje horší výsledky.
- **Embedded methods:** třída metod, která používá feature selection jako součást kostrukce modelu. Časté je použití se Support vector machines, kdy dochází k opakované konstrukci modelu a odstraňování rysů s nejnižšími váhami.

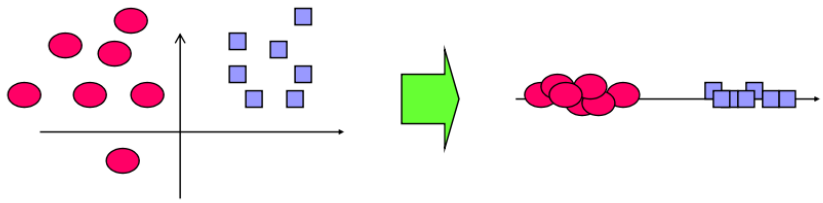
K algoritmům pro feature selection lze zařadit metody postupného výběru / eliminace, simulované žíhání, genetické algoritmy nebo indukci rozhodovacími stromy.

7.3.1 Forward Selection (FS) & Backward elimination (BE)

FS a BE jsou často uplatňované metody pro vyhledávání podmnožiny rysů z vysoce dimenzionálního prostoru. Jako mnoho dalších vyhledávacích algoritmů jsou založeny na technice hladového hill climbingu, a tedy jednotlivé rysy volí sekvenčně ve více iteracích. Rozdíl mezi FS a BE je následující:

- FS začíná s prázdnou podmnožinou a v každé iteraci k ní přidává jeden nový rys, který vede k největšímu zlepšení.
- BE začíná s množinou, rovnající se originální sadě rysů a v každé iteraci odebere jeden rys, jehož odstranění vedlo k největšímu zlepšení.

Hlavním problémem obou přístupů (a mnoha podobných algoritmů) je jejich nenávratnost. Je-li tedy v i-té iteraci vybrán / odstraněn nějaký rys, v následujících iteracích již nemůže být odebrán / vrácen. Tato skutečnost může vést k situacím, kdy zvolíme rysy, které se mohly zdát vhodné v aktuální iteraci, ale v pozdějších iteracích jejich výběr působil rušivě. Pro kvalitní výběr je tedy nezbytné zavést optimalizace.



Obrázek 7.2: Výběr podmnožiny rysů z výchozí datové sady. Převzato z [45].

7.4 Rozhodovací stromy

Rozhodovací stromy jsou pro svou snadnou interpretovatelnost jednou z nejoblíbenějších technik dolování dat. Při výběru rysů mohou sloužit například jako metoda založená na informacích. Každý uzel stromu představuje jeden rys. Jeho konstrukce začíná od kořene, za který je stanoven rys s maximální odlišností od ostatních. Pro určení dalšího větvení stromu se používá hodnoty entropie - míry informační hodnoty daného atributu.

Problémem rozhodovacích stromů je, že mohou vést k výběru množiny, obsahující redundantní rysy. Možným řešením je tzv. regularized tree. Regularized tree je doplněn o penalizace prosřednictvím proměnných, blízkých k proměnným použitých k rozdělení předešlých uzlů stromu až do uzlu aktuálního. Konstrukce stromu probíhá pouze jednou, a tedy lze algoritmus považovat za výpočetně efektivní. Regularized tree navíc řeší vzájemné interakce rysů, nelinearity a je necitlivý k měřítkům vlastností a odlehlým hodnotám (žádá minimální normalizaci). Příkladem takového stromu může být regularized random forest.

7.5 WEKA

WEKA (Waikato Environment for Knowledge Analysis [52]) je balík programů pro strojové učení, implementovaný v jazyce Java. Jedná se o software licencovaný pod GNU, vyvíjený na University of Waikato a široce rozšířený v komerční i akademické sféře. Součástí balíčku jsou nástroje pro datovou analýzu, vizualizaci a prediktivní modelování. V rámci grafického uživatelského prostředí poskytuje úlohy pro předzpracování a dolování z dat, shlukování, klasifikaci, regresi a metody pro výběr rysů.

Kapitola 8

Návrh a implementace

V této kapitole se zaměřím na implementační detaily navrženého prediktoru, založeného na principech fylogenetické analýzy a rozdílnosti ve vybraných vlastnostech aminokyselin. Dále se zaměřím na použité datasety a na návrh provedených experiementů.

8.1 Návrh predikčního nástroje

Nově vytvořený nástroj pro predikci vlivu aminokyselinových substitucí na funkci proteinu byl inspirován algoritmem MAPP [19], publikovaným v roce 2005 E. Stonem a A. Sidowem. Predikční jádro tedy vychází z konceptu fylogenetické analýzy a rozdílnosti ve fyzikálně-chemických vlastnostech mezi původní a substituovanou aminokyselinou. Kroky navrženého algoritmu jsou detailně popsány v následujících kapitolách.

8.1.1 Konstrukce zarovnání a fylogenetického stromu

Fylogenetická analýza pro svou činnost vyžaduje vícenásobné zarovnání homologních sekvencí a k nim zkonstruovaný fylogenetický strom, kde listy stromu odpovídají jednotlivým sekvencím. Součástí fylogenetického stromu musí být také ohodnocení délek jednotlivých větví, které určuje jejich evoluční vzdálenost (sekvence s nižší hodnotou jsou tedy evolučně méně vzdálené).

Jelikož konstrukce vícenásobného zarovnání a fylogenetického stromu není součástí algoritmu, je nezbytné patřičné soubory dodat předpočítány na jeho vstup. Lze využít vybraných nástrojů třetích stran. Syntaktický analyzátor je schopen zpracovat libovolný Newick formát fylogenetického stromu a vícenásobné zarovnání ve formátu FASTA.

Originální implementace algoritmu MAPP využívala nástroj CLUSTALW pro sestavení vícenásobného zarovnání a SEMPHY pro fylogenetický strom. Výsledky, uvedené v kapitole 9, jsem získal s využitím CLUSTALΩ a FastTree pro vícenásobné zarovnání, respektive fylogenetický strom. Pro vyhledávání homologních sekvencí byl použit BLASTp v nastavení na 200 sekvencí s e-value 10^{-12}. Vyhledávání bylo dále zúženo na databázi nr90, popsanou v kapitole 3.3.

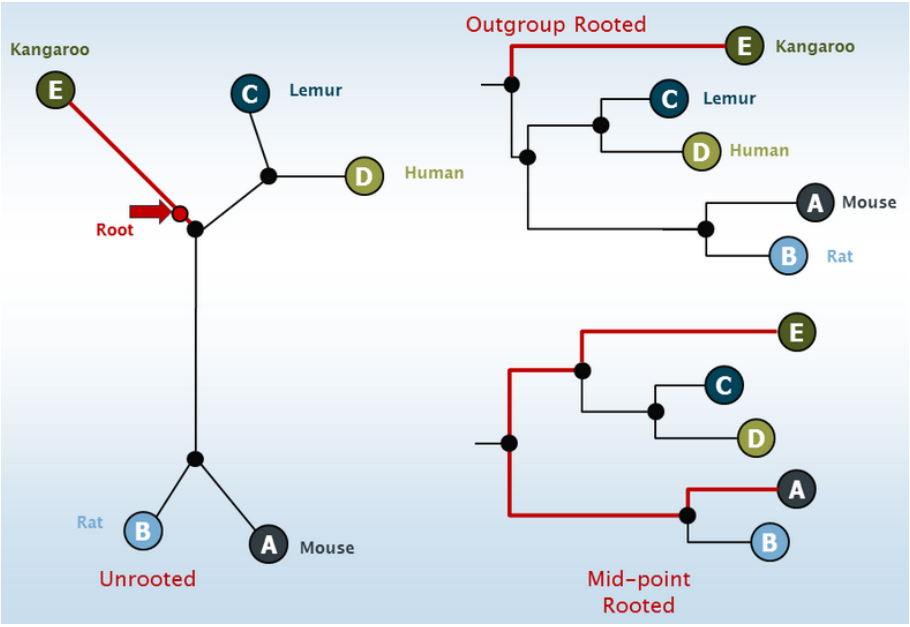
8.1.2 Kořen fylogenetického stromu

Výstupem nástroje FastTree (a většiny ostatních) je nekořenový fylogenetický strom, a tedy ve stromu není vyznačen žádný společný předek pro všechny uzly stromu. Pro uplatnění al-

goritmu, popsaného níže, je však zapotřebí strom nejprve převést do jeho kořenové varianty. K tomuto účelu slouží dvě základní metody:

- **Midpoint rooting:** snaží se ukořenit fylogenetický strom v jeho středu. V prvním kroku tedy spočte vzdálenosti mezi všemi dvojicemi listů a z nich vybere tu nejdelší. Kořen je poté vložen přesně na půl cesty mezi zvolenou dvojicí listů stromu. Tento nově vložený uzel by měl dle myšlenky algoritmu reprezentovat společného předka všech listů stromu. Midpoint rooting je vhodný především pro stromy s relativně konstantní rychlostí evoluce. Dobře pracuje také s vyváženými stromy s blízce příbuznými subtypy, oddělenými od sebe dlouhou středovou větví. Hlavním nedostatkem midpoint algoritmu je náchylnost k velkým odchylkám v evoluční rychlosti, zvláště pokud je fylogenetický strom nevyvážený a tyto velké odchylky se tak vyskytují pouze na jedné straně stromu. Z tohoto důvodu se častěji používá druhá z uvedených metod.
- **Outgroup rooting:** se snaží využít existující znalosti o stromu vložením kořene na vhodné místo prostřednictvím tzv. outgroupu. Outgroup je druh nebo molekula, o níž je předem známo, že je více vzdálená než jakékoliv dva listy stromu (ryba je vzdáleným druhem od fylogenetického stromu savců). Outgroup je ke stromu připojen nejdelší hranou a na tuto je vložen kořen stromu. Outgroup rooting řeší problém stromů s nekonstantní rychlostí evoluce, ale je zapotřebí do stromu vkládat nový, evolučně vzdálený prvek.

Nově implementovaný predikční algoritmus pracuje s fylogenetickým stromem, zkonstruovaným nad množinou homologních sekvencí. Z pochopitelných důvodů tedy byla pro sestavení kořenového stromu použita metoda midpoint rootingu, jelikož rychlost evoluce je s vysokou pravděpodobností relativně konstatní a přidání odlehlé sekvence (outgroup rooting) by mohlo narušit výsledek predikce.



Obrázek 8.1: Převod fylogenetkého stromu na kořenovou formu pomocí metod outgroup a midpoint rooting. Převzato z [66].

8.1.3 Fylogentická analýza

Fylogenetická analýza slouží k určení míry konzervovanosti jednotlivých aminokyselin na určité pozici v sekvenci proteinu, tedy jak často docházelo v průběhu evoluce k jejím mutacím. Jak již bylo vysvětleno v 3.2, z evoluční teorie vyplývá následující: je-li aminokyselina silně konzervovaná, pak jakákoliv mutace na dané pozici vedla ke znevýhodnění organismu a jeho vyřazení z procesu evoluce. Naopak docházelo-li na určité pozici proteinu k častým obměnám, může být tato pozice chápána jako k mutacím otevřená. Míra konzervovanosti je vypočítána z vícenásobného zarovnání homologních sekvencí jednoduchým určením četnosti výskytu jednotlivých aminokyselin ve sloupci zarovnání.

Fylogenetická analýza ale navíc přidává myšlenku, že ne všechny sekvence v zarovnání mají z hlediska evoluce stejnou váhu. Toto poznání vychází z práce Altschul et al. [69], která se zaobírala problematikou vnitřních vztahů a redundance ve zdrojových datech stromu, kdy by několik vysoce korelovaných sekvencí v zarovnání mohlo "přehlasovat"nezávislé sekvence, nesoucí více informací.

