



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ

FACULTY OF INFORMATION TECHNOLOGY

ÚSTAV INFORMAČNÍCH SYSTÉMŮ

DEPARTMENT OF INFORMATION SYSTEMS

**VYUŽITÍ STROJOVÉHO UČENÍ PRO PREDIKCI VLIVU
MUTACÍ NA STABILITU PROTEINŮ**

MACHINE LEARNING AS A TOOL FOR THE PREDICTION OF THE EFFECT OF MUTATIONS ON
PROTEIN STABILITY

BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

AUTOR PRÁCE

AUTHOR

JURAJ ONDREJ DÚBRAVA

VEDOUCÍ PRÁCE

SUPERVISOR

Ing. MILOŠ MUSIL,

BRNO 2018

Abstrakt

Táto práca sa zaoberá predikciou vplyvu aminokyselinových mutácií na stabilitu proteínu. Cieľom bolo vytvorenie nástroja klasifikujúceho výsledný charakter mutácie s využitím kombinácie viacerých metód strojového učenia. Implementovaný prístup kombinujúci metódy SVM a Random Forest dosiahol lepšie výsledky ako použitie metód samostante. Nástroj bol porovnaný s dostupnými nástrojmi na nezávislom datasete na ktorom dosiahol úspešnosť predikcie 67% a koreláciu 0.3.

Abstract

The main focus of this thesis is the prediction of the effect of amino acid substitutions on protein stability. My goal was to develop a predictive tool for the classification of the effect of mutations by combining several machine learning techniques. The implemented predictor, which utilizes SVM and Random forest methods, has achieved higher accuracy than any of the integrated methods. The novel predictive tool was compared with the existing ones using independent testing dataset. The predictor has yield 67% accuracy and MCC 0.3.

Klíčové slová

proteínová stabilita, aminokyselinové substitúcie, mutácie, strojové učenie, ensemble systémy, support vector machines, náhodný les

Keywords

protein stability, amino acid substitutions, mutations, machine learning, ensemble systems, support vector machines, random forest

Citácia

DÚBRAVA, Juraj Ondrej. *Využití strojového učení pro predikci vlivu mutací na stabilitu proteinů*. Brno, 2018. Bakalářská práce. Vysoké učení technické v Brně, Fakulta informačních technologií. Vedoucí práce Ing. Miloš Musil,

Využití strojového učení pro predikci vlivu mutací na stabilitu proteinů

Prehlásenie

Prehlasujem, že som túto bakalársku prácu vypracoval samostatne pod vedením pána Ing. Miloša Musila. Uviedol som všetky literárne pramene a publikácie z ktorých som čerpal.

.....
Juraj Ondrej Dúbrava
6. mája 2018

Podakovanie

Ďakujem pánovi Ing. Milošovi Musilovi za rady a odborné vedenie tejto práce.

Obsah

1	Úvod	3
2	Proteíny	4
2.1	Základné rozdelenie proteínov	4
2.2	Aminokyseliny	5
2.3	Syntéza proteínov	5
2.4	Štruktúra proteínov	6
3	Vplyv aminokyselinových substitúcií na stabilitu proteínu	7
3.1	Stabilita proteínu	7
3.2	Mutácie	8
3.2.1	Vznik mutácií	8
3.2.2	Typy mutácií	8
4	Strojové učenie	11
4.1	Úvod do strojového učenia	11
4.2	Rozhodovacie stromy	12
4.2.1	Algoritmus J48	12
4.2.2	Algoritmus Náhodný strom (Random Tree)	12
4.2.3	Algoritmus Náhodný les (Random Forest)	12
4.3	Support vector machines (SVM)	14
4.3.1	Jadrové funkcie	14
4.3.2	Algoritmus SMO	15
4.4	Algoritmus Naive Bayes	15
5	Ensemble metódy	16
5.1	Rôznorodosť v ensemble systémoch	17
5.2	Tvorba ensemble systémov	18
5.2.1	Bagging	18
5.2.2	Boosting	18
5.3	Spojenie klasifikátorov	19
5.3.1	Trénovateľné metódy	19
5.3.2	Netrénovateľné metódy	20
6	Nástroje na predikciu stability	21
6.1	Strojové učenie	21
6.1.1	AUTO-MUTE	21
6.1.2	I-Mutant	21

6.1.3	iPTREE-STAB	22
6.1.4	EASE-AA	22
6.1.5	mCSM	22
6.1.6	MAESTRO	22
6.1.7	ELASPIC	22
6.2	Energetická funkcia - fyzikálny potenciál	23
6.2.1	CC/PBSA	23
6.2.2	ERIS	23
6.2.3	Rosetta	23
6.2.4	CUPSAT	23
6.3	Energetická funkcia - štatistický potenciál	23
6.3.1	PopMuSiC	24
6.3.2	DMutant	24
6.3.3	FoldX	24
7	Dátová sada a jej parametre	25
7.1	Parametre záznamu mutácie	26
7.2	Felsteinov algoritmus	27
8	Implementácia	28
8.1	Testovanie vo WEKE	28
8.2	Príprava záznamu mutácie v Pythone	29
8.3	Testovanie metód strojového učenia v Pythone	30
8.4	Ensemble systém v Pythone	31
8.5	Porovnanie s inými nástrojmi	33
9	Záver	36
	Literatúra	37
A	Obsah DVD	41
B	Manuál	42

Kapitola 1

Úvod

Proteíny patria k základným stavebným prvkom každého živého organizmu. V ňom sa podieľajú na množstve rôznych funkcií; ako trávenie alebo prenos kyslíka. Tieto rozličné funkcie z nich robia významný cieľ skúmania. V dnešnej dobe je snahou najmä vylepšovanie rôznych vlastností s cieľom vytvoriť hodnotnejšie proteíny pre rôzne oblasti. Touto problematikou sa zaoberá oblasť proteínového inžinierstva.

Jednou z vlastností, ktorá je u proteínov veľmi dôležitá, je stabilita. Stabilný proteín je pripravený lepšie zvládať nepriaznivé podmienky svojho okolia, napríklad vydržať vyššiu teplotu alebo rôzne chemické vlastnosti prostredia. Oblasť návrhu a výroby stabilnejších proteínov je v dnešnej dobe zaujímavá najmä pre priemysel, napríklad pri výrobe rôznych biotechnológií. Zaujímavou je aj pre farmaceutický priemysel, pretože stabilnejšie proteíny by viedli k výrobe účinnejších liečiv.

V proteínoch však dochádza k mutáciám aminokyselín tvoriacich proteínový reťazec. Mutácie sú predovšetkým dielom evolúcie, ale taktiež sú uskutočňované umelo s určitým zámerom. Jedna zmena aminokyseliny v reťazci môže znamenať zmenu stability proteínu v pozitívnom, ale aj negatívnom zmysle. Preto je snahou predikovať vplyv takýchto zmien. V poslednom období vzniklo množstvo rôznych nástrojov zaoberajúcich sa predikciou vplyvu mutácií na stabilitu, pričom využívajú rôzne prístupy pri predikcii.

Cieľom tejto práce je vytvorenie, otestovanie a vyhodnotenie predikčného nástroja kombinujúceho viaceré metódy strojového učenia.

Prvé kapitoly teoretickej časti sa zaoberajú proteínmi a aminokyselinami. Obsahujú informácie o vlastnostiach proteínov a aminokyselín. Taktiež vysvetľujú pojem stability a jej súvis s mutáciami.

Zvyšné kapitoly teoretickej časti sú venované strojovému učeniu, tzv. ensemble systémom a existujúcim nástrojom. Táto časť obsahuje základnú klasifikáciu metód strojového učenia spoločne s podrobnejšou charakteristikou rôznych metód. Časť o ensemble systémoch vysvetľuje dôvody a výhody použitia týchto systémov.

Posledná časť je venovaná tvorbe dátovej sady a samotnej implementácii nástroja. Bližšie sú uvedené parametre mutácie, výber metód strojového učenia, tvorba ensemble systému a jeho vyhodnotenie v porovnaní s inými nástrojmi.

Kapitola 2

Proteíny

Proteíny (bielkoviny) môžeme charakterizovať ako základné stavebné prvky všetkých živých organizmov. Nie sú však iba stavebnými prvkami bunky, zabezpečujú väčšinu bunecných funkcií. Pochopenie procesu vzniku proteínov a ich funkcie nachádza široké uplatnenie v odvetviach ako medicína, poľnohospodárstvo, priemysel a mnohé ďalšie. V tejto kapitole sa budem zaoberať základným rozdelením proteínov, procesom ich vzniku z DNA a štruktúrou.

2.1 Základné rozdelenie proteínov

Proteíny sú biopolyméry tvorené z jedného alebo viacerých polypeptidových reťazcov, ktoré je možné chápať ako sekvenciu polymérov aminokyselín navzájom spojených kovalentnou peptidovou väzbou. Proteíny sa skladajú do množstva komplikovaných tvarov a ich funkcie súvisia s konkrétnym priestorovým usporiadaním (konformáciou), pričom sa snažia zaujať čo najlepšiu konformáciu z energetického hľadiska. Konformácia vychádza z primárnej štruktúry, ktorú je možné chápať ako reťazec aminokyselín v danom poradí [1]. Podľa funkcie môžeme proteíny rozdeliť do niekoľkých skupín [1]:

- **Enzýmy:** ich funkciou je katalýza rozpadu a tvorba kovalentných väzieb. Príkladom môže byť napríklad pepsín, ktorý sa podieľa na odbúraní bielkovín pri trávení.
- **Štruktúrne proteíny:** tvoria základné stavebné jednotky buniek a tkanív, poskytujú im mechanickú oporu. Príkladom je keratín tvoriaci základnú zložku vlasov a nechtov.
- **Transportné proteíny:** prenášajú malé molekuly a ióny v organizme. Príkladom sú proteín hemoglobín ako nosič kyslíka v krvnom obeh a proteín transferrin prenášajúci železo.
- **Pohybové proteíny:** sú pôvodcami pohybu buniek a tkanív. Príkladom takýchto proteínov sú kinesin a myosin.
- **Zásobné proteíny:** slúžia na skladovanie malých molekúl a iónov. Kasein v mlieku poskytuje zdroj aminokyselín pre novonarodené živočíchy.
- **Signálne proteíny:** ich funkciou je prenos informačných signálov medzi bunkami. Do tejto skupiny patrí známy proteín inzulín regulujúci hladinu cukru v krvi.
- a ďalšie.

2.2 Aminokyseliny

Aminokyseliny sú odvodené od organických kyselín a predstavujú rôzne triedy molekúl s jednou spoločnou vlastnosťou; všetky vlastnia karboxylovú (COOH) a aminovú (NH_2) skupinu. Tieto skupiny sú naviazané k jednému uhlíkovému atómu, ktorý je označovaný ako α uhlík. Rôznorodosť jednotlivých aminokyselín spočíva v postrannom reťazci (R) určujúcom chemické vlastnosti aminokyselín, resp. proteínov. Jednotlivé aminokyseliny sú v proteínovej molekule vzájomne spojené peptidovou väzbou, ktorá prepojuje karboxylovú skupinu jednej aminokyseliny s amino skupinou druhej. Reťazec viacerých aminokyselín je označovaný ako peptidový reťazec (polypeptid). Celkovo existuje 20 rôznych aminokyselín, ktoré môžeme na základe chemických vlastností postranných reťazcov rozdeliť na šesť základných skupín [46]:

- **Aminokyseliny s alifatickým postranným reťazcom:** alanin (Ala), valin (Val), leucin (Leu), isoleucin (Ile), glycín (Gly)
- **Bazické skupiny s aminovou skupinou na postrannom reťazci:** arginin (Arg), lysín (Lys)
- **S aromatickým jadrom alebo hydroxylovou skupinou na postrannom reťazci:** histidin (His), fenylalanín (Phe), serín (Ser), threonín (Thr), tyrosín (Tyr), tryptofán (Trp)
- **Kyslé skupiny s karboxylovou alebo aminovou skupinou na postrannom reťazci:** kyselina asparagová (Asp), asparagín (Asn), kyselina glutamová (Glu), glutamín (Gln)
- **So sírou v postrannom reťazci:** methionín (Met), cysteín (Cys)
- **Obsahujúce sekundárny amin:** prolin (Pro)

2.3 Syntéza proteínov

Proteíny vznikajú z DNA v procese nazývanom proteosyntéza. Tento proces sa skladá z 2 hlavných častí, ktorými sú transkripcia a translácia.

