VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
FACULTY OF INFORMATION TECHNOLOGY

ÚSTAV POČÍTAČOVÉ GRAFIKY A MULTIMÉDIÍ
DEPARTMENT OF COMPUTER GRAPHICS AND MULTIMEDIA

NÁZEV PRÁCE

THESIS TITLE

BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

AUTOR PRÁCE **JMÉNO PŘÍJMENÍ**
AUTHOR

VEDOUCÍ PRÁCE **Doc. RNDr. JMÉNO PŘÍJMENÍ, Ph.D.**
SUPERVISOR

BRNO 2018

Abstrakt

Do tohoto odstavce bude zapsán výtah (abstrakt) práce v českém (slovenském) jazyce.

Abstract

Do tohoto odstavce bude zapsán výtah (abstrakt) práce v anglickém jazyce.

Kľúčové slová

Sem budou zapsána jednotlivá klíčová slova v českém (slovenském) jazyce, oddělená čárkami.

Keywords

Sem budou zapsána jednotlivá klíčová slova v anglickém jazyce, oddělená čárkami.

Citácia

PŘÍJMENÍ, Jméno. *Název práce*. Brno, 2018. Bakalářská práce. Vysoké učení technické v Brně, Fakulta informačních technologií. Vedoucí práce Doc. RNDr. Jméno Příjmení, Ph.D.

Název práce

Prehlásenie

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně pod vedením pana X... Další informace mi poskytli... Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

.....
Jméno Příjmení
13. apríla 2018

Poďakovanie

V této sekci je možno uvést poděkování vedoucímu práce a těm, kteří poskytli odbornou pomoc (externí zadavatel, konzultant, apod.).

Obsah

Kapitola 1

Úvod

Proteíny patria k najzložitejším známym molekulám. Tieto zložité molekuly sú základnými stavebnými prvkami každého živého orgamizmu, čo z nich robí významný cieľ skúmania, najmä z hľadiska ich rôznych funkcií v organizme.

V proteínoch dochádza k mutáciam aminokyselín, ktoré môžu ovplyvniť stabilitu proteínu v pozitívnom alebo negatívnom zmysle. Táto vlastnosť proteínov je veľmi dôležitá v mnohých oblastiach priemyslu, najmä toho farmaceutického. V poslednom období vzniklo množstvo rôznych nástrojov zaoberajúcich sa predikciou vplyvu mutácií na stabilitu využívajúc rôznych prístupov pri predikcií.

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

Druhá kapitola sa zaoberá proteínmi a aminokyselinami. Bližšie sú popísané základné vlastnosti ako štruktúra a rozdelenie proteínov, rozdelenie aminokyselín a základná dogma molekulárnej biológie.

Tretia kapitola je určená mutáciam v aminokyselinách a stabilite proteínu. Bližšie je uvedená charakteristika stability a jej súvis s mutáciami, typy mutácií a ich vzik.

Štvrtá kapitola podáva stručný úvod do oblasti strojového učenia. Informuje o základnom rozdelení metód strojového učenia a ďalej obsahuje podrobnejšiu charakteristiku rôznych klasifikačných algoritmov.

Piata kapitola je venovaná tzv. ensemble metódam slúžiacim na vylepšenie výsledkov pri rôznych problémoch s ktorými sa stretávame pri využití strojového učenia. Konktrétne sú opísané dôvody na použitie takýchto metód a najpoužívanejšie techniky na ich tvorbu.

V šiestej kapitole sú vymenované dostupné nástroje na predikciu stability proteínov. Jednotlivé nástroje obsahujú stručnú charakteristiku a zaradenie do príslušnej skupiny podľa spôsobu predikcie.

Siedma kapitola sa venuje tvorbe dátovej sady, problémov pri jej tvorbe a jednotlivým parametrom mutácie použitých pri predikcií. Jednotlivé parametre sú stručne charakterizované.

V ôsmej kapitole je obsiahnutá implementácia nástroja. Kapitola obsahuje bližšie informácie o postupe výberu metód strojového učenie pre nástroj, tvorbu tréningových podsád, stručnú charakteristiku použitých technológií a vyhodnotenie nástroja na testovacej sade. Prítomné sú aj výsledky porovnania implementácie s vybranými nástrojmi.

Záverečná deviata kapitola podáva zhrnutie dosiahnutých výsledkov v rámci testovania nástroja.

Kapitola 2

Proteíny

Proteíny (bielkoviny) môžeme charakterizovať ako základné stavebné prvky všetkých živých organizmov. Nie sú však iba stavebnými prvkami bunky, zabezpečujú väčšinu bunečných 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 ionty v organizme. Príkladom sú proteín hemoglobín ako nosič kyslíka v krvnom obehu 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 iontov. Kasein v mlieku poskytuje zdroj aminokyselín pre novonarodené živočíchy.
- **Signálne proteíny:** ich funkcoiu 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 [37]:

- **Aminokyseliny s alifatickým postranným reťazcom:** alanin (Ala), valin (Val), leucin (Leu), isoleucin (Ile), glycin (Gly)
- **Bazické skupiny s aminovou skupinou na postrannom reťazci:** arginin (Arg), lysin (Lys)
- **S aromatickým jadrom alebo hydroxylovou skupinou na postrannom reťazci:** histidin (His), fenylalanin (Phe), serin (Ser), threonin (Thr), tyrosin (Tyr), tryptofan (Trp)
- **Kyslé skupiny s karboxylovou alebo aminovou skupinou na postrannom reťazci:** kyselina asparagová (Asp), asparagin (Asn), kyselina glutamová (Glu), glutamin (Gln)
- **So sírou v postrannom reťazci:** methionin (Met), cystein (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.

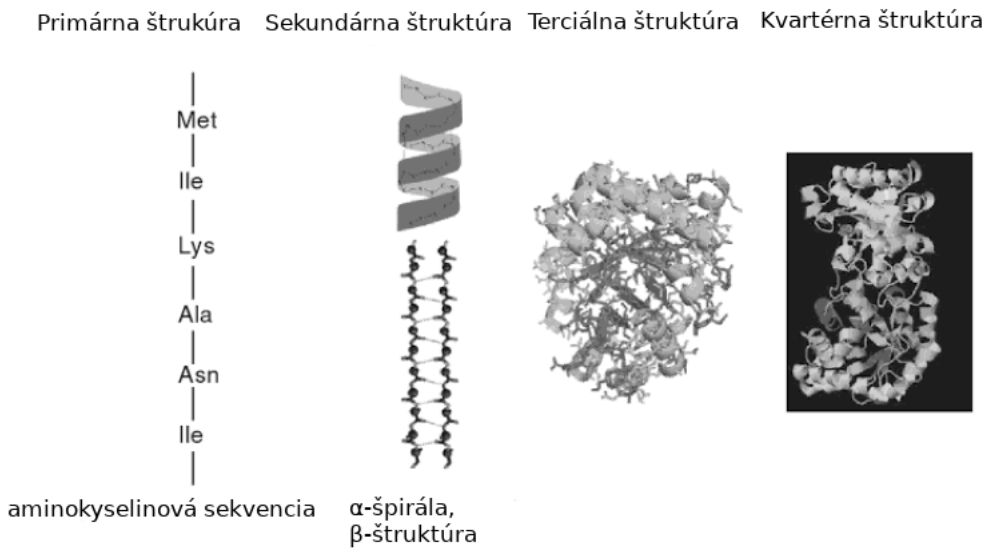
- **Transkripcia:** pri procese transkripcie dochádza k prepisu časti nukleotidovaj sekvencie DNA (génu) do molekuly RNA. Dôležitú úlohu zohráva enzým RNA-polymeráza, ktorá musí pred začiatkom traskripcie nájsť oblasť tzv. promotoru obsahujúcu informáciu o začiatku transkripcie a následne sa na túto 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 trojrozmernej štruktúry proteínov môžeme podľa [37] rozdeliť do štyroch úrovní 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 [37]. V prípade β-štruktúry prebiehajú úseky reťazca paralelne vedľa seba a sú stabilizované vodíkovými mostíkmi medzi susediacimi úsekmi.
- **Terciálna štruktúra:** reprezentuje trojrozmerné priestorové usporiadanie zloženého polypeptidového reťazca [37]. 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 polypetidového 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 [15].

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 oblastich ako je medicína, kde chceme dosiahnuť výrobu účinnejších 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 van der Waalsove sily. Naopak, nestabilný proteín sa nachádza v denaturovanom stave, kde dominuje entropická a neentropická voľná energia [15].

