VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
FACULTY OF INFORMATION TECHNOLOGY

ÚSTAV INFORMAČNÍCH SYSTÉMŮ
DEPARTMENT OF INFORMATION SYSTEMS

TECHNIKY KLASIFIKACE PROTEINŮ

TECHNIQUES OF PROTEIN CLASSIFICATION

DIPLOMOVÁ PRÁCE
MASTER'S THESIS

AUTOR PRÁCE — **Bc. LUKÁŠ DEKRÉT**
AUTHOR

VEDOUCÍ PRÁCE — **Ing. IVANA BURGETOVÁ, Ph.D.**
SUPERVISOR

BRNO 2020

Vysoké učení technické v Brně
Fakulta informačních technologií

Ústav informačních systémů (UIFS) Akademický rok 2019/2020

Zadání diplomové práce

Student: **Dekrét Lukáš, Bc.**
Program: Informační technologie Obor: Bezpečnost informačních technologií
Název: **Techniky klasifikace proteinů**
Protein Classification Techniques
Kategorie: Bioinformatika

Zadání:

1. Seznamte se s různými klasifikačními metodami.
2. Seznamte se s různými způsoby a technikami klasifikace proteinů.
3. Po dohodě s vedoucí vyberte vhodný klasifikační model pro klasifikaci proteinů na základě zvolených vlastností.
4. Vytvořte vhodnou datovou sadu pro testování zvoleného klasifikačního modelu.
5. Zvolený model implementujte a otestujte.
6. Zhodnoťte dosažené výsledky.

Literatura:

- Han, J., Kamber, M.: Data Mining: Concepts and Techniques, Third Edition, Morgan Kaufmann Publishers, 2012, 703 p., ISBN 978-0-12-381479-1
- Golan, Y.: Introduction to Computational Proteomics, Chapman and Hall/CRC, 2010, 767 p., ISBN 9781584885559

Při obhajobě semestrální části projektu je požadováno:

- Body 1 až 3.

Podrobné závazné pokyny pro vypracování práce viz https://www.fit.vut.cz/study/theses/

Vedoucí práce: **Burgetová Ivana, Ing., Ph.D.**
Vedoucí ústavu: Kolář Dušan, doc. Dr. Ing.
Datum zadání: 1. listopadu 2019
Datum odevzdání: 20. května 2020
Datum schválení: 24. října 2019

Abstrakt

Hlavným cieľom klasifikácie proteínov do rodín je pochopiť štruktúrne, funkčné a evolučné vzťahy medzi jednotlivými proteínmi, ktoré nie sú jednoducho odvoditeľné z dostupných dát. Keďže štruktúra a funkcia proteínov spolu úzko súvisia, zisťovanie funkcie vychádza hlavne zo štruktúrnych vlastností, ktoré sa dajú so súčasnými prostriedkami získať pomerne jednoducho. Klasifikácia proteínov má svoje miesto taktiež pri vývoji špeciálnych liekov, diagnostikovaní klinických chorôb alebo v personalizovanej zdravotnej starostlivosti, vďaka čomu sa do nej investuje značné množstvo zdrojov. V práci som vytvoril nový hierarchický nástroj pre klasifikáciu proteínov, ktorý dosahuje lepšie výsledky ako niektoré existujúce riešenia. Realizácii nástroja predchádzalo oboznámenie s vlastnosťami proteínov, preskúmanie existujúcich prístupov klasifikácie, tvorba rozsiahlej dátovej sady, uskutočnenie experimentov a výber výsledných klasifikátorov hierarchického nástroja.

Abstract

Main goal of classifying proteins into families is to understand structural, functional and evolutionary relationships between individual proteins, which are not easily deducible from available data. Since the structure and function of proteins are closely related, determination of function is mainly based on structural properties, that can be obtained relatively easily with current resources. Protein classification is also used in developement of special medicines, in the diagnosis of clinical diseases or in personalized healthcare, which means a lot of investment in it. I created a new hierarchical tool for protein classification that achieves better results than some existing solutions. The implementation of the tool was preceded by acquaintance with the properties of proteins, examination of existing classification approaches, creation of an extensive data set, realizing experiments and selection of the final classifiers of the hierarchical tool.

Kľúčové slová

proteín, aminokyselina, klasifikácia, neurónová sieť, SVM, náhodný les, vektor príznakov

Keywords

protein, amino acid, classification, neural net, SVM, random forest, feature vector

Citácia

DEKRÉT, Lukáš. *Techniky klasifikace proteinů*. Brno, 2020. Diplomová práce. Vysoké učení technické v Brně, Fakulta informačních technologií. Vedoucí práce Ing. Ivana Burgetová, Ph.D.

Techniky klasifikace proteinů

Prehlásenie

Prehlasujem, že som túto prácu vypracoval samostatne pod vedením pani Ing. Ivany Burgetovej Ph.D. Uviedol som všetky literárne pramene a publikácie z ktorých som čerpal.

.....
Lukáš Dekrét
2. júna 2020

Poďakovanie

Veľké poďakovanie patrí Ing. Ivane Burgetovej Ph.D. za odborné vedenie tejto práce, cenné rady a ústretovosť pri konzultáciách. Ďalej sa chcem poďakovať organizácii MetaCentrum a Google Colaboratory za poskytnutie výpočtových zdrojov.

Obsah

1 Úvod **3**

2 Proteíny **5**
2.1 Vznik proteínov 5
2.2 Aminokyseliny 6
2.3 Štruktúra proteínov 7
2.4 Fyzikálne, chemické a biologické vlastnosti aminokyselín 7
2.5 Základné delenie proteínov 8

3 Strojové učenie **10**
3.1 Delenie metód strojového učenia 10
3.2 Neurónové siete 11
3.3 Rozhodovacie stromy 14
3.4 Support Vector Machines (SVM) 16
3.5 Metriky hodnotenia algoritmov strojového učenia 18

4 Klasifikácia proteínov **20**
4.1 Reprezentácia proteínov 20
4.2 Výber vektora príznakov 23
4.3 Redukcia príznakov 27
4.4 Existujúce klasifikačné nástroje 28

5 Návrh a implementácia **30**
5.1 Požiadavky implementovaného programu 30
5.2 Program na extrakciu príznakov a trénovanie modelov 31
5.3 Hierarchický nástroj na klasifikáciu proteínov 37

6 Experimenty **39**
6.1 Výber dátovej sady 39
6.2 Overenie naimplementovaných metód extrakcie príznakov 42
6.3 Experimenty pre výber metód extrakcie príznakov 43
6.4 Experimenty pre výber klasifikátorov do hierarchického nástroja 50
6.5 Výsledky hierarchického nástroja 52

7 Záver **56**

Literatúra **58**

A Obsah priloženého DVD **63**

B Parametre algoritmov strojového učenia použité pri experimentoch **65**

C Dátové sady použité v diplomovej práci **67**
C.1 Dátová sada použitá na experimenty pre výber metód extrakcie príznakov . 67
C.2 Dátová sada použitá na výber klasifikátorov do hierarchického nástroja . . . 69

Kapitola 1

Úvod

Proteíny sú základné stavebné jednotky všetkých živých organizmov. Zohrávajú dôležitú úlohu pri väčšine bunečných funkciách. Zabezpečujú pohyb, vnútorné dýchanie, či katalyzáciu chemických reakcií. Získať informácie o proteínovej sekvencii nie je pri súčasných prostriedkoch zložité. Problém nastáva pri určovaní konkrétnej funkcie proteínov. Jedným z možných riešení je ich klasifikácia. Kvôli vysokej variabilite a počtu proteínov sa proces klasifikácie stále viac automatizuje. Automatizácia procesu klasifikácie proteínov otvára možnosti pokročilej tvorbe liekov, diagnostikovaniu klinických chorôb ako napríklad Alzheimerova alebo neurodegeneratívna choroba, alebo vzniku personalizovanej zdravotnej starostlivosti. Presná metóda klasifikácie proteínov by nám okrem iného pomohla pochopiť aj evolučné vzťahy medzi jednotlivými proteínmi a živočíšnymi druhmi. Táto práca skúma možnosti efektívnej a presnej klasifikácie proteínov s využitím komplexných klasifikačných algoritmov. Schopnosť klasifikovať a rozlišovať objekty medzi sebou je pre informačné technológie veľmi dôležitá. Potrebujeme ju s rôznymi obmenami vo všetkých systémoch, ktoré pracujú so štrukturovanými dátami.

Práca je členená do piatich hlavných kapitol. Kapitola 2 sa venuje popisu biologických procesov vzniku proteínov, ich základným stavebným jednotkám nazývaným aminokyseliny a popisu štruktúry proteínov. V tejto kapitole sú taktiež uvedené fyzikálne, chemické a biologické vlastnosti aminokyselín, ktoré sú pre túto prácu dôležité. Posledná časť kapitoly 2 uvádza základné delenie proteínov.

Kapitola 3 je venovaná metódam strojového učenia, ktoré sú využívané pri klasifikácii. Konkrétne je uvedená metóda neurónových sietí, rozhodovacie stromy, náhodný les a Support Vector Machines (SVM). Ďalšou časťou tejto kapitoly je popis metrík hodnotenia algoritmov strojového učenia.

Kapitolu 4 tvorí popis klasifikácie proteínov. Dôraz sa kladie na spôsoby reprezentácie proteínov, metódam na výber príznakov a metódam redukcie príznakov. Táto kapitola tvorí jadro teoretickej časti diplomovej práce, pretože kvalitné spracovanie vstupných dát je pri trénovaní kvalitného klasifikátora kľúčové. Okrem toho sa v nej popisujú niektoré z veľkého množstva klasifikačných nástrojov, ktoré sú dostupné na webových serveroch.

V kapitole 5 sa nachádza popis požiadavkov na implementovaný program, ktoré plynú zo zadania diplomovej práce a popis návrhu a implementácie programu na extrakciu príznakov a trénovanie modelov. Záver kapitoly je venovaný úvodu do hierarchického nástroja na škálovateľnú klasifikáciu proteínov.

Posledná kapitola obsahuje kontrolu správnosti naimplementovaných metód pomocou výsledkov z referenčných článkov, popis výberu dátových sád, popis experimentov pre výber metód extrakcie príznakov a popis experimentov pre výber klasifikátorov do hierarchického

nástroja. V poslednej časti kapitoly 6 sú uvedené výsledky hierarchického nástroja a porovnanie so súčasnými nástrojmi.

Kapitola 2

Proteíny

Proteíny, nazývané taktiež bielkoviny, sú základné stavebné jednotky všetkých živých organizmov. Zohrávajú dôležitú úlohu pri väčšine bunečných funkciách. V tejto kapitole je popísaná ich tvorba, štruktúra, delenie a niektoré fyzikálne, chemické a biologické vlastnosti aminokyselín, z ktorých sa proteíny skladajú.

2.1 Vznik proteínov

Pre vznik proteínov je veľmi dôležitá deoxyribonukleová kyselina (DNA). Tá je tvorená dvoma polynukleotidovými reťazcami, ktoré spolu držia vodíkovími väzbami, pričom vytvárajú pravotočivú špirálovitú štruktúru. Polynukleotidový reťazec vzniká spojením nukleotidov cez cukor-fosfátovú kostru. Nukleotid je zložený z päťuhlíkovej molekuly monosacharidu (deoxyribózy), fosfátovej skupiny a nukleovej dusíkatej bázy. V DNA sa vyskytujú štyri typy dusíkatých báz: adenín (A), guanín (G), tymín (T) a cytozín (C). Tie sa ďalej delia na dve skupiny. Purínová skupina obsahuje adenín a guanín, pyrimidínová skupina tymín a cytozín. Vodíkové väzby medzi polynukleotidovími reťazcami DNA vznikajú spojením tymínu s adenínom a cytozínu s guanínom. Hovoríme teda, že tymín je komplementárny ku adenínu a cytozín ku guanínu. Proces vzniku proteínov sa nazýva proteosyntéza a skladá sa z transkripcie a translácie.

Transkripcia je proces, pri ktorom dochádza k prepisu časti DNA do molekuly RNA (ribonukleová kyselina). RNA je tvorená jedným polynukleotidovým reťazcom. Oproti DNA sa líši tým, že má namiesto tymínu uracil (U) a namiesto deoxyribózy ribózu. Je taktiež niekoľkonásobne kratšia ako DNA. Pred začiatkom transkripcie enzým RNA-polymeráza nájde oblasť promotoru obsahujúci informáciu o začiatku potrebnej sekvencie DNA pre tvorbu proteínu (tzv. génu), a na to miesto sa naviaže. V tom mieste sa polynukleotidové reťazce DNA začnú rozvoľňovať a vďaka komplementarite nukleotidov sa začne tvoriť RNA. Proces končí, keď RNA-polymeráza narazí na sekvenciu, ktorá ukončuje daný gén. Výsledná RNA molekula sa volá mediátorová RNA (mRNA).

Translácia je proces, pri ktorom sa z informácie uloženej v mRNA vytvorí polypeptidový reťazec aminokyselín (proteín). Hlavnú úlohu pri translácii majú ribozómy, tvorené ribozomálnou RNA a proteínmi 5S a 23S. Ribozómy postupne čítajú mRNA po trojiciach nukleotidov (tzv. kodónoch), kde každá trojica kóduje jednu aminokyselinu, ktorá je pripojená do vznikajúceho polypeptidového reťazca. Štartovací kodón mRNA má tvar AUG a ukončujúce stop kodóny tvar UAA, UAG alebo UGA. AUG je taktiež tvar kodónu pre aminokyselinu *methionin*. Nezanedbateľnú úlohu pri translácii hrá transferová RNA (tRNA),

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

Tabuľka 2.1: Kombinácie nukleotidov pre aminokyseliny podľa článku [52]. Zelenou farbou je vyznačený štartovací kodón a červenou farbou stop kodóny.

ktorá prenáša aminokyseliny v bunke a odovzdáva ich do reťazca aminokyselín, ktoré sa spájajú peptidovou väzbou. Máme 64 možných kombinácii nukleotidov. Ich tvar a výslednú aminokyselinu vidíme v tabuľke 2.1.