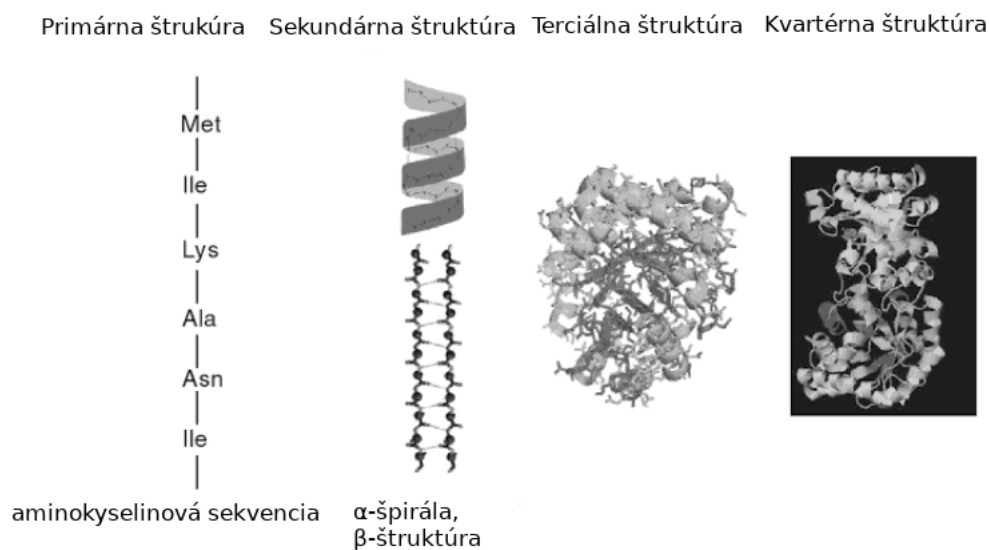
- **Transkripcia:** pri procese transkripcie dochádza k prepisu časti nukleotidovej sekvencie DNA (génu) do molekuly RNA. Dôležitú úlohu zohráva enzým RNA-polymeráza, ktorá musí pred začiatkom transkripcie nájsť oblasť tzv. promotoru obsahujúcu informáciu o začiatku transkripcie a následne sa na túto oblasť naviazať. Proces prepisu končí, keď RNA-polymeráza narazí na sekvenciu tzv. terminátoru. Výsledná molekula RNA sa označuje ako mediátorová RNA (mRNA).
- **Translácia:** pri procese translácie dochádza k prenosu informácie z mRNA do polypeptidového reťazca aminokyselín. Sekvencia nukleotidov RNA sa postupne číta po trojiciach (tzv. kodónoch), pričom každý kodón je preložený na jednu z dvadsiatich aminokyselín. Trojica nukleotidov umožňuje vytvoriť 64 možných kombinácií, takže jedna aminokyselina môže byť reprezentovaná viacerými kodónmi. Výsledkom translácie je proteín.

2.4 Štruktúra proteínov

Popis trojrozsomernej štruktúry proteínov môžeme podľa [46] rozdeliť do štyroch úrovni organizácie:

- **Primárna štruktúra:** sekvencia aminokyselín v polypeptidovom reťazci
- **Sekundárna štruktúra:** zachytáva elementy, ktoré na krátkych úsekoch v sekvencií proteínu zaujímajú podobnú konformáciu. Ide najmä o α -špirálu (α -helix) a β -štruktúra alebo β -skladaný list. α -špirála je také priestorové usporiadanie, kedy reťazec vytvára špirálu. Táto konformácia je stabilizovaná vodíkovými mostíkmi medzi peptidovými väzbami ležiacimi nad sebou [46]. V prípade β -štruktúry prebiehajú úseky reťazca paralelne vedľa seba a sú stabilizované vodíkovými mostíkmi medzi susediacimi úsekmi.
- **Terciálna štruktúra:** reprezentuje trojrozsmerne priestorové usporiadanie zloženého polypeptidového reťazca [46]. Na podobe výslednej terciálnej štruktúry majú vplyv chemické vlastnosti aminokyselín a ich usporiadanie v reťazci.
- **Kvartérna štruktúra:** popisuje usporiadanie jednotlivých polypeptidových reťazcov v molekule proteínu. Týka sa to však iba tzv. oligomerných proteínov, ktoré sú tvorené z viac ako jedného polypeptidového reťazca.

Primárnu, sekundárnu, terciálnu a kvartérnu štruktúru je možné vidieť na obrázku 2.1:



Obr. 2.1: Primárna, sekundárna, terciálna a kvartérna štruktúra. Prevzaté a upravené z [19].

Kapitola 3

Vplyv aminokyselinových substitúcií na stabilitu proteínu

Stabilita je jednou z najdôležitejších vlastností proteínu. Motivácia skúmania stability je dnes veľká, pretože táto vlastnosť proteínov je dôležitá v mnohých oblastiach ako je medicína, kde je snahou dosiahnuť výrobu účinnějších liečiv, v oblasti priemyslu a poľnohospodárstva. Stabilný proteín dokáže lepšie znášať nepriaznivé podmienky okolitého prostredia, akými sú vyššie teploty alebo chemické vlastnosti okolia v ktorom sa proteín nachádza. Na stabilitu proteínu však vplývajú aminokyselinové substitúcie, ktoré môžu proteín stabilizovať, ale aj destabilizovať. Preto vzniká potreba skúmať vplyv substitúcií na stabilitu. V tejto kapitole sa zmienim o stabilite proteínu, čo stabilitu určuje a o dôvodoch vzniku a vplyve mutácií.

3.1 Stabilita proteínu

Stabilita proteínu je určená množinou vzájomne pôsobiacich a ovplyvňujúcich sa síl. Tieto sily určujú, či sa proteín nachádza vo svojom pôvodnom zloženom alebo rozloženom (denaturovanom) stave. Stabilita je úzko prepojená so stavom, v ktorom sa proteín nachádza. Stabilný proteín sa nachádza v zloženom stave, ktorý je stabilizovaný rôznymi vzájomnými interakciami, kde patria hydrofóbne, elektrostatické, vodíkové väzby alebo aj van der Waalsove sily. Naopak, nestabilný proteín sa nachádza v denaturovanom stave, kde dominuje entropická a neentropická voľná energia [19].

Stabilitu proteínu je možné reprezentovať ako zmenu tzv. Gibbsovej (voľnej) energie (ΔG) potrebnej na prechod proteínu zo zloženého do denaturovaného stavu, alebo naopak. Gibbsova voľná energia je termodynamický potenciál vyjadrujúci maximálne množstvo reverzibilnej práce, ktorá môže byť uskutočnená termodynamickým systémom pri konštantnej teplote a tlaku. Gibbsova voľná energia je definovaná nasledujúcim vzťahom [45]:

$$G = H - TS, \quad (3.1)$$

kde H predstavuje entalpiu, T teplotu a S entropiu.

Zmena voľnej Gibbsovej energie (ΔG) je daná vzťahom

$$\Delta G = G_{folded} - G_{unfolded}, \quad (3.2)$$

kde G_{folded} predstavuje voľnú energiu v zloženom a $G_{unfolded}$ energiu v nezloženom stave. Existuje niekoľko laboratórnych metód na určenie stability ako napríklad cirkulárny dichroizmus (CD), diferenciálna skenovacia kalorimetria (DSC), fluorescencia (FI), absorpcia svetla (Abs), nukleárna magnetická rezonancia (NMR) [19].

Určenie stability bez použitia niektorej z laboratórnych metód je možné uskutočniť výpočtom jedného z existujúcich silových polí (Talaris, Score12, ...). Výpočet takéhoto pola ukazuje nasledujúci jednoduchý príklad [39] [19]:

Voľná energia v zloženom stave je daná vzťahom

$$G_F = G_{hy} + G_{el} + G_{hb} + G_{vw} + G_{ss}, \quad (3.3)$$

kde G_{hy} , G_{el} , G_{hb} , G_{ss} , G_{vw} sú hydrofóbne, elektrostatické, vodíkové, disulfidické a van der Waalove voľné energie.

Elektrostatickými interakciami najviac prispievajú nabité postranné reťazce reziduí Lysine, Histodine, Arginine, kyselina asparagová a glutamová.

Vodíkové väzby sú jednými z hlavných zúčastnených pri tvorbe sekundárnej štruktúry proteínu. Výpočet ich príspevku je založený najmä na ich geometrických informáciách.

Voľná energia v nezloženom stave je daná vzťahom

$$G_U = G_{en} + G_{ne}, \quad (3.4)$$

kde G_{en} a G_{ne} sú entropické a neentropické voľné energie.

3.2 Mutácie

Mutácie sú náhodné alebo cielené zmeny v DNA. Sú naprosto nevyhnutné pre biologickú evolúciu, bez nich by sa skôr či neskôr zastavila. Ak by sa výraznejšie zmenili podmienky vonkajšieho prostredia, organizmy by bez mutácií nemuseli na zmeny zareagovať a pravdepodobne by vyhynuli. Za mutácie sú považované všetky také zmeny týkajúce sa genetickej informácie, ktoré nie sú výsledkom segregácií a rekombinácií už existujúcich častí genotypov [50]. Podľa úrovne, na ktorej sa mutácia vyskytla, môžeme rozlišovať [16]:

- **Génové mutácie:** zmena v stavbe DNA, ktorá je reprezentovaná zmenou nukleotidovej sekvencie na určitom mieste. Nazývajú sa tiež bodovými mutáciami a z hľadiska predikcie sú najzásadnejšie.
- **Chromozómové mutácie:** mení sa štruktúra chromozómu.
- **Genómové mutácie:** mení sa počet chromozómov.

3.2.1 Vznik mutácií

Mutácie nevznikajú náhodne, každá mutácia má svoju príčinu za ktorú stojí pôsobenie tzv. mutagénnych faktorov. Medzi najdôležitejšie patria chemické a fyzikálne faktory.

Medzi fyzikálne faktory patria rôzne zdroje žiarenia, najmä ionizujúce a ultrafialové. Poškodenie štruktúry DNA je priamo úmerné množstvu absorbovaného žiarenia.

Medzi chemické faktory môžeme zaradiť genotoxické látky, tzv. genotoxíny. Takýchto látok je veľké množstvo a patria medzi ne napríklad pesticídy, herbicídy, niektoré farbivá, konzervačné a dezinfekčné látky [50].

3.2.2 Typy mutácií

Podľa [16] sa rozlišujú tri základné typy génových mutácií:

- **Substitúcia:** jedná sa o zmenu jedného alebo viacerých nukleotidov. V tomto prípade sa nemení dĺžka pôvodného proteínu. Novovzniknutý proteín sa oproti pôvodnému obvykle líši v jednej aminokyseline.
- **Vloženie:** jedná sa o vloženie jedného alebo viacerých nukleotidov do pôvodnej sekvencie, spôsobuje zväčšenie dĺžky sekvencie.
- **Odstránenie:** odstránenie jedného alebo viacerých po sebe nasledujúcich nukleotidov, mení dĺžku sekvencie rovnako ako vloženie.



Obr. 3.1: Jednotlivé typy mutácií¹.

V prípade, že k mutácií dôjde v kódujúcej oblasti, mutácie je možné rozlíšiť na [16]:

- **Synonymné:** vychádzajú z tzv. degenerovanosti genetického kódu. Zmena nukleotidu v kodóne sa tak na štruktúre proteínu nemusí vôbec prejavovať a vyzerá to tak, ako keby k mutácií vôbec nedošlo.
- **Nesynonymné:** pri zmene nukleotidu v kodóne dochádza k zmene aminokyseliny a rovnako aj k zmene konformácie proteínu.
- **Posunové:** spôsobujú zmenu čítacieho rámca a často vedú k predčasnemu ukončeniu prekladu proteínu.
- **Nezmyselné:** vytvárajú STOP kodón a tým spôsobujú predčasné ukončenie prekladu proteínu.

¹Zdroj: <https://www.bbc.co.uk/education/guides/zc499j6/revision/2>

Na stabilitu proteínu vplývajú aminokyselinové mutácie, ktoré môžu spôsobiť to, že proteín sa stane nestabilným. Preto do hlavnej oblasti skúmania stability patrí predikcia zmeny stability na základe aminokyselinovej mutácie. Jedná sa o predikciu zmeny Gibbsovej voľnej energie ($\Delta\Delta G$) medzi voľnou energiou pôvodného a zmutovaného proteínu. Môžeme ju definovať nasledujúcim vzťahom:

$$\Delta\Delta G = G_{mutant} - G_{wild_type}, \quad (3.5)$$

Podľa tejto hodnoty je možné rozdeliť mutácie na stabilizujúce, neutrálne a destabilizujúce. Väčšia snaha pri predikcií môže viesť k zlepšeniu návrhu nových odolnejších proteínov alebo pri štúdií rozličných chorôb.

Kapitola 4

Strojové učenie

Strojové učenie je v súčasnej dobe chápané ako disciplína umelej inteligencie. Základnou technikou strojového učenia je prehľadávanie stavového priestoru. K charakteristickým vlastnostiam patrí využívanie znalostí, práca so symbolickými či štruktúrovanými premennými [31]. Pojem strojové učenie takisto označuje počítačové metódy pracujúce s obrovským množstvom dát, v ktorých sa hľadajú rôzne vzory a pravidelnosti. Takéto metódy nachádzajú svoje uplatnenie pri hľadaní riešení v mnohých odvetviach akými sú počítačové videnie, rozpoznávanie reči a takisto bioinformatika. Keďže strojové učenie nachádza v mnohých odvetviach čoraz väčšie uplatnenie, je potrebné brať do úvahy jeho výhody a rovnako aj nevýhody. Medzi výhody patrí automatické hľadanie spojitostí vo veľkom množstve dát. Medzi hlavné nevýhody metód patrí neschopnosť správnej analýzy dát pri nevyváženosti predložených dát, nesprávne výsledky pri malom množstve tréningových dát pre metódu, alebo príliš veľký počet parametrov vstupných dát.

V tejto kapitole sa budem venovať základným technikám strojového učenia a popisom používaných algoritmov.

4.1 Úvod do strojového učenia

Podľa [2] je možné algoritmy strojového učenia rozdeliť do 3 základných skupín:

- **Klasifikácia:** Rieši problém priradenia výstupnej triedy vstupným dátam, ktoré môžu byť reprezentované vektorom hodnôt. Ako príklad si je možné predstaviť zatriedenie žiadateľov o pôžičku do tried s vysokým alebo nízkym rizikom toho, že pôžičku nebudú schopní splácať na základe rôznych údajov o žiadateľoch.
- **Regresia:** Regresné metódy na rozdiel od klasifikačných nepriradujú vstupom výstupnú triedu, ale snažia sa určiť priamo číselnú hodnotu výstupu. Príkladom môže byť určenie ceny ojazdeného auta na základe parametrov ako počet najazdených kilometrov, značka, rok výroby.
- **Hľadanie asociácií:** Asociačné pravidlá slúžia na hľadanie zaujímavých asociácií vo veľkom objeme dát. Pri ich hľadaní nás zaujíma podmienená pravdepodobnosť, ktorá sa uvádza vo forme $P(X|Y)$, Y je produkt podmienený výskytom produktu X .