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 [36]:

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

kde H predstavuje entalpiu, T teplotu a S entropiu.

Zmena voľnej Gibbsovej energie (ΔG) je daná vzťahom

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

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), diferencilna skenovacia kalorimetria (DSC), fluorescencia (Fl), absorpcia svetla (Abs), nukleárna magnetická rezonancia (NMR) [15].

Určenie stability bez použitia niektorej z laboratórnych metód je možné uskutočniť výpočtom jedného z existujúcich silových polí (Talaris, Score12, ...). Výpočet takéhoto poľa ukazuje nasledujúci jednoduchý príklad [30] [15]:

Voľná energia v zloženom stave je daná vzťahom

$$G_F = G_{hy} + G_{el} + G_{hb} + G_{vw} + G_{ss}, \tag{3.3}$$

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

Elektrostatickými interakciami najviac prispievajú nabité postranné reťazce reziduí Lysine, Hystodine, 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

$$G_U = G_{en} + G_{ne}, \tag{3.4}$$

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. Mutáciami sú označované všetky také zmeny genetickej informácie, ktoré nie sú výsledkom segregácií a rekombinácií už existujúcich častí genotypov [40]. Podľa úrovne, na ktorej sa mutácia vyskytla, môžeme rozlišovať [12]:

- **Génové mutácie:** zmena v stavbe DNA, ktorá je reprezentovaná zmenou nukleotidovej sekvencie na určitom mieste [40]. Nazývajú sa tiež bodovými mutáciami a z hľadiska predikcie sú najzásadnejšie.
- **Chromozómové mutácie:** mení sa štruktúra chromozómu.
- **Genónové mutácie:** mení sa počet chromozómov.

3.2.1 Vznik mutácií

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

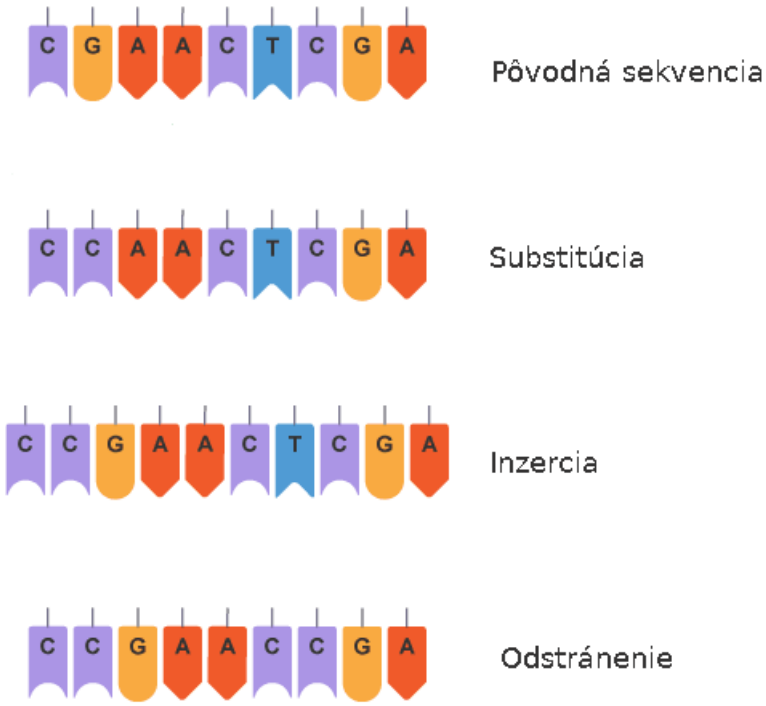
Medzi fyzikálne faktory patria rôzne zdroje žiarenia, najmä ionizujúce a ultrafialové. Poškodenie štruktúry DNA je priamo úmerné množstvu absorbovaného žiarenia.

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 [40].

3.2.2 Typy mutácií

Podľa [12] rozlišujeme tri základné typy génových mutácií:

- **Substitúcia:** jedná sa o zámenu jedného alebo viacerých párov po sebe nasledujúcich bází inými [40]. V tomto prípade sa nemení dĺžka pôvodného proteínu. Novovziknutý proteín sa oproti pôvodnému obvykle líši v jednej aminokyseline.
- **Vloženie:** jedná sa o vloženie jedného alebo viacerých nových párov bází do pôvodnej sekvencie, spôsobuje zväčšenie dĺžky sekvencie.
- **Odstránenie:** odstránenie jedného alebo viacerých po sebe nasledujúcich párov bází, mení dĺžku sekvencie rovnako ako vloženie.



Obr. 3.1: Jednotlivé typy mutácií[1].

V prípade, že k mutácií dôjde v kódujúcej oblasti, môžeme mutácie rozlíšiť na [12]:

- **Synonymné:** vychádzajú z tzv. degenerovanosti genetického kódu. Zámena nukleotidu v kodóne sa tak na štruktúre proteínu nemusí vôbec prejaviť 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 predčasnému ukončeniu prekladu proteínu.
- **Nezmyselné:** vytvárajú STOP kodón a tým spôsobujú predčasné ukončenie prekladu proteínu.

[1] Zdroj: https://www.bbc.co.uk/education/guides/zc499j6/revision/2

Na stabilitu proteínu vplývajú aminokyselinové mútacie, 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:

$$\Delta\Delta G = G_{mutant} - G_{wild\_type}, \tag{3.5}$$

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 [25]. Pojem strojové učenie takisto označuje počítačové metódy pracujúce s obrovským množstvom dát, medzi ktorými je snahou nájsť vzťahy. 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 vzťahov vo veľkom množstve dát, čo by bolo pri mechanickom hľadaní takmer nemožné. Medzi hlavné nevýhody metód patrí neschopnosť správnej analýzy dát pri nevyváženosti predložených dát, nesprávne výsledky pri malom množstve tréningových dát pre metódu alebo nemožnosť práce s dátami obsahujúcimi veľké množstvo parametrov.

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 narozdiel od klasifikačných nepriraďujú 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 učenie môžeme ďalej rozdeliť na základe spôsobu, akým sa učia. Podľa [2] ich rozdeľujeme na:

- **Učenie s učiteľom:** Pri tomto type učenia je nutné mať k dispozícií vstupné aj výstupné dáta. Cieľom je nájsť vzťahy medzi vstupom a výstupom, ktoré slúžia na naučenie metódy. Medzi algoritmy patriace do tejto skupiny radíme regresiu aj klasifikáciu.
- **Učenie bez učiteľa:** V tomto type učenie nie sú k dispozícií 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. Algoritmus produkuje klasifikačno-rozhodovací strom pre poskytnuté dáta rekurzívnym rozdeľovaním dát. Pri rozhodovaní je využitá stratégia depth-first. Algortimus berie do úvahy všetky možné testy, ktoré môžu rozdeliť dáta a vyberá test udávajúci najlepšiu informačnú hodnotu. Pre každý diskrétny 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 rovankú šancu výberu. Kombinácia veľkého počtu náhodných stromov obvykle vedie k správnemu modelu.

4.2.3 Algoritmus Náhodný les (Random Forest)

Random Forest [8] je metóda založená na kombinácií 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 $\{h(x, \Theta_k), k = 1, ...\}$, kde $\{k\}$ sú nezávisle identicky rozdelené náhodné vektory a každý strom hlasuje jednotlivo o najpopulárnejšej triede vo vstupe x.