Metoda, popsaná v práci Altschul et al., vychází z kořenového fylogenetického stromu. Předpokládá tedy společného předka, od kterého byli potomci odvozeni stochastickými mutacemi, podléhajícími zákonům Brownova pohybu s "časem"úměrným délce hran evolučního stromu. Výpočet vah sekvencí stromu je založen na modifikaci Felsensteinova algoritmu [37] při němž je počítán vážený průměr "nejlepších vah", získaných vkládáním kořene na všechny větve stromu. Výpočet vah s každou pozicí kořene ve stromu může být proveden s lineární časovou složitostí. Po výpočtu těchto vah algoritmus spojí dvojici potomků do jednoho uzlu a posouvá se o úroveň výše, až dokud nezůstane pouze jediný uzel stromu. Vektor, popisující tento poslední uzel, zachycuje váhy všech sekvencí fylogenetického stromu. Modifikovaný Felsensteinův algoritmus se provede pouze jednou na začátku algoritmu a získané váhy jsou posléze využity pro výpočty na všech pozicích vícenásobného zarovnání.

Míra konzervovanosti aminokyselin je z vah a sloupce zarovnání následně spočtena jako $c_m = \sum_{i=1}^{n} w_i 1_{(A_i=\alpha_m)}$, kde c_m značí konzervovanost aminokyseliny m ve zpracovávaném sloupci zarovnání a w_i značí váhu sekvence v zarovnání. Míra konzervovanosti aminokyselin je ovlivněna koeficientem výskytu mezer ve vícenásobném zarovnání jako $c' = c/(1-g)$, kde koeficient g je spočten analogicky k dříve uvedenému vzorci pro výpočet konzervovanosti jako $g = \sum_{i=1}^{n} w_i 1_{(A_i=GAP)}$. Váha mezer je tímto způsobem distribuována proporčně do koeficientu konzervovanosti všech aminokyselin. Závěrem této fáze výpočtu je míra konzervovanosti aminokyselin převedena do nového profilu p jako $p = (1-0,01)c' + (0,01)(1/20)$. Důvodem této úpravy je možnost uniformního zacházení i pro případ plně konzervovaných reziduí.

8.1.4 Fyzikálně-chemické vlastnosti aminokyselin

Společně s analýzou konzervovanosti byla v návrhu predikčního jádra použita míra rozdílnosti ve fyzikálně-chemických vlastnostech aminokyselin. Zde se vychází z předpokladu, že vzájemně podobné aminokyseliny, jako je leucin a isoleucin budou mít menší dopad na funkci proteinu, než aminokyseliny zcela rozdílné, jako je pro příklad glycin a tryptofan. V neoptimalizované verzi implementovaného algoritmu bylo využito šest expertně vybraných vlastnostní, vycházejících z práce Stona a Sidowa (již popsány v oddíle 6.3.9). Tato množina byla později nahrazena výběrem dvanácti nových rysů, popsaných v oddíle 9.6.

V tomto kroku jsou vlastnosti aminokyselin nejprve zaneseny do matice $20 * x$, kde x je počet použitých fyzikálně chemických vlastností. V následujícím kroku je spočten průměr a odchylka sloupce zarovnání. Průměr je zde vypočten jako $M = O^T * W * X$, kde O je

jedničkovým vektorem $20*1$, W bude dále značit matici o rozměrech $20*20$, v níž diagonála odpovídá vektoru konzervovanosti aminokyselin p z oddílu 8.1.3 a X je maticí aminokyselinových vlastností. Odchylka sloupce je dle zavedeného značení následně spočtena jako diagonála z matice $D = (X^T * W * X) - (X^T * W * O * O^T * W * X)$.

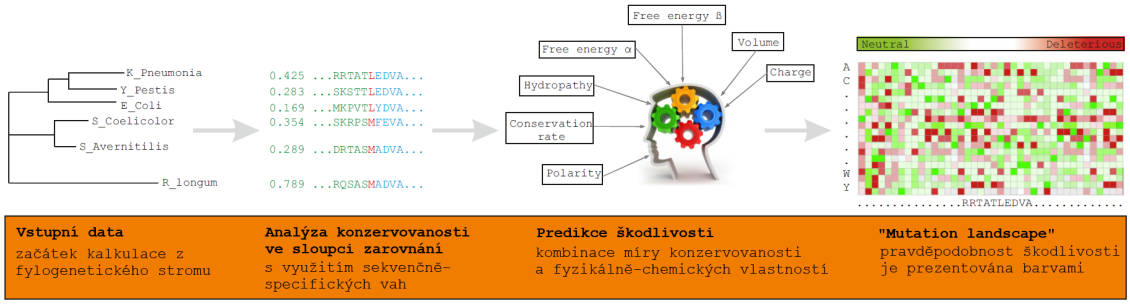
Označíme-li vektor odchylek písmenem d a d_k bude symbolizovat k-tý prvek vektoru, pak d_m budeme považovat za vektor hodnot $1/\sqrt{(d_k)}$. Skóre každé mutace na dané pozici je pak stanoveno diagonálou matice spočtenou jako $S = T * d_m * R^{-1} * d_m * T^T$, kde T značí vzdálenost vlastností dané aminokyseliny od váženého průměru, stanovenou jako $T = X - (O * M)$ a R^{-1} je invertovanou korelační maticí vlastností aminokyselin. Ostatní použité symboly se drží dříve zavedených definic.

V závěrečném kroku je takto spočtený skórovací vektor (určený diagonálou matice S) zapotřebí převést na pravděpodobnostní vyjádření, tzv. impact skóre. Impact skóre (dále označováno jako p-value) značí hodnotu statistického t-testu, v němž nulová hypotéza zní "mutace je neutrální". Tento převod je proveden prostřednictvím Fisherovy distribuce [50].

8.1.5 Rozhraní nástroje

Jak již bylo uvedeno dříve, vstupem nástroje jsou dva povinné soubory: a) vícenásobné zarovnání sekvencí ve formátu FASTA, b) fylogenetický strom v libovolném formátu Newick. K nim může být volitelně doplněn soubor ve standardním textovém formátu, obsahující uživatelsky definovanou sadu fyzikálně-chemických vlastností. Hodnoty pro jednotlivé aminokyseliny musí být pro každou vlastnost zaznamenány ve sloupci a od ostatných sloupců odděleny mezerou. Sada šesti fyzikálně-chemických vlastností tedy musí obsahovat šest sloupců po dvaceti řádcích. Vstupní hodnoty lze nástroji předávat jako parametry prostřednictvím příkazového řádku.

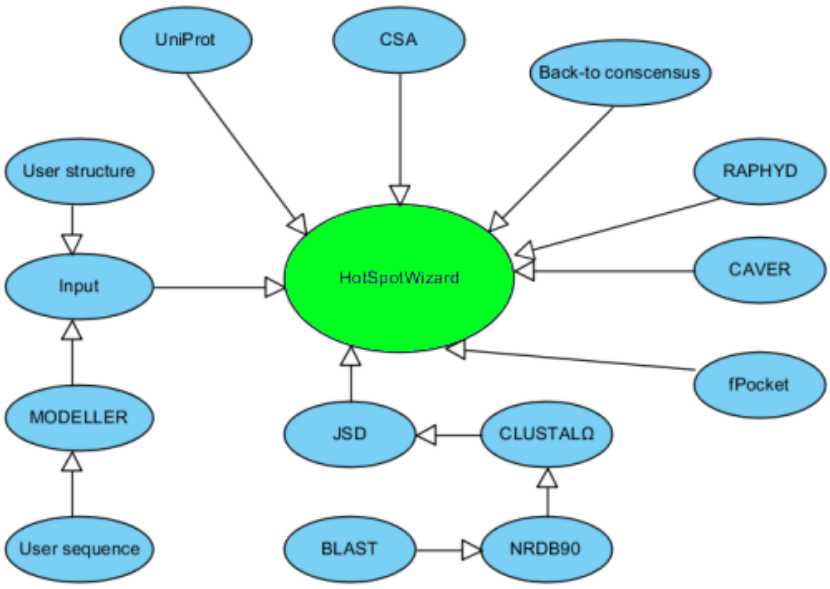
Výstupem programu je sada p-value hodnot vždy pro 20 aminokyselin na jednom řádku, kde číslo řádku odpovídá pozici sloupce ve vícenásobném zarovnání. Formát jednoho řádku výstupu je následující: $aminokyselina : pValue|aminokyselina : pValue|...$ Výstup nástroje lze chápat jako tzv. mutation landscape, tedy matici p-value hodnot o rozměrech 20 * délka zarovnání, kde každá buňka udává pravděpodobnost, že zvolená mutace bude na dané pozici neutrální. Vhodný rozhodovací práh, oddělující mutace predikované jako neutrální od mutací potenciálně škodlivých, je třeba zvolit se zohledněním cílové aplikace. Zjednodušené schéma navrženého algoritmu je graficky znázorněno na obrázku 8.2.



Obrázek 8.2: Průběh výpočetu nástroje RAPHYD.

8.2 HotSpot Wizard

Nově implementovaný predikční nástroj, nesoucí označení RAPHYD, byl integrován do nástroje HotSpot Wizard 2.0, vyvíjeného Loschmidthovými laboratořemi (aktuálně dostupná starší verze 1.7 na adrese http://loschmidt.chemi.muni.cz/hotspotwizard/). HotSpot Wizard [5] je integrační nástroj pro využití v oblasti proteinového inženýrství. Od doby spuštění bylo nástrojem analyzováno přes 5 000 proteinů 800 unikátními uživateli. Jeho účelem je identifikace proteinových reziduí, vhodných k mutagenezi. Tato rezidua mohou být součástí vazebných míst, tunelů, apod. K predikci používá jak sekvenčních, tak i strukturních informací o proteinu. Schéma integrovaných nástrojů je znázorněno níže na obrázku 8.3 společně s jejich stručným popisem.



Obrázek 8.3: Schéma nástrojů integrovaných do nástroje HotSpot Wizard.

Seznam nástrojů a databází, integrovaných do prostředí HotSpot Wizard 2.0:

- **UniProt/SwissProt [2]:** volně dostupná databáze proteinových sekvencí a informací o jejich funkci. Informace o biologické funkci proteinů jsou derivovány z odborné literatury.
- **Catalytic Site Atlas (CSA) [57]:** databáze, dokumentující vazebná místa a katalytická rezidua ve 3D strukturách enzymů. Obsahuje záznamy derivované z literatury a záznamy odvozené srovnáváním homologních sekvencí.
- **Back to consensus:** založeno na myšlence, že vyskytuje-li se na pozici v zarovnání s velkou četností aminokyselina jiná, než jaká je umístěna v originální sekvenci, pak mutace za tuto často se vyskytující aminokyselinu způsobí pravděpodobněji zvýšení stability.
- **RAPHYD:** nově vyvíjený nástroj pro predikci efektu aminokyselinových substitucí, který je náplní této práce.

- **CAVER [21]:** nástroj pro analýzu a vizualizaci tunelů (cesty vedoucí od dutiny uvnitř proteinu k jeho povrchu) a kanálů (cesty vedoucí napříč celým proteinem) v proteinových strukturách.
- **fPocket [79]:** nástroj pro detekci aktivních míst a dutin na povrchu proteinů.
- **JSD [32]:** slouží k analýze konzervovanosti.
- **CLUSTALΩ:** nástroj pro konstrukci vícenásobného zarovnání (viz 4.3.1).
- **NRDB90:** databáze, obsahující sekvence s maximálně devadesáti procentní identitou (viz 3.3).
- **BLAST:** nástroj pro vyhledávání homologních sekvencí v určené databázi (viz 3.4.1).
- **MODELLER [4]:** program, sloužící k homologním modelování trojrozměrné struktury proteinů.