Metódy strojového učenia môžeme ďalej rozdeliť na základe spôsobu, akým sa učia. Podľa [2] sa rozdeľujú na:

- **Učenie s učiteľom:** Pri tomto type učenia je nutné mať k dispozícii vstupné aj výstupné dáta. Cieľom je nájsť vzťahy medzi vstupom a výstupom, ktoré slúžia na naučenie metódy. Medzi algoritmy patriace do tejto skupiny sa radí regresia aj klasifikácia.
- **Učenie bez učiteľa:** V tomto type učenia nie sú k dispozícii referenčné výstupné dáta, ale len vstupné. Snahou je nachádzať pravidelnosti vo vstupných dátach. Medzi takéto metódy patria rôzne typy zhlukovania.

4.2 Rozhodovacie stromy

Rozhodovací strom je hierarchický model so stromovou štruktúrou. Metódy tohto typu používajú učenie s učiteľom a môžeme ich použiť na klasifikáciu aj regresiu. Štruktúra stromu je tvorená z dvoch typov uzlov, vnútorných (nelistových) a listových uzlov. Každý z nelistových uzlov obsahuje testovaciu funkciu. Po vyhodnotení tejto funkcie sa vyberie nasledujúci uzol, v ktorom sa bude pokračovať. Tento proces začína v koreňovom uzle a pokračuje rekurzívne až do dosiahnutia listového uzlu. Listový uzol obsahuje označenie triedy, do ktorej bude zaradený vstupný vektor, alebo číselnú hodnotu.

4.2.1 Algoritmus J48

Algoritmus J48 patrí k metódam rozhodovacích stromov. Jedná sa o modifikovanú implementáciu algoritmu C4.5 v nástroji WEKA. Algoritmus produkuje klasifikačno-rozhodovací strom pre poskytnuté dáta rekurzívnym rozdeľovaním dát. Pri rozhodovaní je využitá stratégia depth-first. Algoritmus berie do úvahy všetky možné testy, ktoré môžu rozdeliť dáta, a vyberá test udávajúci najlepšiu informačnú hodnotu. Pre každý diskretný atribút je zvážený jeden test s počtom výsledkov, ktorý zodpovedá počtu rôznych hodnôt atribútov.

4.2.2 Algoritmus Náhodný strom (Random Tree)

Pri tomto algoritme je strom náhodným stromom vytvoreným náhodne z množstva všetkých možných stromov. Každý list obsahuje k náhodných parametrov. Náhodné vytvorenie stromu v tomto kontexte znamená, že každý strom v množine stromov má rovnakú šancu výberu. Kombinácia veľkého počtu náhodných stromov obvykle vedie k správne mu modelu.

4.2.3 Algoritmus Náhodný les (Random Forest)

Random Forest [9] je metóda založená na kombinácii viacerých rozhodovacích stromov. Každý strom závisí na hodnotách náhodného vektora hodnôt navzorkovaného nezávisle a s rovnakým rozložením pre všetky stromy v tzv. lese stromov. Obecne môžeme metódu popísať nasledujúcou definíciou:

Náhodný les (random forest) je klasifikátor tvorený kolekciou klasifikátorov so stromovou štruktúrou $\{h(x, \Theta_k), k = 1, \dots\}$, kde $\{k\}$ sú nezávisle identicky rozdelené náhodné vektory a každý strom hlasuje jednotlivo o najpopulárnejšej triede vo vstupe x .

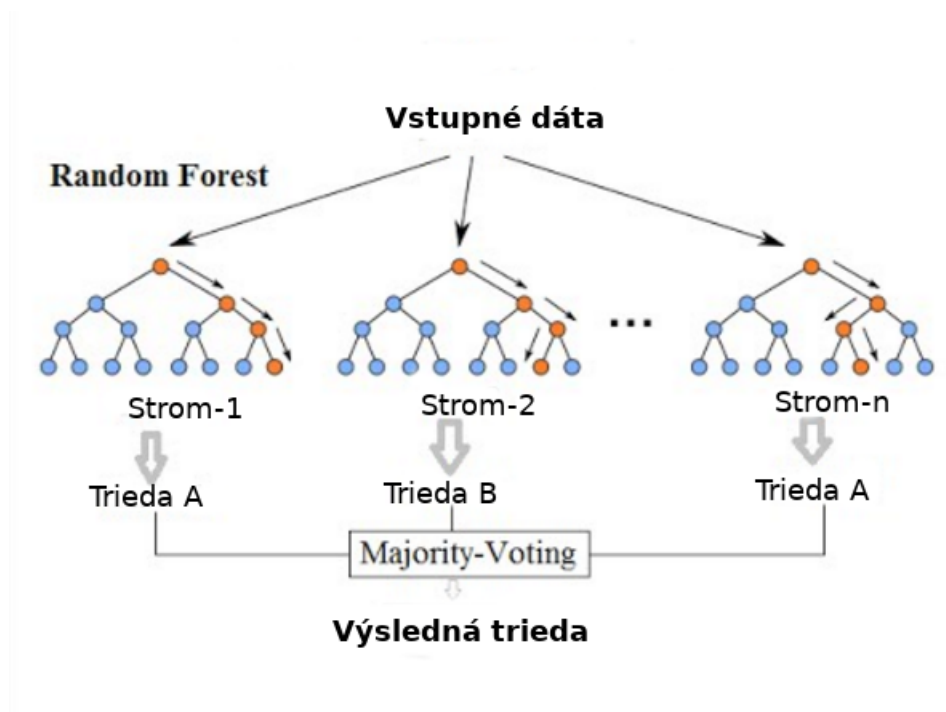
Najväčšiu pozornosť tejto metódy tvoria 3 vlastnosti:

- poskytuje presnú predikciu pre mnohé typy aplikácií
- je schopná merať dôležitosť jednotlivých parametrov pri tréovaní modelu
- blízkosť medzi vzorkami môže byť meraná tréovaným modelom

Algoritmus náhodný les pre klasifikáciu aj regresiu je možné zjednodušene popísať nasledovne, pričom je uvažovaných M rozhodovacích stromov. Schéma metódy je znázornená na obrázku 4.1:

- Pre každý z M rozhodovacích stromov sa vytvorí sada tréovacích dát z originálnych dát. Na ich výber slúži metóda tzv. bagging, ktorá náhodne vyberie zadaný počet tréovacích dát.
- Pre každú množinu tréovacích dát je vytvorený klasifikačný alebo regresný strom, ktorý je následne natréovaný na M -tej náhodnej množine tréovacích dát. V tejto metóde je každý uzol rozdelený najlepším rozdelením spomedzi podmnožiny prediktorov náhodne vybraných v tomto uzle. Naopak, pri klasických stromoch je uzol rozdelený na základe najlepšieho rozdelenia medzi všetkými premennými.
- Predikcia nových dát spojením výsledkov predikcie M stromov, napríklad hlasovaním väčšiny (tzv. majority voting) pri klasifikácii alebo priemerom hodnôt pri regresii.

Rozšírenie algoritmu náhodný les je momentálne veľmi aktívnou oblasťou vo výpočetnej biológii. Metóda nachádza veľké uplatnenie v bioinformatike, napríklad aj pri nástrojoch predikujúcich stabilitu proteínov.



Obr. 4.1: Metóda Random Forest¹.

4.3 Support vector machines (SVM)

Support vector machines patria k najnovším metódam strojového učenia. Prvýkrát boli publikované v prácach [7] [13]. Tieto metódy uskutočňujú klasifikáciu konštruovaním N -dimenzionálnej nadroviny, ktorá optimálne rozdeľuje dáta do dvoch kategórií. Cieľom je nájsť takú nadrovinu, ktorá rozdelí vstupné vektory tak, že jedna skupina vektorov je na jednej strane roviny a druhá na strane opačnej. Vektory nachádzajúce sa blízko nadroviny sú označované ako tzv. podporné vektory (support vectors).

Ak sú tréningové dáta lineárne rozdeliteľné, potom pár (\mathbf{w}, b) existuje ako

$$\mathbf{w}^T \mathbf{x}_i + b \geq 1, \text{ pre všetky } \mathbf{x}_i \in P$$

$$\mathbf{w}^T \mathbf{x}_i + b \leq -1, \text{ pre všetky } \mathbf{x}_i \in N$$

s rozhodovacím pravidlom daným vzťahom $f_{\mathbf{w},b}(x) = \text{sgn}(\mathbf{w}^T x + b)$, kde \mathbf{w} je váhový vektor a b je odchýlka (tzv. bias).

V prípade, že dáta sú lineárne rozdeliteľné do dvoch tried, optimálnu nadrovinu je možné nájsť minimalizovaním štvorcovej normy rozdeľujúcej nadroviny. Jedná sa o konvexný kvadratický programovací problém. Pri možnosti lineárneho rozdelenia sa snaží SVM nájsť 1-dimenzionálnu nadrovinu (priamku), ktorou rozdelí skupiny vstupných vektorov. Po rozdelení dát priamkou metóda zistí vzdialenosť priamky od najbližších podporných vektorov. Táto vzdialenosť sa označuje ako tzv. krajná hranica (margin), pričom sa hľadá najväčšia vzdialenosť medzi podpornými vektormi.

Lineárne rozdeliteľné dáta sú však len ideálnym príkladom. Ak by analýza pozostávala len z premenných z dvoch kategórií, dvoch predikovaných premenných a množiny bodov rozdeliteľných priamkou, bolo by to veľmi jednoduché. V skutočnosti sa tieto metódy musia vysporiadať s viac ako dvomi predikovanými premennými, rozdelením dát nelineárnymi krivkami alebo množinami dát, ktoré nie je možné úplne rozdeliť.

4.3.1 Jadrové funkcie

Pri väčšine reálnych problémov neexistuje lineárna nadrovina rozdeľujúca pozitívne a negatívne vzorky v tréningových dátach. Jedným z riešení je prenesenie dát do priestoru, ktorý má viac dimenzií a definovať rozdeľujúcu nadrovinu v tomto priestore. Takýto viacdimenzionálny priestor sa nazýva priestor transformovaných vlastností. S vhodne vybraným priestorom transformovaných vlastností dostatočnej dimenzie je možné rozdeliť ľubovoľnú tréningovú dátovú sadu. Mapovanie dát do iného (potencionálne nekonečného) Hilbertovho priestoru H je definované ako $\Phi: R^d \rightarrow H$. Tréningový algoritmus bude potom závisieť len na dátach cez bodové produkty v H , napríklad na funkciách v tvare $\Phi(x_i) \cdot \Phi(x_j)$. Ak by existovala tzv. jadrová funkcia K taká, že $K(x_i, x_j) = \Phi(x_i) \cdot \Phi(x_j)$, v algoritme by sme potrebovali iba K .

Jadrové funkcie sú špeciálnou triedou funkcií umožňujúce výpočet vnútorných produktov priamo v priestore vlastností bez nutnosti mapovania dát tak, ako to bolo popísané vyššie. Akonáhle je vytvorená nadrovina, jadrová funkcia je použitá na mapovanie nových bodov do priestoru vlastností pre klasifikáciu.

¹Prevzaté a upravené z <http://bit.ly/2rpjSDK>

Výber vhodnej aproximačnej jadrovej funkcie je dôležitý, pretože funkcia definuje transformovaný priestor vlastností v ktorom budú klasifikované trénovacie dáta. Medzi najpoužívanejšie patria tieto jadrové funkcie [2]:

- Polynomiálna funkcia stupňa P , ktorý je zvolený užívateľom

$$K(x, y) = (x \cdot y + 1)^P \quad (4.1)$$

- Radiálne-bázová funkcia (RBF)

$$K(x, y) = e^{-\|x-y\|^2/2\sigma^2} \quad (4.2)$$

- Sigmoidná funkcia

$$K(x, y) = \tanh(\kappa x \cdot y - \delta)^P \quad (4.3)$$

4.3.2 Algoritmus SMO

Sekvenčná minimalizačná optimalizácia (SMO) [37] je algoritmus na trénovanie SVM, ktorý jednoducho rieši kvadratický SVM problém. SMO uskutočňuje dekompozíciu celkového problému na podproblémy riešené analyticky. Metóda si v každom kroku vyberá na vyriešenie najmenší optimalizačný problém. Pre typický kvadratický SVM problém, najmenší možný optimalizačný problém zahŕňa dva Lagraengove násobitele. V každom kroku si metóda vyberie dva tieto násobitele na spoločnú optimalizáciu, nájde pre ne optimálne hodnoty a aktualizuje SVM. Výhodou tohto algoritmu je, že množstvo potrebnej pamäte pri trénovacej sade je lineárne, čo umožňuje algoritmu pracovať s veľkými trénovacími sadami.

4.4 Algoritmus Naive Bayes

Naive Bayes je klasifikačným algoritmom pre klasifikačné problémy, kde sa vyskytujú dve alebo viac tried. Je založený na Bayesovom teoréme s nezávislými predpokladmi medzi prediktormi. Klasifikátor predpokladá, že efekt hodnoty x prediktora na danú triedu c je nezávislý na hodnotách ďalších prediktorov. Predpoklad sa nazýva triedna podmienená nezávislosť.

Tento model je jednoduchý na vytvorenie, no napriek svojej jednoduchosti klasifikátor často poskytuje dobré výsledky a je široko používaný, pretože v mnohých prípadoch prekonáva komplikovanejšie klasifikačné metódy.

Kapitola 5

Ensemble metódy

V posledných rokoch sa metódy strojového učenia využívajú vo veľkej miere v oblasti bioinformatiky. Napriek ich výhodám vznikajú rôzne problémy spojené najmä s nedostatkom dát, veľkou rôznorodosťou dát, alebo tzv. preučeníím metód. Jedným z riešení, ktoré vykazujú dobré výsledky, je využitie výstupov z viacerých klasifikátorov namiesto použitia jedného samostatného klasifikátora. Takáto kombinácia môže niekedy poskytnúť protichodné výsledky a tým pomôže zvýšiť presnosť a robustnosť predikcie. Prístup využitia viacerých klasifikátorov je označovaný ako tzv. ensemble stratégia.