2.2 Aminokyseliny

Aminokyseliny sú organické zlúčeniny, v ktorých sa vyskytujú najmä atómy uhlíka (C), vodíka (H), kyslíka (O) a dusíka (N). Obsahujú karboxylovú (-COOH) a aminovú ($-NH_2$) skupinu, ktoré sú naviazané na uhlík atómu, nazývanému taktiež α-uhlík. Okrem týchto zlúčenín sa v aminokyselinách nachádza postranný reťazec, na základe ktorého ich vieme rozlíšiť. Postranný reťazec taktiež určuje chemické vlastnosti aminokyselín. Celkovo poznáme 20 aminokyselín, ktoré sa nachádzajú v genetickom kóde. Niektoré zdroje [51] uvádzajú aj ďalšie, nazývané seleno-aminokyseliny (SeCys alebo SeMet). Tie sa však nenachádzajú vo všetkých organizmoch a tak sú v práci zanedbané. Na základe fyzikálnych vlastností delíme aminokyseliny na [54]:

- **Nepolárne (hydrofóbne):** glycin (Gly, G), alanin (Ala, A), valin (Val, V), leucin (Leu, L), isoleucin (Ile, I), methionin (Met, M), fenilalanin (Phe, F), tryptofan (Trp, W), prolin (Pro, P), cystein (Cys, C)
- **Polárne s kladným nábojom (zásadité):** arginin (Arg, R), histidin (His, H), lysine (Lys, K)
- **Polárne so záporným nábojom (kyslé):** kyselina asparagová (Asp, D), kyselina glutamová (Glu, E)
- **Polárne bez náboja:** serin (Ser, S), threonin (Thr, T), asparagin (Asn, N), glutamin (Gln, Q), tyrosin (Tyr, Y)

2.3 Štruktúra proteínov

Proteíny sú makromolekuly zložené z aminokyselín, ktoré sú vzájomne spojené peptidovou väzbou. Tá prepojuje aminovú skupinu jednej aminokyseliny s karboxylovou skupinou druhej aminokyseliny. Reťazec takýchto aminokyselín sa nazýva polypeptid. V článku [50] bolo experimentálne zistené, že funkcia proteínov priamo súvisí s ich aminokyselinovou sekvenciou a štruktúrou. Proteín môžeme organizovať do štyroch základných úrovní štruktúry [54, 56]:

- **Primárna štruktúra:** sekvencia aminokyselín, ktoré čítame od N-konca (voľná aminová skupina) po C-koniec (voľná karboxylová skupina).
- **Sekundárna štruktúra:** zachytáva lokálne usporiadanie na sekvencii proteínu. Najčastejšie usporiadania sú α-špirála a β-skladaný list. Pri α-špirále ide o vodíkové mostíky medzi jednotlivými aminokyselinami, ktoré sú od seba vzdialené 3-4 aminokyseliny. β-skladané listy sú tvorené taktiež vodíkovými väzbami tak, že sa jeden úsek polypeptidového reťazca spojí s iným niekde ďalej v reťazci.
- **Terciárna štruktúra:** reprezentuje trojrozmernú priestorovú pozíciu každého atómu v polypeptidovom reťazci. Podoba výslednej terciárnej štruktúry závisí predovšetkým od chemických vlastností aminokyselín [27]. Nepolárne (hydrofóbne) aminokyseliny sa viac balia do vnútra proteínu a ostatné na okraje, aby mohli lepšie interagovať s okolím.
- **Kvartérna štruktúra:** popisuje počet a relatívnu pozíciu polypeptidových reťazcov, ktoré sú spojené nekovalentnými väzbami do väčšieho komplexu.

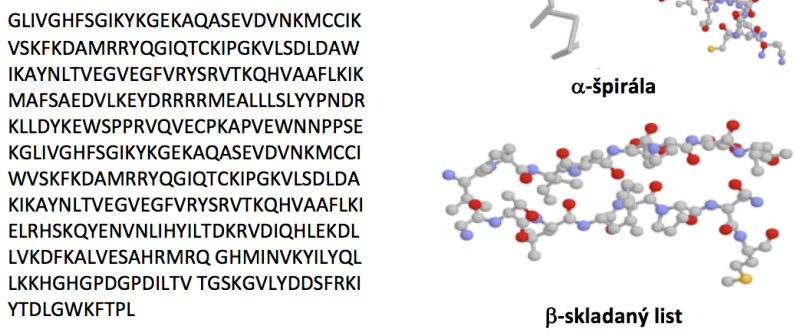
Primárna, sekundárna, terciárna a kvartérna štruktúra je zobrazená na obrázku 2.1.

2.4 Fyzikálne, chemické a biologické vlastnosti aminokyselín

O sekvencii, tvare a vlastnostiach proteínov do nemalej miery rozhodujú aj fyzikálne, chemické a biologické vlastnosti aminokyselín, ktoré tvoria daný proteín [27]. Všetky vlastnosti sa dajú nájsť v databáze GenomeNet [8], a sú dostupné cez LinkDB systém. Databáza je organizovaná do dvoch skupín: aminokyselinové indexy a mutačné aminokyselinové matice. Pri aminokyselinových indexoch má každý záznam prístupové číslo, krátky popis a číselné hodnoty každej z dvadsiatich aminokyselín pre danú vlastnosť. Vlastnosti uložené v tejto databáze môžeme rozdeliť do štyroch základných zhlukov [22]:

- **α-špirála a sklony aminokyselín ku otočeniu**
- **sklony ku vytvoreniu β-skladaného listu**
- **hydrofóbnosť**
- **iné fyzikálno-chemické vlastnosti**

Mutačné aminokyselinové matice môžu byť buď symetrické alebo nesymetrické. Formát každého záznamu je podobný ako pri aminokyselinových indexoch. Rozdiel je len v počte číselných hodnôt. Pre symetrické matice ich je 210 a pre nesymetrické je ich 400 a viac. V práci budem využívať len aminokyselinové indexy. Vlastnosti som vyberal rovnomerne



(a) Primárna štruktúra - sekvencia aminokyselín

(b) Sekundárna štruktúra

(c) Terciárna štruktúra

(d) Kvartérna štruktúra

Obr. 2.1: Obrázky štruktúr proteínov prevzaté a upravené z [46].

z celej databázy. Tri z nich skúmajú polaritu a vlastnosti popísané v sekcii 2.2: *hydrofóbnosť aminokyselín, hydrofilnosť aminokyselín* a *polarita aminokyselín*. Ďalej som vybral po jednej vlastnosti súvisiacej s α-špirálou a β-skladaným listom: *priemerná relatívna pravdepodobnosť vzniku alfa-špirály* a *priemerná relatívna pravdepodobnosť vzniku beta skladaného listu*. Ďalšou vlastnosťou je *objem bočného reťazca aminokyselín*, ktorý úzko súvisí s priestorovým usporiadaním aminokyselín v reťazci a posledné dve vlastnosti sú *alfa-NH chemické posuvy* a *membránová preferencia cytochrómu*. Hodnoty týchto vlastností sú zobrazené v tabuľke 2.2.

2.5 Základné delenie proteínov

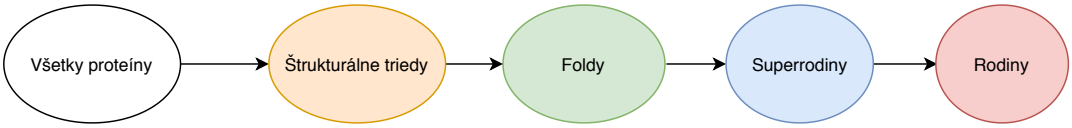
Proteíny klasifikujeme do rodín na základe ich sekvenčnej, funkčnej a sekundárnej alebo terciárnej štruktúrnej podobnosti. Proteíny nachádzajúce sa v jednej rodine majú spoločného predka, od ktorého sa odlíšili časom pomocou mutácie. V súčastnosti poznáme viac ako 60 tisíc rodín proteínov. Tie sa zhlukujú do väčších celkov, ktoré sa nazývajú superrodiny. Superrodiny sa ďalej zhlukujú do foldov (angl. *folds*) a tie do štrukturálnych tried. Rozli-

	A	R	N	D	C	Q	E	G	H	I
hydrofóbnosť aminokyselín	0,87	0,85	0,09	0,66	1,52	0,0	0,67	0,1	0,87	3,15
hydrofilnosť aminokyselín	-0,5	3,0	0,2	3,0	-1,0	0,2	3,0	0,0	-0,5	-1,8
polarita aminokyselín	8,1	10,5	11,6	13,0	5,5	10,5	12,3	9,0	10,4	5,2
objem bočného reťazca	27,5	105	58,7	40,0	44,6	80,7	62,0	0,0	79,0	93,5
pravdep. vzniku alfa-špirály	1,36	1,00	0,89	1,04	0,82	1,14	1,48	0,63	1,11	1,08
pravdep. vzniku beta skladaného listu	0,81	0,85	0,62	0,71	1,17	0,98	0,53	0,88	0,92	1,48
membránová preferencia cytochrómu	1,56	0,59	0,51	0,23	1,80	0,39	0,19	1,03	1,00	1,27
alfa-NH chemické posuvy	8,25	8,27	8,74	8,41	8,31	8,41	8,37	8,39	8,42	8,19
	L	**K**	**M**	**F**	**P**	**S**	**T**	**W**	**Y**	**V**
hydrofóbnosť aminokyselín	2,17	1,64	1,67	2,87	2,77	0,07	0,07	3,77	2,67	1,87
hydrofilnosť aminokyselín	-1,8	3,0	-1,3	-2,5	0,0	0,3	-0,4	-3,4	-2,3	-1,5
polarita aminokyselín	4,9	11,3	5,7	5,2	8,0	9,2	8,6	5,4	6,2	5,9
objem bočného reťazca	93,50	100,0	94,10	115,5	41,90	29,30	51,30	145,5	117,3	71,50
pravdep. vzniku alfa-špirály	1,21	1,22	1,45	1,05	0,52	0,74	0,81	0,97	0,79	0,94
pravdep. vzniku beta skladaného listu	1,24	0,77	1,05	1,20	0,61	0,92	1,18	1,18	1,23	1,66
membránová preferencia cytochrómu	1,38	0,15	1,93	1,42	0,27	0,96	1,11	0,91	1,10	1,58
alfa-NH chemické posuvy	8,42	8,41	8,42	8,22	0,0	8,38	8,24	8,09	8,18	8,44

Tabuľka 2.2: Hodnoty použitých fyzikálno-chemickych vlastností.

šujeme štyri základné štrukturálne triedy: *alfa proteíny, alfa a beta proteíny (alfa + beta), alfa a beta proteíny (alfa / beta)* a *beta proteíny*. Názornú hierarchiu vidíme na obrázku 2.2.

Najznámejšia databáza, ktorá má v sebe uložené rodiny proteínov sa nazýva Pfam (angl. *Protein families*) [12]. Obsahuje jednotlivé sekvencie proteínov ako aj ich viacnásobné zarovnanie. Verzia 32.0, ktorá bola vydaná v Septembri v roku 2018, obsahuje takýchto rodín skoro 18 tisíc. Okrem rodín proteínov sa v nej nachádzajú aj rodiny domén proteínov, ktoré sú konzervovanými časťami proteínových sekvencií a môžu fungovať nezávisle na zvyšku sekvencie. Každá rodina má priradený identifikátor, ktorý začína reťazcom *PF*, za ktorým nasleduje päť číslic (napr. PF05528).



Obr. 2.2: Stromová hierarchia proteínov.

V práci budem prezentovať nie len vybrané rodiny samotné, ale aj ich celú stromovú hierarchiu od štrukturálnych tried. Hierarchické informácie o proteínoch som získal z databázy SCOP (angl. *Structural Classification of Proteins*) [2, 3], ktorej cieľom je určiť evolučné vzťahy medzi proteínmi.

Kapitola 3

Strojové učenie

Strojové učenie je jedna z disciplín umelej inteligencie. Charakteristické vlastnosti metód strojového učenia, sú využívanie znalostí a práca so symbolickými či štruktúrovanými dátami [25]. Tieto metódy sa snažia nájsť spojitosti v obrovských množstvách dát, ktoré nie sú na prvý pohľad viditeľné. Okrem bioinformatiky sa využívajú v počítačovom videní, spracovaní a rozpoznávaní reči, hraní hier a v rôznych ďalších oblastiach. Jedne z hlavných nevýhod metód, je výrazné zhoršenie hľadania vzorov pri nevyvážených dátach alebo malom množstve dát. Ďalší problém strojového učenia popísaný v článkoch [17, 39] známy ako *The Curse of dimensionality*, vzniká pri veľkom množstve vstupných parametrov, ktoré nemajú žiadnu rozlišovaciu hodnotu. Okrem tohto môže dochádzať k preučeniu (angl. *overfitting*), čo vzniká v momente, keď je metóda strojového učenia tak vytrénovaná, že reaguje len na trénovaciu sadu. V tejto kapitole sú popísané základné delenia metód strojového učenia, niektoré z používaných algoritmov a metriky hodnotenia týchto algoritmov.

3.1 Delenie metód strojového učenia

Základné delenie metód strojového učenia spočíva v tom, akým spôsobom získavajú nové informácie a vylepšujú svoje výsledky. Podľa článku [53] ich delíme na tieto základné časti:

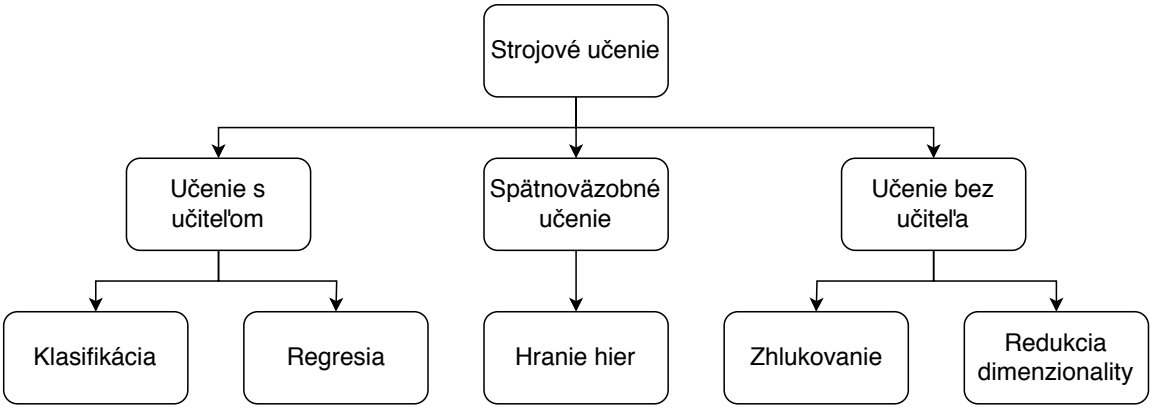
- **Učenie s učiteľom:** Pri použití tohto typu učenia je treba mať ku vstupným dátam priradený odpovedajúci výstup. Cieľom je nájsť vzťah medzi vstupnými a výstupnými dátami.
- **Učenie bez učiteľa:** Tento typ učenia sa snaží pochopiť štruktúru vstupných dát a tak zistiť ich rozdelenie zmysluplne. Dopredu nie je jasné na koľko tried budú vstupné dáta rozdelené.
- **Spätnoväzobné učenie:** Vyžaduje spätnú väzbu z okolia, ktorá závisí na predchádzajúcich akciách. Spätná väzba je častokrát vyjadrená odmenou. Algoritmus sa snaží dostať čo najlepšiu odmenu a učí sa, ktoré akcie mu ku tomu pomáhajú.