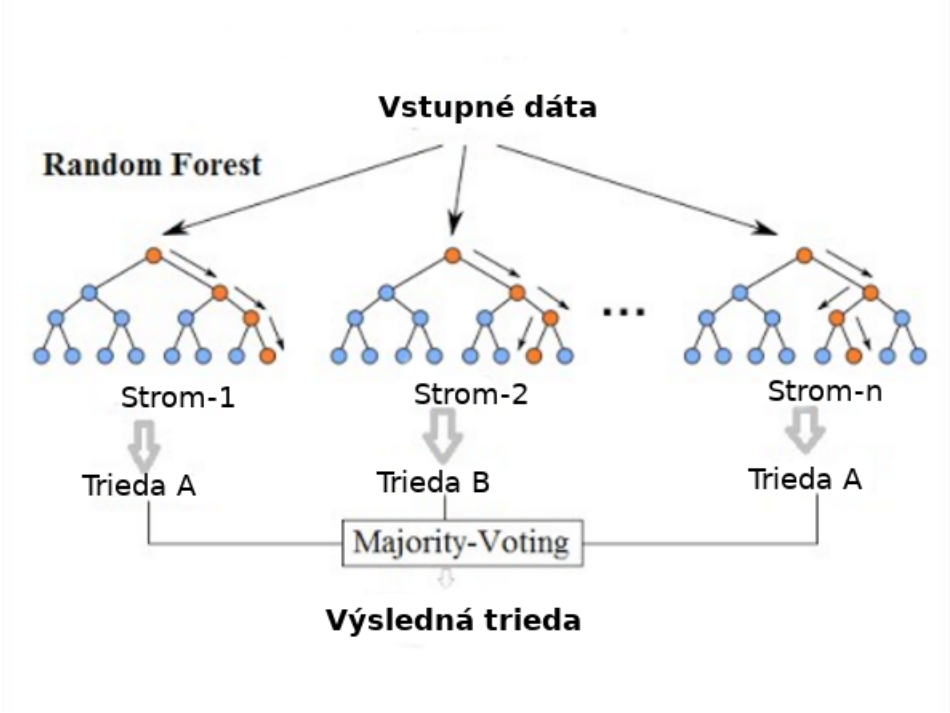
Najväčšiu pozornosť tejto metódy tvoria 3 vlastnosti:

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

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

- Pre každý z M rozhodovacích stromov vytvoríme sadu tréningových dát z originálnych dát. Na ich výber slúži metóda tzv. bagging, ktorá náhodne vyberie zadaný počet trénovacích dát.
- Pre každú množinu tréningových dát vytvoríme klasifikačný alebo regresný strom, ktorý je následne natrénovaný na M-tej náhodnej množine tréningových 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 klasifikácií alebo priemerom hodnôt pri regresí.

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



Obr. 4.1: Metóda Random Forest[1].

4.3 Support vector machines (SVM)

Support vector machines patria k najnovším metódam strojového učenia. 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 označujeme ako tzv. podporné vektory (support vectors).

Ak sú trénovacie dáta lineárne rozdeliteľné, potom pár $(\mathbf{w}, b)$ existuje ako
$\mathbf{w}^T\mathbf{x}_i + b \geq 1$, pre všetky $\mathbf{x}_i \in P$
$\mathbf{w}^T\mathbf{x}_i + b \leq -1$, pre všetky $\mathbf{x}_i \in N$
s rozhodovacím pravidlom daným vzťahom $f_{\mathbf{w},b}(x) = sgn(\mathbf{w}^T x + b)$, 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 dimenzí 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 $\Phi : R^d \to H$. Trénovací algoritmus bude potom závisieť len na dátach skrz 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.

[1] Prevzaté a upravené z http://bit.ly/2rpjSDK

Výber vhodnej aproximačnej jadrovej funkcie je dôležitý, pretože funkcia definuje transformovaný priestor vlastností v ktorom budú klasifikované tréningové dáta. Medzi najpoužívanejšie jadrové funkcie patria

- $K(x, y) = (x \cdot y + 1)^P$
- $K(x, y) = e^{-\|x-y\|^2/2\sigma^2}$
- $K(x, y) = tanh(\kappa x \cdot y - \delta)^P$

4.3.2 Algoritmus SMO

Sekvenčná minimalizačná optimalizácia (SMO) 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 vyberia 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 tréningovými 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 prediktoru 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 narážame na rôzne problémy spojené najmä s nedostatkom dát, veľkou rôznorodosťou dát alebo tzv. preučením 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 označujeme ako tzv. ensemble stratégiu.

Podľa [29] existuje mnoho teoretických a praktických dôvodov na použitie ensemble systémov:

- **Štatistické dôvody:** Dobrý výkon metódy na trénovacích dátach nemusí predpovedať dobrú výkonnosť zovšeobecňovania. Množina klasifikátorov s podobným výkonom na trénovacích 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 klasifikátoru.

- **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 mmnožstva 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 hra-

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

5.1 Rôznorodosť v ensemble systémoch

V posedných rokoch niekoľko štúdií teoreticky [?] aj prakticky [?] ukázalo, že použitie ensemble systému viacerých klasifikátorov prekonáva výsledky samostatného klasifikátora keď členovia systému sú dostatočne presní a robia niekoľko zhodných chýb. Natrénovanie klasifikátora, ktorý by dokonale zvládol rozdeliť stavový priestor bez výskytu chyby, potrebujeme 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 []. 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 rozmanitosť jednotlivcov:

- **Rôzne tréningové dátové sady:** Tréningová dátová sada môže byť rozdelená na menšie celky a každá takáto menšia podsada je použitá na natrénovanie jedného samostatného klasifikátora. Na rozdelenie dát môže byť použitá technika z niekoľkých dostupných prevzorkovávacích techník. Jednou z najbežnejších takýchto techník je tzv. bootstrap technika. Exituje však niekoľko ďalších, napríklad metódy ako tzv. jack-knife.

- **Rôzne tréningové parametre:** Použitie rôznych tréningových parametrov pre jednotlivé klasikikátory je tiež jednou z možných ciest. Napríklad viacvrstvová neurónová sieť môže byť trénovaná 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 predikcií:** Lineárny klasifikátor môže mať niekedy problém s nachádzaním vzorov u parametrov, ktoré sú veľmi vzdialené. Pre riešenie tohto problému môže byť prospešné experimentovanie s rozličnými podsadami predikovaných parametrov generovaných buď metódami učenia s učiteľom alebo bez neho. []

- **Rôzne typy klasikikátorov:** Neexistuje žiadny nadradený klasifikátor, ktorý by poskytoval najlepšie výsledky bez ohľadu na oblasť odborníka alebo množstva tréningových dát. Na druheje strane, na každý problém zvyčajne existuje niekoľko vhodných klasifikátorov, preto výber niekoľkých z nich a kombinácia ich výstupov je možnou cestou ako vytvoriť množinu klasifikátorv 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

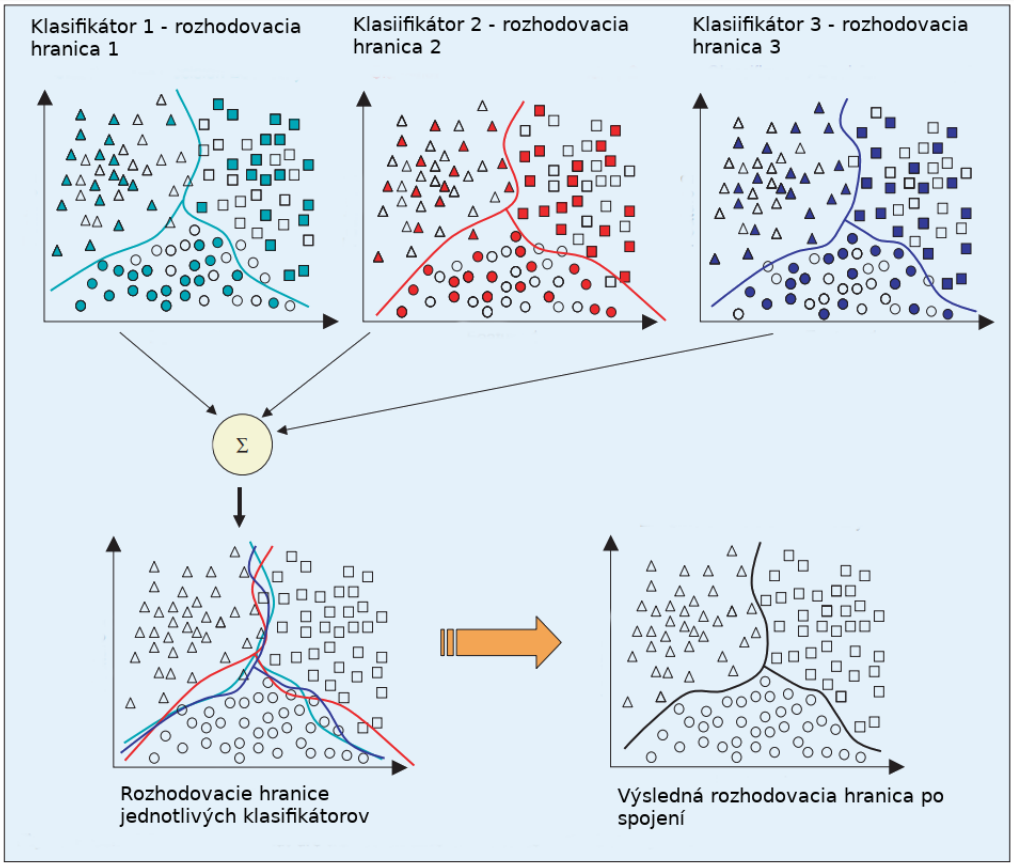
Bagging [7] alebo aj tzv. bootstrap aggregation je jednou z najstarších, ľahko implementovateľných ensemble stratégií. Bola navrhnutá na zlepšenie presnosti algoritmov strojového učenia používaných pri štatistickej klasifikácií a regresií. Rôznorodosť je dosiahnutá využitím tzv. bootstrap podmnožín tréningových dát, pričom rôzne tréningové množiny sú vybrané náhodne s náhradou z celého množstva tréningových dát. Každá takáto podmnožina je určená na trénovanie iného klasifikátora rovnakého typu. Nakoniec sa využije hlasovania 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 zaisteneie 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 zaistí 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 nestabilnosť je kontrolovateľná výberom ich parametrov.