Podľa [38] existuje mnoho teoretických a praktických dôvodov na použitie ensemble systémov:

- **Štatistické dôvody:** Dobrý výkon metódy na tréningových dátach nemusí predpovedať dobrú výkonnosť zovšeobecňovania. Množina klasifikátorov s podobným výkonom na tréningových dátach môže mať rôznu zovšeobecňovaciu výkonnosť. Tento poznatok je možné vidieť najmä, ak testovacie dáta určené na rozlíšenie schopnosti zovšeobecňovať nie sú dostatočne reprezentatívne. V takýchto prípadoch môže spriemerovanie výstupov niekoľkých klasifikátorov znížiť riziko zlého výberu alebo slabého klasifikátora.
- **Veľký objem dát:** V určitých oblastiach môže nastať problém príliš veľkého objemu dát, ktorý má byť spracovaný. To nie je niekedy možné efektívne uskutočniť len pomocou jedného klasifikátora. V takýchto situáciách je vhodné zvážiť rozdelenie dát na menšie celky, trénovať klasifikátory na menších celkoch a skombinovať ich výstupy pomocou vhodného kombinačného pravidla.
- **Malý objem dát:** Dostupnosť dostatočného a reprezentatívneho množstva tréningových dát je dôležitá pre klasifikačný algoritmus, aby bol schopný dosiahnuť úspešného naučenia rozdelenia dát. Pri nedostatku tréningových dát sa osvedčili techniky prevzorkovania, ktoré sú vhodné na vytvorenie náhodných prekrývajúcich sa podmnožín dostupných dát a každú je možné použiť na učenie iného klasifikátora.
- **Rozdeľuj a panuj:** Napriek dostatku dát sú niektoré problémy pre klasifikátory príliš zložité, napríklad rozhodovacia hranica rozdeľujúca dáta môže byť veľmi komplexná, alebo leží mimo oblasti funkcií dostupných pre vybraný klasifikačný model. Ako príklad je možné uviesť dáta, ktoré nie je možné lineárne rozdeliť. Žiadny lineárny klasifikátor nie je schopný dáta rozdeliť, avšak vhodná kombinácia lineárnych klasifikátorov tzv. ensemble systému by bola schopná naučiť sa túto nelineárnu hranicu.

Dáta sú v tomto prípade rozdelené na menšie celky, pričom každý klasifikátor sa učí jednu z častí.

5.1 Rôznorodosť v ensemble systémoch

V posledných rokoch niekoľko štúdií teoreticky [21] aj empiricky [32] ukázalo, že použitie ensemble systému viacerých klasifikátorov prekonáva výsledky samostatného klasifikátora, keď členovia systému sú dostatočne presní a robia niekoľko zhodných chýb. Pretože je takmer nemožné natréňovať klasifikátor, ktorý by dokonale zvládol rozdeliť stavový priestor bez výskytu chyby, sú potrebné klasifikátory, ktoré robia chyby na odlišných častiach tréningových dát. Ak sú splnené tieto podmienky, je možné dosiahnuť robustnejšej a presnejšej predikcie.

Ako príklad tohto faktu je možné uviesť demonštráciu uvedenú v jednej zo Zhangových štúdií [49]. V nej je demonštrované, že ensemble systém pozostávajúci z troch rovnakých klasifikátorov s presnosťou 95% je horší ako ensemble pozostávajúci z klasifikátorov, ktorých presnosť je len 67% a najmenšou chybou medzi dvojicou klasifikátorov. Na základe týchto poznatkov je možné usúdiť, že rôznorodosť medzi členmi ensemble systému je rozhodujúca pri zlepšovaní predikcie.

Existuje niekoľko možností ako dosiahnuť vysokú rôznorodosť jednotlivcov:

- **Rôzne tréningové dátové sady:** Tréningová dátová sada môže byť rozdelená na menšie celky a každá takáto menšia podsada je použitá na natréňovanie jedného samostatného klasifikátora. Na rozdelenie dát môže byť použitá technika z niekoľkých dostupných prevzorkovacích techník. Jednou z najbežnejších takýchto techník je tzv. bootstrap technika. Existuje však niekoľko ďalších, napríklad metódy ako tzv. jack-knife.
- **Rôzne tréningové parametre:** Použitie rôznych tréningových parametrov pre jednotlivé klasifikátory je tiež jednou z možných ciest. Napríklad viacvrstvá neurónová sieť môže byť tréňovaná využitím rôznych vstupných váh vrstiev a uzlov v každej vrstve. Tento prístup umožňuje kontrolu nestability klasifikátorov a ako následok prispieva k ich rozmanitosti.
- **Rôzne parametre pri predikcii:** Lineárny klasifikátor môže mať niekedy problém s nachádzaním vzorov u parametrov, ktoré sú veľmi vzdialené. Pre riešenie tohto problému môže byť prospešné experimentovanie s rôznymi podsadami predikovaných parametrov generovaných buď metódami učenia s učiteľom alebo bez neho [44].
- **Rôzne typy klasifikátorov:** Neexistuje žiadny nadradený klasifikátor, ktorý by poskytoval najlepšie výsledky bez ohľadu na oblasť odborníka alebo množstva tréningových dát. Na druhej strane, na každý problém zvyčajne existuje niekoľko vhodných klasifikátorov, preto výber niekoľkých z nich a kombinácia ich výstupov je možnou cestou ako vytvoriť množinu klasifikátorov s rôznymi rozhodovacími hranicami.

5.2 Tvorba ensemble systémov

Existujú dva základné spôsoby, ako vybrať členov ensemble systému: bagging a boosting. Príklad ensemble systému je možné vidieť na obrázku 5.1.

5.2.1 Bagging

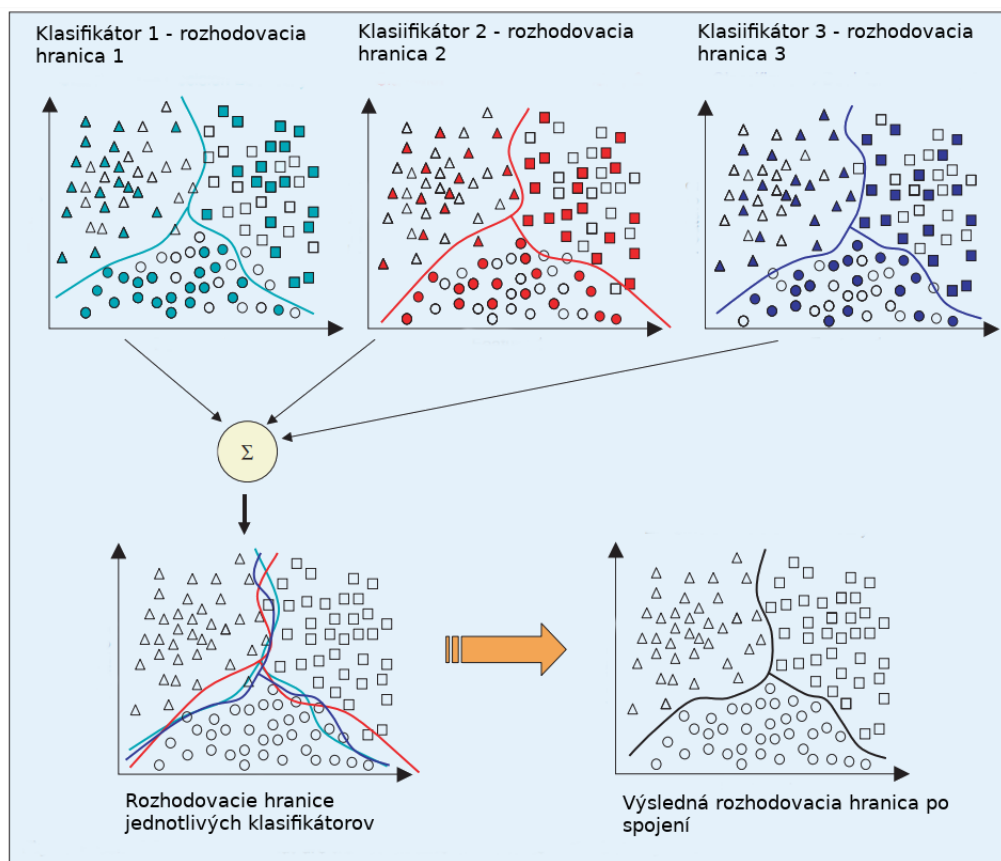
Bagging [8] alebo aj tzv. bootstrap aggregation je jednou z najstarších, ľahko implementovateľných ensemble stratégií. Bola navrhnutá na zlepšenie presnosti algoritmov strojového učenia používaných pri štatistickej klasifikácii a regresii. Rôznorodosť je dosiahnutá využitím tzv. bootstrap podmnožín tréningových dát, pričom rôzne tréningové množiny sú vybrané náhodne s náhradou z celého množstva tréningových dát. Každá takáto podmnožina je určená na tréning iného klasifikátora rovnakého typu. Nakoniec sa využije hlasovanie väčšiny na výsledkoch jednotlivých klasifikátorov. Trieda vybraná najväčším počtom klasifikátorov sa stáva konečným rozhodnutím ensemble systému.

Bagging je výkonným mechanizmom najmä pri obmedzenom množstve spoľahlivých dát. Na zaistenie dostatočného množstva tréningových vzorkov v podmnožinách je približne 75 až 100% vzorkov prítomných v každej podmnožine. Nastáva tak veľký prekryv medzi tréningovými podmnožinami a mnoho vzorkov sa nachádza viackrát v danej podmnožine. V takomto prípade sa rôznorodosť zaisťuje použitím nestabilného klasifikačného modelu pre ktorý je možné získať rôzne rozhodovacie hranice s rôznymi tréningovými dátami. Techniky ako neurónové siete alebo rozhodovacie stromy sú dobrými príkladmi na tento účel, pretože ich nestabilitnosť je kontrolovateľná výberom ich parametrov.

5.2.2 Boosting

Boosting [41] je ďalším algoritmom na výber členov ensemble systému. Bolo dokázané, že slabý klasifikátor (iba o niečo úspešnejší ako náhodné hádanie) je možné pretvoriť na silný klasifikátor, ktorý poskytuje správne predikcie pre všetky prípady z ľubovoľne malej časti prípadov. Rovnako ako bagging, boosting vytvára súbor klasifikátorov prevzorkovaním dát s použitím hlasovania väčšiny. Prevzorkovanie je pri boostingu zlepšené, aby každému klasifikátoru boli poskytnuté najviac informatívne tréningové dáta. Boosting vytvára tri slabé klasifikátory:

- Prvý klasifikátor C_1 je tréningovaný na náhodnej podmnožine dostupných dát.
- Tréningová podsada pre klasifikátor C_2 je vybraná ako najinformatívnejšia podsada vo vzťahu k C_1 . Klasifikátor C_2 je teda tréningovaný na dátach, na ktorých mal C_1 polovičnú úspešnosť.
- Tretí klasifikátor C_3 je tréningovaný na dátach, kde C_1 a C_2 nesúhlasili.
- Nakoniec sú klasifikátory skombinované hlasovaním väčšiny.



Obr. 5.1: Ensemble systém. Prevzaté a upravené z [38].

5.3 Spojenie klasifikátorov

Po natrénovaní jednotlivých klasifikátorov je potrebné ich spojiť najvhodnejším spôsobom. Tieto metódy môžeme rozdeliť do dvoch kategórií: netrénovateľné a trénovateľné. Netrénovateľné sú použiteľné, ak samostatný klasifikátor poskytuje porovnateľné výsledky vo väčšine častí priestoru vlastností. Trénovateľné metódy dynamicky menia rozhodovacie pravidlá podľa špecifického typu klasifikovaného prípadu, sú užitočné v prípadoch, keď klasifikátory konštantne správne alebo nesprávne klasifikujú určité prípady.

5.3.1 Trénovateľné metódy

Medzi trénovateľné metódy patria nasledovné:

- *Stacked generalization*: Snahou je vytvoriť metaklasifikátor s využitím poznatkov o presnosti klasifikátorov. Dostupné dáta sú rozdelené na tréningovú a testovaciu sadu. Všetky nástroje sú trénované na tréningovej sade, testovacia sada je použitá na určenie výkonnosti klasifikátorov a tvorbu metaklasifikátoru.
- *Rozhodcovské stromy*: Jedná sa o prístup zdola nahor pri tvorbe ensemble systému. V prvom kroku sú dáta rozdelené do konečného počtu neprekrývajúcich sa podsád a každá z nich je určená na tréning toho istého typu klasifikátora. Pre každý pár

je vytvorený tzv. rozhodca a proces sa rekurzívne opakuje pokým nezostane len jeden klasifikátor na aktuálnej úrovni.

5.3.2 Netrénovateľné metódy

Do kategórie netrénovateľných metód môžeme zaradiť nasledujúce metódy:

- *Hlasovanie väčšiny*: Najjednoduchšia metóda, ktorá priradí objektu triedu na základe počtu hlasov jednotlivých klasifikátorov. Existujú 3 typy: jednotné hlasovanie, kde sa všetky klasifikátory musia zhodnúť na predikcii; jednoduché hlasovanie v ktorom sa aspoň polovica klasifikátorov musí zhodnúť na rozhodnutí; väčšinové hlasovanie, kde je výsledok daný podľa počtu hlasov.
- *Váňované hlasovanie väčšiny*: Tento prístup je vylepšením predchádzajúceho prístupu, jednotlivé klasifikátory majú rôznu váhu v konečnom rozhodovaní a tieto váhy by mali byť prispôsobené ich presnosti.
- *Bayesova kombinácia*: Váňovaná metóda, váha spojená s klasifikátorom je tzv. posterior pravdepodobnosť klasifikátora na trérovacej sade.
- *Váňovanie entropie*: Rovnako sa jedná o váňovanú metódu a váhy klasifikátorov sú nepriamo úmerné entropiám klasifikačného vektoru.