8.3 Použité datasety

Implementace byla testována na čtyř datasetech, celkově čítajících 74 192 mutací nad 16 256 sekvencemi. Tabulka 8.1 poskytuje numerický přehled jednotlivých datasetů. Jejich podrobný popis je pak uveden v oddílech níže.

Tabulka 8.1: Přehled testovacích datasetů.

Dataset	**Mutace**			**Sekvence**
	Neutrální	**Škodlivé**	**Celkem**	
MMP	7 538	4 456	11 994	13
PMD	1 248	2 249	3 497	1 406
BSIFT	3 081	11 738	14 819	4 036
PredictSNP	24 082	19 800	43 882	10 801
Celkem	34 921	43 729	74 192	16 256

8.3.1 Dataset PredictSNP

Dataset PredictSNP [33] byl zkompilován z pěti různých zdrojů. Konkrétně se jedná o:

- SNPs&GO dataset, obsahující celkem 58 057 mutací, získaných z databáze SwissProt.
- MutPred dataset, čítající 65 654 mutací zkompilovaných z databáze SwissProt a HGMD.
- PON-P dataset, obsahující 39 670 mutací, vzniklý spojením dbSNP, PhenCode, IDbases a šestnácti nezávislými místně-specifickými databázemi.
- HumVar dataset, čítající 41 918 mutací ze SwissProt a dbSNP.
- Humsavar, obsahující 36 994 mutací, nacházejících se v záznamech UniprotKB/SwissProt.

Spojení všech uvedených datasetů vedlo na velké množství duplicit. Pro sestavení plně nezávislého datasetu tedy byly uplatněny následující postupy:

- páry mutací s konfliktem ve funkcionálních anotacích, kdy v jedné byla mutace označena jako škodlivá a ve druhé jako neutrální, byly odstraněny.
- byly odstraněny veškeré duplicitní mutace.
- byly eliminovány všechny mutace, které se vyskytovaly v trénovacích sadách nástrojů, testovaných v rámci studie [33]
- byly odstraněny také všechny mutace, vyskytující se na překrývajících pozicích s mutacemi v trénovacích datasetech (za překrývající se byly považovány fragmenty ve dvou sekvencích zarovnaných nástrojem BLAST s e-value 10^{-10}, kdy zarovnané fragmenty dosahovaly alespoň 50 % identity).
- mutace, vyskytující se v ostatních zmíněných datasetech (PMD, MMP, BSIFT) byly rovněž vyloučeny.

8.3.2 Dataset PMD

Dataset PMD (The Protein Mutant Database) byl derivován ze stejnojmenné databáze [76], jejíž poslední aktualizace byla provedena v roce 2007. Ke konstrukci databáze byly využity informace z více jak 10 000 článků a obsahuje experimentální informace o efektu 165 800 SNP mutací na aktivitu proteinu, jeho stabilitu a spojitosti s případnými genetickými onemocněními. Záznamy v této databázi jsou doplněny o notace "[=]"/ "[+]"/ "[-]". Záznamy označeny jako "[=]"jsou považovány za neutrální, ostatní za škodlivé. Důvodem je, že zvýšená aktivita proteinu může mít v živém organismu stejně negativní dopady jako její snížení.

Při konstrukci cílového datasetu byly z databáze PMD vyřazeny všechny duplicity a dále záznamy, kdy docházelo ke konfliktu v anotacích (např. jestliže existovaly dva duplicitní záznamy, kdy jeden byl označen jako "[=]"a druhý "[+]"). Podobně jako v případě PredictSNP datasetu byly závěrem vyřazeny také všechny mutace, překrývající se pozicemi s trénovacími datasety nástrojů testovaných ve studii [33].

8.3.3 Dataset MMP

Dataset MMP *(Massively Mutated Proteins)* sestává z pouhých třinácti silně promutovaných proteinů. Jedenáct sekvencí vychází ze studie Yampolského a Stolzfuse [49] (dvanáctá byla vyřazena z důvodu krátké délky - pouhých 30 aminokyselin). Zbývající dvě proteinové sekvence, doplňující MMP dataset, byly získány z patentů, vydaných společností Danisco Inc. Konkrétně se jedná o patenty, popisující vliv mutací na serinovou proteázu z organizmu Bacillus subtilis [84] a α-amylázu z organizmu Geobacillus stearothermophilus [83]. Originální dataset těchto třinácti proteinů obsahoval 16 500 mutací a podobně jako v případě PMD datasetu, pouze mutace zachovávající stejnou úroveň proteinové aktivity jako v případě nemutovaného proteinu byly označeny jako neutrální. Mutace, vyskytující se na pozicích, překrývajících se s trénovacími sadami nástrojů, testovaných ve studii [33], byly po vzoru dřívějších datasetů odstraněny.

8.3.4 Dataset BSIFT

Dataset BSIFT vychází ze studie Lee at al. [86] Sestává z množiny experimentálně ověřených mutagenezí, extrahovaných z databáze SwissProt. Rozdělení na neutrální a škodlivé mutace bylo provedeno na základě slov, vyskytujících se v textovém poli MUTAGEN. Rozpoznání slov bylo učiněno pouze na první dvě slova popisu a nebylo-li identifikováno žádné z klíčových slov, mutace byla z cílového datasetu vyřazena. Seznam klíčových slov je uveden v tabulce 8.3.4.

Tabulka 8.2: Přehled použitých klíčových slov.

Podmnožina	**Počet výskytů**	**Klíčová slova**
Aktivační mutace	511	increase, enhance, activat, constitutive acti, restore
Neutrální mutace	2 606	no effect, no change, normal, mild, minimal effect, minor, small effect, wild-typ
Škodlivé mutace	11 867	decrease, inhibit, reduc, loss, lower, abolish, abrogate, inactive, diminish, disrupt, impair, eliminate, no activity, prevent, suppress, increases km, increases the km

8.4 Návrh experimentů

Přesnost nově implementovaného nástroje byla měřena na všech výše uvedených datasetech MMP, PMD, BSIFT a PredictSNP. V úvodním experimentu byl pro vyhledání homologních sekvencí použit nástroj BLASTp v nastavení pro maximální počet 200 homologů a prahem e-value 10^{-12}. Vícenásobné zarovnání sekvencí bylo zkonstruováno prostřednictvím nástroje CLUSTALΩ a k němu náležející fylogenetický strom pomocí FastTree 2.0. Tato výchozí konfigurace se po provedení dalších experimentů ukázala jako nejefektivnější v poměru přesnosti k požadavkům na výpočetní čas. Za účelem dosažení nejvyšší míry přesnosti vyvíjeného nástroje byly vyzkoušeny následující optimalizace:

- **Optimalizace nástrojů třetích stran:** nástroj BLASTp pro vyhledávání homologních sekvencí z databáze nr90 byl testován v osmi různých konfiguracích s prahem e-value nastaveným na hodnotu 10^{-6} / 10^{-12} a maximálním počtem homologům na 50, 100, 150 nebo 200 sekvencí. Nad výstupem všech osmi konfigurací BLASTp byly pro účel volby nástroje pro konstrukci vícenásobného zarovnání otestovány MUSCLE a CLUSTALΩ. Výsledky experimentu jsou dále rozvedeny v oddíle 9.3.

- **Optimalizace rozhodovacího prahu:** záměrem tohoto druhého experimentu bylo nalezení optimálního prahu pro hodnotu p-value, dle které algoritmus rozhoduje, zda mutaci označit za neutrální, či škodlivou. Testované prahové hodnoty byly voleny z intervalu <0,01; 0,1> s krokem 0,01. Detaily experimentu jsou zachyceny v oddíle 9.4.

- **Optimalizace fyzikálně-chemických vlastností aminokyselin:** účelem experimentu bylo vybrat novou, vhodnější sadu vlastností aminokyselin jako alternativu k šesti expertně vybraným rysům, přejatých z práce Stona a Sidowa. Jako zdrojová datová sada byla použita databáze AAIndex, popsaná níže v oddíle 8.5. Nad touto

databází byly uplatněny dva přístupy - a) výběr rysů založený na informacích, b) výběr rysů založený na experimentech. Oba přístupy jsou i s výsledky podrobněji zachyceny v oddíle 9.6.

8.5 Databáze AAIndex

AAIndex [72] je databází fyzikálně-chemických a biochemických vlastností aminokyselin. Sdružuje v sobě výsledky, pocházející z článků datovaných až do roku 1943 a v současné době databáze existuje ve třech variantách:

- **AAIndex1:** základní verze databáze, v současné době čítající celkem 544 záznamů s numerickým vyjádřením vlastností 20 aminokyselin. Databáze je vysoce redundantní. Důvodem redundance je, že mnohé vlastnosti byly převzaty z více různých publikací, popisujících odlišné experimenty z různých laboratoří, apod. Po provedení korelační analýzy lze reálný počet aminokyselinových vlastností odhadnout na počet nižší, než 200.
- **AAIndex2:** k současnému datu (duben 2015) obsahuje 98 záznamů mutačních matic. Formát zápisu je podobný formátu databáze AAIndex1, ale namísto 20 numerických hodnot (jedna pro každou aminokyselinu) obsahuje 210 hodnot (20 na diagonále a $20 * 19/2$ mimo hlavní diagonálu) pro symetrickou matici nebo 400 a více hodnot pro matice nesymetrické.
- **AAIndex3:** k současnému datu (duben 2015) obsahuje 47 záznamů matic kontaktních potenciálů aminokyselin. Formát záznamu je téměř identický s AAIndex2.

Jak již bylo zmíněno v 8.4, v poslední fázi optimalizace nově vyvíjeného prediktoru byla databáze AAIndex použita jako zdrojová množina fyzikálně chemických vlastností pro problém výběru nové množiny rysů, kterým se budu hlouběji zaobírat v oddíle 9.6. Konkrétně pro výběr rysů posloužila databáze AAIndex1, jejíž formát je znázorněn na obrázku 8.4.

```
H ANDN920101
D alpha-CH chemical shifts (Andersen et al., 1992)
R LIT:1810048b PMID:1575719
A Andersen, N.H., Cao, B. and Chen, C.
T Peptide/protein structure analysis using the chemical shift index method:
  upfield alpha-CH values reveal dynamic helices and aL sites
J Biochem. and Biophys. Res. Comm. 184, 1008-1014 (1992)
C BUNA790102    0.949
I    A/L     R/K     N/M     D/F     C/P     Q/S     E/T     G/W     H/Y     I/V
    4.35    4.38    4.75    4.76    4.65    4.37    4.29    3.97    4.63    3.95
    4.17    4.36    4.52    4.66    4.44    4.50    4.35    4.70    4.60    3.95
//
```

Obrázek 8.4: Formát záznamu v databázi AAIndex1. Značení: (H) identifikátor; (D) název vlastnosti včetně autora a roku vydání publikace; (R) identifikace v PubMed; (A) seznam autorů; (T) název článku; (J) vydavatel a rok vydání; (C) seznam dalších záznamů v AAIndex1 s korelačním koeficientem vyšším než 0,8; (I) 20 numerických hodnot popisované vlastnosti pro jednotlivé aminokyseliny.

Kapitola 9

Výsledky

V předcházejícím oddíle jsem rozvedl implementační detaily nově navrženého nástroje pro predikci vlivu aminokyselinových substitucí na funkci proteinu, dále označovaného jako RAPHYD (RApid PHYlogenetic predictor of Deleteriousness). V této závěrečné kapitole shrnu výsledky základního testování, provedeného na nástroji RAPHYD před jeho optimalizací a dále popíši jednotlivé kroky proběhlé optimalizace. Konkrétně byl nástroj optimalizován výběrem nástrojů třetích stran, volbou rozhodovacího prahu a výběrem nové sady fyzikálně-chemických vlastností. Součástí kapitoly je rovněž srovnání s ostatními již existujícími predikčními nástroji.