Ďalší spôsob delenia predstavený v článku [38] je:

- **Klasifikácia:** Výstupné dáta sú triedy, ktoré môžu byť reprezentované vektorom hodnôt a do ktorých sa snaží algoritmus priradiť vstupné dáta. Príkladom je priradenie proteínov do jednotlivých rodín. Klasifikácia využíva učenie s učiteľom.

- **Regresia:** Výstupné dáta sú reálne hodnoty, ktoré sa snaží algoritmus čo najpresnejšie predpovedať. Príklad môže byť odhad ceny pozemku na základe vstupných parametrov. Regresia využíva učenie s učiteľom.
- **Zhlukovanie:** Má za úlohu rozdeliť dáta do niekoľkých skupín tak, aby dáta v jednej skupine boli viac podobné zvyšným dátam z danej skupiny ako z iných skupín. Používa sa napríklad na rozdelenie užívateľov internetu pri cielenom marketingu. Zhlukovanie využíva učenie bez učiteľa.
- **Redukcia dimenzionality:** Je proces, v ktorom sa redukuje počet náhodných premenných, aby nám ostali len tie, ktoré sú principiálne dôležité. Používa sa napríklad pri rozpoznávaní obrazu. Redukcia dimenzionality využíva učenie bez učiteľa.

Spomínané delenie je zobrazené na obrázku 3.1.



Obr. 3.1: Rozdelenie algoritmov strojového učenia podľa článkov [38, 53].

3.2 Neurónové siete

Algoritmus neurónovej siete je inšpirovaný fungovaním biologickej neurónovej siete, ktorá sa nachádza v našich mozgoch. Skladá sa z neurónov a synapsí (prepojení). Základnou stavebnou jednotkou neurónových sietí je neurón. Rovnako ako biologický neurón, i tento umelý má jeden výstup (angl. *axon*), a veľa vstupov (angl. *dendrite*). Výstup sa rovnako ako v biologickom neuróne distribuuje do viacerých miest. Telo neurónu (angl. *soma*) tvorí bázová funkcia $f_b(\vec{x}, \vec{w})$, kde $\vec{x}$ je vstupný vektor príznakov, a $\vec{w}$ je vektor váh. Každý neurón obsahuje taktiež odchýlku b (angl. *bias*). Najznámejšie bázové funkcie sú:

- **lineárna bázová funkcia:**

$$f_b(\vec{x}, \vec{w}) = \sum_{n=1}^{N} (x_n * w_n) + b, \tag{3.1}$$

- **radiálna bázová funkcia:**

$$f_b(\vec{x}, \vec{w}) = \sqrt{\sum_{n=1}^{N} (w_n - x_n)^2} + b. \tag{3.2}$$

Druhá funkcia, ktorá definuje neurón je aktivačná funkcia f_a. Aktivačná funkcia je väčšinou nelineárna a ráta výsledok z hodnoty neurónu. Výsledná hodnota sa teda dá zapísať vzťahom 3.3.

$$y = f_a(f_b(\vec{x}, \vec{w})). \tag{3.3}$$

Najznámejšie aktivačné funkcie sú:

- **Gaussova funkcia:**

$$f_a(x) = ae^{\frac{-(x-b)^2}{2c^2}}, \tag{3.4}$$

- **Heavisideová (kroková) funkcia:**

$$f_a(x) = \begin{cases} 1 & ak \;\; x \geq 0, \\ 0 & inak, \end{cases} \tag{3.5}$$

- **lineárna funkcia:**

$$f_a(x) = ax + b, \tag{3.6}$$

- **rektifikovaná lineárna jednotka:**

$$f_a(x) = max(0, x), \tag{3.7}$$

- **logistická funkcia (sigmoid):**

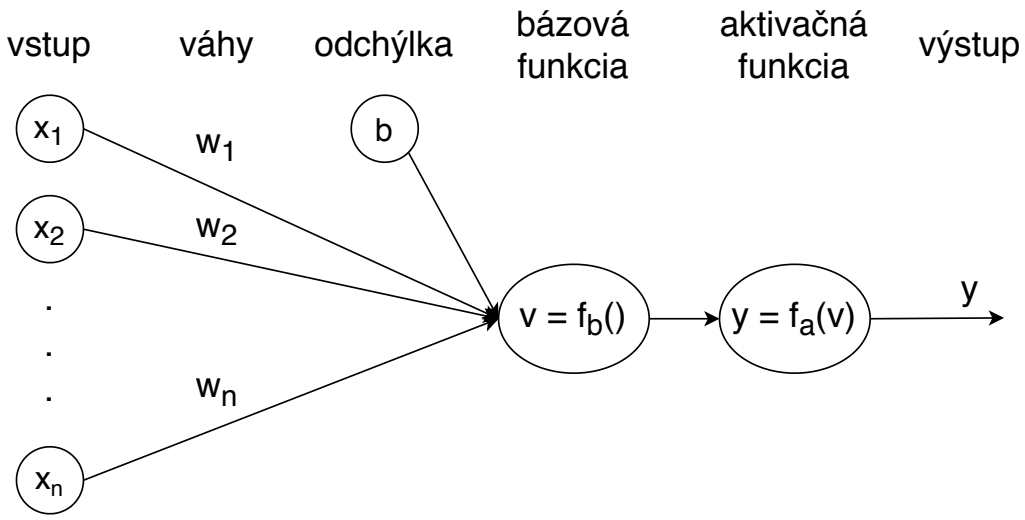
$$f_a(x) = \frac{1}{1 + e^{-x}}, \tag{3.8}$$

- **prahová funkcia:**

$$f_a(x) = \begin{cases} 1 & ak \;\; x \geq \sigma, \\ 0 & inak. \end{cases} \tag{3.9}$$

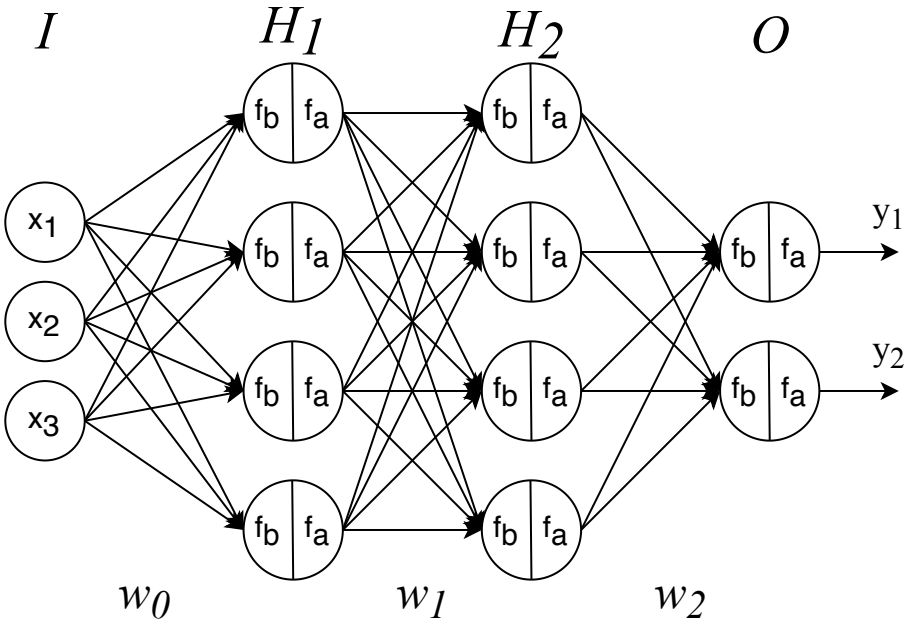
- **normalizovaná exponenciálna funkcia (softmax):** berie na vstupe vektor reálnych hodnôt a normalizuje ich do pravdepodobnostnej distribúcie.

Prvý jednoduchý matematický popis neurónu bol predstavený Warenom McCullochom a Walterom Pittsom v článku [29]. Model tohto neurónu je ukázaný na obrázku 3.2.



Obr. 3.2: Model neurónu, predstavený Warenom McCullochom a Walterom Pittsom v článku [29].

Prepojením viacerých neurónov vzniká neurónová sieť, ktorá je schopná riešiť aj zložitejšie problémy. Príklad takejto siete je ukázaný na obrázku 3.3. V priebehu vývoja sa vytvorilo veľa variánt neurónových sietí, ktoré sa od seba líšia v rôznych aspektoch ako napríklad v topológii, type učenia, aktivačnej funkcii alebo bázovej funkcii. Pri topológii rozlišujeme dopredné neurónové siete (angl. *feedforward neural networks*) a rekurentné neurónové siete (angl. *recurrent neural networks*). Jednou z dôležitých vlastností neurónových sietí je ich Turingovská úplnosť. Dôkaz je predstavený v článku [10].



Obr. 3.3: Umelá neurónová sieť vytvorená prepojením viacerých umelých neurónov. I predstavuje vstupnú vrstvu, O výstupnú vrstvu a H_1 a H_2 predstavujú skryté vrstvy (angl. *hidden layers*). Počet neurónov vo výstupnej vrstve sa rovná počtu výstupov (tried) riešenej úlohy.

3.2.1 Proces učenia

Keďže sa zameriavame na problém klasifikácie, predpokladáme, že existuje trénovacia dátová sada s priradenými triedami ku jednotlivým prvkom. Učenie (prípadne trénovanie) neurónovej siete spočíva v procese nastavovania hodnôt váh, aby tzv. stratová funkcia (angl. *loss function*), ktorá predstavuje rozdiel medzi aktuálnym a očakávaným výstupom bola čo najmenšia. Jedna z príkladov takejto stratovej funkcie je krížová entropia (angl. *cross entropy*). Váhy sú upravované optimalizačným procesom *gradientného zostupu* (angl. *gradient descent*). Cieľom je teda minimalizácia váh pomocou hodnoty gradientu stratovej funkcie, čo reprezentuje vektor parciálnych derivácií straty L vzhľadom na jednotlivé váhy. Túto úpravu môžeme vyjadriť rovnicou 3.10 kde η je koeficient učenia (angl. *learning rate*).

$$w_{i+1} = w_i - \eta \frac{\delta L}{\delta w_i} \tag{3.10}$$

Na výpočet potrebných gradientov jednotlivých váh sa v súčastnosti najviac používa algoritmus spätného šírenia chyby (angl. *backpropagation*). Tento algoritmus využíva reťazové pravidlo derivácie postupne smerom od výstupnej vrstvy siete až po vstupnú vrstvu, čím

ovplyvní každý parameter. Kvôli nutnosti vypočítať gradienty postupne pre každý prvok trénovacej dátovej sady, je najväčším problémom tohto algoritmu práve vysoká výpočtová náročnosť. Optimalizácia spočíva v dávkovom spracovaní trénovacej sady, pri ktorom sa sada rozdelí na malé množiny vzoriek tzv. dávok (angl. *mini-batch*). Výsledná stratová funkcia sa počíta z celej takejto dávky, vďaka čomu neprebieha modifikácia váh pre každý prvok zvlášť, ale len toľkokrát, koľko existuje dávok. Tento princíp bol predstavený v článku [23]. Proces učenia sa opakuje v niekoľkých iteráciách a je dokončený v momente, keď je splnená aspoň jedna podmienka z nasledujúcich:

- Učenie dosiahne maximálneho počtu iterácií.
- Počet zle klasifikovaných označených prvkov klesne pod dopredu určený prah.
- Všetky zmeny váh klesnú pod hodnotu daného prahu.
- Stratová funkcia začne dávať vyššie hodnoty. Na to aby sme predišli uviaznutiu v lokálnom minime sa používa parameter trpezlivosti (angl. *patience*), ktorý určuje počet iterácii, v ktorých sa môže stratová funkcia po sebe zhoršiť.

3.3 Rozhodovacie stromy

Rozhodovací strom predstavený v článku [47] je algoritmus založený na učení s učiteľom. Dá sa použiť pri klasifikácii ako aj pri regresii. Jeho model je usporiadaný hierarchicky do stromovej štruktúry, kde každý uzol znázorňuje vybraný príznak, spojenie rodičovského uzlu s potomkom predstavuje rozhodovacie pravidlo súvisiace s daným príznakom a každý listový uzol je výsledok. Pri klasifikácii sú listy jednotlivé triedy. Príklad takéhoto rozhodovacieho stromu je ukázaný na obrázku 3.4. Algoritmy na vytváranie takýchto stromov sú napríklad:

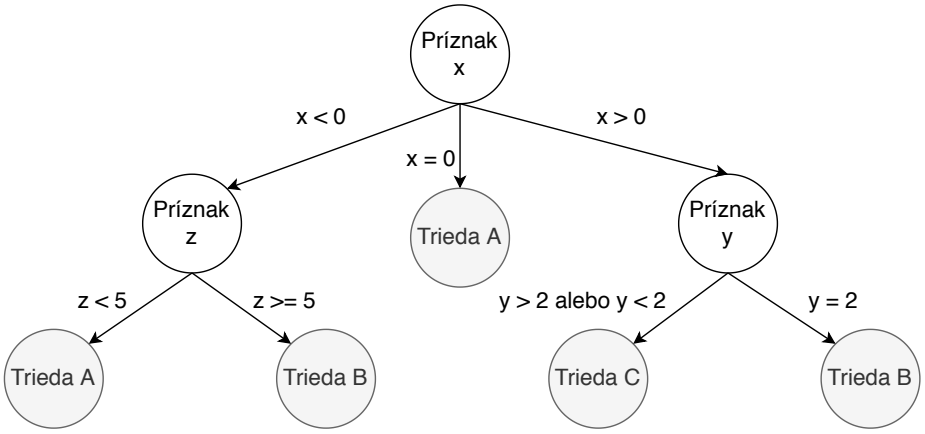
- CART (angl. *Classification and Regression Trees*), ktoré ako metriku používajú Giniho koeficient.
- ID3 (angl. *Iterative Dichotomiser 3*), ktorý ako metriku používa entropickú funkciu.

Metrika rozhodovacieho stromu slúži na to, aby bol schopný vybrať poradie príznakov v strome. Cieľom je vyberať také príznaky, ktoré rozdelia vstupné dáta čo najrovnomernejšie. Metrika algoritmu ID3 má tvar vyjadrený rovnicou 3.11.

$$H(S) = \sum_{c \in C} -p(c) \log_2 p(c), \tag{3.11}$$

kde S je množina dát pre ktorú sa počíta entropia, C je množina tried v S a $p(c)$ je pomer prvkov patriacich do triedy c, ku celkovému počtu prvkov v S. Entropia sa počíta pre každého potomka po vytvorení rodičovského uzlu a teda sa v každej iterácii množina S mení.