Kapitola 6

Nástroje na predikciu stability

Ako je už uvedené v kapitole 4, strojové učenie nachádza čoraz väčšie uplatnenie v rôznych oblastiach akou je aj bioinformatika. V oblasti skúmania a predikcie stability vzniká mnoho nástrojov využívajúcich práve rôznych techník a metód strojového učenia. Takéto nástroje sú však zatiaľ len vo svojich začiatkoch a majú mnoho nedostatkov spojených najmä s potrebnými dátami. Ďalšou skupinou nástrojov na predikciu stability sú také, ktoré využívajú tzv. silové polia. Túto skupinu nástrojov môžeme rozdeliť na také, ktoré využívajú fyzikálne efektívne energetické funkcie simulujúce základné sily pôsobiace medzi atómami a na metódy založené na štatistických potenciáloch, pre ktoré sú energie odvodené z výskytu reziduí alebo atómových spojoch uvedených v dátových sadách pozostávajúcich z experimentálne charakterizovaných mutovaných proteínov. V tejto kapitole sa zameriam na stručné zhodnotenie a opis rôznych predikčných nástrojov patriacich do vyššie uvedených kategórií.

6.1 Strojové učenie

Strojové učenie nachádza čoraz širšie uplatnenie v bioinformatike a mnohé predikčné nástroje ho využívajú ako svoj základ. V tejto sekcii sa nachádza charakteristika nástrojov využívajúcich práve metódy strojového učenia.

6.1.1 AUTO-MUTE

Nástroj využíva klasifikačné metódy na určenie vplyvu mutácie a regresiu na určenie hodnoty $\Delta\Delta G$. Na začiatku je identifikovaných šesť najbližších susedov aminokyseliny na mutovanej pozícii a určia sa ich vlastnosti. Vektor vlastností, tvorený vlastnosťami založenými na energiách, je určený na natrénovanie vybranej metódy zvolenej užívateľom. Nástroj využíva algoritmy z balíka programu WEKA, konkrétne je možné zvoliť metódy Random Forest a SVM pre klasifikáciu alebo Tree Regression a SVM Regression pre regresiu. AUTO-MUTE je dostupný ako webový server alebo ako samostatná aplikácia [30].

6.1.2 I-Mutant

Nástroj poskytuje predikciu znamienka zmeny stability pomocou klasifikácie a tiež určenie hodnoty $\Delta\Delta G$ pomocou regresie. I-Mutant pracuje so štrukturálnymi alebo sekvenčnými informáciami o mutovanej aminokyseline a jej najbližších susedoch. Pre strojové učenie je použitá metóda SVM s jadrovou funkciou RBF (Radial Basis Function). Trénovanie

dáta pozostávajú z podmnožiny dát pochádzajúcich z databázy ProTherm. Vstupný vektor vlastností mutácie tvorí spolu 42 hodnôt. Nástroj je dostupný ako webový server [10].

6.1.3 iPTREE-STAB

iPTREE-STAB slúži na určenie vplyvu mutácie na stabilitu a hodnoty $\Delta\Delta G$ z informácií o sekvenciách. Predikcia je založená najmä na využití rozhodovacích stromov spojených s algoritmom adaptive boosting, a takisto využíva aj regresných a klasifikačných stromov. Dáta slúžiace na natrénovanie metód tvorí databáza ProTherm. Pre predikciu využíva informácie zo susedných aminokyselín, konkrétne používa okno o veľkosti 7 reziduí. Vektor vlastností tvorí 5 hodnôt, ktorými sú aminokyselina po mutácii, pôvodná aminokyselina, hodnota pH, teplota pri ktorej bola meraná stabilita a informácie o troch susedných reziduách. Nástroj je dostupný iba ako webový server [23].

6.1.4 EASE-AA

Nástroj ponúka predikciu vplyvu mutácie a takisto určenie hodnoty $\Delta\Delta G$. Predikcia je založená iba na informáciách o sekvenciách. EASE-AA využíva informácie o konzervovanosti, ktoré sú získané pomocou metódy SIFT. Konzervovanosť je vyjadrená pomocou tzv. SIFT skóre. Ďalej využíva štrukturálne informácie, kde patria sekundárna štruktúra, dostupná povrchová plocha (ASA), pravdepodobnosť poruchy (tzv. disorder probability) a informácie o vlastnostiach aminokyselín, ktoré tvorí 7 hodnôt vlastností. Vektor vlastností je vstupom pre metódu SVM tvoriacu základ nástroja, rovnako ako I-Mutant využíva jadrovú funkciu RBF. Nástroj je dostupný ako samostatná aplikácia [17].

6.1.5 mCSM

Metóda identifikuje pre miesto mutácie všetky atómy v zadanej vzdialenosti od geometrického stredu. Následne je uskutočnený výpočet vzdialenosti medzi jednotlivými atómami prostredia a vygeneruje sa matica atómových vzdialeností. mCSM nakoniec vytvorí grafovo založené vzory atómových vzdialeností. Mutácia je reprezentovaná tzv. podpisovým vektorom, ktorý slúži na tréning a testovanie metód strojového učenia. Nástroj je dostupný ako webová služba [36].

6.1.6 MAESTRO

MAESTRO je multiagentný systém strojového učenia. K predikciám využíva informácie o sekvenciách a štruktúre spoločne s 2 štatistickými funkciami, slúži na predikciu jednobodových ale aj viacbodových mutácií. Systém využíva kombináciu neurónových sietí, SVM a lineárnej regresie, ktoré sú natrénované na nezávislých dátach a nakoniec sú ich výsledky spojené. Vstupným vektorom strojového učenia je 9 hodnôt, ktoré sú rozdelené na štatistické skórovacie funkcie a vlastnosti proteínov, ako napríklad veľkosť proteínu. Špeciálnou vlastnosťou prediktora je možnosť predikcie stabilizujúcich disulfidových mostíkov. Nástroj je dostupný ako samostatná aplikácia a aj ako webový server [29].

6.1.7 ELASPIC

ELASPIC je nástroj využívajúci ensemble stratégie. Slúži na predikciu hodnoty $\Delta\Delta G$. Na začiatku nástroj predikuje proteínové domény a ich interakcie a detekuje proteínové jadrá.

Mutácie sú mapované a molekulárne, evolučné a energetické vlastnosti sú určené na vytvorenie prediktívneho modelu využívajúceho rozhodovacie stromy s algoritmom Stochastic Gradient boosting. ELASPIC je dostupný ako samostatná aplikácia aj ako webový server [47].

6.2 Energetická funkcia - fyzikálny potenciál

Nasledujúca sekcia obsahuje stručnú charakteristiku nástrojov využívajúcich k predikcií fyzikálny potenciál.

6.2.1 CC/PBSA

Jedná sa o štruktúrne založenú metódu pre rýchle a kvantitatívne odhadnutie voľnej energie potrebnej k zloženiu mutovaného proteínu. V prvom kroku metóda uskutočňuje rýchle generovanie alternatívnych proteínových konformácií pomocou programu CONCOORD na navzorkovanie dostupného konfiguračného priestoru. Energetická funkcia založená na silových poliach a riešenie Poisson-Boltzmannovej rovnice sú spriemerované nad vytvoreným celkom. Voľná energia je nakoniec aproximovaná ako suma elektrostatických interakcií, van der Waalsových energií a zmenou entropie. Nástroj je dostupný ako webová služba [5].

6.2.2 ERIS

Nástroj ERIS je zložený na silovom poli Medusa a pozostáva z celoatómového silového poľa, algoritmu obaľovania bočného reťazca a tzv. backbone relaxation metódy. Parametre silového poľa boli trénované s vysokým rozlíšením proteínových štruktúr. ERIS je možné použiť ako webovú službu aj ako samostatný program [48].

6.2.3 Rosetta

Rosetta predstavuje sadu softvéru na makromolekulárne modelovanie. Zahŕňa celú skupinu rôznych silových polí založených na kombinácii prispievateľov voľnej energie ako sú van der Waalsova energia alebo elektrostatické interakcie. Väčšina z protokolov tohto nástroja je veľmi časovo náročná, a preto je nástroj dostupný iba ako samostatná aplikácia [26].

6.2.4 CUPSAT

Metóda je založená na výpočte atómových potenciálov a potenciálov torzných uhlov, ktoré boli získané z proteínových štruktúr pomocou webového serveru PISCES. Hodnoty Boltzmannovej energie sú vypočítané z radiálnej párovej distribúcie atómov aminokyselín a Gaussovská apodizačná funkcia je aplikovaná na priradenie výhodných energetických hodnôt pre susediace orientácie pozorovaných torzných uhlov. Nástroj je dostupný ako webová služba [34].

6.3 Energetická funkcia - štatistický potenciál

Nasledujúca sekcia obsahuje stručnú charakteristiku nástrojov využívajúcich k predikcií štatistický potenciál.

6.3.1 PopMuSiC

Zmena stability danej bodovej mutácie je vypočítaná na základe štruktúry pôvodného nezmutovaného proteínu a množine energetických funkcií. Hodnota $\Delta\Delta G$ je vyjadrená ako lineárna kombinácia 13 štatistických potenciálov. Nástroj je dostupný ako webová služba [15].

6.3.2 DMutant

Nástroj DMutant využíva kombináciu orientácie na základe poznatkov, vzdialenostne závislý potenciál atómov aminokyselín a potenciál torzného uhlu. Tieto časti tvoria diskriminačnú funkciu, ktorej parametre boli optimalizované tréningovou dátovou sadou. Nástroj je dostupný v podobe samostatného programu [22].

6.3.3 FoldX

Energetická funkcia nástroja zahŕňa požiadavky u ktorých bolo zistené, že sú dôležité pre stabilitu. Patria tu van der Waalsovej príspevky, rozdiel solvatačnej energie alebo vodíkové väzby. Energetická funkcia je tvorená 8 časťami, ktoré sú nakoniec lineárne skombinované. Nástroj je dostupný ako samostatná aplikácia [20].

Kapitola 7

Dátová sada a jej parametre

Vytvorenie spoľahlivej, dostatočne veľkej a rozmanitej trérovacej dátovej sady je rozhodujúcou vlastnosťou pri tvorbe každého predikčného nástroja. Dáta pre testovanie a trénovanie prediktoru boli z veľkej časti získané z databázy ProTherm [4]. Jedná sa o najrozsiahlejšiu databázu obsahujúcu termodynamické parametre akými sú Gibbssova voľná energia, tepelná kapacita alebo entalpia. Databáza taktiež obsahuje informácie o použitých metódach a meraniach, experimentálnych podmienkach a informácie o aktivite proteínu, sekundárnej štruktúre a dostupnosti pôvodných reziduí. ProTherm je prepojená aj s ďalšími databázami, napríklad so sekvenčnou SWISS-PROT [6], štrukturálnou PDB [43] alebo funkcionálnou PMD [25].

Databáza ProTherm bola naposledy aktualizovaná vo februári v roku 2013 a aktuálne obsahuje približne 26 000 záznamov jedno a viacbodových mutantov navrhnutých nad viac ako 740 unikátnymi proteínmi získaných rôznymi experimentálnymi technikami. Pre tvorbu testovacích a trérovacích dátových sád pre existujúce nástroje predikujúce vplyv aminokyselínových substitúcií na stabilitu proteínu je najpoužívanejším zdrojom dát práve databáza ProTherm. V súčasnom stave však trpí množstvom seriózných nedostatkov. Aby sa predišlo problémom spojených s touto databázou, vyextrahované boli iba mutácie u ktorých sú uvedené zmeny stability a overené všetky zdroje. Najväčšími problémami pri získavaní dát boli:

- Chýbajúca hodnota $\Delta\Delta G$
- Opačné znamienka hodnoty $\Delta\Delta G$
- Trojstavové skladanie - niektoré publikácie uvažujú 3 hodnoty Gibbsovej voľnej energie: nezložený stav, medzistav a zložený stav. Databáza ProTherm však nerozlišuje medzi prechodom z nezloženého stavu do medzistavu a z medzistavu do zloženého stavu.

Aby bola vytvorená dátová sada spoľahlivá a aby eliminovala možné experimentálne chyby v nameraných hodnotách zmeny Gibbsovej voľnej energie ($\Delta\Delta G$), záznamy s hodnotami $\Delta\Delta G$ z intervalu $[-0.5, 0.5]$ boli odstránené. Záznamy s hodnotou $\Delta\Delta G \geq 0.5 \text{ kcal.mol}^{-1}$ boli označené za destabilizujúce a záznamy s $\Delta\Delta G \leq -0.5 \text{ kcal.mol}^{-1}$ boli označené za stabilizujúce. Tento rozhodovací prah bol zvolený podľa tvrdenia, že experimentálna chyba merania $\Delta\Delta G$ je približne $0.48 \text{ kcal.mol}^{-1}$ [27]. V prípade, že pre jednu mutáciu bolo uskutočnených viac meraní, ponechaný bol iba záznam merania, ktoré prebehlo pod experimentálnou hodnotou pH, ktorá bola blízko fyziologickej hodnote pH 7. Každý záznam bol ďalej rozšírený o štrukturálne informácie.

7.1 Parametre záznamu mutácie

Každý záznam mutácie je tvorený z 8 hodnôt, ktoré tvoria vstupný vektor pre strojové učenie a určenie vplyvu mutácie. V tejto časti sú popísané jednotlivé parametre záznamu.