9.1 Použité metriky

V tomto oddílu textu stručně popíši zkratky a pojmy, používané níže v rámci této kapitoly a dále v přílohách. Jedná se o:

- **Neutrální mutace:** taková, která nepoškodí funkci proteinu.
- **Škodlivá mutace:** taková, která způsobí poškození funkce proteinu.
- **TP:** značí True Positive, tedy škodlivou mutaci, která byla správně predikována jako škodlivá.
- **TN:** značí True Negative, tedy neutrální mutaci, která byla predikována jako neutrální.
- **FP:** značí False Positive, tedy neutrální mutaci predikovanou jako škodlivou.
- **FN:** značí False Negative, tedy škodlivou mutaci predikovanou jako neutrální.
- **Pokrytí:** poměr ohodnocených mutací vůči všem mutacím v datasetu.
- **Přesnost:** poměr správně a chybně predikovaných mutací. Přesnost predikce je spočtena jako $(TP + TN)/(TP + TN + FP + FN)$.
- **Normalizovaná přesnost:** v tabulkách bude dále značena jako P. norm. a je vypočítána jako $[TP/(TP + FN) + TN/(TN + FP)]/2$. Důvodem zavedení normalizované přesnosti je její necitlivost k problému nevyvážených datasetů, obsahujících například výrazně vyšší počet neutrálních mutací než škodlivých.

9.2 Základní testování

Před zahájením optimalizačních experimentů, jejichž výsledky uvedu níže v rámci této kapitoly, byla provedena sada testů nad všemi čtyřmi datasety PredictSNP, MMP, PMD a BSIFT (viz 8.3) v základní konfiguraci nástrojů třetích stran, rozhodovacího prahu a zvolených fyzikálně-chemických vlastností aminokyselin.

Výchozí konfigurace nástrojů třetích stran byla inspirována pracovním prostředím nástroje HotSpot Wizard [5]. Pro vyhledávání homologních sekvencí v databázi nr90 byl použit nástroj BLASTp. BLASTp byl v průběhu experimentu testován celkem v osmi variacích. Prahová e-value byla nastavena na 10^{-6} nebo 10^{-12} a v databázi nr90 bylo k zadané sekvenci vyhledáno maximálně 50, 100, 150, či 200 homologů. Jestliže BLASTp nalezl v databázi více, jak požadovaný maximální počet homologních sekvencí, pak byl maximální počet vybrán uniformně napříč všemi nalezenými sekvencemi, sežazenými dle jejich e-value. Pro konstrukci vícenásobného zarovnání a fylogenetického stromu byly využity nástroje CLUSTALΩ a FastTree 2.

Výchozí hodnota rozhodovacího prahu (p-value threshold) 0,01 a výchozí fyzikálně-chemické vlastnosti aminokyselin byly převzaty z práce Stona a Sidowa [19]. Ve shodě s 6.3.9 se tedy jedná o následujících šest vlastností: hydropatie [39], polarita [42], náboj [42], objem postranního řetězce [1], volná energie α-šroubovicové konformace [81] a volná energie β-listové konformace [81].

Tabulka 9.2 znázorňuje výsledky, sloučené ze všech čtyř datasetů MMP, PMD, PredictSNP a BSIFT, v osmi výše uvedených konfiguracích (maximum 50 / 100 / 150 / 200 homologů, e-value 10^{-6} / 10^{-12}). Detailní výsledky pro nesloučené datasety jsou k nahlédnutí v příloze A. Z výsledků je patrné, že experimenty s BLASTp v konfiguraci 10^{-12} dosahují systematicky lepších výsledků, než při e-value prahu 10^{-6} (normalizovaná přesnost je při sloučení všech datasetů v průměru o 0,01 vyšší ve prospěch e-value prahu 10^{-12}).

Nejlepší výsledky mohou být dále pozorovány ve variantách s BLASTp nastaveným na maximální počet 100 a 200 homologů (při sloučení datasetů je normalizovaná přesnost v obou případech 0,667). Průměrné pokrytí nad sloučenými datasety se pohybuje v rozmezí od 0,704 do 0,788. Důvody pro nízké pokrytí jsou dvojí:

- Pro některé proteiny a konfigurace BLASTp nebyl nástroj schopen dohledat žádné homologní sekvence v nr90 databázi, a tedy nebylo možné zkonstruovat fylogenetický strom a vícenásobné zarovnání, požadované jako vstup predikčního algoritmu.

- Evaluované mutace, vyskytující se na pozicích s více jak 50 % mezer ve sloupci vícenásobného zarovnání byly zanedbány. Z výsledků je zřetelný vliv podmínky, vylučující na mezery bohaté pozice, kdy experimenty s e-value prahem 10^{-12} tíhly k vyšší úrovni pokrytí, v průměru přibližně o 0,03 nad experimenty s e-value prahem 10^{-6}. Tento trend je způsoben faktem, že s méně striktním e-value prahem nachází BLASTp vzdálenější homology s nižší mírou identity a v zarovnání tak vzniká více na mezery bohatých pozic. K podobnému efektu dochází při vyšším počtu homologů ve vícenásobném zarovnání (při maximu 200 homologů a prahu e-value 10^{-12} je při vícenásobném zarovnání originální sekvence o délce 500 aminokyselin roztažena v průměru na 3 500 pozic).

Tabulka 9.1: Výsledky základního testování, sloučené z experimentů nad datasety MMP, PMD, PredictSNP a BSIFT. Pro účely testování byly využity nástroje CLUSTALΩ a FastTree 2.0 ke konstrukci vícenásobného zarovnání a fylogenetického stromu. Byl použit výchozí rozhodovací práh 0,01 a sada šesti expertně zvolených vlastností aminokyselin, převzatých z [19].

Počet/eval.	**50/1e-6**	**50/1e-12**	**100/1e-6**	**100/1e-12**	**150/1e-6**	**150/1e-12**	**200/1e-6**	**200/1e-12**
TN	22 164	22 730	21 458	22 038	20 621	21 319	20 277	20 901
FP	3 670	4 228	3 331	4 165	3 206	4 057	3 152	3 909
TP	13 773	15 218	13 274	15 082	12 926	14 463	12 882	14 332
FN	16 900	16 303	16 462	15 560	16 108	15 363	15 921	14 863
Pokrytí	0,762	0,788	0,735	0,766	0,712	0,744	0,704	0,728
Přesnost	0,636	0,649	0,637	0,653	0,635	0,648	0,635	0,652
P. norm.	0,653	0,663	0,656	0,667	0,655	0,663	0,656	0,667

9.3 Výběr nástrojů třetích stran

Volba nástroje pro konstrukci fylogenetického stromu se opírá o článek [53]. Z tabulky 9.2 je patrné, že nástroj FastTree je výrazně přesnější, než minimum-evolution a parsinomy metody, ale ne tak přesný jako nástroje využívající strojové učení a SPR. Dle tabulky 9.3 je nástroj FastTree naopak řádově rychlejší než nástroje založené na bázi strojového učení. Protože rychlost nově implementovaného prediktoru byla jedním z hlavních kritérií, FastTree byl zvolen jako nástroj s optimálním kompromisem mezi rychlostí zpracování a kvalitou výsledku.

Tabulka 9.2: Srovnání kvality (přesnosti) nástrojů pro konstrukci fylogenetického stromu. Převzato z [53].

Dataset	**250**	**1,250**	**5,000**	**78,132**
Metoda	**a.a.**	**a.a.**	**a.a.**	**nt.**
RAxML 7 (JTT+CAT, SPRs)	90,5%	88,4%	88,4%	-
PhyML 3.0 (JTT+I, SPRs)	89,9%	-	-	-
FastTree 2.0.0	86,9%	83,7%	84,3%	92,1%
PhyML 3.0	86,0%	-	-	-
FastMe	80,5%	78,8%	77,0%	-
FastTree 2.0.0	80,4%	78,3%	76,6%	91,4%
BIONJ	77,7%	73,7%	73,1%	-
Parsinomy (RAxML)	76,8%	76,5%	69,4%	-
Neighbour joining	76,0%	72,6%	71,6%	66,1%
Clearcut	75,5%	72,3%	71,5%	58,1%

Implementovaný predikční nástroj byl dále rozsáhle testován pro určení nejvhodnějšího softwaru ke konstrukci vícenásobného zarovnání. Srovnání bylo provedeno mezi nástroji MUSCLE a CLUSTALΩ. Pro vyhledání homologních sekvencí byl v případě obou nástrojů využit BLASTp v konfiguraci pro 50, 100, 150 a 200 homologů s e-value 10^{-6} a 10^{-12}.

Tabulky 9.4 a 9.5 zobrazují srovnání mezi vybranými nástroji na datasetu MMP. Z vý-

Tabulka 9.3: Srovnání rychlosti nástrojů pro konstrukci fylogenetického stromu. Převzato z [53].

Dataset	Sekvence	FastTree 2.0	RAxML 7	PhyML
16s rRNA, subsets	500	0,02h	2,2h	2,9h
COGs, subsets	500	0,02h	5,2h	7,2h
COGs, subsets	2 500	0,11h	61h	-
Efflux permeases	8 362	0,25h	197h	>1 200h
16S rRNAs, families	15 011	0,66h	64h	>2,000h
ABC transporters	39 092	1,02	-	-
16S rRNAs, all	237 882	21,8h	-	-

sledků je patrné, že nástroj MUSCLE dosáhl nepatrně vyšší přesnosti, v průměru přibližně o 0,001. Přesnost prediktoru dosahuje maxima při konfiguraci na maximálně 200 homologních sekvencích a e-value 10^{-12}, kde MUSCLE vede v přesnosti nad CLUSTALΩ o 0,008. Tento zanedbatelný nárust přesnosti je však vyvážen nepřijatelnými časovými nároky na konstrukci vícenásobného zarovnání.

Na základě provedených experimentů s nástroji třetích stran byly závěrem zvoleny nástroje identické s těmi, uplatněnými již při základním testování. Konkrétně se tedy jedná o nástroj CLUSTALΩ pro sestavení vícenásobného zarovnání a FastTree 2.0 pro konstrukci fylogenetického stromu (nástroje shodné s rozhraním HotSpot Wizard 2.0). Výsledky plynoucí z tohoto kroku optimalizace byly tedy uvedeny již v tabulce 9.2.

Tabulka 9.4: Přesnost nástroje RAPHYD, integrujícího různé nástroje pro vícenásobného zarovnání. Pro účely testování byl použit výchozí rozhodovací práh 0,01 a sada šesti fyzikálně-chemických vlastností.