5.2.2 Boosting

Boosting [33] 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 trénovacie dáta. Boosting vytvára tri slabé klasifikátory:

- Prvý klasifikátor C_1 je trénovaný 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 trénovaný na dátach, na ktorých mal C_1 polovičnú úspešnosť.
- Tretí klasifikátor C_3 je trénovaný na dátach, kde C_1 a C_2 nesúhlasili.
- Nakoniec sú klasifikátory skombinované hlasovaním väčšiny.



Obr. 5.1: Ensemble systém. Prevzaté a upravené z [29].

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 presnoti 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 trénovanie 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 predikcií; jednoduché hlasovanie v ktorom sa aspoň polovica klasifikátorov musí zhodnúť na rozhodnutí; väčšinové hlasovanie kde je výsledok daný podľa počtu hlasov.
- *Váhované 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:* Váhovaná metóda, váha spojená s klasifikátorom je tzv. posterior pravdepodobnosť klasifikátora na tréningovej sade.
- *Váhovanie entropie:* Rovnako sa jedná o váhovanú metódu a váhy klasifikátorov sú nepriamo úmerné entropií 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 atómoých 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 sekcií sa nachádza charakteristika nástrojov využívajúch 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 pozícií a určia sa ich vlastnosti. Vektor vlastností, tvorený vlastnosťami založenými na energií, je určený na natrénovanie vybranej metódy zvolenej užívateľom. Nástroj využíva algoritmy z balíka programu WEKA, konktrétne 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 [24].

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). Tréningové

dáta pozostávajú z podmnožiny dát pochádzajúcich z databázy ProTherm. Vstupný vektor vlastností mutácie tvorí spolu 42 hodnôt. Nástroj je dostupný ako webový server [9].

6.1.3 iPTREE-STAB

iPTREE-STAB slúži na určenie vplyvu mutácie na stabilitu a hodnoty $\Delta\Delta G$ z informácií o sekvencií. 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 mutácií, pôvodná aminokyselina, hodnota pH, teplota pri ktorej bola meraná stabilita a informácie o troch susedných reziduách. Nástroj je dostupný iba ako webový server [18].

6.1.4 EASE-AA

Nástroj ponúka predikciu vplyvu mutácie a takisto určenie hodnoty $\Delta\Delta G$. Predikcia je založená iba na informáciách o sekvencií. EASE-AA využíva informácie o konzervovanosti, ktoré sú získané pomocou metódy SIFT. Konzerovanosť 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 [13].

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 trénovanie a testovanie metód strojového učenia. Nástroj je dostupný ako webová služba [28].

6.1.6 MAESTRO

MAESTRO je multiagentný systém strojového učenia. K predikcií využíva informácie o sekvencií 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 prediktoru je možnosť predikcie stabilizujúcich disulfidových mostíkov. Nástroj je dostupný ako samostatná aplikácia a aj ako webový server [23].

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 Stochaistic Gradient boosting. ELASPIC je dostupný ako samostatná aplikácia aj ako webový server [38].

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 samostantý program [39].

6.2.3 Rosetta

Rosetta predstavuje sadu softvéru na makromolekulárne modelovanie. Zahŕňa celú skupinu rôznych silových polí založených na kombinácií prispievateľov voľnej enrgie ako sú van der Waalsova energia alebo elektrostatické interakcie. Väčšina z protokolov tohto nástroja je veľmi časovo náročných a preto je nástroj dostupný iba ako samostatná aplikácia [20].

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 [27].

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 [11].

6.3.2 DMutant

Nástroj DMutant využíva kombináciu orientácie na základe poznatkov, vdialenostne 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 [17].

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 [16].

Kapitola 7

Dátová sada a jej parametre

Vytvorenie spoľahlivej, dostatočne veľkej a rozmanitej tréningovej 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 Gibssova 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 [35] alebo funkcionálnou PMD [19].

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 tréningových dátových sád pre existujúce nástroje predikujúce vplyv aminokyselinových substitúcií na stabilitu proteínu je najpoužívanejším zdrojom dát práve databáza ProTherm. V súčasnom stave však trpí množstvom serióznych nedostatkov. Aby sme sa vyhli problémom tejto databázy, vyextrahovali sme iba mutácie u ktorých sú uvedené zmeny stability a overili 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 Gibssovej voľnej energie ($\Delta\Delta G$), záznamy s hodnotami $\Delta\Delta G$ z intervalu $[-0.5, 0.5]$ boli odstránené. Záznamy s hodnotou $\Delta\Delta G \geq 0.5$ $kcal.mol^{-1}$ boli označené za destabilizujúce a záznamy s $\Delta\Delta G \leq$ -0.5 $kcal.mol^{-1}$ boli označené za stabilizujúce. Tento rozhodovací prah bol zvolený podľa tvrdenia, že experimentálna chyba merania $\Delta\Delta G$ je približne 0.48 $kcal.mol^{-1}$ [21]. 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 rozšírený o štrukturálne informácie.

7.1 Parametre mutačného záznamu

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 mutácií 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 mutácií. 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 [22].

- **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 sme zisťovali, 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 sme každému proteínu priradili 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 sme opäť použili modul DSSP prítomný v knižnici BioPython.

- **Konzervovanosť:** Jeden z najdôležitejších parametrov v zázname mutácie. Informácia o konzervovanosti pozície na ktorej došlo k mutácií 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 pozícií sú z hľadiska skúmania veľmi zaujímavé, pretože konzerované 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, počítame vzájomnú informáciu, ktorá vyjadruje závislosť dvojice náhodných veličín. Údaj o vzájomnej informácií dostaneme pomocou nasledujúceho vzťahu:

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

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 sekvencií, 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 sekvencií. Vstupom je viacnásobné zarovanie 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

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

- vnútorný uzol

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

- koreňový uzol

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

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. V našom prípade obsahuje pole hodnoty s váhami pre jednotlivé sekvencie vo viacnásobnom zarovnaní. Počas priechodu sa počítajú hodnoty v poli každého uzlu a výsledkzom je pole obsiahnuté v koreni stromu. Pole s váhami sa použije na zistenie konzervovanosti konkrétnej pozície. Počítaním váh sledujeme zmeny na pozícií v priebehu evolúcie proteínu a na základe koeficientov vypočítame 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 [14] 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 vyskušať a porovnať na zvolenej dátovej sade; a nástroje na predspracovanie analyzovaných dát. WEKA poskytuje algoritmy na regresiu, zhlukovanie, 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. 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 boli dáta rozdelené v pomere 66% tréningové dáta a zvyšné záznamy slúžili ako testovacie dáta. Nastavená bola aj tzv. cost–sensitive matica, ktorá slúžila na postihovanie predikcie stabilizujúcich mutácií ako destabilizjúcich, pretože v tréningovej 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 predikcií 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 mutačného záznamu v Pythone

Na implementáciu predikčného nástroja sme zvolili programovací jazyk Python, najmä kvôli dostupnosti mnohých bioinformatických nástrojov a knižníc, ale aj pre jednoduché implementovanie strojového učenia vďaka dostupným knižniciam.