- **Zmena polarity:** Tento údaj poskytuje informáciu o zmene v polarite aminokyseliny po mutácii oproti polarite pôvodnej aminokyseliny. Údaj bol získaný z tabuľkových hodnôt pre skúmané aminokyseliny. Uvažované boli len polárne a nepolárne aminokyseliny.
- **Zmena náboja:** Hodnota vyjadruje informáciu o zmene náboja pôvodnej aminokyseliny a aminokyseliny po mutácii. Výsledný údaj bol získaný z tabuľkových hodnôt pre dané aminokyseliny. Uvažovaný bol negatívny, pozitívny a neutrálny náboj.
- **Zmena indexu hydrofobicity:** Údaj o výslednej zmene hydrofobicity mutovanej a pôvodnej aminokyseliny. Číselný údaj značí rozdiel medzi tabuľkovými hodnotami týchto porovnávaných aminokyselín [28].
- **Zmena veľkosti aminokyseliny:** Údaj o zmene veľkosti pôvodnej aminokyseliny. Aminokyseliny boli rozdelené do 3 intervalov na základe ich veľkosti. V jednom intervale sa tak nachádzajú len aminokyseliny, pre ktoré platí, že rozdiel ich veľkostí je menší alebo rovný hodnote 50. Následne sa zisťovalo do akého z jednotlivých intervalov patrí pôvodná a mutovaná aminokyselina. Z tohto údaju bolo možné určiť zmenu veľkosti z veľkej na malú, opačnú zmenu alebo žiadnu zmenu, ak aminokyseliny patrili do rovnakého intervalu.
- **Sekundárna štruktúra proteínu:** Údaj o sekundárnej štruktúre proteínu bol získaný použitím modulu DSSP prítomného v knižnici BioPython. Pre zjednodušenie bol každému proteínu priradený tento údaj len z 3 možností, ktorými sú helix, coil a sheet.
- **Dostupná povrchová plocha (ASA):** Jedná sa o údaj, ktorý udáva plochu aminokyseliny, ktorú môže dosiahnuť rozpúšťadlo. Na výpočet tohto parametru bol opäť použitý modul DSSP prítomný v knižnici BioPython.
- **Konzervovanosť:** Jeden z najdôležitejších parametrov v zázname mutácie. Informácia o konzervovanosti pozície na ktorej došlo k mutácii je reprezentovaná jednou z 5 hodnôt od 0 po 4, ktoré kódujú percentuálne vyjadrenie miery konzervovanosti danej pozície. Na výpočet bol použitý fylogenetický strom a Felsteinov algoritmus, ktorý je podrobnejšie popísaný v časti 7.2. Mutácie na konzervovanej pozícii sú z hľadiska skúmania veľmi zaujímavé, pretože konzervované aminokyseliny bývajú dôležité pre stabilitu, aktivitu, alebo schopnosť proteínu vytvoriť terciárnu štruktúru.
- **Korelácia:** Výpočet korelácie mutovanej pozície s ostatnými pozíciami vo viacnásobnom zarovnaní. Stĺpec v zarovnaní predstavuje náhodnú veličinu, počíta sa vzájomná informácia, ktorá vyjadruje závislosť dvojice náhodných veličín. Údaj o vzájomnej informácii dostaneme pomocou nasledujúceho vzťahu:

$$I(X; Y) = \sum_{y \in Y} \sum_{x \in X} p(x, y) \log_2 \frac{p(x, y)}{p(x)p(y)} \quad (7.1)$$

Po vypočítaní koeficientov sa zistí, či sa medzi získanými koeficientami nachádza aspoň jeden s hodnotou nad stanoveným prahom. Ak áno, pozícia je korelovaná, v opačnom prípade nie je. Korelácia danej pozície tiež znamená, že sa v priebehu evolúcie menila spoločne s pozíciami v sekvenciách, na ktorých boli získané korelačné koeficienty nad stanoveným prahom.

7.2 Felsteinov algoritmus

Felsteinov algoritmus slúži na riešenie tzv. maličkého likelihood problému a snahou je určiť tzv. likelihood hodnotu stromu pre zadanú pozíciu v sekvenciách. Vstupom je viacnásobné zarovnanie sekvencií, tvar stromu spoločne s ohodnotením vnútorných uzlov a dĺžka hrán. Algoritmus je založený na princípe dynamického programovania a pozostáva z viacerých krokov. Prvým krokom je priechod stromom od listov ku koreňu a vyhodnotenie L-hodnoty pre každý uzol podľa pravidiel, ktoré sú nasledovné:

- listový uzol

$$L_{s_k}(k) = \begin{cases} 0 & k \neq \text{nukleotid v liste} \\ 1 & k = \text{nukleotid v liste} \end{cases} \quad (7.2)$$

- vnútorný uzol

$$L_{s_k}(k) = \left[\sum_{s_i} P_{s_k s_i}(t_i) \cdot L_{s_i}(i) \right] \times \left[\sum_{s_j} P_{s_k s_j}(t_j) \cdot L_{s_j}(j) \right] \quad (7.3)$$

- koreňový uzol

$$L = \sum_{s_0} \pi_{s_0} \cdot L_{s_0}(0) \quad (7.4)$$

L-hodnota je uložená v poli, ktoré obsahuje každý uzol stromu. V tomto prípade každý uzol obsahuje pole s váhami jednotlivých nukleotidov, ktoré sú počítané počas priechodu stromom. Pri použití algoritmu v tejto práci obsahuje pole hodnoty s váhami pre jednotlivé sekvencie vo viacnásobnom zarovnaní. Počas priechodu sa počítajú hodnoty v poli každého uzlu a výsledkom je pole obsiahnuté v koreni stromu. Pole s váhami sa použije na zistenie konzervovanosti konkrétnej pozície. Počítaním váh sa sledujú zmeny na pozíciách v priebehu evolúcie proteínu a na základe koeficientov je vypočítaná konzervovanosť.

Kapitola 8

Implementácia

Finálnou časťou práce je implementácia predikčného nástroja. Táto kapitola bližšie popisuje výber použitých technológií, spôsob realizácie nástroja, dosiahnuté výsledky a porovnanie finálnej implementácie s podobnými nástrojmi.

8.1 Testovanie vo WEKE

Výber vhodnej metódy strojového učenia na základe otestovania rôznych metód na zostavenej dátovej sade predstavoval prvú fázu pri tvorbe predikčného nástroja. Na otestovanie dátovej sady bol zvolený nástroj **WEKA**.

WEKA [18] predstavuje voľne dostupný balík algoritmov strojového učenia napísaného v programovacom jazyku Java. Ide o veľmi populárny nástroj určený na použitie v akademickej ale aj komerčnej sfére. Nástroj poskytuje množstvo algoritmov strojového učenia, ktoré je možné rýchlo vyskúšať a porovnať na zvolenej dátovej sade; a nástroje na predspracovanie analyzovaných dát. WEKA poskytuje algoritmy na regresiu, zhľukovanie, klasifikáciu, získavanie asociačných pravidiel a výber atribútov. Pred testovaním bola dátová sada prevedená do formátu ARFF, natívneho formátu nástroja. Nástroj ponúka nasledovné triedy algoritmov: Bayes, Functions, Lazy, Meta, Mi, Misc, Rules, Trees. Z týchto tried bolo vybraných niekoľko algoritmov, ktoré sú bližšie popísané v kapitole 4. Zvolené boli najmä rôzne stromové algoritmy a rôzne verzie SVM. Pomocou týchto metód bola dátová sada otestovaná a na základe výsledkov boli vybrané najlepšie metódy na ďalšie použitie pri implementácií. Výsledky testovania obsahuje tabuľka 8.1.

Testovanie zvolených metód bolo vykonané na celej dátovej sade, pri testovaní bola použitá krížová validácia a testovacie dáta boli rozdelené do desiatich častí, pričom konečné výsledky sú spriemerované z niekoľkých testovaní. Nastavená bola aj tzv. cost-sensitive matica, ktorá slúžila na postihovanie predikcie stabilizujúcich mutácií ako destabilizujúcich, pretože v trénovacej sade je väčší počet destabilizujúcich mutácií ako stabilizujúcich a poslúžila na vyvažovanie nerovnomerného počtu záznamov. Výsledky testovania jednotlivých metód sú v tabuľke 8.1.

Tabuľka 8.1 obsahuje údaje TP (true positive) rate o presnosti identifikácie stabilizujúcich a destabilizujúcich položiek, ktoré sú skutočne stabilizujúce alebo destabilizujúce, FP (false positive) rate vyjadrenie presnosti určenia stabilizujúcej mutácie ako destabilizujúcej a naopak. Posledným údajom je celková presnosť metódy pri predikcii na testovacej podsade. Na základe tohto údaju boli vybraté najlepšie metódy.

Metóda strojového učenia	TP rate	FP rate	Accuracy
Naive Bayes	0.776	0.626	0.737
LibSVM	0.786	0.706	0.766
SMO	0.774	0.774	0.6
DecisionTable	0.774	0.774	0.6
RandomForest	0.793	0.692	0.797
RandomTree	0.793	0.574	0.766
J48	0.774	0.626	0.74

Tabuľka 8.1: Výsledky testovanie algoritmov strojového učenia

Z tabuľky 8.1 je vidieť, že z metód založených na rozhodovacích stromoch dosiahla najlepšie výsledky metóda Random Forest a z metód implementujúcich SVM mala najlepší výsledok metóda LibSVM. Obidve metódy dosiahli presnosť vyššiu ako 70%. V prípade SVM mal druhý algoritmus SMO horšie výsledky. Metóda Naive Bayes založená na bayesovskej pravdepodobnosti dosiahla len o niečo horšie výsledky ako zvyšné metódy. Potvrdili sa však očakávania dobrých výsledkov algoritmu Random Forest aj SVM, a preto boli tieto metódy určené ako základ vytváraného ensemble systému predikčného nástroja.

8.2 Príprava záznamu mutácie v Pythone

Na implementáciu predikčného nástroja bol zvolený programovací jazyk **Python**, najmä kvôli dostupnosti mnohých bioinformatických nástrojov a knižníc, ale aj pre jednoduché implementovanie strojového učenia vďaka dostupným knižniciam. Program má charakter konzolovej aplikácie ovládanej z príkazového riadku s niekoľkými vstupnými parametrami od užívateľa.

Prvou fázou pri tvorbe nástroja bola implementácia skriptu s prípravou záznamu reprezentujúceho jednu mutáciu, ktorý tvorí vstup pre strojové učenie. Príprava záznamu zahŕňa výpočet jednotlivých parametrov mutácie a celkovo pozostáva z výpočtu 8 hodnôt. Výsledný záznam má podobu CSV súboru s jednotlivými parametrami.

Pre výpočet hodnôt zmien náboja, polarity, indexu hydrofobicity a veľkosti aminokyseliny sú v príslušnej triede prítomné dátové štruktúry obsahujúce hodnoty náboja, polarity, indexu hydrofobicity a veľkosti pre jednotlivé aminokyseliny. Hodnoty parametrov sa vypočítajú na základe zmeny podľa určenia pôvodnej aminokyseliny a aminokyseliny po mutácií.

Výpočet údajov o sekundárnej štruktúre proteínu a hodnote dostupnej povrchovej plochy (ASA) prebieha spoločne. Na ich výpočet bola použitá knižnica **Biopython** [12], konkrétne modul **DSSP**. Pomocou neho je možné získať obidva údaje z PDB súboru. Pre správne fungovanie modulu je však nutné mať prítomný samostatný program **dssp** [24], ktorý modul používa na výpočet.

Proces výpočtu konzervovanosti mutovanej pozície je o niečo zložitejší ako ostatné výpočty. Vstupom výpočtu je **FASTA** súbor so sekvenciou požadovaného typu reťazca a PDB súbor. **FASTA** súbor slúži ako vstup programu **BLAST** [3], ktorý bol použitý na získanie homologných sekvencií, zvolená bola hodnota 250 sekvencií. Výstup v podobe XML súboru je spracovaný do podoby textového súboru slúžiaceho ako vstup programu **Clustal Omega** [42], ktorý vytvára viacnásobné zarovnanie vstupných sekvencií. PDB súbor je určený na získanie počiatočnej pozície rezidua, ktorá spoločne s indexom mutácie slúži na získanie výsledného indexu mutácie vo viacnásobnom zarovnaní. Výpočet je založený na vytvorení fylogenetického stromu a použití Felsteinovho algoritmu. Na vytvorenie stromu je použitý

program **FastTree** [40]. Na implementáciu fylogenetického stromu v Pythone bola použitá knižnica **ete3**¹, ktorá využíva na zostavenie stromu výstup programu **FastTree**. Výsledné váhy získané Felsteinovým algoritmom spoločne s viacnásobným zarovnaním slúžia na konečný výpočet konzervovanosti, ktorá je rozdelená do 5 úrovní.

Posledným parametrom je určenie korelácie mutovanej pozície. Na výpočet je opäť použité viacnásobné zarovnanie sekvencií. Jednotlivé hodnoty parametrov, ktoré nemajú číselnú hodnotu výstupu, sú nakoniec zakódované do svojej číselnej reprezentácie, aby s nimi mohla pracovať metóda strojového učenia.

8.3 Testovanie metód strojového učenia v Pythone

Ďalšou fázou tvorby nástroja bola implementácia metód strojového učenia. Na implementáciu jednotlivých algoritmov som zvolil knižnicu **Scikit-learn** [35]. Scikit-learn je knižnica strojového učenia pre programovací jazyk Python a zahŕňa v sebe rozličné klasifikátory, zhlučovacie algoritmy ako napríklad support vector machines, random forests a gradient boosting.