Nástroj	Počet/eval.	50	100	150	200
CLUSTALΩ	**1e-6**	0,680	0,680	0,678	0,676
	1e-12	0,679	0,702	0,687	0,688
MUSCLE	**1e-6**	0,681	0,680	0,676	0,676
	1e-12	0,695	0,700	0,676	0,696

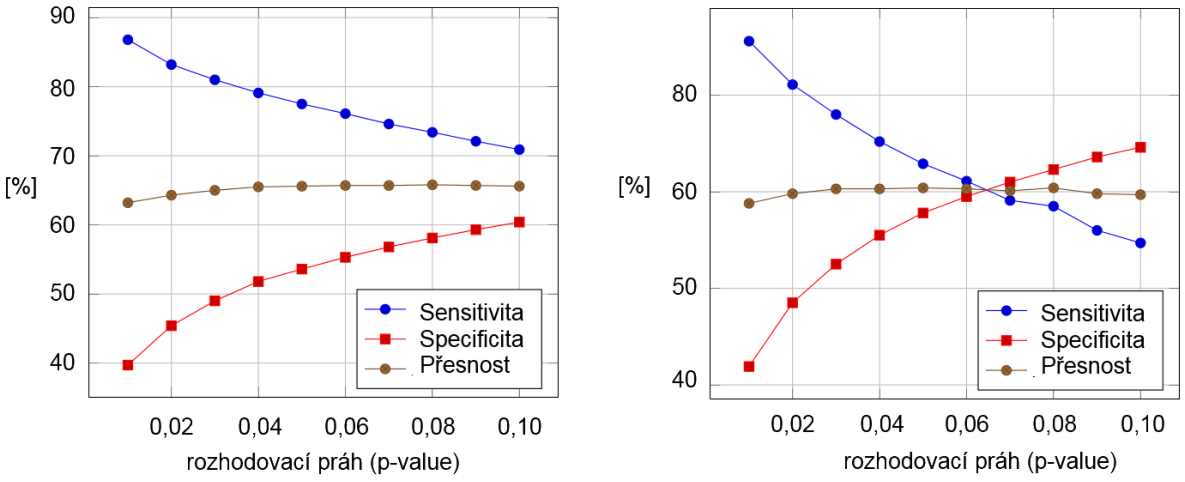
Tabulka 9.5: Přesnost nástroje RAPHYD, integrujícího různé nástroje pro vícenásobného zarovnání. Pro účely testování byl použit výchozí rozhodovací práh 0,01 a sada šesti fyzikálně-chemických vlastností.

Nástroj	Počet/eval.	50	100	150	200
CLUSTALΩ	**1e-6**	15 s	39 s	41 s	57 s
	1e-12	11 s	30 s	36 s	48 s
MUSCLE	**1e-6**	70 s	253 s	435 s	715 s
	1e-12	60 s	211 s	367 s	606 s

9.4 Vliv zvoleného rozhodovacího prahu

Všechny dosud uvedené výsledky byly získány za uplatnění výchozího rozhodovacího prahu 0,01. Z tabulky 9.2 je však patrné, že v této konfiguraci vede prediktor často k falešně negativním výsledkům (škodlivé mutace jsou tedy označovány za neutrální). Tato kapitola se tedy zaobírá otázkou, zda vůbec a nakolik je možné navýšit přesnost prediktoru pouhým zvýšením rozhodovacího prahu. Grafy na obrázku 9.4 znázorňují změnu v přesnosti nad MMP a PredictSNP datasetem s deseti různými rozhodovacími prahy v intervalu ⟨0,01; 0,1⟩. Po deseti hodnotách byly testovány rovněž i intervaly ⟨0,001; 0,01⟩ a ⟨0,1; 1,0⟩, zde však docházelo k rapidnímu poklesu přesnosti predikce.

Obdržené výsledky naznačují, že vyšší rozhodovací práh vede k výraznému navýšení přesnosti oproti predikci se standardním prahem 0,01. Konkrétněji, nárust přesnosti byl až 1,6 % v datasetu MMP s vrcholem okolo hodnoty 0,05 (stejná přesnost byla naměřena i s rozhodovacím prahem 0,08, ale 0,05 se jeví být globálním maximem na intervalu ⟨0,01; 0,1⟩). V případě PredictSNP datasetu došlo k navýšení přesnosti až o 2,6 % s maximem okolo hodnoty 0,08. Z provedeného experimentu lze usoudit, že vliv vyššího rozhodovacího prahu na přesnost predikce je nezanedbatelný. Optimální hodnota rozhodovacího prahu by však měla být vždy nastavena se zohledněním zamýšleného použití. V některých aplikacích je hodnota sensitivity (míry korektně rozpoznaných škodlivých mutací) důležitější, než celková přesnost predikce. Pro příklad ve zdravotnictví je v diagnostických testech důležité vždy odhalit genetické onemocnění způsobující mutace (chybně pozitivní test je přijatelnější, než-li neléčená choroba) a naopak v mnohých aplikacích proteinového inženýrství může být výhodnější vyřadit pouze mutace s nejjistější pravděpodobností jejich škodlivosti. Rozdíly mezi mírou sensitivity a specificity mohou být pozorovány v grafech na obrázku 9.4, případně detailněji v tabulkách v příloze B.



Obrázek 9.1: Grafy, zachycující vývoj sensitivity, specificity a normalizované přesnosti pro různé hodnoty rozhodovacího prahu pro PredictSNP (vlevo) a MMP (vpravo) dataset.

9.5 Vliv sloupců bohatých na mezery

V důsledku konstrukce vícenásobného zarovnání dochází ke vzniku značného množství sloupcových pozic, které jsou pouze z útlého zlomku tvořeny aminokyselinami a u většiny sekvencí v zarovnání byla na danou pozici vložena mezera. Při evaluaci základních experimentů 9.2 byla uplaněna podmínka, aby pouze pozice s méně jak 50 % mezer byly zohledněny. Vzniká otázka, nakolik tato limitace ovlivnila přesnost predikce. Výsledky, zachycující oba případy (exkluze na mezery bohatých pozic / evaluace všech pozic), jsou pro datasety MMP a PredictSNP uvedeny v tabulce 9.6. Rozhodovací práh byl nastaven na hodnotu 0,1.

V MMP datasetu došlo pouze k zanedbatelnému rozdílu v přesnosti predikce. V případě PredictSNP datasetu pak rozdíl představuje přibližně 1 % ve prospěch uplatnění podmínky, vyřazující z evaluace sloupce s alespoň 50 % mezer. Z uvedených výsledků tedy můžeme usoudit, že stanovené omezení výrazně neovlivňuje přesnost predikce. Uplatnění podmínky navíc vede k neohodnocení přibližně třetiny testovaných mutací.

Tabulka 9.6: Vliv podmínky eliminace na mezery bohatých pozic ve vícenásobném zarovnání (G značí použití této podmínky, N pak její vynechání).

Dataset	**MMP/N**	**MMP/G**	**PredictSNP/N**	**PredictSNP/G**
TP	4 936	4 658	17 464	10 313
FN	2 602	2 541	6 447	4 235
TN	3 294	3 244	11 038	8 280
FP	1 162	1 104	8 634	5 431
Pokrytí	1,000	0,963	1,000	0,640
Přesnost	0,686	0,684	0,654	0,658
P. norm.	0,697	0,697	0,646	0,656

9.6 Výběr rysů

Dosud byly výsledky všech experimentů získány s výchozí sadou fyzikálně-chemických vlastnostní aminokyselin, popsaných již v 6.3.9. V této sekci se zaměřím na otázku, zda vůbec a nakolik je možné navýšit přesnost prediktoru změnou množiny nástrojem zohledňovaných vlastností aminokyselin.

Pro výběr rysů byl zvolen soubor fyzikálně-chemických vlastnostní z databáze AAIndex1 (viz 8.5). Tato databáze obsahuje celkem 544 rysů a vyznačuje se vysokou redundancí, způsobenou vícenásobnými výskyty instancí jednoho rysu, pocházejících z experimentů provedených různými laboratořemi, týmy, či v jiném roce, atd. Redundantní rysy jsou vzájemně vysoce korelované, ale nejsou identické (odchylky přístroje, měření, ...). Navíc v databázi nenesou stejné označení a je tudíž obtížné související rysy slučovat. Pro výběr rysů nad uvedenou databází byly navrženy dvě techniky, založené na experimentech a informační hodnotě jednotlivých rysů.

Metoda založená na informacích

První uplaněný přístup vycházel z myšlenky, že výběr podmnožiny atributů s vysokou vzájemnou ortogonalitou povede k lepším výsledkům. Pro ověření této hypotézy bylo 544 vlastností z databáze AAIndex nejprve rozděleno do pěti až deseti shluků. Pro roztřídění vlastností do požadovaného počtu shluků bylo využito algoritmu K-means.

K-means [43] je nehierarchický algoritmus, v němž je nezbytné zadat počet shluků před zahájením algoritmu. K-means reprezentuje třídu shluku pomocí "fiktivního"centrálního bodu. Na počátku náhodně vybere k objektů, které reprezentují jednotlivé shluky a ostatní prvky rozdělí do jednotlivých shluků na základě podobnosti (v tomto případě je rys tvořen 20 reálnými čísly, odpovídajícími hodnotám fyzikálně-chemické vlastnosti 20 standardních aminokyselin). Ve druhém kroku algoritmus určí nové středy shluků, které spočte jako průměrnou hodnotu rysů uvnitř shluku. Algoritmus znovu roztřídí rysy do shluků dle podobnosti (blízkosti ke středu shluku) a ukončí se, pokud žádný objekt nebyl přeřazen do jiného shluku nebo algoritmus dosáhl maximálního počtu iterací.

Po rozdělení množiny 544 rysů do 5, 6, 7, 8, 9 nebo 10 shluků na základě jejich podobnosti (dle naměřených parametrů 20 standarních aminokyselin) byl z každého shluku vybrán právě jeden rys. Pro šest variací experimentu s pěti až deseti shluky bylo provedeno vždy 20 000 testů a vybrána nejlepší nalezená řešení. Interval pěti až deseti shluků byl zvolen na základě existujících nástrojů pro predikci vlivu aminokyselinových substitucí, které zpravidla využívají počet fyzikálně-chemických vlastností z uvedeného rozsahu.

Tabulka 9.7 znázorňuje nejlepší dosaženou přesnost prediktoru s parametry vybranými z 5, 6, 7, 8, 9 a 10 shluků v porovnání s přesností algoritmu se šesti expertně vybranými fyzikálně-chemickými vlastnostmi. Experiment byl proveden s výchozím rozhodovacím prahem 0,01 na MMP datasetu. Z tabulky 9.7 je patrné, že nejvyšší úspěšnosti dosahoval prediktor s pouze pěti vybranými rysy. Důvodem je zřejmě skutečnost, že tento počet je minimálním pro kalkulaci přijatelného množství informace o diverzifikaci jednotlivých aminokyselin (rychlý experiment se třemi a čtyřmi podával systematicky výsledky blížící se k úspěšnosti 0,5, tedy k přesnosti rovnající se náhodě) a současně maximální počet, aby do podmnožiny rysů nebyl náhodným výběrem zanesen nevhodný rys. Závěrem experimentu však lze prohlásit, že ani nejlepší dosažený výsledek s pěti zvolenými rysy se nevyrovnal přesnosti predikce s použitím šesti expertně zvolených vlastnostní (0,663 oproti 0,688, tedy pokles přibližně o 0,025) a v důsledku lze tuto metodu označit za neefektivní.

Tabulka 9.7: Souhrn nejlepších výsledků, dosažených s využitím $\{5, 6, 7, 8, 9, 10\}$ shluků (měřeno na MMP datasetu). Predikční nástroj byl použit s výchozím rozhodovacím prahem.

Shluky	**Orig.**	**5**	**6**	**7**	**8**	**9**	**10**
P. norm.	0,688	0,663	0,651	0,652	0,660	0,658	0,656

Metoda založená na experimentech

Tato druhá metoda byla založena na kombinaci dvou tradičních algoritmů pro výběr rysů - feature selection (dále jen FS) & backward elimination (dále jen BE), které byly detailněji popsány již v kapitole 7.3.1. Hlavním důvodem uplatnění níže popsané metody byla snaha o odstranění problému nenávratnosti FS a BE (jednou vybraný rys nemůže být odstraněn z / navrácen do podmnožiny). Konkrétním příkladem může být použití metody FS na MMP

dataset, která s prahem 0,1 vedla k výběru vlastnosti "Principal component II"[63] v první iteraci algoritmu. S využitím této jediné vlastnosti prediktor dosahoval maximální přesnosti 0,657, avšak přidání jakéhokoliv dalšího rysu (případně i množiny rysů) způsobilo pokles normalizované přesnosti pod 0,640. Jinými slovy tento rys z databáze AAIndex působil rušivě v kombinaci s ostatními vlastnostmi, přestože stojící samostatně byl algoritmem FS ohodnocen jako nejlepší (a z vybrané podmnožiny tedy již neodebratelný).