Prvou fázou pri tvorbe nástroja bola implementácia prípravy mutačného záznamu, 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 hydrofobity 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[1], konktrétne modul DSSP. Pomocou neho je možné získať obidva údaje z PDB súboru. Pre správne fungovanie modulu je však nutné mať prítomný samostatný program **dssp**, 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 získane homológnych sekvencií. Výstup v podobe XML súboru je spracovaný do podoby textového súboru slúžiaceho ako vstup programu **Clustal Omega** [34], 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 Felsteinového algoritmu. Na vytvorenie stromu je použitý program **FastTree** [31]. Na im-

[1]Dostupné na www.biopython.org

plementáciu fylogenetického stromu v Pythone bola použitá knižnica **ete3**[2], 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 zarovananie sekvencií. Jednotlivé hodnoty parametov, 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 sme zvolili knižnicu **Scikit-learn**[3]. Scikit-learn je knižnica strojového učenia pre programovací jazyk Python a zahŕňa v sebe rozličné klasifikátory, zhlukovacie 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 sme otestovali 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. Tréningová 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	0.35
SVM	0.81	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 0.35. Pri testovaní metódy SVM bol použitý polynóm ako 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.

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 sme použili techniky bagging s prekryvom dát v podsadách. Na zostavenie podsád

[2]Dostupné na http://etetoolkit.org/
[3]Dostupné na http://scikit-learn.org/stable/

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. Podsady boli testované s prítomnosťou všetkých parametrov, v ďalšom testovaní sme pristúpili k odstráneniu niektorých parametrov, ktoré by nemuseli mať taký výrazný vplyv na predikciu. Testovali sme podsady 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.

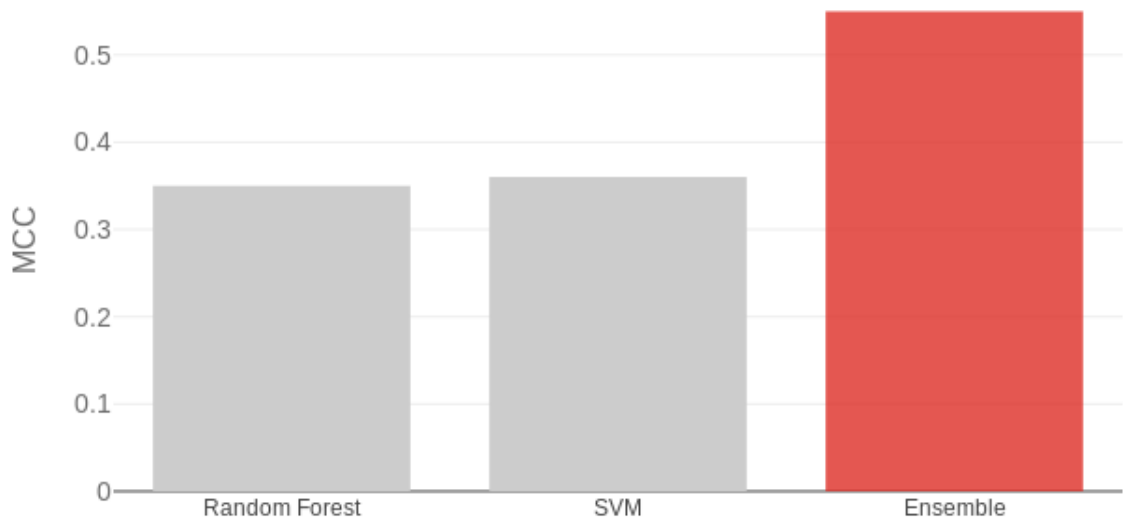
Dataset	MCC (SVM)			MCC (RF)		
1.	0.3	0.27	0.32	0.17	0.2	0.2
2.	0.31	0.33	0.4	0.19	0.25	0.21
3.	0.28	0.4	0.38	0.18	0.2	0.13
4.	0.36	0.36	0.39	0.21	0.29	0.22
5.	0.41	0.42	0.41	0.23	0.28	0.24
6.	0.44	0.41	0.4	0.29	0.3	0.25
7.	0.4	0.29	0.39	0.28	0.26	0.3
8.	0.47	0.36	0.42	0.29	0.28	0.29
9.	0.45	0.36	0.41	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 údaju o korelačnom koeficiente. Na základe získaných údajov boli vybrané podsady s najlepšími 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 sme získali údaj o výslednej triede, nakoniec sme mali sedem hodnôt predikovanej triedy pre mutáciu. Výslednú hodnotu výslednej triedy sme získali hlasovaním väčšiny, výsledkom bola trieda, ktorá sa medzi hodnotami nachádzala častejšie.

Ensemle systém bol vo finálnej implementácií 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.



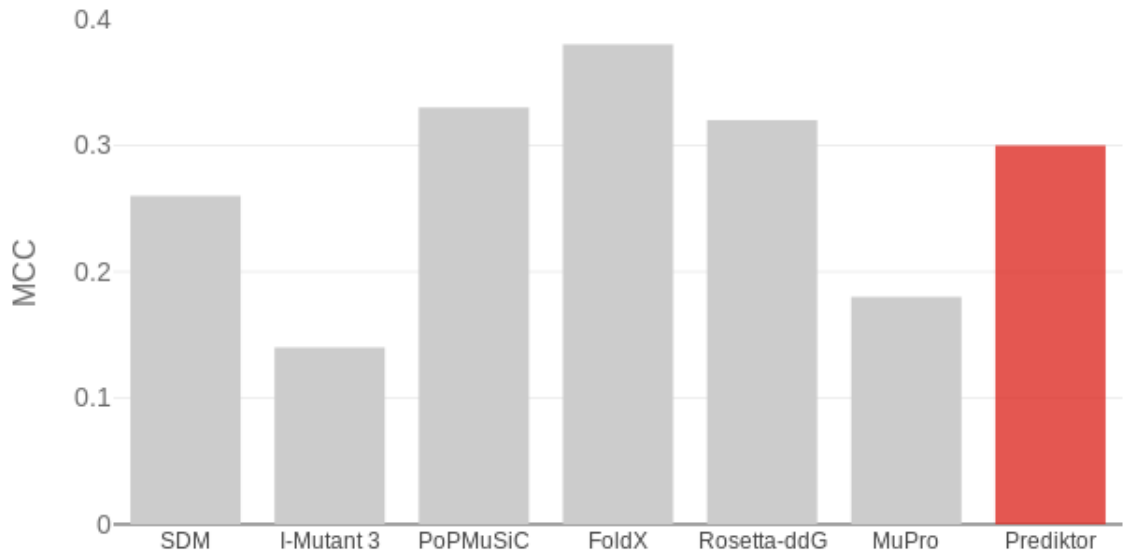
Obr. 8.1: Graf s porovnaním hodnôt MCC na testovacej sade

8.5 Porovnanie s inými nástrojmi

Po implementovaní prediktora sme jeho predikčnú silu porovnali s niekoľkými dostupnými nástrojmi. Tieto nástroje boli zvolené podľa toho, či boli otestované na testovacej sade

S350, ktoré mnohé nástroje používajú ako svoju testovaciu sadu. Táto sada obsahuje 350 záznamov mutácií, z ktorých je 95 stabilizujúcich a 255. Porovnávanie spočívalo v otestovaní implementovaného prediktora na sade S350 a porovnaní výsledkov s nástrojmi, ktoré boli otestované na tejto sade. Daný postup sme zvolili najmä preto, že natrénovanie a otestovanie iných nástrojov na nami vytvorenej dátovej sade by nebolo úplne možné, pretože mnoho nástrojov je dostupných len v podobe webovej aplikácie.

Zo sady S350 sme neodstránili záznamy s hodnotou $\Delta\Delta G$ v intervale $[-0.5, 0.5]$, ktoré sa v našej dátovej sade nevyskytujú. Záznamy s týmito hodnotami sme ponechali nakoľko by porovanie 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 taká, že mutácie s $\Delta\Delta G >= 0$ boli označené ako stabilizujúce a zvyšné mutácie ako destabilizujúce. Hranice sú naopak oproti našej 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 spomedzi uvedených patrí Rosetta.

Zvyšné zvolené prediktory dosiahli menšiu hodnotu korelácie, najlepšie výsledky z nich dosiahol nástroj SDM [26] založený na získavaní znalostí. Zostávajúcimi nástrojmi sú I-Mutant3 a MuPro [10] . Obidva uvedené nástroje zhodne využívajú strojové učenie a konktrétne 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 sme dosiali lepšej korelácie ako nástroj I-Mutant3 aj ako MuPro. Korelácia nástroja I-Mutant3 dosiahla hodnotu 0.14 a hodnota u MuPro bola 0.18. Implementovaný prediktor dosiahol na dase S350 korelácie 0.3, čo je v porovaní s týmito hodnotami zlepšenie o 0.16, resp. 0.12.

Náš prediktor dosiahol hodnoty korelácie 0.3, čo spomedzi testovaných nástrojov predstavuje dobrú hodnotu, podobnú testovaným. Najlepšie výsledky pri predikcií 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 trénovanie 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 slúžiacich na rozhodovanie.