Podľa výsledkov testovania rôznych metód uvedených v časti 8.1 dosiahli najlepšie výsledky metódy SVM a Random Forest. Na začiatku tejto fázy vývoja som otestoval implementácie týchto metód z knižnice Scikit-learn na dátovej sade, z ktorej boli odstránené záznamy slúžiace ako testovacie dáta. Testovacia sada obsahuje 250 záznamov a bola vytvorená náhodným výberom z pôvodnej sady. Je tvorená zo 45 stabilizujúcich mutácií a 206 destabilizujúcich mutácií. Trénovacia sada obsahuje 1315 záznamov. Výsledky testovania sú reprezentované dvomi hodnotami, presnosťou (accuracy) a hodnotou korelácie vyjadrenej Matthewsovým korelačným koeficientom (MCC). Výsledky testovania sú prítomné v tabuľke 8.2.

Metóda	Accuracy	MCC
Random Forest	0.712	0.34
SVM	0.813	0.36

Tabuľka 8.2: Výsledky testovanie algoritmov z knižnice Scikit-learn

Tabuľka 8.2 ukazuje, že v presnosti dosiahla metóda SVM veľmi dobré výsledky, presnosť bola 81%. Metóda Random Forest dosiahla presnosti približne 71%. Obe metódy dosiahli podobných výsledných hodnôt Matthewsovho korelačného koeficientu, SVM dosiahla hodnoty 0.36 a Random Forest 0.34. Pri testovaní metódy SVM bolo použitých niekoľko jadrových funkcií a najlepšie výsledky dosiahla polynomiálna jadrová funkcia. Hodnota MCC je pre vyhodnotenie metódy smerodatnejšia a vyjadruje vzťah predikovaných a skutočných hodnôt. Dosiahnuté hodnoty korelácie budú slúžiť na porovnanie výsledkov samostatne použitej metódy a vytvoreného ensemble systému.

¹Dostupné na <http://etetoolkit.org/>

8.4 Ensemble systém v Pythone

Základom ensemble systému je použitie viacerých klasifikátorov a spojenie ich výsledkov. Pred natrénovaním jednotlivých metód bolo potrebné zostaviť tréningové dáta pre tento systém. Vytvorenie dát spočívalo vo vytvorení menších podsád z pôvodnej sady. Na ich vytvorenie som využil techniku bagging s prekryvom dát v podsadách. Na zostavenie podsád bola použitá sada 1315 záznamov z predchádzajúceho testovania. Celkovo bolo vytvorených 14 tréningových podsád, ktoré boli najskôr samostatne otestované na jednotlivých metódach.

Dátová sada	Stabilizujúce záznamy	Destabilizujúce záznamy	MCC (SVM)	MCC (RF)
1.	100	100	0.4	0.25
2.	100	150	0.42	0.29
3.	130	100	0.36	0.24
4.	150	100	0.41	0.31
5.	150	150	0.43	0.3
6.	150	200	0.45	0.34
7.	200	150	0.41	0.29
8.	200	200	0.43	0.34
9.	200	250	0.52	0.29
10.	200	250	0.48	0.32
11.	250	200	0.53	0.37
12.	250	200	0.47	0.34
13.	250	250	0.47	0.34
14.	250	250	0.48	0.29

Tabuľka 8.3: Výsledky testovania podsád metódami Random Forest a SVM

Tabuľka 8.3 ukazuje výsledky testovania jednotlivých podsád na vybraných metódach. Ako je možné vidieť, niektoré veľkosti podsád sa opakujú a to z dôvodu obmeny záznamov v týchto sadách a zaistenia väčšieho rozptylu u destabilizujúcich záznamov. Podsady boli testované s prítomnosťou všetkých parametrov, v ďalšom testovaní došlo k odstráneniu niektorých parametrov, cieľom bolo zistiť ako takéto kombinácie parametrov budú pôsobiť na predikciu. Vytvorené podsady boli otestované bez parametra zmeny náboja, veľkosti dostupnej povrchovej plochy a kombinácií parametrov zmeny indexu hydrofobicity a dostupnej povrchovej plochy. Výsledky testovania sú uvedené v tabuľke 8.4.

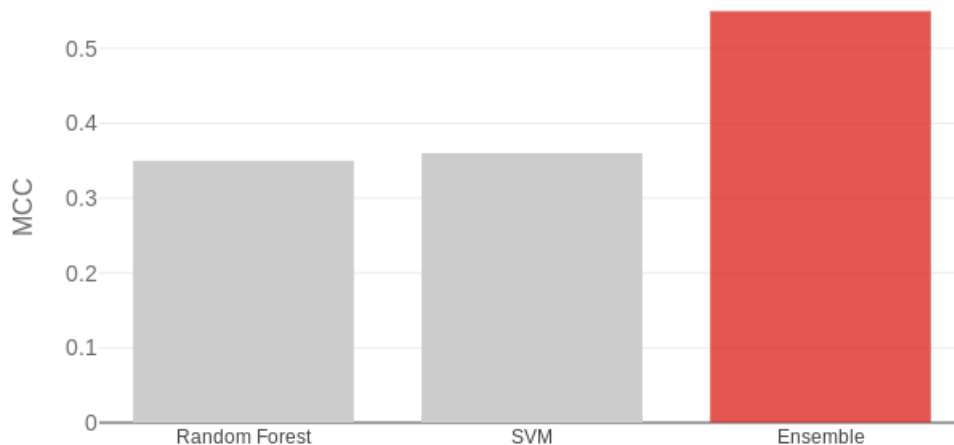
Dataset	MCC (SVM)			MCC (RF)		
1.	0.3	0.27	0.32	0.17	0.2	0.2
2.	0.31	0.33	0.4	0.19	0.25	0.21
3.	0.28	0.4	0.38	0.18	0.2	0.13
4.	0.36	0.36	0.39	0.21	0.29	0.22
5.	0.41	0.42	0.41	0.23	0.28	0.24
6.	0.44	0.41	0.4	0.29	0.3	0.25
7.	0.4	0.29	0.39	0.28	0.26	0.3
8.	0.47	0.36	0.42	0.29	0.28	0.29
9.	0.45	0.36	0.41	0.26	0.27	0.31
10.	0.36	0.42	0.46	0.28	0.29	0.32
11.	0.43	0.39	0.42	0.32	0.32	0.27
12.	0.44	0.4	0.43	0.31	0.33	0.32
13.	0.41	0.36	0.41	0.28	0.27	0.33
14.	0.44	0.42	0.44	0.3	0.28	0.31

Tabuľka 8.4: Výsledky testovania podsád bez parametra dostupnej povrchovej plochy, zmeny náboja a kombinácie zmeny indexu hydrofobicity a dostupnej povrchovej plochy.

Tabuľka 8.4 ukazuje výsledky testovania podsád bez odstránených parametrov. Testovania so všetkými parametrami ale aj s odstránenými slúžili na získanie údajov o korelačnom koeficiente. Na základe získaných údajov boli vybrané podsady s najlepšimi výsledkami. Tieto podsady sú základom ensemble systému a každá z nich slúži na natrénovanie jednej metódy strojového učenia.

Implementovaný ensemble systém pozostáva so spojenia metód Random Forest a SVM, konkrétne bola metóda Random Forest natrénovaná trikrát a metóda SVM štyrikrát. Každým natrénovaním bol získaný údaj o výslednej triede, nakoniec bolo k dispozícii sedem hodnôt predikovanej triedy pre mutáciu. Výsledná hodnota predikovanej triedy bola získaná hlasovaním väčšiny, výsledkom bola trieda, ktorá sa medzi hodnotami nachádzala častejšie.

Ensemble systém bol vo finálnej implementácii natrénovaný na podsadách č. 8, 9, 11, 14. Metóda Random Forest bola natrénovaná na podsadách č. 8, 11, 13 so všetkými parametrami. Metóda SVM bola natrénovaná na podsade č. 9, 11 a 14 so všetkými parametrami a podsade č. 8 bez parametra dostupnej povrchovej plochy. Finálna hodnota korelačného koeficientu na testovacej sade dosiahla hodnotu 0.55.

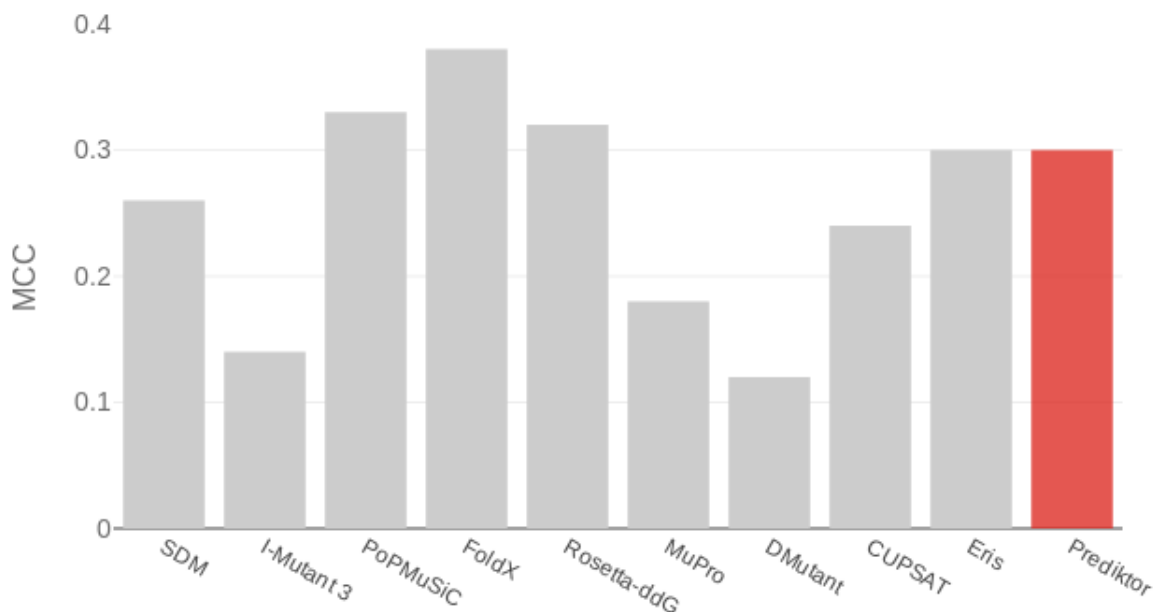


Obr. 8.1: Graf s porovnaním hodnôt MCC na testovacej sade.

8.5 Porovnanie s inými nástrojmi

Po implementovaní prediktora som pristúpil k porovaniu predikčnej sily s niekoľkými dostupnými nástrojmi. Tieto nástroje boli zvolené podľa toho, či boli otestované na testovacej sade S350 [14], ktoré mnohé nástroje používajú ako svoju testovaciu sadu. Táto sada obsahuje 350 záznamov mutácií, z ktorých je 95 stabilizujúcich a 255 destabilizujúcich. Porovnávanie spočívalo v otestovaní implementovaného prediktora na sade S350 a porovnaní výsledkov s nástrojmi, ktoré boli otestované na tejto sade. Daný postup bol zvolený najmä preto, že natrénovanie a otestovanie iných nástrojov na vytvorenej dátovej sade by nebolo úplne možné, pretože mnoho nástrojov je dostupných len v podobe webovej aplikácie.

Zo sady S350 neboli odstránené záznamy s hodnotou $\Delta\Delta G$ v intervale $[-0.5, 0.5]$, ktoré sa v dátovej sade nevyskytujú. Záznamy s týmito hodnotami boli ponechané, pretože bez nich by porovnanie nebolo uskutočnené na rovnakej dátovej sade. Každému záznamu bolo dodatočne nutné určiť príslušnú triedu, pretože sada obsahuje iba hodnotu $\Delta\Delta G$. Rozhodovacia hranica bola stanovená tak, že mutácie s $\Delta\Delta G \geq 0$ boli označené ako stabilizujúce a zvyšné mutácie ako destabilizujúce. Hranice sú naopak oproti dátovej sade, upravené boli po dodatočnom skúmaní sady S350.



Obr. 8.2: Graf s porovnaním hodnôt MCC vybraných nástrojov

Graf na obrázku 8.2 ukazuje porovnanie hodnôt korelačného koeficientu u rôznych nástrojov na sade S350. Z grafu vidieť, že najlepšiu hodnotu dosiahol nástroj FoldX, ktorý patrí do kategórie nástrojov využívajúcich štatistický potenciál. Medzi ďalšie nástroje s vysokou hodnotou korelácie patria nástroje PoPMuSiC rovnako patriaci do kategórie nástrojov využívajúcich štatistický potenciál a nástroj Rosetta využívajúci fyzický potenciál. Všetky uvedené nástroje dosiahli koreláciu väčšiu ako 0.3, pričom medzi najlepšie nástroje zo všetkých uvedených patrí Rosetta.

Zvyšné zvolené prediktory dosiahli menšiu hodnotu korelácie, najlepšie výsledky z nich dosiahli nástroje SDM [33], CUPSAT a Eris. SDM predstavuje nástroj založený na získavaní znalostí, CUPSAT využíva na výpočet potenciál atómov a torzných uhlov, Eris využíva fyzický potenciál. Zostávajúcimi nástrojmi sú I-Mutant3 a MuPro [11]. Obidva uvedené nástroje zhodne využívajú strojové učenie a konkrétne metódu SVM.

U mnohých ďalších nástrojov je problém v získaní údajov o použitej dátovej sade na trénovanie a rovnako spôsob validácie nástroja, takže ich nie je možné brať do úvahy pri porovnávaní nástrojov.