K vyřešení problému nenávratnosti byla navržena zde uvedená metoda. Jedná se o iterativní metodu, při níž dochází k pravidelnému střídání algoritmů FS a BE. Jednotlivé kroky by bylo možné popsat následujícím postupem:

- Inicializuje se maximální množina 544 vlastností z databáze AAIndex a aktuální podmnožina vybraných rysů (prázdná).
- Pomocí algoritmu FS je z maximální množiny postupně do aktuální podmnožiny přeřazeno 25 rysů. Těchto 25 rysů je z maximální množiny trvale odstraněno.
- Algoritmem BE je aktuální podmnožina postupně zredukována na pouhých 5 rysů. V průběhu redukce je v paměti držena "záložní"podmnožina, která dosáhla nejvyšší přesnosti predikce.
- Byl-li dovršen nastavený maximální počet iterací, algoritmus se ukončí a vrátí "záložní"podmnožinu. V opačném případě je aktuální podmnožina nahrazena zálohou a algoritmus se vrací na druhý bod. Byla-li nejvyšší přesnost ve třetím bodu dosažena pro příklad s 10 rysy, pak FS startuje již s 10 vybranými a z maximální množiny tedy přebírá jen 15 dalších.

Pro proces FS byla horní hranice 25 rysů zvolena jako přijatelný kompromis mezi nároky na výpočetní čas a dostatečnou velikostí prostoru rysů pro následné BE. Spodní hranice 5 rysů pak byla stanovena jako minimální počet vlastností, schopných poskytnout kvalitní výsledky (vychází z dříve popsané informačně založené metody), přičemž tato hodnota je jen o málo nižší, než řady v současné době existujících nástrojů (6 fyzikálně-chemických vlastností v nástroji MAPP, 7 v nástroji PASE, 8 v SNPdrayd, ...). Rozhodovací práh byl při procesu výběru rysů nastaven na hodnotu 0,1. Tato hodnota byla vybrána experimentálně, jelikož se při ní nejvíce projevoval vliv jednotlivých rysů při malých množinách (při výběru prvního rysu byla s ostatními prahy většina rysů vyhodnocena s přesností 0,5, tedy s pravděpodobností náhody).

Popsaný algoritmus je výpočetně velmi náročný a žádá si masivní paralelizaci, ale již po čtyřech iteracích dospěl k výběru 12 fyzikálně-chemických vlastností, s nimiž přesnost predikce dosáhla na 0,721 na MMP datasetu při rozhodovacím prahu 0,1. Rysy, uvedené v tabulce 9.6 tedy vedly k navýšení přesnosti přibližně o 0,024 (0,697 versus 0,721).

9.7 Srovnání s ostatními nástroji

V tomto oddíle uvedu srovnání nově vyvinutého nástroje RAPHYD s ostatními již existujícími. Tabulka 9.9 zachycuje nástroj RAPHYD ve dvou variantách (ve variantě s originálními šesti vlastnostmi a s 12 vlastnostmi zvolenými v rámci procesu výběru rysů) na MMP datasetu s rozhodovacím prahem 0,1 pro obě situace 6 / 12 fyzikálně-chemických vlastností.

Z tabulky 9.9 je dále patrné, že přesnost nástroje RAPHYD je srovnatelná s ostatními (při všech testovaných hodnotách rozhodovacího prahu, jelikož nejhorší naměřená přesnost

Tabulka 9.8: Sada fyzikálně-chemických vlastností vybraných kombinací FS a BE.

AAIndex ID	**Název vlastnosti**	**Autor**
ISOY800104	Normalized relative frequency of bend [90]	Isogai et al.
ROBB760113	Information measure for loop [8]	Robson-Suzuki
ROBB760111	Information measure for C-terminal [8]	Robson-Suzuki
ROBB760112	Information measure for coil [8]	Robson-Suzuki
GUOD860101	Retention coefficient at pH 2 [15]	Guo et al.
BROC820102	Retention coefficient in HFBA [9]	Browne et al.
JACR890101	Weights from the IFH scale [67]	Jacobs-White
MEEJ800102	Retention coefficient in HPLC, pH2.1 [41]	Meek
FAUJ880110	Number of full nonbonding orbitals [40]	Fauchere et al.
RACS820103	Average relative fractional occurrence in AL(i) [74]	Rackovsky-Scheraga
MITS020101	Amphiphilicity index [73]	Mitaku et al.
OOBM850104	Optimized average non-bonded energy per atom [54]	Oobatake et al.

byla 0,688 při prahu 0,01). Ve variantě RAPHYD-6 dosahují vyšší přesnosti pouze nástroje SNAP a PredictSNP, který je konsenzem více nástrojů. Ve variaci RAPHYD-12 pak nástroj na datasetu MMP dosahuje dokonce výrazně vyšší úspěšnosti, než ostatní z testovaných.

Srovnání rychlosti dále ukazuje, že nástroj RAPHYD patří v obou variantách 6 / 12 k nejrychlejším. Uvedených 182 sekund v sobě zahrnuje rovněž i čas pro vyhledání homologních sekvencí nástrojem BLASTp v konfiguraci prahu e-value 10^{-12} a maximálního počtu 200 homologních sekvencí a čas potřebný pro konstrukci vícenásobného zarovnání sekvencí a fylogenetického stromu. Samotná kalkulace nástroje RAPHYD po vložení vstupních souborů se pohybuje pod hranicí jedné sekundy. Z tabulky 9.9 vyplývá, že z testovaných nástrojů dosahují vyšší rychlosti pouze nsSNPAnalyzer a PANTHER, kde je však rychlost vyvážena nízkou přesností predikce.

Tabulka 9.9: Srovnání nástroje RAPHYD s ostatními nástroji, evaluovanými ve studii [33] na MMP datasetu. RAPHYD je zde uveden ve dvou variantách se 6 / 12 fyzikálně-chemickými vlastnostmi.

Nástroj	nsSNPAnalyzer	PANTHER	PhD-SNP	PPH-1	PPH-2
TN	4 264	4 336	3 739	4 390	3 518
FP	2 687	834	3 798	3 053	3 925
TP	2 510	329	3 399	3 330	3 769
FN	1 518	1 428	1 058	944	505
Rychlost	20 s	20 s	892 s	479 s	512 s
Pokrytí	0,915	0,619	1,000	0,977	0,977
Přesnost	0,617	0,695	0,595	0,659	0,622
P. norm.	0,618	0,603	0,629	0,684	0,677
Nástroj	SIFT	SNAP	PredictSNP	RAPHYD - 6	RAPHYD -12
TN	2 887	5 338	4 291	4 658	5 719
FP	4 463	2 200	3 247	2 541	1 480
TP	3 675	3 163	3 773	3 244	2 820
FN	416	1 293	683	1 104	1 528
Rychlost	408 s	1 429 s	-	182 s	182 s
Pokrytí	0,954	1,000	1,000	0,963	0,963
Přesnost	0,574	0,709	0,672	0,684	0,739
P. norm.	0,646	0,709	0,708	0,697	0,721

Kapitola 10

Závěr

V úvodu teoretické části této práce jsem poskytl náhled do biologické problematiky proteinů a mutací, které ovlivňují jejich funkci. V následujících kapitolách jsem se pak zaměřil na metody a nástroje, využívané pro vyhledávání homologních sekvencí v datově objemných biologických databázích a pro konstrukci vícenásobného zarovnání a fylogenetických stromů z nalezené množiny záznamů. Samostatná kapitola byla rovněž věnována také technikám pro predikci efektu aminokyselinových substitucí na funkci proteinu a existujícím nástrojům. V závěru teoretické části jsem pak uvedl několik algoritmů pro redukci dimenzionality prostřednictvím výběru a extrakce rysů.

V první kapitole praktické části jsem se zaměřil na metody, použité při návrhu a implementaci mnou vyvinutého nástroje RAPHYD. Popsal jsem zde způsoby pro určení kořene fylogenetického stromu, fylogenetickou analýzu a postup určení rozdílností ve fyzikálně-chemických vlastnostech aminokyselin. Stručně jsem zmínil také nástroj HotSpot Wizard, do kterého byl RAPHYD úspěšně integrován a čtyři datasety určené pro testování.

Druhá kapitola pak poskytla shrnutí dosažených výsledků a provedených optimalizací. Prvním krokem byla optimalizace nástrojů třetích stran pro konstrukci vícenásobného zarovnání a fylogenetického stromu a nastavení nástroje BLASTp pro vyhledávání homologních sekvencí v databázi nr90. Na základě výsledků jsem jako optimální kompromis mezi rychlostí a kvalitou určil nástroje CLUSTALΩ a FastTree 2.0 pro sestavení vícenásobného zarovnání, respektive fylogenetického stromu. Pro BLASTp jsem pak zvolil konfiguraci e-value prahu na hodnotu 10^{-12} při vyhledávání maximálně 200 homologních sekvencí. Ve druhém kroku jsem optimalizoval rozhodovací práh, dělící mutace do neutrálních a škodlivých podmnožin. Zde výsledky odhalily, že optimálním prahem je hodnota 0,08, při níž prediktor dosahuje v závislosti na datasetu až o 2,6 % lepších výsledků oproti výchozí hodnotě 0,01. Dále jsem se zabýval otázkou, jakého vlivu na přesnost predikce dosahuje podmínka ignorování mutací na pozicích s více jak 50 % mezer ve sloupci vícenásobného zarovnání. Rozdíl zde činil méně než 1 % a lze ho tak považovat za zanedbatelný. V posledním kroku optimalizace jsem se snažil nalézt novou sadu fyzikálně-chemických vlastností z databáze AAIndex. Uplatil jsem zde dvě metody: založenou na informacích a na experimentech, z nichž druhá ze zmíněných vedla ke zvýšení přesnosti až o 2,4 % (v závislosti na datasetu) při dvanácti vybraných vlastnostech.

Závěrem jsem nově vyvinutý nástroj RAPHYD srovnal s osmi již existujícími nástroji pro predikci škodlivosti aminokyselinových substitucí. Ze srovnání vyplývá, že při vysoké rychlosti výpočtu je jeho přesnost po optimalizacích srovnatelná s nejlepšími z nástrojů.

Literatura

[1] A. A. Zamyatnin: Protein volume in solution. *Progress in Biophysics and Molecular Biology*, ročník 24, 1972: s. 107–123.

[2] A. Bairoch, B. Boeckmann: The SWISS-PROT protein sequence data bank. *Nucleic Acids Research*, ročník 19, 1991: s. 2247–2249.

[3] A. D. Johnson, R. E. Handsaker, S. L. Pulit, M. M. Nizzan, Ch. J. O Donnell, P. I. W. de Bakker: SNAP: a web-based tool for identification and annotation of proxy SNPs using HapMap. *Bioinformatics*, ročník 24, 2008: s. 2938–3939.

[4] A. Fiser, A. Šali: Modeller: Generation and refinement of homology-based protein structure models. *Methods in Enzymology*, ročník 374, 2003: s. 461–469.

[5] A. Pavelka, E. Chovancova, J. Damborsky: HotSpot Wizard: a web server for identification of hot spots in protein engineering. *Nucleic Acids Research*, ročník 43, 2009: s. 1–8.