Pseudokód algoritmu 1 ukazuje základný priebeh vytvorenia rozhodovacieho stromu. Metóda označená ako *attributeSelectionMethod*, vyberá najvhodnejší atribút pre testovanie na základe výberovej metriky ako je napríklad spomínaný Giniho index.



Obr. 3.4: Príklad rozhodovacieho stromu s troma triedami.

Algorithm 1 Generovanie rozhodovacieho stromu z trénovacích dvojíc pre časť dát D [18, 44]

Vstup:

- Časť dát D, je množina označených n-tíc trénovacích dát.
- List kandidátnych atribútov (príznakov): *attributeList*.
- Metóda pre výber atribútu *attributeSelectionMethod* určujúca rozdeľovacie kritérium, ktoré najlepším spôsobom rozdeľuje vstupné dáta do jednotlivých tried. Toto kritérium sa skladá z rozdeľovacieho atribútu (*splittingAttribute* a prípadne $split - point$ alebo *splittingsubset*.

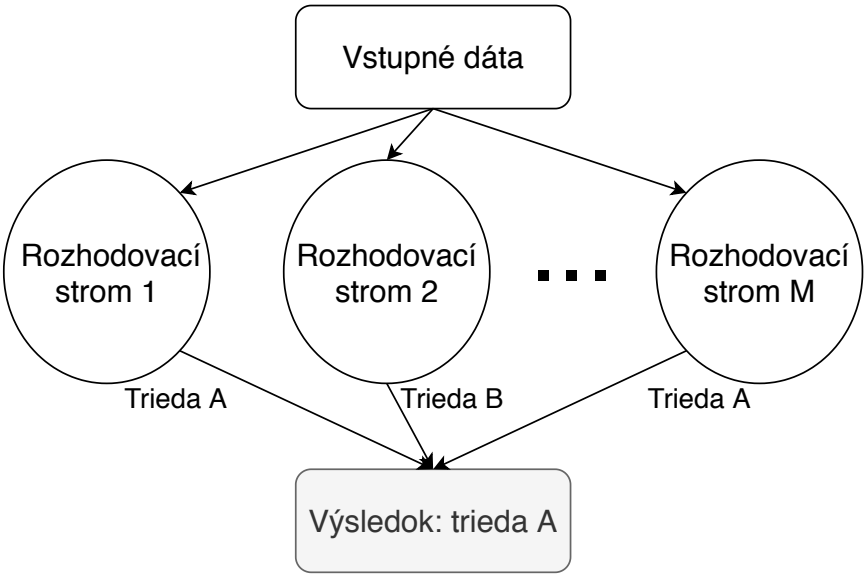
Výstup: rozhodovací strom

Metóda: GennerateDecisionTree
vytvor uzol N;
if všetky n-tice v D patria do rovnakej triedy C **then**
return N ako listový uzol triedy C;
if *attributeList* je prázdny **then**
return N ako listový uzol prevažujúci triedou C v D;
aplikuj $attributeSelectionMethod(D, attributeList)$ pre nájdenie rozdelovacieho kritéria;
označ uzol N rozdelovacím kritériom; // atributom
$attributeList \leftarrow attributeList - splittingAttribute$
for all i **in** rezdeľovacie kritérium **do**
nech D_i je monžina n-tíc z D vyhovujúcich i;
if D_i je prázdne **then**
pripoj list k uzlu N a označ ho prevažujúci triedou C v D_i
else
pripoj uzol vráteny z $GennerateDecisionTree(D_i, attributeList)$ k uzlu N hranou označenou i;
return N;

3.3.1 Náhodný les

Náhodný les publikovaný v článku [6] sa skladá z veľkého množstva (M) jednotlivých rozhodovacích stromov. Pomocou algoritmu *bagging (bootstrap aggregation)* sa vstupné dáta rozdelia na M množín tak, že každá množina obsahuje určitý počet percent (napríklad 70%) pôvodnej množiny vstupných dát. Na nich sa potom natrénuje rozhodovací strom, ktorého každý uzol je rozdelený najlepším rozdelením spomedzi podmnožiny príznakov náhodne vybraných v tomto uzle. Predikcia nových dát takto zloženého náhodného lesa sa vyberie spojením výstupov jednotlivých stromov. Pri klasifikácii to je trieda, ktorá bola predikovaná najčastejšie. Príklad takéhoto náhodného lesa s M rozhodovacími stromami je uvedený na obrázku 3.5. Pri regresii býva výsledok vypočítaný napríklad spriemerovaním hodnôt jednotlivých rozhodovacích stromov. Koncept za týmto algoritmom je jednoduchý. Veľké množstvo relatívne nezávislých modelov (rozhodovacích stromov) prekonajú ľubovoľný model, ktorý pracuje sám, pretože pravdepodobnosť, že sa pomýli väčšina modelov je menšia ako pravdepodobnosť, že sa pomýli jeden model. Výhody tejto metódy sú:

- poskytuje presnú predikciu pre širokú škálu aplikácií,
- blízkosť vzoriek je merateľná pomocou trénovaných modelov,
- trénovanie je jednoducho paralelizovateľné,
- je schopná presne odhadnúť váhu jednotlivých príznakov.



Obr. 3.5: Náhodný les použitý na klasifikáciu s M rozhodovacími stromami.

3.4 Support Vector Machines (SVM)

Support Vector Machines publikovaný v článku [9], patrí ku najznámejším metódam strojového učenia. Klasifikácia sa uskutočňuje konštruovaním n-dimenzionálnej nadroviny, ktorá rozdeľuje vstupné vektory do dvoch kategórií. Tie vstupné vektory, ktoré sa nachádzajú najbližšie pri nadrovine sa nazývajú podporné vektory (angl. *support vectors*).

Základná varianta algoritmu SVM pracuje s dvoma triedami, ktoré sa často nazývajú *pozitívna* (P) a *negatívna* (N) trieda. Nech vstupné dáta sú x_i, $(i = 1, 2, ..., M)$, kde M je počet vzoriek. Označme *pozitívnu* triedu $P = 1$ a *negatívnu* triedu $N = -1$. V prípade, že sú vstupné dáta lineárne rozdeliteľné, platia pre všetky vstupné vektory vzťahy vyjadrené rovnicami 3.12 a 3.13.

$$w^T x_i + b \geq 1, \text{pre všetky } x_i \in P, \tag{3.12}$$

$$w^T x_i + b \leq -1, \text{pre všetky } x_i \in N, \tag{3.13}$$

kde w je váhový vektor a b je odchýlka. Rozhodovacia funkcia je $f(x) = sgn\left(w^T x + b\right)$. Algoritmus sa pri takto lineárne rozdeliteľných vstupných dátach snaží rozdeliť dáta priamkou, teda 1-dimenzionálnou nadrovinou. Po rozdelení metóda zistí vzdialenosť priamky od najbližších podporných vektorov, pričom v procese učenia hľadá také podporné vektory, aby medzi nimi bola čo najväčšia vzdialenosť. SVM môžu byť použité aj na zložitejšie a nelineárne rozdeliteľné vstupné vektory. V týchto situáciách sa musia vstupné vektory namapovať na vysoko-dimenzionálny priestor príznakov. Potom vstupy môžeme znovu klasifikovať lineárne. Použitím nelineárnej vektorovej funkcie $\Phi(x) = (\varphi_1(x), ..., \varphi_m(x))$ na mapovanie n-dimenzionálneho vstupného vektoru x, dostaneme m-dimenzionálny priestor príznakov. Rozhodovacia funkcia má tvar 3.14:

$$f(x) = sgn\left(\sum_{i,j=1}^{M} \alpha_i y_i \left(\Phi^T(x_i) \cdot \Phi(x_j)\right) + b\right). \tag{3.14}$$

Pracovanie vo vysokých dimenziách avšak vytvára problém nadmerného prispôsobenia (angl. *overfitting*) na trénovacie dáta. Ten sa rieši pomocou jadrových funkcií. Jadrová funkcia je funkcia, ktorá vracia skalárny súčin priestoru príznakov vstupných vektorov, vyjadrená ako $K(x_i, x_j) = \left(\Phi^T(x_i) \cdot \Phi(x_j)\right)$. Medzi najpoužívanejšie jadrové funkcie patria:

- **polynomiálna funkcia stupňa P:**

$$K(x, y) = (x \cdot y + 1)^P, \tag{3.15}$$

- **sigmoidná funkcia:**

$$K(x, y) = tanh(\alpha x \cdot y - c), \tag{3.16}$$

- **radiálne-bázová funkcia (RBF):**

$$K(x, y) = e^{-||x-y||^2/2\sigma^2}. \tag{3.17}$$

3.4.1 Viac triedne SVM

V reálnych problémoch potrebujeme vedieť klasifikovať dáta do viac ako dvoch tried. Aby to bolo možné, používajú sa viaceré prístupy. Najznámejšie z nich sú:

- **Jeden proti všetkým**, ktorý trénuje k klasifikátorov, kde k je počet tried. Každý klasifikátor považuje jednu triedu za *pozitívnu* a všetky ostatné za *negatívne*. Výsledky všetkých klasifikátorov potom vhodne skombinuje a dostane výslednú triedu.

- **Jeden proti jednému**, ktorý trénuje $k * (k - 1)/2$ klasifikátorov, kde každý z nich je trénovaný na dvoch vstupných triedach.

3.5 Metriky hodnotenia algoritmov strojového učenia

Algoritmy strojového učenia nie sú dokonalé, vďaka čomu dochádza ku chybám pri klasifikácii. Na to, aby sme ohodnotili výkonnosť daného algoritmu nám slúžia metriky. Tie pracujú so štyrmi základnými výsledkami systému (popis je písaný z pohľadu triedy A):

- **Správne pozitívne** - (angl. *True positive*) (TP) nastáva, keď algoritmus klasifikuje prvok z triedy A, do triedy A.
- **Falošne pozitívne** - (angl. *False positive*) (FP) nastáva, keď algoritmus klasifikuje prvok z inej triedy do triedy A.
- **Správne negatívne** - (angl. *True negative*) (TN) nastáva, keď algoritmus klasifikuje prvok, ktorý nepatrí do triedy A, do inej triedy.
- **Falošne negatívne** - (angl. *False negative*) (FN) nastáva, keď algoritmus klasifikuje prvok, ktorý patrí do triedy A, do inej triedy.

Existuje veľké množstvo metrík, pričom každá skúma iné vlastnosti. Pre rôzne typy úloh sú niektoré metriky dôležitejšie ako iné. Pre klasifikáciu proteínov je najdôležitejšia presnosť. Najznámejšie používané metriky na hodnotenie algoritmov strojového učenia sú [16, 37]:

- **Senzitivita** - (angl. *Sensitivity*) (TPR) je miera systému správne klasifikovať prvky z triedy A.
$$\mathrm{TPR} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FN}} \tag{3.18}$$
- **Špecifita** - (angl. *Specificity*) (TNR) je miera systému správne klasifikovať prvky, ktoré nepatria do triedy A, do iných tried ako do A.
$$\mathrm{TNR} = \frac{\mathrm{TN}}{\mathrm{TN} + \mathrm{FP}} \tag{3.19}$$
- **Zhodnosť** - (angl. *Precision*) (PPV) je miera správnosti pri klasifikácii prvku do triedy A.
$$\mathrm{PPV} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FP}} \tag{3.20}$$
- **Presnosť** - (angl. *Accuracy*) (ACC) je miera správnej klasifikácie všetkých prvkov.
$$\mathrm{ACC} = \frac{\mathrm{TP} + \mathrm{TN}}{\mathrm{TP} + \mathrm{TN} + \mathrm{FP} + \mathrm{FN}} \tag{3.21}$$
- **F1 skóre** - (angl. *F1 score*) (F_1) kombinuje metriku zhodnosti so senzitivitou.
$$\mathrm{F}_1 = \frac{2 * \mathrm{TP}}{2 * \mathrm{TP} + \mathrm{FP} + \mathrm{FN}} \tag{3.22}$$
- **Mathewov korelačný koeficient** - (angl. *Matthews correlation coefficient*) (MCC) dosahuje hodnôt v intervale $MCC \in [-1, 1]$. Hodnota $MCC = -1$ indikuje úplne zlý klasifikátor a $MCC = 1$ úplne správny klasifikátor.
$$\mathrm{MCC} = \frac{\mathrm{TP} * \mathrm{TN} - \mathrm{FP} * \mathrm{FN}}{\sqrt{(\mathrm{TP} + \mathrm{FP})(\mathrm{TP} + \mathrm{FN})(\mathrm{TN} + \mathrm{FP})(\mathrm{TN} + \mathrm{FN})}} \tag{3.23}$$

- **Matica zámeny** - (angl. *Confusion matrix*) je matica, ktorá obsahuje v riadkoch skutočnú hodnotu predpovedaného prvku a v stĺpcoch predpovede klasifikátora. V práci budem uvádzať percentuálne hodnoty jednotlivých položiek matice. Príklad matice zámen z pohľadu triedy D vidíme v tabuľke 3.1.

<table>
<tr><th>trieda</th><th>A</th><th>B</th><th>C</th><th>D</th><th>E</th></tr>
<tr><td>A</td><td colspan="3" rowspan="3">Správne negatívne (TN)</td><td rowspan="3">Falošne pozitívne (FP)</td><td rowspan="3">Správne negatívne (TN)</td></tr>
<tr><td>B</td></tr>
<tr><td>C</td></tr>
<tr><td>D</td><td colspan="3">Falošne negatívne (FN)</td><td>Správne pozitívne (TP)</td><td>Falošne negatívne (FN)</td></tr>
<tr><td>E</td><td colspan="3">Správne negatívne (TN)</td><td>Falošne pozitívne (FP)</td><td>Správne negatívne (TN)</td></tr>
</table>

Tabuľka 3.1: Matica zámeny z pohľadu triedy D.

Kapitola 4

Klasifikácia proteínov

Klasifikácia proteínov sa používa v mnohých oblastiach medicíny. Dôležité miesto má napríklad pri vývoji liekov a vakcín [35]. Na začiatku vývoja sekvencovania proteínov, sa proteíny klasifikovali manuálne pomocou biologických experimentov. To bolo drahé aj časovo náročné a preto sa v posledných rokoch kládol dôraz na automatizáciu tohto procesu. Automatizácia sa týka hlavne počítačových nástrojov a algoritmov. Jeden z najznámejších spôsobov klasifikácie bol založený na zarovnávaní aminokyselinových sekvencií. V súčastnosti sa však do popredia dostávajú algoritmy bez zarovnávania kvôli ich rýchlosti a efektivite. Jedne z najznámejších metód, ktoré nepotrebujú zarovnanie sú metódy založené na strojovom učení. Dĺžka proteínov je však variabilná a preto ak chceme použiť metódy strojového učenia, potrebujeme nájsť taký vektor príznakov, ktorý bude mať fixnú dĺžku bez straty dôležitej informácie. Jednotlivé kroky testovacej a trénovacej fázy sú zobrazené na obrázku 4.1.