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 natrénovanie 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 dobre 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 vylepšenie určenia konzervovanosti pomocou zostavenie fylogenetického stromu a použitia Felsteinovho algoritmu.

Kapitola 10

Typografické a jazykové zásady

Při tisku odborného textu typu *technická zpráva* (anglicvky *technical report*), ke kterému patří například i text kvalifikačních prací, se často volí formát A4 a často se tiskne pouze po jedné straně papíru. V takovém případě volte levý okraj všech stránek o něco větší než pravý – v tomto místě budou papíry svázány a technologie vazby si tento požadavek vynucuje. Při vazbě s pevným hřbetem by se levý okraj měl dělat o něco širší pro tlusté svazky, protože se stránky budou hůře rozevírat a levý okraj se tak bude oku méně odhalovat.

Horní a spodní okraj volte stejně veliký, případně potištěnou část posuňte mírně nahoru (horní okraj menší než dolní). Počítejte s tím, že při vazbě budou okraje mírně oříznuty.

Pro sazbu na stránku formátu A4 je vhodné používat pro základní text písmo stupně (velikosti) 11 bodů. Volte šířku sazby 15 až 16 centimetrů a výšku 22 až 23 centimetrů (včetně případných hlaviček a patiček). Proklad mezi řádky se volí 120 procent stupně použitého základního písma, což je optimální hodnota pro rychlost čtení souvislého textu. V případě použití systému LaTeX ponecháme implicitní nastavení. Při psaní kvalifikační práce se řiďte příslušnými závaznými požadavky.

Stupeň písma u nadpisů různé úrovně volíme podle standardních typografických pravidel. Pro všechny uvedené druhy nadpisů se obvykle používá polotučné nebo tučné písmo (jednotně buď všude polotučné nebo všude tučné). Proklad se volí tak, aby se následující text běžných odstavců sázel pokud možno na *pevný rejstřík*, to znamená jakoby na linky s předem definovanou a pevnou roztečí.

Uspořádání jednotlivých částí textu musí být přehledné a logické. Je třeba odlišit názvy kapitol a podkapitol – píšeme je malými písmeny kromě velkých začátečních písmen. U jednotlivých odstavců textu odsazujeme první řádek odstavce asi o jeden až dva čtverčíky (vždy o stejnou, předem zvolenou hodnotu), tedy přibližně o dvě šířky velkého písmene M základního textu. Poslední řádek předchozího odstavce a první řádek následujícího odstavce se v takovém případě neoddělují svislou mezerou. Proklad mezi těmito řádky je stejný jako proklad mezi řádky uvnitř odstavce

Při vkládání obrázků volte jejich rozměry tak, aby nepřesáhly oblast, do které se tiskne text (tj. okraje textu ze všech stran). Pro velké obrázky vyčleňte samostatnou stránku. Obrázky nebo tabulky o rozměrech větších než A4 umístěte do písemné zprávy formou skládanky všité do přílohy nebo vložené do záložek na zadní desce.

Obrázky i tabulky musí být pořadově očíslovány. Číslování se volí buď průběžné v rámci celého textu, nebo – což bývá praktičtější – průběžné v rámci kapitoly. V druhém případě se číslo tabulky nebo obrázku skládá z čísla kapitoly a čísla obrázku/tabulky v rámci kapitoly – čísla jsou oddělena tečkou. Čísla podkapitol nemají na číslování obrázků a tabulek žádný vliv.

Tabulky a obrázky používají své vlastní, nezávislé číselné řady. Z toho vyplývá, že v odkazech uvnitř textu musíme kromě čísla udat i informaci o tom, zda se jedná o obrázek či tabulku (například „... *viz tabulka 2.7* ...“). Dodržování této zásady je ostatně velmi přirozené.

Pro odkazy na stránky, na čísla kapitol a podkapitol, na čísla obrázků a tabulek a v dalších podobných příkladech využíváme speciálních prostředků DTP programu, které zajistí vygenerování správného čísla i v případě, že se text posune díky změnám samotného textu nebo díky úpravě parametrů sazby. Příkladem takového prostředku v systému LaTeX je odkaz na číslo odpovídající umístění značky v textu, například návěští (`\ref{navesti}` – podle umístění návěští se bude jednat o číslo kapitoly, podkapitoly, obrázku, tabulky nebo podobného číslovaného prvku), na stránku, která obsahuje danou značku (`\pageref{navesti}`), nebo na literární odkaz (`\cite{identifikator}`).

Rovnice, na které se budeme v textu odvolávat, opatříme pořadovými čísly při pravém okraji příslušného řádku. Tato pořadová čísla se píší v kulatých závorkách. Číslování rovnic může být průběžné v textu nebo v jednotlivých kapitolách.

Jste-li na pochybách při sazbě matematického textu, snažte se dodržet způsob sazby definovaný systémem LaTeX. Obsahuje-li vaše práce velké množství matematických formulí, doporučujeme dát přednost použití systému LaTeX.

Mezeru neděláme tam, kde se spojují číslice s písmeny v jedno slovo nebo v jeden znak – například *25krát*.

Členicí (interpunkční) znaménka tečka, čárka, středník, dvojtečka, otazník a vykřičník, jakož i uzavírací závorky a uvozovky se přimykají k předcházejícímu slovu bez mezery. Mezera se dělá až za nimi. To se ovšem netýká desetinné čárky (nebo desetinné tečky). Otevírací závorka a přední uvozovky se přimykají k následujícímu slovu a mezera se vynechává před nimi – (takto) a „takto“.

Pro spojovací a rozdělovací čárku a pomlčku nepoužíváme stejný znak. Pro pomlčku je vyhrazen jiný znak (delší). V systému TeX (LaTeX) se spojovací čárka zapisuje jako jeden znak „pomlčka“ (například „Brno-město“), pro sázení textu ve smyslu intervalu nebo dvojic, soupeřů a podobně se ve zdrojovém textu používá dvojice znaků „pomlčka“ (například „zápas Sparta – Slavie“; „cena 23–25 korun“), pro výrazné oddělení části věty, pro výrazné oddělení vložené věty, pro vyjádření nevyslovené myšlenky a v dalších situacích (viz Pravidla českého pravopisu) se používá nejdelší typ pomlčky, která se ve zdrojovém textu zapisuje jako trojice znaků „pomlčka“ (například „Další pojem — jakkoliv se může zdát nevýznamný — bude neformálně definován v následujícím odstavci.“). Při sazbě matematického mínus se při sazbě používá rovněž odlišný znak. V systému TeX je ve zdrojovém textu zapsán jako normální mínus (tj. znak „pomlčka“). Sazba v matematickém prostředí, kdy se vzoreček uzavírá mezi dolary, zajistí vygenerování správného výstupu.

Lomítko se píše bez mezer. Například školní rok 2008/2009.

Pravidla pro psaní zkratek jsou uvedena v Pravidlech českého pravopisu [**?**]. I z jiných důvodů je vhodné, abyste tuto knihu měli po ruce.

10.1 Co to je normovaná stránka?

Pojem *normovaná stránka* se vztahuje k posuzování objemu práce, nikoliv k počtu vytištěných listů. Z historického hlediska jde o počet stránek rukopisu, který se psal psacím strojem na speciální předtištěné formuláře při dodržení průměrné délky řádku 60 znaků a při 30 řádcích na stránku rukopisu. Vzhledem k zápisu korekturních značek se používalo řádkování 2 (ob jeden řádek). Tyto údaje (počet znaků na řádek, počet řádků a proklad

mezi nimi) se nijak nevztahují ke konečnému vytištěnému výsledku. Používají se pouze pro posouzení rozsahu. Jednou normovanou stránkou se tedy rozumí 60*30 = 1800 znaků. Obrázky zařazené do textu se započítávají do rozsahu písemné práce odhadem jako množství textu, které by ve výsledném dokumentu potisklo stejně velkou plochu.