[6] A. Stamatakis1, T. Ludwig, H. Meier: RAxML-III: a fast program for maximum likelihood-based inference of large phylogenetic trees. *Bioinformatics*, ročník 21, 21. 4, 2005: s. 456–463.

[7] B. Alberts, et al.: *Essential cell biology*. Garland Science, 2004.

[8] B. Robson, E. Suzuki: Conformational properties of amino acid residues in globular proteins. *Molecular Biology*, ročník 107, 1976: s. 327–356.

[9] C. A. Browne, H. P. J. Bennett, S. Solomon: The isolation of peptides by high-performance liquid chromatography using predicted elution positions. *Analytical Biochemistry*, ročník 124, 1982: s. 201–208.

[10] C. Branden, J. Tooze: *Introduction to protein structure*. Garland Publishing, 1999.

[11] C. N. Pauline, S. Henikoff: SIFT: predicting amino acid changes that affect protein function. *Nucleic Acids Research*, ročník 31, 2003: s. 3812–3814.

[12] C. Notredame, D. G. Higgins, J. Heringa: T-coffee: a novel method for fast and accurate multiple sequence alignment. *Journal of Molecular Biology*, ročník 302, 302. 1, 2000: s. 205–217.

[13] D. G. Higgins, A. J. Bleasby, R. Fuchs: CLUSTAL V: improved software for multiple sequence alignment. *Comput. Appl. Biosci.*, ročník 8, 8. 2, 1992: s. 189–191.

[14] D. G. Higgins, P. M. Sharp : CLUSTAL: a package for performing multiple sequence alignment on a microcomputer. *Gene*, ročník 73, 1988: s. 237–244.

[15] D. Guo, C. T. Mant, A. K. Taneja, J. M. Parker, R. S. Hodges: Prediction of peptide retention times in reversed-phase high-performance liquid chromatography. *Chromatography*, ročník 359, 1986: s. 499–517.

[16] D. Plotree, D. Plotgram: PHYLIP-phylogeny inference package (version 3.2). *Cladistics*, ročník 5, 5. 2, 1989: s. 163–166.

[17] D. W. Mount: Comparison of the PAM and BLOSUM amino acid substitution matrices. *PubMed*, 2008.

[18] D. Whitford: *Proteins: Structure and function.* Wiley, 2005.

[19] E. A. Stone, A. Sidow: Physicochemical constraint violation by missense substitutions mediates impairment of protein function and disease severity. *Genome Research*, ročník 15, 2005: s. 978–986.

[20] E. Capriotti, R. Calabrese, R. Casadio: Predicting the insurgence of human genetic diseases associated to single point protein mutations with support vector machines and evolutionary information. *Bioinformatics*, ročník 22, 22. 22, 2006: s. 2729–2734.

[21] E. Chovancová, A. Pavelka, P. Beneš, O. Strnad, J. Brezovský, B. Kozlíková, A. Gora, V. Šustr, M. Klvaňa, P. Medek, L. Biedermannová, J. Sochor, J. Damborský: CAVER 3.0: A tool for the analysis of transport pathways in dynamic protein structures,. *PLoS Computational Biology*, ročník 8, 2012.

[22] E. V. Koonin: Orthologs, paralogs, and evolutionary genomics. *Annual Review of Genetics*, ročník 39, 2005: str. 309–338.

[23] F. Sievers, A. Wilm, D. Dineen, et al.: Fast, scalable generation of high-quality protein multiple sequence alignments using Clustal Omega. *Molecular Systems Biology*, ročník 7, 2011: s. 539–545.

[24] G. Wainreb, H. Ashkenazy, Y. Bromberg, A. Starovolsky-Shitrit , T. Haliloglu, E. Ruppin, K. B. Avraham, B. Rost, N. Ben-Tal: MuD: an interactive web server for the prediction of non-neutral substitutions using protein structural data. *Nucleic Acids Res.*, ročník 38, 38. 21, 2010: str. 7869.

[25] H-D. Jakubke, N. Sewald: *Peptides from A to Z: A concise encyclopedia.* Wiley, 2008.

[26] I. Burgetová, T. Martínek: Základy molekulární biologie [pdf]. FIT VUT Brno, [cit. 2015-05-12].

[27] I. Gronau, S. Moran: Optimal implementations of UPGMA and other common clustering algorithms. *Information Processing Letters*, ročník 104, 104. 6, 2007: s. 205–210.

[28] I. Guyon, A. Elisseeff: An introduction to variable and feature selection. *Journal of Machine Learning Research*, ročník 3, 2003: s. 1157–1182.

[29] I. Jolliffe: *Principal component analysis.* Wiley Online Library, 2014.

[30] I. T. Jolliffe: A note on the use of principal components in regression. *Journal of the Royal Statistical Society*, ročník 31, 31. 3, 1982: s. 300–303.

[31] International Human Genome Sequencing Consortium: Initial sequencing and analysis of the human genome. *Nature*, ročník 409, 2001: s. 860–921.

[32] J. A. Capra, M. Singh: Predicting functionally important residues from sequence conservation. *Bioinformatics*, ročník 23, 23. 15, 2007: s. 1875–1882.

[33] J. Bendl, J. Stourac, O. Salanda, A. Pavelka, E. D. Wieben, J. Zendulka, et al.: PredictSNP: Robust and accurate consensus classifier for prediction of disease-related mutations. *PLoS Comput. Biol.*, ročník 10: e1003440., 2014.

[34] J. C. Whisstock, A. M. Lesk: Prediction of protein function from protein sequence and structure. *Quarterly Reviews of Biophysics*, ročník 36, 36. 3, 1985: s. 307–340.

[35] J. D. Thompson, D.G. Higgins, T. J. Gibson: CLUSTAL W: Improving the sensitivity of progressive multiple sequence alignment through sequence weighting, position-specific gap penalties and weight matrix choice. *Nucleic Acids Research*, ročník 22, 1994: s. 4673–4680.

[36] J. D. Thompson, T. J. Gibson, F. Plewniak, F. Jeanmougin, D. G. Higgins: The CLUSTAL X windows interface: flexible strategies for multiple sequence alignment aided by quality analysis tools. *Nucleic Acids Research*, ročník 25, 25. 24, 1997: s. 4876–4882.

[37] J. Felsenstein: Evolutionary trees from DNA sequences: A maximum likelihood approach. *Journal of Molecular Evolution*, ročník 17, 1981: s. 368–376.

[38] J. Flegr: *Úvod do evoluční biologie*. Academia, 2007.

[39] J. Kyte , R. F. Doolittle: A simple method for displaying the hydropathic character of a protein. *Journal of Molecular Evolution*, ročník 157, 1982: s. 105–132.

[40] J. L. Fauchere, M. Charton, L. B. Kier, A. Verloop, V. Pliska: Amino acid side chain parameters for correlation studies in biology and pharmacology. *International Journal of Peptide and Protein Research*, ročník 32, 1988: s. 269–278.

[41] J. L. Meek: Prediction of peptide retention times in high-pressure liquid chromatography on the basis of amino acid composition. *PNAS*, ročník 77, 1980: s. 1632–1636.

[42] J. M. Berg, J. L. Tymoczko, L. Stryer, J. M. Berg, J. L. Tymoczko: *Biochemistry*. W. H. Freeman, 2002.

[43] J. MacQueen: *Some methods for classification and analysis of multivariate observations*. University of California, 1967.

[44] J. Pevsner: *Bioinformatics and functional genomics*. Wiley, 2009.

[45] J. Zendulka: Předzpracování dat [pdf]. `FIT VUT Brno`, [cit. 2015-05-12].

[46] K-Ch. Wong, Z. Zhang: SNPdryad: Predicting deleterious non-synonymous human SNPs using only orthologous protein sequences. *Bioinformatics Advance Access*, 2014.

[47] K. Katoh, K. Misawa, K. Kuma, T. Miyata: MAFFT: a novel method for rapid multiple sequence alignment based on fast Fourier transform. *Nucleic Acids Research*, ročník 30, 30. 14, 2002: s. 3059–3066.

[48] K. Q. Weinberger, L. K. Saul : Unsupervised learning of image manifolds by semidefinite programming. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2004.

[49] L. Y. Yampolsky, A. Stoltzfus: The exchangeability of amino acids in proteins. *Genetics*, ročník 170, 2005: s. 1459–1472.

[50] M. Abramowitz, I. A. Stegun: *Handbook of mathematical functions with formulas, graphs, and mathematical tables*. New York: Dover, 1965.

[51] M. Cargill: Characterization of single-nucleotide polymorphisms in coding regions of human genes. *Nature Genetics*, ročník 23, 23. 3, 1999: s. 373–373.

[52] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, I. H. Witten: The WEKA data mining software: An update. *SIGKDD Explorations*, ročník 11, 11. 1, 2009.

[53] M. N. Price, P. S. Dehal, A. P. Arkin: FastTree 2 - Approximately maximum-likelihood trees for large alignments. *Molecular Systems Biology*, ročník 5, 5. e9490, 2010.

[54] M. Oobatake, Y. Kubota, T. Ooi: Optimization of amino acid parameters for correspondence of sequence to tertiary structures of proteuins. *Bull. Inst. Chem. Res.*, ročník 63, 1985: s. 82–94.

[55] M. Scholz: *Approaches to analyse and interpret biological profile data*. Universitat Potsdam, 2006.

[56] N. Friedman, M. Ninio, I. Pe'er, T. Pupko: A structural EM algorithm for phylogenetic inference. *Nucleic Acids Research*, ročník 9, 2002: s. 331–353.

[57] N. Furnham, G. L. Holliday, T. A. de Beer, J. O. Jacobsen, W. R. Pearson, J. M. Thornton: The Catalytic Site Atlas 2.0: cataloging catalytic sites and residues identified in enzymes. *Nucleic Acids Res.*, 2014.

[58] N. Saitou, Ma. Nei: The Neighbor-joining method: A new method for reconstructing phylogenetic trees. *Molecular Biology and Evolution*, ročník 4, 4. 4, 1987: s. 406–425.

[59] NCBI Resource Coordinators: Database resources of the National Center for Biotechnology Information. *Nucleic Acids Research*, ročník 41, 2013: s. D8–D20.

[60] OpenStax College: *Anatomy and physiology*. Rice University, 2013.

[61] P. D. Thomas, M. J. Campbell, A. Kejariwal, H. Mi, B. Karlak, R. Daverman, K. D. A. Muruganujan, A. Narechania: PANTHER: A library of protein families and subfamilies indexed by function. *Genome research*, ročník 13, 2003: s. 2129–2141.

[62] P. J. Russell: *iGenetics*. 2014.

[63] P.H.A. Sneath: Relations between chemical structure and biological activity in peptides. *Journal of Theoretical Biology*, ročník 12, 12. 2, 1966: s. 157–195.

[64] R. C. Edgar: MUSCLE: multiple sequence alignment with high accuracy and high throughput. *Nucl. Acids Res.*, ročník 32, 32. 5, 2004: s. 1792–1797.

[65] R. D. Finn, J. Clements, S. R. Eddy: HMMER web server: interactive sequence similarity searching. *Nucleic acids research*, ročník 25, 25. 17, 2011: s. 3389–3402.

[66] R. Edwards: How to root a phylogenetic tree [online]. `http://cabbagesofdoom.blogspot.cz/2012/06/how-to-root-phylogenetic-tree.html`, [cit. 2015-05-12].

[67] R. Jacobs, S. H. White: The nature of the hydrophobic bonding of small peptides at the bilayer interface: implications for the insertion of transbilayer helices. *Biochemistry*, ročník 28, 1989: s. 3421–3437.

[68] S. B. Needleman, Ch. D. Wunsch: A general method applicable to the search for similarities in the amino acid sequence of two proteins. *Journal of Molecular Biology*, ročník 3, 1970: s. 443–453.