V tejto kapitole sú popísané spôsoby reprezentácie proteínov, výber príznakov a ich redukcia. Záver kapitoly je venovaný už existujúcim nástrojom na klasifikáciu proteínov.

4.1 Reprezentácia proteínov

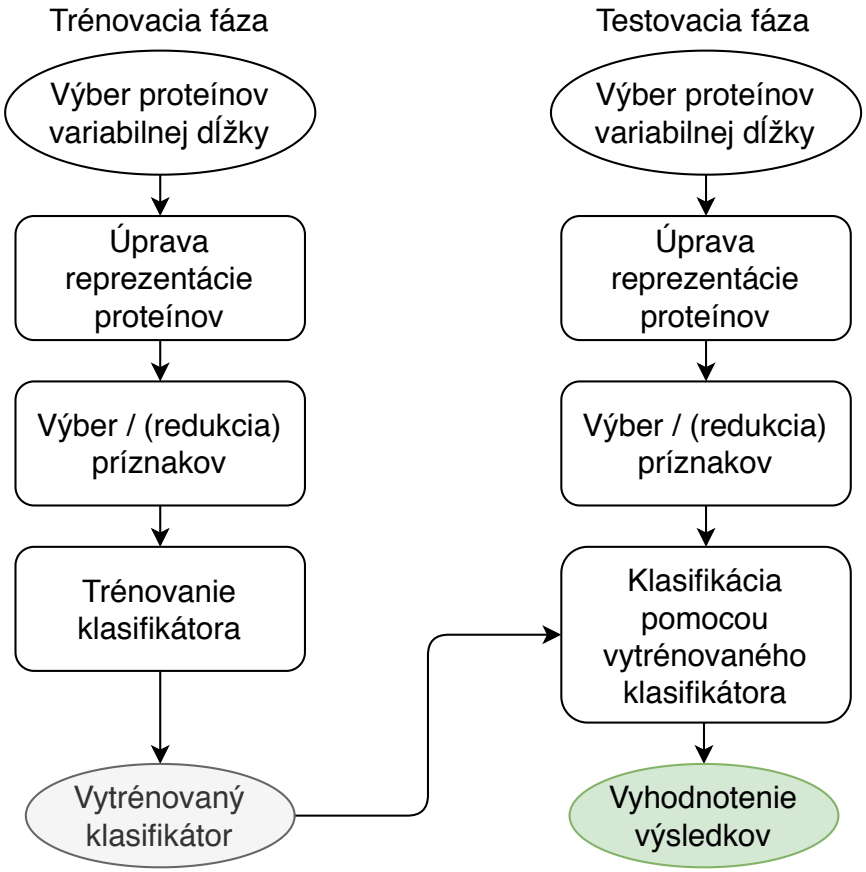
Reprezentácia proteínov môže byť buď sekvenčná alebo maticová. Medzi sekvenčné patrí aminokyselinová sekvencia (AKS) a kódovanie aminokyselín (KA). Medzi maticové pozične-špecifická skórovacia matica (PSSM), reprezentácia substitučnou maticou (RSM) alebo matica fyzikálno-chemických vlastností (FCV). Existujú taktiež spôsoby reprezentácie, ktoré využívajú terciárnu štruktúru. Jednotlivé spôsoby reprezentácie proteínov sú popísané v nasledujúcich podkapitolách.

4.1.1 Aminokyselinová sekvencia

Aminokyselinová sekvencia (AKS) je najrozšírenejšia reprezentácia proteínov. Proteín P sa môže zapísať v tvare:

$$P = (p_1, p_2, ..., p_n); \quad p_i \in \{A, C, D, E, F, G, H, I, K, L, M, N, P, Q, R, S, T, V, W, Y\},$$

kde n je počet aminokyselín daného proteínu a $A, C, D, E, ..., Y$ sú jedno-znakové skratky aminokyselín prezentované v sekcii 2.2. Táto reprezentácia je ekvivalentná k primárnej štruktúre proteínu.



Obr. 4.1: Kroky trénovacej a testovacej fázy klasifikácie proteínov.

4.1.2 Kódovanie aminokyselín

Kódovanie aminokyselín (KA) je reprezentácia, ktorá zoskupuje jednotlivé aminokyseliny do celkov, ktorým priradí špeciálny znak. Takto sa zredukuje kardinalita znakov v aminokyselinách a zrýchli sa trénovanie klasifikátorov. Rozdelenie aminokyselín je založené na vysokej evolučnej podobnosti medzi nimi. Často sa stáva, že to má za následok zlepšenie výsledkov a výrazné zrýchlenie klasifikácie. Jedno z možných kódovaní predstavené v článku [28] je:

1. $\mathbf{e}_1 = \{\text{H, R, K}\}$
2. $\mathbf{e}_2 = \{\text{D, E, N, Q}\}$
3. $\mathbf{e}_3 = \{\text{C}\}$
4. $\mathbf{e}_4 = \{\text{S, T, P, A, G}\}$
5. $\mathbf{e}_5 = \{\text{M, I, L, V}\}$
6. $\mathbf{e}_6 = \{\text{F, Y, W}\}$

Tento spôsob reprezentácie aminokyselín je veľmi podobný predchádzajúcej reprezentácii, s tým, že každý symbol je nahradený symbolom skupiny, do ktorej je daná aminokyselina zaradená. Aminokyselinová sekvencia *'FYHREK'* sa pomocou tohto kódovania prepíše na $'e_6 e_6 e_1 e_1 e_2 e_1$ '.

4.1.3 Pozične-špecifická skórovacia matica

Pozične-špecifická skórovacia matica (PSSM) [14] sa tvorí zo skupiny sekvencií proteínov, ktoré boli pred tým zarovnané či už podľa štrukturálnej alebo sekvenčnej podobnosti. PSSM berie do úvahy štyri základné parametre: pozíciu (angl. *position*), sondu (angl. *probe*), profil (angl. *profile*) a konsenzus (angl. *consensus*). Pozícia je index každej aminokyseliny v sekvencii po viacnásobnom zarovnaní, sonda je skupina sekvencií ktoré sú zarovnané na základe sekvenčnej alebo štrukturálnej podobnosti, profil je matica s dvadsiatimi stĺpcami (pre každú aminokyselinu jeden) a konsenzus je sekvencia aminokyselín, ktorá je najpodobnejšia všetkým zarovnaným sekvenciám. PSSM reprezentácia proteínu s dĺžkou N je matica $N \times 20$ kde PSSM(i, j) sa vypočíta pomocou vzťahu:

$$\text{PSSM}(i,j) = \sum_{k=1}^{20} v(i,k) * D(j,k); \quad i \in \{1,...,N\},\ j \in \{1,...,20\}, \tag{4.1}$$

kde $v(i,k)$ je pomer počtu k-tej aminokyseliny na pozícii i jednotlivých sekvencií v sonde a celkového počtu sekvencií. $D(j,k)$ je hodnota Dayhoffovej mutačnej (substitučnej) matice predstavenej v článku [11] medzi j-tou a k-tou aminokyselinou. Najznámejšia substitučná matica je BLOSUM62 a je zobrazená v tabuľke 4.1.

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

Tabuľka 4.1: Substitučná matica BLOSUM62 prevzatá z článku [40].

Po skonštruovaní PSSM matice z nej môžeme vyčítať, ako veľmi sú jednotlivé aminokyseliny zakonzervované. To znamená, ako daná aminokyselina počas vývoja podliehala mutácii. Vysoké číslo v PSSM znamená vysokú konzerváciu aminokyseliny na danej pozícii. Nízke číslo naopak nízku konzerváciu.

4.1.4 Reprezentácia substitučnou maticou

Reprezentácia substitučnou maticou (RSM) prvý krát predstavená v článku [57] je matica $N \times 20$, ktorá sa pre proteín $P = (p_1, p_2, ..., p_N)$ vytvorí pomocou nasledujúceho vzťahu:

$$\text{RSM}(i, j) = M(p_i, j), \quad i \in \{1, ..., N\},\ j \in \{1, ..., 20\}, \tag{4.2}$$

kde $M(p, j)$ je 20×20 matica, ktorej elementy predstavujú pravdepodobnosť mutácie aminokyseliny p na aminokyselinu j.

4.1.5 Matica fyzikálno-chemických vlastností

Matica fyzikálno-chemických vlastností (FCV) [33] sa tvorí pre fyzikálno-chemickú vlastnosť d a daný proteín $P = (p_1, p_2, ..., p_N)$. Matica $FCV^d(i, j) \in \mathbb{R}^{N \times N}$ vznikne sčítaním hodnoty fyzikálno-chemickej vlastnosti d aminokyseliny na pozícii i a j proteínu, teda:

$$\text{FCV}^d(i, j) = index(p_i, d) + index(p_j, d), \quad i, j \in \{1, ..., N\}, \tag{4.3}$$

kde $index(x, d)$ je hodnota vlasntosti d aminokyseliny x.

4.2 Výber vektora príznakov

Ďalšou dôležitou činnosťou pri klasifikácii proteínov je výber vektora príznakov z reprezentácie uvedenej vyššie. Vektor príznakov nám slúži ako deskriptor proteínu, ktorý je vstupom algoritmov strojového učenia. Niektoré zo základných prístupov výberu vektora príznakov sú: aminokyselinová kompozícia (AK), rozdelená aminokyselinová kompozícia (RAK), pseudo-aminokyselinová kompozícia (PAK), dvojice (D2), vzdialené páry (VP), AAIndexLoc (AA), globálne kódovanie (GE), fyzikálno-chemické dvojice (PCD2), n-tice (NT1, NT2) a pseudo-PSSM (PP).

4.2.1 Aminokyselinová kompozícia

Aminokyselinová kompozícia (AK) je jeden z najjednoduchších spôsobov výberu príznakov. Je založený na sčítaní výskytov jednotlivých aminokyselín a určenie jej pravdepodobnosti výskytu. Výpočet pre proteín p o dĺžke N aminokyselín sa zapíše nasledovne:

$$\text{AK}(x) = \frac{c(x)}{N}, \quad x \in \{1, ..., 20\}, \tag{4.4}$$

kde $c(x)$ značí počet výskytov danej aminokyseliny v proteíne p. Výsledný vektor má 20 hodnôt v intervale $[0, 1]$. Táto metóda sa opiera o fakt, že aminokyselinová kompozícia reflektuje fyzikálno-chemické vlastnosti. Hlavným problémom tejto metódy je, že zanedbáva poradie aminokyselín v sekvencii. Upravená verzia tohto spôsobu výberu príznakov, ktorá pracuje s poradím aminokyselín sa volá pseudo-aminokyselinová kompozícia.

4.2.2 Rozdelená aminokyselinová kompozícia

Rozdelená aminokyselinová kompozícia (RAK) [24], spočíva v rozdelení sekvencie na tri časti. Prvá časť sa skladá z 25 aminokyselín z N-konca aminokyselinovej sekvencie, druhá z 25 aminokyselín z C-konca aminokyselinovej sekvencie a tretia časť z aminokyselín nachádzajúcich sa medzi prvými dvoma časťami. Takto vzniknú tri samostatné sekvencie, z ktorých sa následne spočíta aminokyselinová kompozícia. Dokopy má vstupný vektor príznakov 60 hodnôt.

4.2.3 Pseudo-aminokyselinová kompozícia

Pseudo-aminokyselinová kompozícia (PAK) [58] je kompozícia, ktorá sa skladá z normálnej aminokyselinovej kompozície a autokovariantných premenných. Výpočet autokovariantných premenných má dva kroky. Prvý krok je vybranie q fyzikálno-chemických vlastností a ich následná normalizácia pomocou vzťahu 4.5:

$$h_j(R_i) = \frac{h_j^0(R_i) - h_j^0}{SD(h_j^0)}, \tag{4.5}$$

kde $h_j^0(R_i)$ je pôvodná hodnota j-tej fyzikálno-chemickej vlastnosti pre aminokyselinu R_i, h_j^0 je priemerná hodnota aminokyselín j-tej fyzikálno-chemickej vlastnosti a $SD(h_j^0)$ je smerodajná odchýlka. Následne sa použije vzťah 4.6 na vypočítanie konkrétnych autokovariantných premenných:

$$AC_{m,j} = \frac{1}{n-m} \sum_{i=1}^{n-m} \left(P_{i,j} - \frac{1}{n} \sum_{i=1}^{n} P_{i,j} \right) \left(P_{(i+m),j} - \frac{1}{n} \sum_{i=1}^{n} P_{i,j} \right), \tag{4.6}$$

kde $m \in 1, ..., M$ je vzdialenosť medzi dvoma aminokyselinami v proteíne (v článku z ktorého čerpám použili hodnotu $M = 13$, no dá sa použiť ľubovoľná hodnota, menšia ako dĺžka najkratšej aminokyselinovej sekvencie z dátovej sady), j reprezentuje j-tú fyzikálno-chemickú vlastnosť, i je pozícia aminokyseliny v proteíne P, n je dĺžka sekvencie. Z toho vychádza, že počet nových príznakov vo výslednom vektore narastie o $M * q$. Výsledný vektor príznakov sa vypočíta pomocou vzťahu 4.7:

$$\text{PAK}(k) = \begin{cases} \frac{f_k}{\sum_{i=1}^{20} f_i + w \sum_{j=1}^{M*q} x_j}, & ak \;\; k \in \{1, ..., 20\}, \\ \frac{wp_{(k-20)}}{\sum_{i=1}^{20} f_i + w \sum_{j=1}^{M*q} x_j}, & ak \;\; k \in \{21, ..., 21 + M * q\}, \end{cases} \tag{4.7}$$

kde f_i je počet výskytov i-tej aminokyseliny v proteíne P, x_j je j-ta autokovariantná premenná a w je váhový faktor. V článku z ktorého čerpám bol použitý $w = 0{,}05$, ktorý je veľmi používaný kvôli tomu, že je to jedna dvadsatina z čísla 1, pričom aminokyselín je 20. Metóda je však stavaná všeobecne a dajú sa v nej využiť rôzne hodnoty váhového faktoru, ktoré sa zisťujú experimentálne pre konkrétnu dátovú sadu.

4.2.4 Dvojice

Dvojice (D2) sa zakladajú na podobnom princípe ako aminokyselinová kompozícia. Rozdiel je v tom, že namiesto jednotlivých aminokyselín sa berú do úvahy dvojice:

$$\text{D2}(x) = \frac{c(i,j)}{N}; \quad i, j \in \{1, ..., 20\}, \; x = j + 20 * (i - 1), \tag{4.8}$$

kde $c(i,j)$ je počet výskytov všetkých dvojíc aminokyselín i a j, ktoré sa nachádzajú bezprostredne v proteíne za sebou. N je počet všetkých aminokyselín v proteíne. Takýto vektor príznakov má 400 hodnôt.