Orientační rozsah práce v normostranách lze v programu Microsoft Word zjistit pomocí funkce *Počet slov* v menu *Nástroje*, když hodnotu *Znaky (včetně mezer)* vydělíte konstantou 1800. Do rozsahu práce se započítává pouze text uvedený v jádru práce. Části jako abstrakt, klíčová slova, prohlášení, obsah, literatura nebo přílohy se do rozsahu práce nepočítají. Je proto nutné nejdříve označit jádro práce a teprve pak si nechat spočítat počet znaků. Přibližný rozsah obrázků odhadnete ručně. Podobně lze postupovat i při použití OpenOffice. Při použití systému LaTeX pro sazbu je situace trochu složitější. Pro hrubý odhad počtu normostran lze využít součet velikostí zdrojových souborů práce podělený konstantou cca 2000 (normálně bychom dělili konstantou 1800, jenže ve zdrojových souborech jsou i vyznačovací příkazy, které se do rozsahu nepočítají). Pro přesnější odhad lze pak vyextrahovat holý text z PDF (např. metodou cut-and-paste nebo *Save as Text...*) a jeho velikost podělit konstantou 1800.

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] Breiman, L.: Bagging Predictors. *Machine Learning*, 1996.

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

[9] Capriotti, E.; Fariselli, P.; Rossi, I.; aj.: A Three-State Prediction of Single Point Mutations on Protein Stability Changes. *BMC Bioinformatics*, 2008.

[10] Cheng, J.; Randall, A.; Baldi, P.: Prediction of Protein Stability Changes for Single-Site Mutations Using Support Vector Machines. *Proteins*, ročník 62, 2006.

[11] 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.

[12] Flegr, J.: *Úvod do evoluční biologie*. Academia, 2007, ISBN 978-80-200-1539-6.

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

[14] Frank, E.; Hall, M.; Trigg, L.; aj.: Data Mining in Bioinformatics using Weka. *Bioinformatics*, 2004.

[15] Gromiha, M. M.: *Protein Bioinformatics: From sequence to function*. Elsevier, 2010, ISBN 978-81-312-2297-3.

[16] 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.

[17] Hoppe, C.; Schomburg, D.: Prediction of protein thermostability with a direction- and distance-dependent knowledge-based potential. *Protein Society*, 2005.

[18] 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.

[19] Kawabata, T.; Ota, M.; Nishikawa, K.: The protein mutant database. *Nucleic Acids Research*, 1999.

[20] Kellogg, E.; Leaver-Fay, A.; Baker, D.: Role of conformational sampling in computing mutationinduced changes in protein structure and stability. *Proteins*, 2011.

[21] Khatum, J.; Khare, S. D.; Dokholyan, N. V.: Can contact potentials reliably predict stability of proteins? *Journal of Molecular Biology*, 2004.

[22] Kyte, J.; Doolittle, R. F.: A simple method for displaying the hydropathic character of a protein. *Journal of Molecular Biology*, 1982.

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

[24] 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.

[25] Mařík, V.; Štěpánková, O.; Lažanský, J.: *Umělá inteligence. 1.* Academia, 1993, ISBN 80-200-0496-3.

[26] Pandurangan, A. P.; Ochoa-Montaño, B.; Ascher, D. B.; aj.: SDM: a server for predicting effects of mutations on protein stability. *Nucleic Acid Research*, ročník 45, 2017.

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

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

[29] Polikar, R.: Ensemble based systems in decision making. *IEEE Circuits Syst Mag.*, 2006.

[30] Ponnunswamy, P.; Gromiha, M.: On the conformational stability of folded proteins. *Journal of theoretical biology*, ročník 166, č. 1, 1994: s. 63–74.

[31] Price, M. N.; Dehal, P. S.; Arkin, A. P.: FastTree 2–approximately maximum-likelihood trees for large alignments. *PLoS One*, ročník 5, 2010.

[32] Rábová, Z.; Hanáček, P.; Peringer, P.; aj.: *Užitečné rady pro psaní odborného textu.* FIT VUT v Brně, Listopad 2008, [Online; navštíveno 12.05.2015].
URL http://www.fit.vutbr.cz/info/statnice/psani_textu.html

[33] Schapire, R. E.: The Strength of Weak Learnability. *Machine Learning*, 1990.

[34] 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.

[35] Sussman, J. L.; Lin, D.; Jiang, J.; aj.: Protein Data Bank (PDB): Database of Three-Dimensional Structural Information of Biological Macromolecules. *Acta Cryst.*, 1998.

[36] 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.

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

[38] 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.

[39] Yin, S.; Ding, F.; Dokholyan, N. V.: Eris: an automated estimator of protein stability. *Nature methods*, 2007.

[40] Řehout, V.; Čítek, J.; Bláhová, B.; aj.: *Základy genetiky a poradenství.* Únor 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

Jak pracovat s touto šablonou

V této kapitole je uveden popis jednotlivých částí šablony, po kterém následuje stručný návod, jak s touto šablonou pracovat.

Jedná se o přechodnou verzi šablony. Nová verze bude zveřejněna do konce roku 2017 a bude navíc obsahovat nové pokyny ke správnému využití šablony, závazné pokyny k vypracování bakalářských a diplomových prací (rekapitulace pokynů, které jsou dostupné na webu) a nezávazná doporučení od vybraných vedoucích, která již teď najdete na webu (viz odkazy v souboru s literaturou). Jediné soubory, které se v nové verzi změní, budou `projekt-01-kapitoly-chapters.tex` a `projekt-30-prilohy-appendices.tex`, jejichž obsah každý student vymaže a nahradí vlastním. Šablonu lze tedy bez problémů využít i v současné verzi.

Popis částí šablony

Po rozbalení šablony naleznete následující soubory a adresáře:

bib-styles Styly literatury (viz níže).

obrazky-figures Adresář pro Vaše obrázky. Nyní obsahuje placeholder.pdf (tzv. TODO obrázek, který lze použít jako pomůcku při tvorbě technické zprávy), který se s prací neodevzdává. Název adresáře je vhodné zkrátit, aby byl jen ve zvoleném jazyce.

template-fig Obrázky šablony (znak VUT).

fitthesis.cls Šablona (definice vzhledu).

Makefile Makefile pro překlad, počítání normostran, sbalení apod. (viz níže).

projekt-01-kapitoly-chapters.tex Soubor pro Váš text (obsah nahraďte).

projekt-20-literatura-bibliography.bib Seznam literatury (viz níže).

projekt-30-prilohy-appendices.tex Soubor pro přílohy (obsah nahraďte).

projekt.tex Hlavní soubor práce – definice formálních částí.

Výchozí styl literatury (czechiso) je od Ing. Martínka, přičemž anglická verze (englishiso) je jeho překladem s drobnými modifikacemi. Oproti normě jsou v něm určité odlišnosti, ale

na FIT je dlouhodobě akceptován. Alternativně můžete využít styl od Ing. Radima Loskota nebo od Ing. Radka Pyšného[1]. Alternativní styly obsahují určitá vylepšení, ale zatím nebyly řádně otestovány větším množstvím uživatelů. Lze je považovat za beta verze pro zájemce, kteří svoji práci chtějí mít dokonalou do detailů a neváhají si nastudovat detaily správného formátování citací, aby si mohli ověřit, že je vysázený výsledek v pořádku.

Makefile kromě překladu do PDF nabízí i další funkce:

- přejmenování souborů (viz níže),
- počítání normostran,
- spuštění vlny pro doplnění nezlomitelných mezer,
- sbalení výsledku pro odeslání vedoucímu ke kontrole (zkontrolujte, zda sbalí všechny Vámi přidané soubory, a případně doplňte).

Nezapomeňte, že vlna neřeší všechny nezlomitelné mezery. Vždy je třeba manuální kontrola, zda na konci řádku nezůstalo něco nevhodného – viz Internetová jazyková příručka[2].

Pozor na číslování stránek! Pokud má obsah 2 strany a na 2. jsou jen „Přílohy“ a „Seznam příloh“ (ale žádná příloha tam není), z nějakého důvodu se posune číslování stránek o 1 (obsah „nesedí“). Stejný efekt má, když je na 2. či 3. stránce obsahu jen „Literatura“ a je možné, že tohoto problému lze dosáhnout i jinak. Řešení je několik (od úpravy obsahu, přes nastavení počítadla až po sofistikovanější metody). **Před odevzdáním proto vždy překontrolujte číslování stran!**

Doporučený postup práce se šablonou

1. **Zkontrolujte, zda máte aktuální verzi šablony.** Máte-li šablonu z předchozího roku, na stránkách fakulty již může být novější verze šablony s aktualizovanými informacemi, opravenými chybami apod.

2. **Zvolte si jazyk**, ve kterém budete psát svoji technickou zprávu (česky, slovensky nebo anglicky) a svoji volbu konzultujte s vedoucím práce (nebyla-li dohodnuta předem). Pokud Vámi zvoleným jazykem technické zprávy není čeština, nastavte příslušný parametr šablony v souboru projekt.tex (např.: `documentclass[english]{fitthesis}` a přeložte prohlášení a poděkování do angličtiny či slovenštiny.