[69] S. F. Altschul, R. J. Carroll, D. J. Lipman: Weights for data related by a tree. *Journal of Molecular Biology*, ročník 207, 207. 4, 1989: s. 647–653.

[70] S. F. Altschull, T. L. Madden, A. A. Schaffer, J. Zhang, Z. Zhang, W. Miller, D. J. Lipman: Gapped BLAST and PSI-BLAST: a new generation of protein database search programs. *Nucleic acids research*, ročník 25, 25. 17, 1997: s. 3389–3402.

[71] S. Guindon, J-F. Dufayard , V. Lefort , M. Anisimova, W. Hordijk, O. Gascuel: New algorithms and methods to estimate Maximum-likelihood phylogenies: Assessing the performance of PhyML 3.0. *System Biology*, ročník 59, 59. 3, 2010: s. 307–321.

[72] S. Kawashima, H. Ogata, M. Kanehisa: AAindex: Amino acid index database. *Nucleic Acids Research*, ročník 27, 1999: s. 368–369.

[73] S. Mitaku, T. Hirokawa, T. Tsuji: Amphiphilicity index of polar amino acids as an aid in the characterization of amino acid preference at membrane-water interfaces. *Bioinformatics*, ročník 18, 2002: s. 608–616.

[74] S. Rackovsky, H. A. Scheraga: Differential geometry and polymer conformation. 4. Conformational and nucleation properties of individual amino acids. *Macromolecules*, ročník 15, 1982: s. 1340–1346.

[75] T. F. Smith, M. S. Waterman: Identification of common molecular subsequences. *Journal of Molecular Biology*, ročník 147, 1981: str. 195–197.

[76] T. Kawabata, M. Ota, K. Nishikawa: The protein mutant database. *Nucleic Acids Research*, ročník 27, 1999: s. 355–357.

[77] T. Lassmann, E. L. L. Sonnhammer : Kalign - an accurate and fast multiple sequence alignment algorithm. *BMC Bioinformatics*, ročník 6, 6. 298, 2005.

[78] T. Martínek: Zarovnání sekvencí [pdf]. `FIT VUT Brno`, [cit. 2015-05-12].

[79] V. Le Guilloux, P. Schmidtke, P. Tuffery: Fpocket: An open source platform for ligand pocket detection. *BMC Bioinformatics*, ročník 10, 10. 168, 2009.

[80] V. Moulton: *Algorithms in bioinformatics*. WABI, 2010.

[81] V. Munoz, L. Serrano: Intrinsic secondary structure propensities of the amino acids, using statistical Phi-Psi matrices: Comparison with experimental scales. *Proteins Structure, Function and Bioinformatics*, ročník 20, 1994: s. 301–311.

[82] V. Ramensky, P. Bork, S. Sunyaev: Human non-synonymous SNPs: server and survey. *Nucleic Acids Research*, ročník 30, 30. 17, 2002: s. 3894–3900.

[83] W. A. Cuevas, D. E. Estell, S. H. Hadi, S-K. Lee, S. W. Ramer, et al.: Geobacillus stearothermophilus Alpha-amylase (AmyS) variants with improved properties, uS Patent US8084240.

[84] W. Aehle, L. G. Cascao-pereira, D. A. Estell, F. Goedegebuur, J. T. Kellis Jr., A. J. Poulose, et al.: Compositions and methods comprising serine protease variants, uS Patent 20150031589.

[85] W. J. Kent: BLAT - The BLAST-Like alignment tool. *Genome Res.*, ročník 12, 12. 4, 2002: s. 656–664.

[86] W. Lee, Y. Zhang, K. Mukhyala, R. A. Lazarus, Z. Zhang: Bi-Directional SIFT predicts a subset of activating mutations. *PLoS ONE*, ročník 4, 4. e8311, 2009.

[87] W. M. Fitch, J. S. Farris: Evolutionary trees with minimum nucleotide replacements from amino acid sequences. *Journal of Molecular Evolution*, ročník 3, 3. 4, 1974: s. 263–278.

[88] W. R. Pearson: Rapid and sensitive sequence comparison with FASTP and FASTA. 1990.

[89] X. Li: PASE: a novel method for functional prediction of amino acid substitutions based on physicochemical propertie. *Frontiers in genetics*, ročník 4, 2013.

[90] Y. Isogai, G. Nemethy, S. Rackovsky, S. J. Leach, H. A. Scheraga: Characterization of multiple bends in proteins. *Biopolymers*, ročník 19, 1980: s. 1183–1210.

Příloha A

Výsledky základního testování

Tabulka A.1: Vyhodnocení přesnosti predikčního nástroje RAPHYD na MMP datasetu. Použit byl výchozí rozhodovací práh 0,01 a sada šesti fyzikálně-chemických vlastností.

Počet/eval.	**50-1e-6**	**50/1e-12**	**100/e1-6**	**100/1e-12**	**150/1e-6**	**150/1e-12**	**200/1e-6**	**200/1e-12**
TN	5 312	6 203	6 483	6 328	6 410	6 226	6 403	6 165
FP	872	872	839	1 120	790	1 122	796	1 034
TP	2 120	2 120	2 092	2 462	2 039	2 333	2 013	2 257
FN	2 282	2 282	2 308	1 973	2 341	2 096	2 333	2 091
Pokrytí	0,883	0,957	0,977	0,911	0,965	0,982	0,963	0,963
Přesnost	0,728	0,725	0,732	0,740	0,730	0,727	0,729	0,729
P. norm.	0,680	0,679	0,680	0,702	0,678	0,687	0,676	0,688

Tabulka A.2: Vyhodnocení přesnosti predikčního nástroje RAPHYD na PMD datasetu. Použit byl výchozí rozhodovací práh 0,01 a sada šesti fyzikálně-chemických vlastností.

Počet/eval.	**50-1e-6**	**50/1e-12**	**100/e1-6**	**100/1e-12**	**150/1e-6**	**150/1e-12**	**200/1e-6**	**200/1e-12**
TN	653	632	666	619	616	590	621	628
FP	210	248	196	237	192	235	190	218
TP	929	1 059	878	971	847	936	828	952
FN	858	734	881	806	858	796	861	785
Pokrytí	0,758	0,764	0,749	0,753	0,719	0,731	0,715	0,739
Přesnost	0,597	0,633	0,589	0,604	0,582	0,597	0,580	0,612
P. norm.	0,638	0,654	0,636	0,635	0,630	0,628	0,628	0,645

Tabulka A.3: Vyhodnocení přesnosti predikčního nástroje RAPHYD na PredictSNP datasetu. Použit byl výchozí rozhodovací práh 0,01 a sada šesti fyzikálně-chemických vlastností.

Počet/eval.	**50-1e-6**	**50/1e-12**	**100/e1-6**	**100/1e-12**	**150/1e-6**	**150/1e-12**	**200/1e-6**	**200/1e-12**
TN	13 585	14 338	12 767	13 558	12 109	13 020	11 759	12 628
FP	1 918	2 321	1 665	2 051	1 579	1 969	1 533	1 928
TP	5 283	6 105	5 018	5 819	4 832	5 482	4 826	5 417
FN	9 408	9 211	9 002	8 850	8 761	8 519	8 586	8 119
Přesnost	0,688	0,729	0,648	0,690	0,622	0,661	0,609	0,640
Pokrytí	0,625	0,639	0,640	0,640	0,621	0,638	0,621	0,642
P. norm.	0,618	0,630	0,633	0,633	0,620	0,630	0,622	0,634

Tabulka A.4: Vyhodnocení přesnosti predikčního nástroje RAPHYD na BSIFT datasetu. Použit byl výchozí rozhodovací práh 0,01 a sada šesti fyzikálně-chemických vlastností.

Počet/eval.	**50-1e-6**	**50/1e-12**	**100/e1-6**	**100/1e-12**	**150/1e-6**	**150/1e-12**	**200/1e-6**	**200/1e-12**
TN	1 614	1 557	1 542	1 533	1 486	1 483	1 494	1 480
FP	670	787	631	757	645	731	633	723
TP	5 441	5 934	5 286	5 830	5 208	5 712	5 215	5 706
FN	4 352	4 076	4 271	3 931	4 148	3 952	4 141	3 868
Pokrytí	0,815	0,834	0,791	0,813	0,775	0,801	0,775	0,795
Přesnost	0,584	0,606	0,582	0,611	0,583	0,606	0,584	0,610
P. norm.	0,631	0,629	0,631	0,633	0,627	0,630	0,630	0,634

Příloha B

Vliv rozhodovacího prahu

Tabulka B.1: Vliv různých rozhodovacích prahů na přesnost nástroje RAPHYD (měřeno na MMP datasetu).

Roz. práh	**0,01**	**0,02**	**0,03**	**0,04**	**0,05**	**0,06**	**0,07**	**0,08**	**0,09**	**0,10**
TN	6 165	5 837	5 618	5 415	5 249	5 119	4 975	4 863	4 754	4 658
FP	1 034	1 362	1 581	1 784	1 950	2 080	2 224	2 236	2 445	2 541
TP	2 257	2 545	2 716	2 847	2 948	3 022	3 087	3 142	3 200	3 244
FN	2 091	1 803	1 632	1 501	1 400	1 326	1 261	1 206	1 145	1 104
Senzitivita	0,856	0,811	0,780	0,752	0,729	0,711	0,691	0,685	0,660	0,647
Specificita	0,519	0,585	0,625	0,655	0,678	0,695	0,710	0,723	0,736	0,746
Přesnost	0,729	0,726	0,722	0,716	0,710	0,705	0,698	0,699	0,689	0,684
P. norm.	0,688	0,698	0,703	0,703	0,704	0,703	0,701	0,704	0,698	0,697

Tabulka B.2: Vliv různých rozhodovacích prahů na přesnost nástroje RAPHYD (měřeno na PredictSNP datasetu).

Roz. práh	**0,01**	**0,02**	**0,03**	**0,04**	**0,05**	**0,06**	**0,07**	**0,08**	**0,09**	**0,10**
TN	12 623	12 111	11 780	11 507	11 281	11 066	10 859	10 675	10 489	10 313
FP	1 925	2 437	2 768	3 041	3 267	3 482	3 689	3 873	4 059	4 235
TP	5 447	6 223	6 721	7 109	7 347	7 579	7 790	7 970	8 134	8 280
FN	8 264	7 488	6 990	6 602	6 364	6 132	5 921	5 741	5 577	5 431
Senzitivita	0,868	0,832	0,810	0,791	0,755	0,761	0,746	0,734	0,721	0,709
Specificita	0,397	0,454	0,490	0,518	0,536	0,553	0,568	0,581	0,593	0,604
Přesnost	0,639	0,649	0,655	0,659	0,659	0,660	0,660	0,660	0,659	0.658
P. norm.	0,632	0,643	0,650	0,655	0,656	0,657	0,657	0,658	0,657	0,656

Příloha C

Obsah DVD

Na přiloženém DVD lze nalézt:

- Soubor README s popisem adresářů a pokyny ke spuštění ukázkového vzorku.
- Adresář se zdrojovými soubory nástroje RAPHYD.
- Adresář se sadou scriptů, použitých při testování nástroje a výběru rysů.
- Adresář s přeloženým nástrojem RAPHYD ve formátu .jar, soubory s vícenásobným zarovnáním sekvencí, fylogenetickým stromem proteinu LacI a sadu fyzikálně-chemických parametrů (ukázkový vzorek).
- Adresář s kompletními tabulkami výsledků a tabulkou vybraných fyzikálně-chemických vlastností.
- Adresář s použitými datovými sadami (PredictSNP, MMP, PMD, BSIFT, AAIndex).