4.2.5 Vzdialené páry

Vzdialené páry (VP) [32] na výber vektoru príznakov z proteínov sú vo veľkej miere inšpirované Yuanovými Markovými reťazcami [15]. Najskôr sa vyberie fyzikálno-chemická

vlastnosť d, ktorá sa skombinuje s nenulovými vzdialenými pármi. Parameter m značí vzdialenosť medzi aminokyselinami. Hodnoty $m \leq 3$ sú dostatočné na reprezentáciu proteínu. Matematicky to vieme zapísať nasledujúcim spôsobom:

$$\mathrm{VP}_m^d(k) = \frac{1}{N-m} \sum_{n=1}^{N-m} H_{i,j}(n, n+m, d) + H_{j,i}(n+m, n, d), \tag{4.9}$$

$$i, j \in \{1, ..., 20\}, \quad k = j + 20 * (i-1) + (m-1) * 400,$$

kde i a j reprezentujú 20 rôznych aminokyselín. Funkciu $H_{x,y}$ vieme vyjadriť takto:

$$H_{x,y}(a, b, d) = \begin{cases} index(x, d), & ak \;\; P_a = x \wedge P_b = y, \\ 0, & inak. \end{cases} \tag{4.10}$$

Funkcia $index(a, d)$ vracia hodnotu a-tej aminokyseliny proteínu P (teda P_a), pre fyzikálno-chemickú vlastnosť d. Výsledný vektor tejto metódy pre $m = 3$ má 1200 hodnôt.

4.2.6 AAIndexLoc

AAIndexLoc (AA) [49] je zložený z aminokyselinovej kompozície, teda zlomku počtu výskytov jednotlivých aminokyselín ku celkovému počtu aminokyselín v reťazci (20 hodnôt), váhovej aminokyselinovej kompozícii vytvorenej vynásobením hodnôt aminokyselinovej kompozície funkciou $index(i, d)$, kde d je vybraná fyzikálno-chemická vlastnosť (20 hodnôt) a päť-úrovňovej dipeptidovej kompozície. Tá vznikne tak, že algoritmom k-means rozdelíme aminokyseliny na päť skupín na základe fyzikálno-chemickej vlastnosti d, vytvoria sa všetky možné dvojice (25 dvojíc) a počíta sa pomer výskytu týchto dvojíc ku všetkým dvojiciam proteínu (25 hodnôt). Celkovo teda deskriptor obsahuje 65 hodnôt.

4.2.7 Globálne kódovanie

Globálne kódovanie (GE) [26] je metóda, ktorá delí aminokyseliny do šiestich základných skupín: $A_1 = \{A, V, L, I, M, C\}$, $A_2 = \{F, W, Y, H\}$, $A_3 = \{S, T, N, Q\}$, $A_4 = \{K, R\}$, $A_5 = \{D, E\}$ a $A_6 = \{G, P\}$. Tieto skupiny sa rozdelia do všetkých kombinácií dvoch množín po troch skupinách (napr.: $\{A_1, A_2, A_3\}$, $\{A_4, A_5, A_6\}$). Každá aminokyselina v proteíne sa potom prepíše na reprezentáciu 0 a 1 na základe príslušnosti do množín. Ak aminokyselina patrí do prvej množiny, nahradíme ju číslom 0, a ak patrí do druhej množiny, tak číslom 1. Existuje 10 rôznych kombinácii na vytvorenie takýchto množín. Prepísaním proteínovej sekvencie na 0 a 1 teda dostaneme 10 rôznych reťazcov. GE deskriptor sa skladá z frekvencií 0 a 1 v každom reťazci (20 hodnôt) a z frekvencií prechodov medzi 0 a 1 v danom reťazci (20 hodnôt).

4.2.8 Fyzikálno-chemické dvojice

Fyzikálno-chemické dvojice (PCD2) [34] je prístup ktorý kombinuje hodnoty fyzikálno-chemickej vlastnosti d aminokyselín s metódou dvojíc (D2). Sú definované nasledujúcim spôsobom:

$$\mathrm{PCD2}^d(k) = \left(\frac{c(i,j) * index(i,d)}{N-1}, \frac{c(i,j) * index(j,d)}{N-1} \right), \tag{4.11}$$

$$i, j \in \{1, 2, ..., 20\}, \quad k = j + 20 * (i-1),$$

kde i a j je 20 aminokyselín, N je dĺžka proteínu, funkcia $index(i, d)$ vracia hodnotu vlastnosti d aminokyseliny i a funkcia $c(i, j)$ počíta počet výskytov dvojice i a j. Výsledný PCD2 deskriptor má 400 dvojíc a teda 800 hodnôt.

4.2.9 N-tice

N-tica (NT1) [31] je metóda podobná štandardnej metóde dvojíc (D2) s tým rozdielom, že sa berú rôzne aminokyselinové abecedy. Pre túto metódu bolo vytvorených 5 abecied. Pre každú z nich sa vytvorí samostatný klasifikátor. Abecedy sú tieto:

- $A_1 = \{G, I, V, F, Y, W, A, L, M, E, Q, R, K, P, N, D, H, S, T, C\}$,
- $A_2 = \{\{L, V, I, M\}, C, A, G, S, T, P, \{F, Y\}, W, E, D, N, Q, \{K, R\}, H\}$,
- $A_3 = \{\{L, V, I, M, C\}, \{A, G\}, \{S, T\}, P, \{F, Y, W\}, \{E, D, N, Q\}, \{K, R\}, H\}$,
- $A_4 = \{\{L, V, I, M, C\}, \{A, S, G, T, P\}, \{F, Y, W\}, \{E, D, N, Q\}, \{K, R, H\}\}$,
- $A_5 = \{\{L, V, I, M, C\}, \{A, S, G, T, P\}, \{F, Y, W\}, \{E, D, N, Q, K, R, H\}\}$.

Každý proteín sa najskôr preloží do týchto piatich abecied. Následne sa na proteíny preložené do abecied A_1, A_2 použije metóda dvojíc (D2) a na proteíny preložené do abecied A_3, A_4 a A_5 podobná metóda ale pre trojice. Výsledok všetkých deskriptorov je vynásobený váhou pre danú abecedu a následne sčítaný do jednej hodnoty. Pre A_1, A_2 a A_3 je váha 1, pre A_4 váha 0,5 a pre A_5 váha 0,25. Toto prerozdelenie je založené na matici mutácií PAM [11], ktorá štatisticky popisuje pravdepodobnosti substitúcie jednej aminokyseliny druhou.

Druhá varianta n-tíc (NT2) [20] na rozdiel od (NT1) používa základnú abecedu dvadsiatich aminokyselín. Sekvencia sa postupne číta po $1, 2, 3, ..., m$ aminokyselinách a počíta sa ich kompozícia. Metóda tak využíva znalosť, že pravdepodobnosť nájdenia identickej sekvencie po sebe idúcich aminokyselín je nízka. Pri čítaní po jednom znaku nám vo výslednom vektore príznakov vyjde 20 hodnôt. Pri čítaní po dvoch znakoch 20^2 hodnôt a tak podobne až po čítanie po m znakoch s 20^m hodnotami. Výsledná veľkosť vektora je teda:

$$N = 20 + 20^2 + ... + 20^m = \frac{20 - 20^{m-1}}{1 - 20}. \tag{4.12}$$

V článku [20] rátajú s $m = 3$, pre ktoré je $N = 8420$.

4.2.10 Pseudo-PSSM

Pseudo-PSSM (PP) [21] je jedna z najznámejších metód založených na maticovej reprezentácii na výber príznakov proteínov. Využíva reprezentáciu pozične-špecifickej skórovacej matice popísanej v sekcii 4.1.3. Zachováva informáciu o aminokyselinovej sekvencii tak, že berie do úvahy pseudo-aminokyselinovú kompozíciu:

$$\mathrm{PP}(k) = \begin{cases} \frac{1}{N} \sum_{i=1}^{N} E(i, j), & ak \;\; k \in \{1, ..., 20\}, \\ \frac{1}{N-lag} \sum_{i=1}^{N-lag} \left[E(i, j) - E(i + lag, j)\right]^2, & \\ \quad ak \;\; j \in \{1, ..., 20\}, \; lag \in \{1, ..., 15\}, k = 20 + j + 20 * (lag - 1), \end{cases} \tag{4.13}$$

kde k je pomocný index na prechádzanie po bunkách matice $PSSM$, lag je vzdialenosť medzi dvoma reziduami a N je dĺžka sekvencie proteínu. $E \in R^{N\times 20}$ je normalizovaná verzia matice $PSSM$, definovaná nasledovne:

$$E(i,j) = \frac{PSSM(i,j) - (1/20)\sum_{v=1}^{20} PSSM(i,v)}{\sqrt{(1/20)\sum_{u=1}^{20}\left(PSSM(i,u) - (1/20)\sum_{v=1}^{20} PSSM(i,v)\right)^2}}, \tag{4.14}$$

$$i \in \{1,2,...,N,\}, \quad j \in \{1,...,20\}.$$

Výsledný vektor príznakov má 320 hodnôt, čo vyplýva aj zo vzťahu 4.13.

4.3 Redukcia príznakov

Často sa stáva, že algoritmus na výber príznakov vyprodukuje vektor s veľkým množstvom hodnôt. To vedie ku výraznému predĺženiu trénovacieho procesu ako aj ku zníženiu presnosti klasifikácie. Veľa z hodnôt vo vektore je vzhľadom na celkový výsledok klasifikácie irelevantných a tak sa vyvinuli algoritmy, ktoré tieto redundantné príznaky redukujú. Príkladom takýchto algoritmov je napríklad analýza hlavných komponent - (angl. *Principal Component Analysis*, PCA).

4.3.1 Analýza hlavných komponent

Cieľom algoritmu analýzy hlavných komponent [4] (tiež známa ako *Karhunen-Loeveová transformácia*), je znížiť dimenzionalitu ortogonálnou transformáciou pôvodne d-dimenzionálnych dátových objektov do nového k-dimenzionálneho priestoru, pričom platí že $k \leq d$. Novo vzniknutý priestor je definovaný takzvanými *hlavnými komponentami*, ktoré sú lineárne nezávislé a ktorých chyba projekcie by mala byť čo najmenšia.

Nech $\Gamma_1, \Gamma_2 ... \Gamma_M$ je trénovacia množina proteínov, na ktorých bola použitá niektorá z metód výberu príznakov, a nech každý vektor príznakov Γ_m obsahuje d príznakov. M je počet proteínov v použitej trénovacej množine. *Priemerný tvar* Ψ vektoru príznakov takto definovanej trénovacej množiny je možné vypočítať nasledujúcim spôsobom:

$$\Psi = \frac{1}{M}\sum_{i=1}^{M}\Gamma_i. \tag{4.15}$$

Každý vektor príznakov Γ_i v tejto množine sa líši od priemerného vektoru o vektor Φ_i, ktorý je vyjadrený nasledujúcim vzorcom:

$$\Phi_i = \Gamma_i - \Psi. \tag{4.16}$$

Využitím algoritmu analýzy hlavných komponent je možné nájsť vektory, ktoré najviac prispievajú variácii jednotlivých proteínov. Tieto vektory u_j vzniknú lineárnou kombináciou vektorov príznakov $\Gamma_1, \Gamma_2 ... \Gamma_M$, a sú vlastnými vektormi matice kovariancie C. Tá značí závislosť medzi jednotlivými prvkami Φ_i. Kovariančná matica C je definovaná vzťahom:

$$C = \frac{1}{M}\sum_{i=1}^{M}\Phi_i\Phi_i^T = AA^T, \tag{4.17}$$

kde matica $A = [\Phi_1, \Phi_2, ...\Phi_M]$ s rozmermi $d \times M$, a A^T označuje transponovaný vektor A. Následne vypočítame vlastné vektory a ku nim odpovedajúce vlastné hodnoty tejto matice. Vlastné hodnoty usporiadame vzostupne podla vlastných hodnôt, čím zistíme prínos každého vlastného vektoru. Prvých k vlastných vektorov potom reprezentuje dimenzie nového vektorového priestoru.

4.4 Existujúce klasifikačné nástroje

Klasifikácia proteínov je oblasť, do ktorej sa investuje značné množstvo zdrojov. V súčasnosti existuje niekoľko nástrojov na klasifikáciu proteínov. Táto podkapitola zhŕňa základné informácie o vybraných nástrojoch. Jednotlivé nástroje sú v texte usporiadané vzostupne podľa dátumu ich publikácie.

4.4.1 ProCANS

ProCANS [55] (*Protein Classification Artificial Neural System*) je nástroj, ktorý beží na superpočítači Cray a slúži na rýchlu klasifikáciu superrodín. Systém využíva funkciu založenú na n-ticiach 4.2.9, pomocou ktorej kóduje sekvencie proteínov. Na reprezentáciu proteínu využíva obyčajnú aminokyselinovú sekvenciu 4.1.1, ako aj kódovanie aminokyselín 4.1.2. Systém bol trénovaný a testovaný na 620 superrodinách, pričom dosiahol prediktívnu presnosť 90%. Rýchla a presná klasifikácia superrodín by sa dala použiť na organizáciu databáz proteínových sekvencií a na rozpoznávanie génov vo veľkých projektoch sekvencovania. Používa algoritmus neurónových sietí s 200 skrytými neurónmi, pričom mieru učenia (*learning rate*) má nastavenú medzi hodnotou 0,2 a 0,8.

4.4.2 GPCRTree

GPCRTree [30] klasifikuje funkciu receptorov spojených s G proteínmi (angl. *G protein-coupled receptors - GPCR*), ktoré majú dôležité fyziologické úlohy nosiace extracelulárne signály do intraceluláryných odpovedí. Klasifikácia GPCR superrodiny je jedna z najväčších výziev v bioinformatike, vďaka veľkej diverzite jednotlivých sekvencií. Jeden z hlavných dôvodov prečo sa snažíme presne klasifikovať práve túto rodinu je fakt, že približne 50% všetkých predávaných liekov sa zameriava práve na ňu. GPCRTree je verejne prístupný internetový server, ktorý implementuje algoritmus klasifikácie GPCR na úrovni foldov, superrodín a rodín. Využíva selektívny klasifikátor zhora-nadol, ktorý priraďuje sekvencie v rámci hierarchie GPCR. Na každej úrovni je natrénovaný buď SVM klasifikátor 3.4 alebo HMM (*Hidden Markov Model*). Na reprezentáciu proteínov používa aminokyselinovú sekvenciu 4.1.1 a na extrakciu príznakov aminokyselinovú kompozíciu 4.2.1 v kombinácii s dvojicami 4.2.4. Ich výsledná presnosť pri foldoch dosahuje 97%, pri superrodinách 84% a pri rodinách 75%.