Pri porovnaní implementovaného nástroja s inými nástrojmi využívajúcimi strojové učenie je vidieť, že implementovaný prediktor využívajúci ensemble prístup dosiahol lepšej korelácie ako nástroje I-Mutant3, MuPro, DMutant, CUPSAT a Eris. Korelácia nástroja I-Mutant3 dosiahla hodnotu 0.14, MuPro dosiahol korelácie 0.18, DMutant dosiahol hodnoty 0.12. CUPSAT a Eris mali najlepšie hodnoty, CUPSAT 0.24 a Eris 0.3. Implementovaný

prediktor dosiahol na dátovej sade S350 koreláciu 0.3, čo je v porovnaní s týmito hodnotami zlepšenie o 0.16, 0.12, 0.18, 0.06 a pri Erise boli dosiahnuté hodnoty totožné.

Výsledky ensemble systému boli porovnané aj s hodnotami dosiahnutými použitím samostatných metód Random Forest a SVM. Samotný Random Forest dosiahol na sade S350 koreláciu 0.26 a SVM 0.28.

Prediktor dosiahol hodnotu korelácie 0.3, čo spomedzi testovaných nástrojov predstavuje dobrú hodnotu, podobnú testovaným. Najlepšie výsledky pri predikcii však dosahujú nástroje založené na využití štatistického alebo fyzického potenciálu, aj keď hodnota MCC nie je najvyššia.

Kapitola 9

Záver

Hlavnou náplňou tejto práce bolo vytvorenie predikčného nástroja využívajúceho metódy strojového učenia, ktorý by určoval vplyv aminokyselinových mutácií na stabilitu proteínu.

Keďže ide o predikčný nástroj, v prvom kroku bolo potrebné získať dáta určené na tréning a testovanie nástroja. Zostavený dataset obsahuje 1564 jednobodových mutácií získaných z databázy *ProTherm*. Jedná sa o unikátny dataset, očistený od mutácií obsahujúce rôzne nedostatky pri uvádzaných hodnotách mutácií. Každá mutácia je doplnená o 8 parametrov určených k vytvoreniu rozhodovacej hranice.

V nasledujúcom kroku bolo potrebné zvoliť metódy strojového učenia. Pomocou nástroja WEKA bolo na zostavenom datasete odskúšaných viacero dostupných metód a nakoniec boli vybrané metódy Random Forest a SVM. Tieto metódy sa stali základom implementovaného ensemble systému.

Nástroj obsahuje ensemble systém kombinujúci metódy Random Forest a SVM na predikciu výsledného vplyvu mutácie. Zvolený prístup slúži na vysporiadanie sa s menším množstvom dát a rovnako na zlepšenie robustnosti predikcie. Na vytvorenie ensemble stratégie bolo potrebné vytvoriť menšie podsady, ktoré slúžili na natréning jednotlivých metód. Takýto prístup predikcie dosiahol na testovacej dátovej sade o veľkosti 250 mutácií vytvoreného z pôvodnej dátovej sady hodnotu korelačného koeficientu 0.55. Oproti použitiu samostanej metódy Random Forest, ktorá dosiahla na zostavenej sade koreláciu 0.34 alebo metóde SVM s koreláciou 0.36 ide o značné zlepšenie tejto hodnoty.

Na záver bol nástroj porovnaný s vybranými nástrojmi na testovacej sade mutácií S350. Na tejto sade dosiahol prediktor hodnotu korelačného koeficientu 0.3 a presnosť dosiahla 67%.

Návrhom zlepšenia predikčného nástroja je určite získanie väčšieho množstva relevantných dát, čo ale v dnešnej dobe nie je jednoduché, pretože databáza *ProTherm* obsahuje rôzne nedostatky a dáta nepribúdajú dostatočne rýchlo. Ďalším návrhom je možnosť zvolenia iných parametrov pre predikciu, ktoré by mohli mať dôležitý vplyv na stabilitu, alebo zlepšenie ich výpočtu. V rámci zlepšenia výpočtu bolo v tejto práci implementované vyhodnotenie konzervovanosti pomocou zostavenia fylogenetického stromu a použitia Felsteinovho algoritmu.

Literatúra

- [1] Alberts, B.: *Základy buněčné biologie: úvod do molekulární biologie buňky*. 1998, ISBN 80-902-9062-0.
- [2] Alpaydin, E.: *Introduction to Machine Learning*. MIT Press, 2010, ISBN 978-0-262-01243-0.
- [3] Altschul, S. F.; Madden, T. L.; Schäffer, A. A.; aj.: Gapped BLAST and PSI-BLAST: a new generation of protein database search programs. *Nucleic Acids Research*, ročník 25, č. 17, 1997: s. 3389–3402.
- [4] Bava, K. A.; Gromiha, M. M.; Uedaira, H.; aj.: ProTherm, version 4.0: Thermodynamic Database for Proteins and Mutants. *Nucleic Acids Research*, 2004.
- [5] Benedix, A.; Becker, C. M.; Groot, B. L.; aj.: Predicting free energy changes using structural ensembles. *Nature methods*, 2009.
- [6] Boeckmann, B.; Bairoch, A.; Apweiler, R.; aj.: The SWISS-PROT protein knowledgebase and its supplement TrEMBL in 2003. *Nucleic Acids Research*, 2003.
- [7] Boser, B.; Guyon, I.; N. Vapnik, V.: A Training Algorithm for Optimal Margin Classifier. ročník 5, 1996.
- [8] Breiman, L.: Bagging Predictors. *Machine Learning*, 1996.
- [9] Breiman, L.: Random Forests. *Machine Learning*, ročník ročník 45, 2001.
- [10] Capriotti, E.; Fariselli, P.; Rossi, I.; aj.: A Three-State Prediction of Single Point Mutations on Protein Stability Changes. *BMC Bioinformatics*, 2008.
- [11] Cheng, J.; Randall, A.; Baldi, P.: Prediction of Protein Stability Changes for Single-Site Mutations Using Support Vector Machines. *Proteins*, ročník 62, 2006.
- [12] Cock, P. J. A.; Antao, T.; Chang, J. T.; aj.: Biopython: freely available Python tools for computational molecular biology and bioinformatics. *Bioinformatics*, ročník 25, č. 11, 2009: s. 1422–1423.
- [13] Cortes, C.; Vapnik, V.: Support-Vector Networks. *Machine Learning*, ročník 20, č. 3, 1995: s. 273–297.
- [14] Dehouck, Y.; Grosfils, A.; Folch, B.; aj.: Fast and accurate predictions of protein stability changes upon mutations using statistical potentials and neural networks: PoPMuSiC-2.0. *Bioinformatics*, ročník 25, č. 19, 2009: s. 2537–2543.

- [15] Dehouck, Y.; Kwasigroch, J. M.; Gilis, D.; aj.: PoPMuSiC 2.1: a web server for the estimation of protein stability changes upon mutation and sequence optimality. *BMC Bioinformatics*, 2011.
- [16] Flegel, J.: *Úvod do evoluční biologie*. Academia, 2007, ISBN 978-80-200-1539-6.
- [17] Folkman, L.; Stantic, B.; Sattar, A.: Towards sequence-based prediction of mutation-induced stability changes in unseen non-homologous proteins. *BMC Genomics*, 2014.
- [18] Frank, E.; Hall, M.; Trigg, L.; aj.: Data Mining in Bioinformatics using Weka. *Bioinformatics*, 2004.
- [19] Gromiha, M. M.: *Protein Bioinformatics: From sequence to function*. Elsevier, 2010, ISBN 978-81-312-2297-3.
- [20] Guerois, R.; Nielsen, J. E.; Serrano, L.: Predicting changes in the stability of proteins and protein complexes: a study of more than 1000 mutations. *Journal of Molecular Biology*, 2002.
- [21] Hansen, L. K.; Salamon, P.: Neural Network Ensembles. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, ročník 12, 1990: s. 993 – 1001.
- [22] Hoppe, C.; Schomburg, D.: Prediction of protein thermostability with a direction- and distance-dependent knowledge-based potential. *Protein Society*, 2005.
- [23] Huang, L. T. .; Gromiha, M. M.; Ho, S. Y.: iPTREE-STAB: interpretable decision tree based method for predicting protein stability changes upon mutations. *Bioinformatics*, 2007.
- [24] Kabsch, W.; Sander, C.: Dictionary of protein secondary structure: pattern recognition of hydrogen-bonded and geometrical features. *Biopolymers*, ročník 22, 1983: s. 2577–2637.
- [25] Kawabata, T.; Ota, M.; Nishikawa, K.: The protein mutant database. *Nucleic Acids Research*, 1999.
- [26] Kellogg, E.; Leaver-Fay, A.; Baker, D.: Role of conformational sampling in computing mutation-induced changes in protein structure and stability. *Proteins*, 2011.
- [27] Khatun, J.; Khare, S. D.; Dokholyan, N. V.: Can contact potentials reliably predict stability of proteins? *Journal of Molecular Biology*, 2004.
- [28] Kyte, J.; Doolittle, R. F.: A simple method for displaying the hydropathic character of a protein. *Journal of Molecular Biology*, 1982.
- [29] Laimer, J.; Hofer, H.; Fritz, M.; aj.: MAESTRO - multi agent stability prediction upon point mutations. *BMC Bioinformatics*, 2015.
- [30] Masso, M.; Vaisman, I. I.: AUTO-MUTE 2.0: A Portable Framework with Enhanced Capabilities for Predicting Protein Functional Consequences upon Mutation. *Adv Bioinformatics*, 2014.

- [31] Mařík, V.; Štěpánková, O.; Lažanský, J.: *Umělá inteligence. 1.* Academia, 1993, ISBN 80-200-0496-3.
- [32] Opitz, D.; Maclin, R.: Popular Ensemble Methods: An Empirical Study. *Journal of Artificial Intelligence Research*, ročník 11, 1999: s. 993 – 1001.
- [33] Pandurangan, A. P.; Ochoa-Montaña, B.; Ascher, D. B.; aj.: SDM: a server for predicting effects of mutations on protein stability. *Nucleic Acid Research*, ročník 45, 2017.
- [34] Parthiban, V.; Gromiha, M. M.; Schomburg, D.: CUPSAT: prediction of protein stability upon point mutations. *Nucleic Acids Research*, 2006.
- [35] Pedregosa, F.; Varoquaux, G.; Gramfort, A.; aj.: Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research*, ročník 12, 2011: s. 2825–2830.
- [36] Pires, D. E.; Ascher, D. B.; Blundell, T. L.: mCSM: predicting the effects of mutations in proteins using graph-based signatures. *Bioinformatics*, 2014.
- [37] Platt, J. C.: Sequential Minimal Optimization: A Fast Algorithm for Training Support Vector Machines. Technická správa, ADVANCES IN KERNEL METHODS - SUPPORT VECTOR LEARNING, 1998.
- [38] Polikar, R.: Ensemble based systems in decision making. *IEEE Circuits Syst Mag.*, 2006.
- [39] Ponnunswamy, P.; Gromiha, M.: On the conformational stability of folded proteins. *Journal of theoretical biology*, ročník 166, č. 1, 1994: s. 63–74.
- [40] Price, M. N.; Dehal, P. S.; Arkin, A. P.: FastTree 2—approximately maximum-likelihood trees for large alignments. *PLoS One*, ročník 5, 2010.
- [41] Schapire, R. E.: The Strength of Weak Learnability. *Machine Learning*, 1990.
- [42] Sievers, F.; Higgins, D. G.: Clustal Omega, accurate alignment of very large numbers of sequences. Multiple sequence alignment methods. *Methods Mol. Biol.*, ročník 1079, 2014: s. 105–116.
- [43] Sussman, J. L.; Lin, D.; Jiang, J.; aj.: Protein Data Bank (PDB): Database of Three-Dimensional Structural Information of Biological Macromolecules. *Acta Cryst.*, 1998.
- [44] Tang, J.; Alelyani, S.; Liu, H.: Data Classification. *Chapman and Hall/CRC*, 2014.
- [45] Voet, D.; Voet, J. G.; Pratt, C. W.: *Fundamentals of Biochemistry: Life at the Molecular Level, 3rd Edition*. John Wiley and Sons, Inc., 2008, ISBN 0470129301.
- [46] Whitford, D.: *Proteins: Structure and function*. Wiley, 2005.
- [47] Witvliet, D.; Strokach, A.; Giraldo-Forero, A. F.; aj.: ELASPIC web-server: proteome-wide structure based prediction of mutation effects on protein stability and binding affinity. *Bioinformatics*, 2016.
- [48] Yin, S.; Ding, F.; Dokholyan, N. V.: Eris: an automated estimator of protein stability. *Nature methods*, 2007.

- [49] Zhang, Y.; Burer, S.; Street, W. N.: Ensemble Pruning Via Semi-definite Programming. *Journal of Machine Learning Research*, ročník 7, 2006: s. 1315–1338.
- [50] Řehout, V.; Čítek, J.; Bláhová, B.; aj.: *Základy genetiky a poradenství*. Február 2003, [Online; cit. 17.12.2017].
URL http://www.zsf.jcu.cz/cs/katedra/katedra-klinickych-a-preklinickych-oboru/import/ucebni_texty/zaklady-genetiky-a-poradenstvi

Príloha A

Obsah DVD

Priložené DVD obsahuje nasledujúce adresáre a súbory:

- spustiteľný skript nástroja `stability_predictor.py`
- skripty `mutation_record.py` a `blosum.py`
- súbor README s popisom spustenia nástroja
- adresár `/models` so serializovanými modelmi použitých metód strojového učenia
- adresár `/tools` s potrebnými programami tretích strán
- adresár `/source` s textom práce
- adresár `/data` s použitými dátovými sadami

Príloha B

Manuál

Nástroj má formu konzolovej aplikácie s niekoľkými vstupnými parametrami. Príklad spustenia je nasledovný: `./stability_predictor.py 1LNI A 79 D K`

Popis jednotlivých parametrov:

- PDB_ID PDB identifikátor proteínu
- CHAIN typ reťazca
- POSITION pozícia mutácie
- WILD_TYPE aminokyselina pred mutáciou
- MUTANT aminokyselina po mutácii
- -h|- -help spustenie nápovedy k programu