3. **Přejmenujte soubory.** Po rozbalení je v šabloně soubor `projekt.tex`. Pokud jej přeložíte, vznikne PDF s technickou zprávou pojmenované `projekt.pdf`. Když vedoucímu více studentů pošle `projekt.pdf` ke kontrole, musí je pracně přejmenovávat. Proto je vždy vhodné tento soubor přejmenovat tak, aby obsahoval Váš login a (případně zkrácené) téma práce. Vyhněte se však použití mezer, diakritiky a speciálních znaků. Vhodný název může být např.: „`xlogin00-Cisteni-a-extrakce-textu.tex`“. K přejmenování můžete využít i přiložený Makefile:

   ```
   make rename NAME=xlogin00-Cisteni-a-extrakce-textu
   ```

[1] BP Ing. Radka Pyšného http://www.fit.vutbr.cz/study/DP/BP.php?id=7848
[2] Internetová jazyková příručka http://prirucka.ujc.cas.cz/?id=880

4. Vyplňte požadované položky v souboru, který byl původně pojmenován `projekt.tex`, tedy typ, rok (odevzdání), název práce, svoje jméno, ústav (dle zadání), tituly a jméno vedoucího, abstrakt, klíčová slova a další formální náležitosti.

5. Nahraďte obsah souborů s kapitolami práce, literaturou a přílohami obsahem svojí technické zprávy. Jednotlivé přílohy či kapitoly práce může být výhodné uložit do samostatných souborů – rozhodnete-li se pro toto řešení, je doporučeno zachovat konvenci pro názvy souborů, přičemž za číslem bude následovat název kapitoly.

6. Nepotřebujete-li přílohy, zakomentujte příslušnou část v `projekt.tex` a příslušný soubor vyprázdněte či smažte. Nesnažte se prosím vymyslet nějakou neúčelnou přílohu jen proto, aby daný soubor bylo čím naplnit. Vhodnou přílohou může být obsah přiloženého paměťového média.

7. Nascanované zadání uložte do souboru `zadani.pdf` a povolte jeho vložení do práce parametrem šablony v projekt.tex (`documentclass[zadani]{fitthesis}`).

8. Nechcete-li odkazy tisknout barevně (tedy červený obsah – bez konzultace s vedoucím nedoporučuji), budete pro tisk vytvářet druhé PDF s tím, že nastavíte parametr šablony pro tisk: (`documentclass[zadani,print]{fitthesis}`). Barevné logo se nesmí tisknout černobíle!

9. Vzor desek, do kterých bude práce vyvázána, si vygenerujte v informačním systému fakulty u zadání. Pro disertační práci lze zapnout parametrem v šabloně (více naleznete v souboru fitthesis.cls).

10. Nezapomeňte, že zdrojové soubory i (obě verze) PDF musíte odevzdat na CD či jiném médiu přiloženém k technické zprávě.

Obsah práce se generuje standardním příkazem `\tableofcontents` (zahrnut v šabloně). Přílohy jsou v něm uvedeny úmyslně.

Pokyny pro oboustranný tisk

- **Oboustranný tisk je doporučeno konzultovat s vedoucím práce.**
- Je-li práce tištěna oboustranně a její tloušťka je menší než tloušťka desek, nevypadá to dobře.
- Zapíná se parametrem šablony: `\documentclass[twoside]{fitthesis}`
- Po vytištění oboustranného listu zkontrolujte, zda je při prosvícení sazební obrazec na obou stranách na stejné pozici. Méně kvalitní tiskárny s duplexní jednotkou mají často posun o 1–3 mm. Toto může být u některých tiskáren řešitelné tak, že vytisknete nejprve liché stránky, pak je dáte do stejného zásobníku a vytisknete sudé.
- Za titulním listem, obsahem, literaturou, úvodním listem příloh, seznamem příloh a případnými dalšími seznamy je třeba nechat volnou stránku, aby následující část začínala na liché stránce (\cleardoublepage).
- Konečný výsledek je nutné pečlivě překontrolovat.

Styl odstavců

Odstavce se zarovnávají do bloku a pro jejich formátování existuje více metod. U papírové literatury je častá metoda s použitím odstavcové zarážky, kdy se u jednotlivých odstavců textu odsazuje první řádek odstavce asi o jeden až dva čtverčíky (vždy o stejnou, předem zvolenou hodnotu), tedy přibližně o dvě šířky velkého písmene M základního textu. Poslední řádek předchozího odstavce a první řádek následujícího odstavce se v takovém případě neoddělují svislou mezerou. Proklad mezi těmito řádky je stejný jako proklad mezi řádky uvnitř odstavce. [32] Další metodou je odsazení odstavců, které je časté u elektronické sazby textů. První řádek odstavce se při této metodě neodsazuje a mezi odstavce se vkládá vertikální mezera o velikosti 1/2 řádku. Obě metody lze v kvalifikační práci použít, nicméně často je vhodnější druhá z uvedených metod. Metody není vhodné kombinovat.

Jeden z výše uvedených způsobů je v šabloně nastaven jako výchozí, druhý můžete zvolit parametrem šablony „`odsaz`“.

Užitečné nástroje

Následující seznam není výčtem všech využitelných nástrojů. Máte-li vyzkoušený osvědčený nástroj, neváhejte jej využít. Pokud však nevíte, který nástroj si zvolit, můžete zvážit některý z následujících:

MikTeX LaTeX pro Windows – distribuce s jednoduchou instalací a vynikající automatizací stahování balíčků.

TeXstudio Přenositelné opensource GUI pro LaTeX. Ctrl+klik umožňuje přepínat mezi zdrojovým textem a PDF. Má integrovanou kontrolu pravopisu, zvýraznění syntaxe apod. Pro jeho využití je nejprve potřeba nainstalovat MikTeX.

WinEdt Ve Windows je dobrá kombinace WinEdt + MiKTeX. WinEdt je GUI pro Windows, pro jehož využití je nejprve potřeba nainstalovat MikTeX či TeX Live.

Kile Editor pro desktopové prostředí KDE (Linux). Umožňuje živé zobrazení náhledu. Pro jeho využití je potřeba mít nainstalovaný TeX Live a Okular.

JabRef Pěkný a jednoduchý program v Javě pro správu souborů s bibliografií (literaturou). Není potřeba se nic učit – poskytuje jednoduché okno a formulář pro editaci položek.

InkScape Přenositelný opensource editor vektorové grafiky (SVG i PDF). Vynikající nástroj pro tvorbu obrázků do odborného textu. Jeho ovládnutí je obtížnější, ale výsledky stojí za to.

GIT Vynikající pro týmovou spolupráci na projektech, ale může výrazně pomoci i jednomu autorovi. Umožňuje jednoduché verzování, zálohování a přenášení mezi více počítači.

Overleaf Online nástroj pro LaTeX. Přímo zobrazuje náhled a umožňuje jednoduchou spolupráci (vedoucí může průběžně sledovat psaní práce), vyhledávání ve zdrojovém textu kliknutím do PDF, kontrolu pravopisu apod. Zdarma jej však lze využít pouze s určitými omezeními (někomu stačí na disertaci, jiný na ně může narazit i při psaní bakalářské práce) a pro dlouhé texty je pomalejší.

Pozn.: Overleaf nepoužívá Makefile v šabloně – aby překlad fungoval, je nutné kliknout pravým tlačítkem na `projekt.tex` a zvolit „Set as Main File“.

Užitečné balíčky pro LaTeX

Studenti při sazbě textu často řeší stejné problémy. Některé z nich lze vyřešit následujícími balíčky pro LaTeX:

- `amsmath` – rozšířené možnosti sazby rovnic,
- `float, afterpage, placeins` – úprava umístění obrázků,
- `fancyvrb, alltt` – úpravy vlastností prostředí Verbatim,
- `makecell` – rozšíření možností tabulek,
- `pdflscape, rotating` – natočení stránky o 90 stupňů (pro obrázek či tabulku),
- `hyphenat` – úpravy dělení slov,
- `picture, epic, eepic` – přímé kreslení obrázků.

Některé balíčky jsou využity přímo v šabloně (v dolní části souboru fitthesis.cls). Nahlédnutí do jejich dokumentace může být rovněž užitečné.

Sloupec tabulky zarovnaný vlevo s pevnou šířkou je v šabloně definovaný „L“ (používá se jako „p“).