4.4.3 LipocalinPred

LipocalinPred je webový nástroj založený na SVM algoritme 3.4, ktorý slúži na identifikáciu proteínových sekvencií lipocalinu. Predikuje lipocaliny pomocou viacerých vytrénovaných klasifikátorov. Tie sú založené na rôznych metódach extrakcie príznakov. Používajú aminokyselinovú kompozíciu 4.2.1, dvojice 4.2.4, pssm profil podobný metóde popísanej v 4.2.10 a kompozícii založenej na sekundárnej štruktúre. Výsledné modely boli trénované na vektoroch príznakov vygenerovaných zo sekvencie aminokyselín. Webový server umožňuje uží-

vateľom skenovať množinu sekvencií, ktoré sú príbuzné lipocalinu. Nástroj bol publikovaný v článku [42] a dosahuje presnosť väčšiu ako 90%.

4.4.4 SECLAF

SECLAF je webový nástroj používajúci hlboké neurónové siete 3.2 na klasifikáciu hierarchickej biologickej sekvencie. Využíva podobný princíp hierarchického klasifikovania ako nástroj GPCRTree. Podľa článku [48], to bol v roku 2018 najpresnejší proteínový klasifikátor. Je implementovaný v jazyku python a používa knižnicu Tensorflow na neurónové siete, ktorá bola vytvorená spoločnosťou Google. Jedne z mnohých spôsobov extrakcie vektoru príznakov je aminokyselinová kompozícia 4.2.1 a globálne kódovanie 4.2.7. Pri extrakcii využíva fyzikálno-chemické vlastnosti aminokyselín. Podobným spôsobom vie nástroj kódovať aj sekvenciu DNA. Pri klasifikácii 698 rodín z databázy UniProt sa jeho presnosť dostala na približne 96%.

4.4.5 Zhrnutie nástrojov

Z nástrojov vidíme, že najčastejšie používanými metódami na extrakciu príznakov sú práve dvojice a aminokyselinová sekvencia v kombinácii s niektorými sofistikovanejšími metódami. Najúspešnejším nástrojom je SECLAF, ktorý používa neurónové siete. Z hľadiska škálovateľnosti vyzerá najschopnejšie hierarchický prístup klasifikácie. Medzi prvými nástrojmi ktoré tento prístup využili je GPCRTree, ktorý sa vďaka tomu dá rozšíriť na ľubovoľné množstvo rodín bez výraznejšej zmeny časovej náročnosti trénovania. Trénovanie klasifikátorov v hierarchickom usporiadaní je veľmi jednoducho paralelizovateľné, keďže jednotlivé klasifikátory nie sú na seba závislé. V práci sa budem snažiť skombinovať osvedčené prístupy na dosiahnutie čo najlepších výsledkov presnosti a možnosti rozšírenia.

Kapitola 5

Návrh a implementácia

Práca obsahuje dva hlavné implementačné systémy. Prvý systém slúži na zmenu reprezentácie aminokyselinových sekvencií, extrakciu príznakov spojenú s fyzikálno-chemickými vlastnosťami aminokyselín, trénovanie klasifikátorov, ich testovanie a celkové vyhodnotenie vybraných metód pomocou metrík na hodnotenie. Implementuje metódy extrakcie príznakov použité v nástrojoch GPCRTree popísanom v sekcii 4.4.2, LipocalinPred popísanom v sekcii 4.4.3 a SECLAF popísanom v sekcii 4.4.4. Druhý program slúži na hierarchickú klasifikáciu väčšieho množstva rodín, ktorý implementuje upravenú verziu hierarchickej klasifikácie z nástroja GPCRTree.

Táto kapitola obsahuje návrh a popis implementácie jednotlivých častí aplikácie na extrakciu príznakov a trénovanie modelov spolu s popisom implementácie hierarchického nástroja. Okrem toho budú uvedené formáty očakávaných vstupných dát ako aj formát výstupu jednotlivých aplikácií.

5.1 Požiadavky implementovaného programu

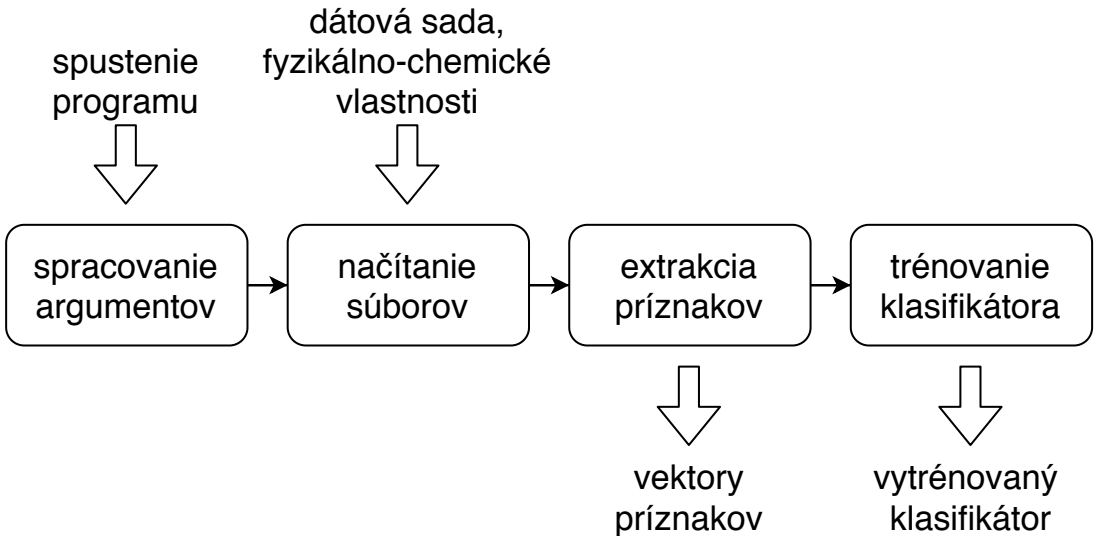
Cieľom diplomovej práce bolo vytvoriť konzolovú aplikáciu, ktorá bude vedieť spracovávať proteíny a sprístupňovať rôzne techniky ich klasifikácie. Základné požiadavky aplikácie sú:

- aplikácia bude vedieť pracovať s proteínmi v známom formáte FASTA
- užívateľ si bude vedieť zvoliť množinu proteínov na klasifikovanie
- aplikácia bude poskytovať viacero možností extrakcie príznakov s dôrazom na efektivitu výpočtu
- extrahované vektory príznakov aplikácia uloží do súboru na to určenému, aby s ním bolo možné ďalej pracovať
- aplikácia bude implementovať viacero metód strojového učenia
- natrénovaný model aplikácia uloží do súboru na to určenému, aby s ním bolo možné ďalej pracovať
- aplikácia poskytne užívateľovi prehľad výsledkov metrík natrénovaného klasifikátora
- aplikácia umožní užívateľovi vyskúšať vytrénovaný klasifikátor
- výstup aplikácie bude vo svetovom jazyku

5.2 Program na extrakciu príznakov a trénovanie modelov

Program na extrakciu príznakov a trénovanie modelov môžeme rozdeliť do štyroch základných blokov. Tie sú názorne ukázané na obrázku 5.1, spolu s ich vstupmi a výstupmi. Celý program bol naimplementovaný v jazyku `python` s verziou `3.5`. Jeden z hlavných dôvodov prečo som si vybral práve jazyk `python` sú jeho knižnice:

- `NumPy` [36] je open-source knižnica ktorá je z veľkej časti naprogramovaná v jazykoch C/C++. Má za sebou širokú komunitu ľudí, ktorí ju udržujú. Podporuje efektívne dátové štruktúry pre veľké multi-dimenzionálne polia a matice, spolu s veľkým množstvom vysoko úrovňových matematických funkcií na prácu s týmito poliami. Vďaka jej efektivite a jednoduchosti použitia sa pri veľkých vektoroch príznakov a pri obsiahlejšej dátovej sade stala pre moju aplikáciu veľmi užitočnou.
- `TensorFlow` [1] je voľne dostupná open-source knižnica, ktorá sa využíva v aplikáciách strojového učenia. Je vyvíjaná spoločnosťou Google a používa sa či už vo výskume ako aj v produkcii. V implementácii som použil najnovšiu verziu `2.0`, ktorá je zabalená v knižnici s názvom `Keras`.
- `Scikit learn` [41] je taktiež voľne dostupná knižnica strojového učenia. Špecializuje sa na viacero klasifikačných, regresných a zhlukovacích algoritmov, vrátane Support Vector Machines a náhodného lesa. Je navrhnutá tak, aby pracovala s vedeckými knižnicami ako `NumPy` a `SciPy`. V implementácii som použil najnovšiu verziu `0.20`.



Obr. 5.1: Základné bloky programu na extrakciu príznakov a trénovanie modelov.

5.2.1 Spracovanie argumentov

Prvý blok sa zameriava na spracovávanie argumentov. Cieľom bolo, aby mal užívateľ čo najväčšiu flexibilitu pri práci s navrhnutou aplikáciou a preto program reaguje na vysoké množstvo argumentov:

- Užívateľ si môže zvoliť vlastnú vstupnú dátovú sadu tak, že do argumentu *-i* zadá zložku so súbormi, kde každý súbor obsahuje proteíny jednej rodiny vo formáte FASTA. Všetky dátové sady použité v práci sú umiestnené na priloženom DVD nosiči.

- Užívateľ si môže zvoliť, prípadne zadefinovať, vlastnú fyzikálno-chemickú vlastnosť, ktorá sa použije v metódach na extrakciu príznakov, ktoré tieto vlastnosti používajú. Formát súboru na definovanie fyzikálno-chemických vlastností je popísaný v nasledujúcej sekcii. Všetky fyzikálno-chemické vlastnosti spomínané v práci sú priložené v súbore `properties.txt`, ktorý sa nachádza pri zdrojových kódoch aplikácie.

- K dispozícii sú rôzne metódy extrakcie príznakov (aminokyselinová kompozícia, rozdelená aminokyselinová kompozícia, vzdialené páry, dvojice, fyzikálno-chemické dvojice a globálne kódovanie), prípadne algoritmy strojového učenia (neurónové siete, Support Vector Machines a náhodný les), ktoré sa budú dať taktiež zvoliť a ľubovoľne kombinovať.

- Užívateľ má k dispozícii škálu parametrov strojového učenia, ktoré sa dajú nastaviť. Pri neurónových sieťach ide o šírku skrytej vrstvy siete a veľkosť dávky, pri Support Vector Machines ide o parametre C (čo definuje cenu nesprávnej klasifikácie prvkov trénovacej dátovej sady) a gamma (ktorá definuje vplyv jedného trénovacieho prvku na výsledný model klasifikátora), pri náhodnom lese je to maximálna hĺbka stromu, minimálny počet prvkov na rozdelenie uzlu a minimálny počet prvkov, ktorý môže obsahovať list stromu.

- Ako posledné má užívateľ možnosť použiť vytrénovaný klasifikátor. Pomocou parametru *-e* dá aplikácii najavo, že má záujem vyskúšať natrénovaný model. Parametrom *-i* zadá zložku so súbormi vo FASTA formáte, v ktorom sú proteíny, ktoré chce klasifikovať. Model sa zadáva v parametri *-c* a aby aplikácia vedela ako má spracovať vstupné proteíny, musí sa uviesť aj odpovedajúca metóda pomocou parametru *-m*.

Tieto argumenty spracováva trieda `IOHandler`, ktorá pracuje s knižnicou `argparse`.

5.2.2 Načítanie súborov

Druhý blok programu sa stará o načítavanie súborov. Implementácia vie načítať dátovú sadu proteínov, ktorá sa nachádza v zložke plnej súborov, kde každý súbor obsahuje proteíny jednej rodiny vo formáte FASTA. Proteín v tomto formáte je reprezentovaný sériou riadkov. Prvý riadok začína znakom nerovnosti (>), za ktorým nasleduje jednoznačný identifikátor knižnice, v ktorom sa daný proteín nachádza, prípadne dodatočné informácie. Zvyšné riadky sú reťazce aminokyselín samotného proteínu reprezentované jedným znakom, pričom šírka každého reťazca je obmedzená na šesťdesiat až osemdesiat aminokyselín. V spomínanej databáze Pfam sú tieto reťazce obmedzené na 60 aminokyselín. Proteín vo FASTA formáte je ukázaný na obrázku 5.2.

Načítavanie súborov má na starosti trieda `FileHandler`, ktorá najskôr prevedie proteíny z FASTA formátu do spojitých reťazcov aminokyselín, a vytvorí pre každú rodinu objekt triedy `Family`. Súbory rodín vo vstupnej zložke sa pred spracovaním abecedne zoradia, čo ovplyvňuje výsledné poradie rodín pri klasifikácií. Pri načítavaní proteínov sa taktiež kladie dôraz na ich validitu. Stáva sa, že v proteínových sekvenciách sa vyskytnú aminokyseliny s jednoznakovým označením *X*, v ktorom ide o ľubovoľnú aminokyselinu (angl. *any amino acid*), prípadne *U*, ktoré reprezentujú už spomínané selenocysteiny. V obidvoch prípadoch

```
>A0A0E0I9P4_ORYNI/7-152
FECLLKLLNFVLTVAGLAMVGYGIYLLVEWMRISIGGGGMPPSPAPPAELLMFGRPMLTA
VALGDGGSFFDKLPKAWFIYLFIGVGAGIFVGSLFGCGGAATRNTCCLCCYAFLVGLLGL
VEAGAAAFIFFDESWKDVIPVDKTEN
```

Obr. 5.2: Príklad proteínu vo formáte FASTA.

tieto aminokyseliny vynechávam kvôli tomu, že metódy reprezentácie proteínov a fyzikálno-chemické vlastnosti rátajú len s dvadsiatimi aminokyselinami. Ďalšou možnosťou by bolo zahodiť celé sekvencie, ktoré obsahujú znaky týchto aminokyselín, no v tom prípade by sme pri niektorých rodinách mohli prísť o značnú časť proteínov.

Druhý súbor, ktorý sa do programu načítava je súbor fyzikálno-chemických vlastností, ktorý sa skladá z minimálne dvoch riadkov. Na prvom riadku sa nachádza zoznam jednoznakových aminokyselín, oddelených čiarkou. Tie nemusia byť nutne v abecednom poradí. Na zvyšných riadkoch sa nachádzajú priamo hodnoty odpovedajúce aminokyselinám z prvého riadku. Príklad takéhoto súboru vidíme na obrázku 5.3. Fyzikálno-chemické vlastnosti prezentované v práci sa nachádzajú v súbore `properties.txt`.

```
A,R,N,D,C,Q,E,G,H,I,L,K,M,F,P,S,T,W,Y,V
1,2,1,4,1,5,2,3,6,2,1,4,1,5,2,5,2,4,1,2
0,1,2,4,5,6,0,0,0,1,2,0,1,2,0,5,0,4,0,6
```

Obr. 5.3: Príklad súboru s dvoma fyzikálno-chemickými vlastnosťami.

5.2.3 Extrakcia príznakov

Tretí blok programu sa stará o extrakciu príznakov. Tento blok je výpočtovo náročný a tak som na jeho realizáciu použil najmä knižnicu `NumPy` [36]. Prvý dôležitý krok extrakcie vektoru príznakov je zvoliť si vhodnú reprezentáciu proteínov. V podkapitole 4.1 som popísal viacero spôsobov, ktoré sa používajú. V mojej implementácii som použil klasickú aminokyselinovú sekvenciu 4.1.1 (AKS) a kódovanie aminokyselín 4.1.2 (KA), ktoré je využité v popisovaných nástrojoch. Z metód na výber príznakov som vybral šesť metód:

- **aminokyselinová kompozícia** 4.2.1 (AK)
- **rozdelená aminokyselinová kompozícia** 4.2.2 (RAK), ktorá potrebuje, aby dĺžka proteínu bola aspoň 50 aminokyselín. V prípade že je proteín kratší, vypíše sa upozornenie pre užívateľa a pokračuje sa ďalším proteínom.
- **dvojice** 4.2.4 (D2)
- **vzdialené páry** 4.2.5 (VP)
- **globálne kódovanie** 4.2.7 (GE)
- **fyzikálno-chemické dvojice** 4.2.8 (PCD2)

Pred extrakciou príznakov sa názvy súborov rodín zoradia alfa-numericky, čo udáva výsledné zoradenie rodín pri trénovaní. Metódy vzdialených párov a fyzikálno-chemických dvojíc vo svojom výpočte využívajú fyzikálno-chemické vlastnosti. V prípade, že by užívateľ neuviedol súbor týchto vlastností, program vypíše chybové hlásenie a ukončí sa. Väčšina použitých metód používa do istej miery početnosť jednotlivých aminokyselín v reťazci. Na to aby som mohol použiť priame indexovanie do poľa v ktorom sa tieto počty výskytov rátali, som musel pole doplniť znakmi abecedy, ktoré nereprezentujú žiadnu aminokyselinu. Pri hľadaní indexu do poľa mi teda stačilo odpočítať z každej aminokyseliny hodnotu 65 (ascii 'A'). Na konci sčítavania som už len vymazal umelo pridané prvky. Výstupom tohto bloku programu je súbor vo formáte `.npz`, ktorý v sebe ukladá polia hodnôt. Vstupné vektory príznakov sú v súbore pomenované ako *inputs* a k nim priradené výstupy ako *targets*. Extrakciu má na starosti trieda `ExtractorHandler`, ktorú volá trieda `Extractor`. Modul extrakcie príznakov je tvorený tak, aby sa dal jednoducho použiť v iných aplikáciách, ktoré potrebujú extrahovať vektory príznakov pomocou vybraných metód. UML návrh triedneho diagramu modulu na extrakciu vektorov príznakov vidíme na obrázku 5.4.

5.2.4 Trénovanie klasifikátorov

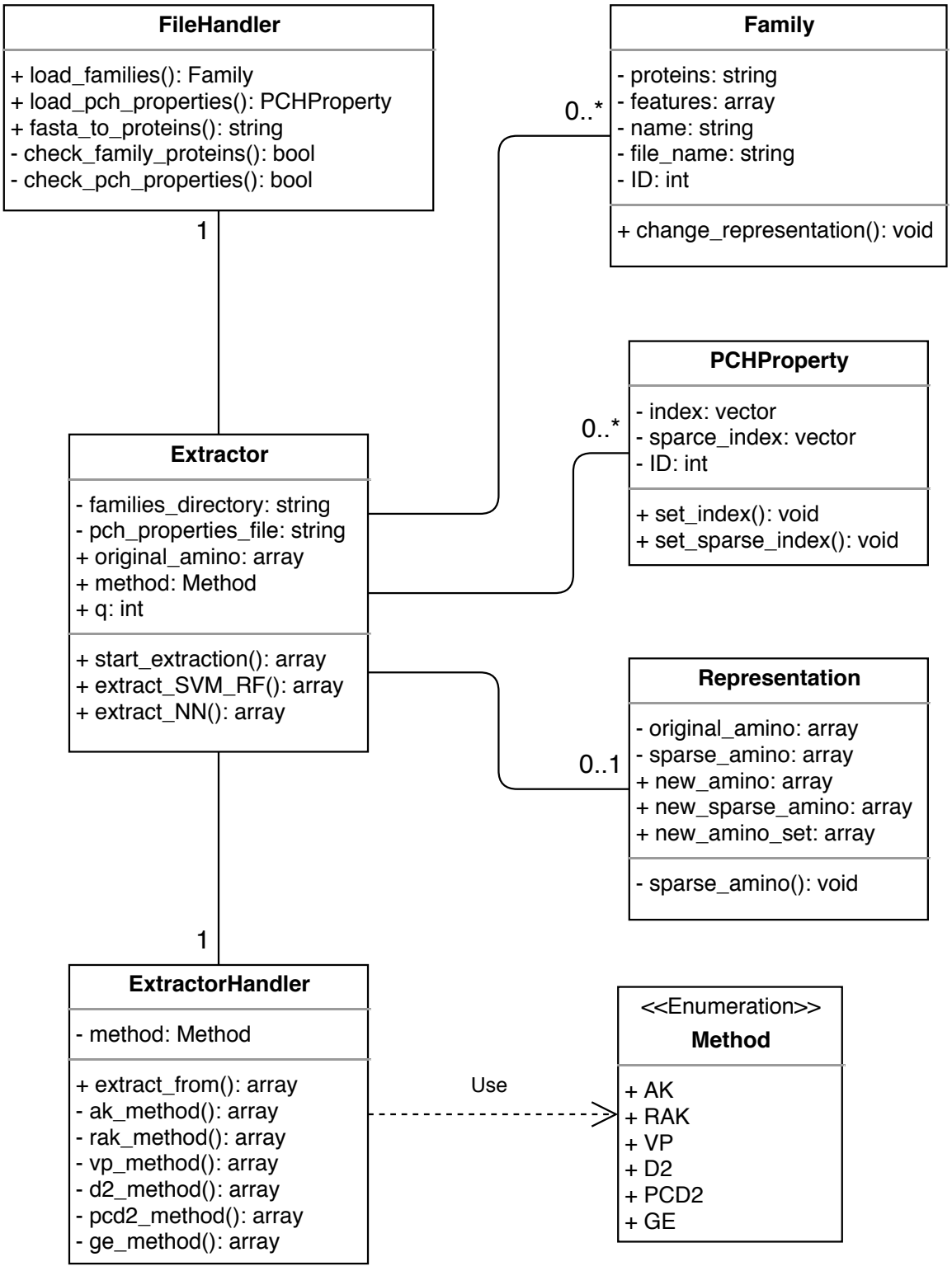
Posledný štvrtý blok programu sa stará o trénovanie klasifikátorov. Vo svojej implementácii využívam najmä knižnice, ktoré sú zamerané na trénovanie algoritmov strojového učenia. Medzi najhlavnejšie patria `TensorFlow` [1] a `Scikit learn` [41]. V práci implementujem tri prezentované algoritmy strojového učenia. Každý obsahuje množinu parametrov, ktoré sa môžu pri trénovaní nastaviť. Parametre, ktoré som v závislosti na metódach extrakcie menil, budem prezentovať v sekcii experimentov. Zároveň sú uvedené v prílohe B.

Pre neurónové siete som zvolil fixný počet dvoch skrytých vrstiev, kde každá mala aktivačnú funkciu s názvom rektifikovaná lineárna jednotka (angl. *rectified linear unit*). Vybral som ju na základe predbežných experimentálnych výsledkov. Výstupná vrstva pracovala s normalizovanou exponenciálnou aktivačnou funkciou (angl. *softmax*), ktorá sa v tejto vrstve využíva najčastejšie, kvôli tomu, že výstup je vo forme pravdepodobnostnej distribúcie. Za optimalizátor trénovania som zvolil rozšírenú variantu *gradientného zostupu* s názvom *Adam* (skratka pre *Adaptive Moment Estimation*), ktorý je v súčastnosti jeden z najpoužívanejších optimalizátorov. Stratová funkcia vo všetkých experimentoch bola krížová entropia (angl. *cross entropy*). Medzi premenlivé parametre, ktoré môže užívateľ meniť patrí veľkosť dávky a šírka skrytej vrstvy neurónovej siete. Modul, ktorý implementuje neurónové siete má názov `neural_network.py`.

Pri algoritme náhodného lesa som fixne určil množstvo rozhodovacích stromov na 100 a konštantu kontrolujúcu náhodnosť pri konštruovaní stromu na 42. Na meranie kvality výberu príznakov do jednotlivých uzlov stromu som na základe predbežných výsledkov použil metriku Giniho koeficientu. Medzi premenlivé parametre, ktoré môže užívateľ meniť patrí maximálna hĺbka stromov, hodnota určujúca minimálny počet prvkov vo validnom liste a v uzle stromu ktorý sa môže rozdeliť. Modul, ktorý implementuje náhodný les má názov `random_forest.py`.

Algoritmus Support Vector Machines má pri všetkých metódach na pevno určenú len jadrovú funkciu. Konkrétne ide o radiálne-bázovú funkciu (angl. *Radial Basis Function*). Implementácia modulu pre Support Vector Machines sa nachádza v súbore `svm.py`.

Súbory na trénovanie klasifikátorov obsahujú aj iné možnosti využitia. Ide o možnosť vytvorenia čiastočne redukovanej dátovej sady, ktorá sa použila v experimentoch, a možnosť vytvorenia dátovej sady slúžiacej na klasifikáciu proteínov do štrukturálnych tried.



Obr. 5.4: Triedny diagram modulu na extrakciu vektoru príznakov.

Podrobné informácie ako použiť tieto možnosti sú uvedené v súbore `README.md`, ktorý je priložený pri programe. Súbor `classifier.py` slúži na vyskúšanie vytrénovaného klasifikátora.

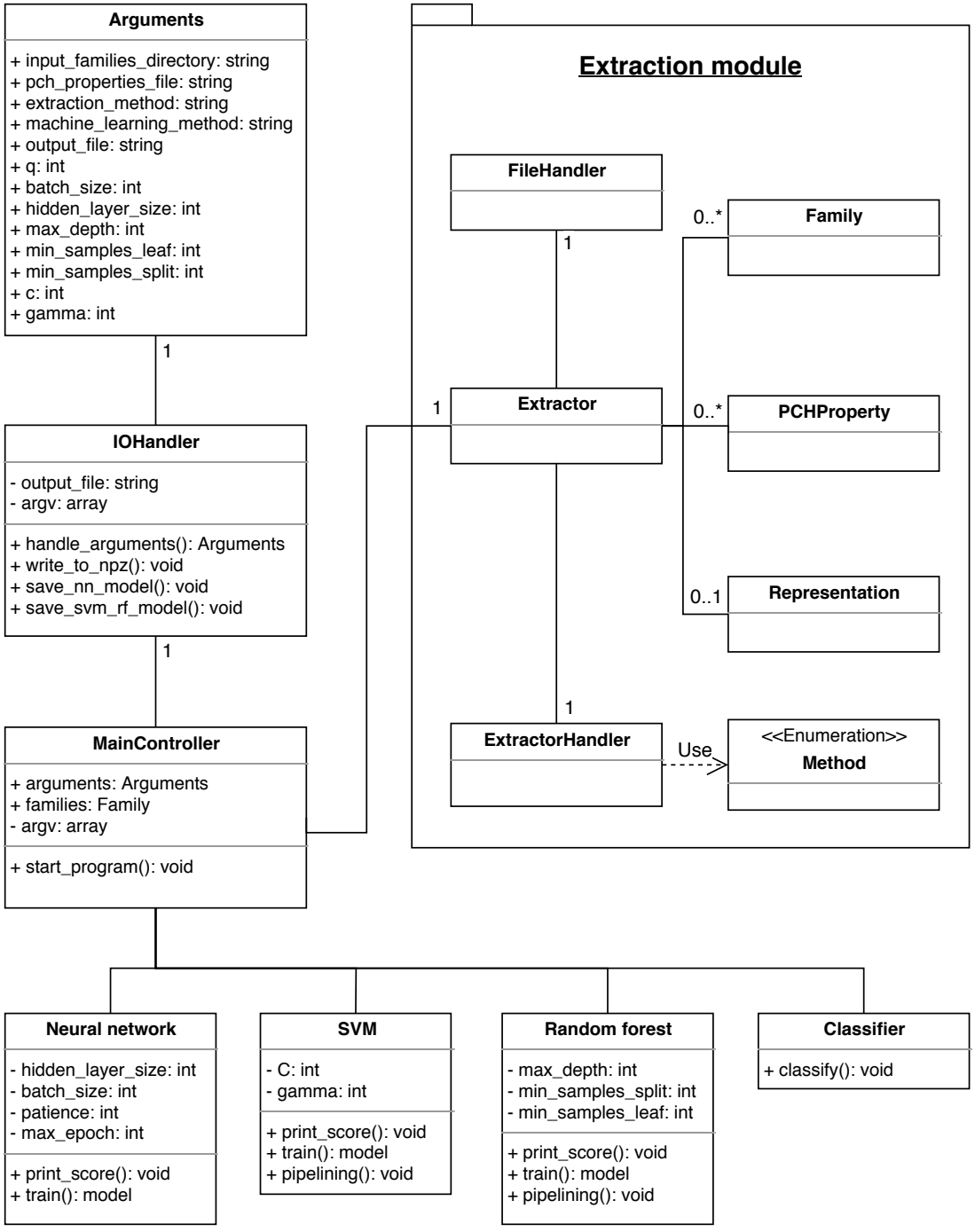
Výstupom tohto bloku programu je súbor vo formáte `.h5` pre neurónové siete a `.sav` pre Support Vector Machines a náhodný les. Okrem toho sa pri každom trénovaní na výstupe objaví aj výpis testov vytrénovaného klasifikátora, ktorý môžeme vidieť na obrázku 5.5.

```
Test Result:
Accuracy score: 0.9330
MCC Score:  0.9107

Classification Report:
                   precision    recall  f1-score   support

               0       0.95      0.94      0.94      1997
               1       0.93      0.93      0.93      2036
               2       0.91      0.93      0.92      1981
               3       0.95      0.94      0.94      1986

        accuracy                           0.93      8000
       macro avg       0.93      0.93      0.93      8000
    weighted avg       0.93      0.93      0.93      8000

Confusion Matrix:
 [[1872   37   67   21 ]
 [   30 1886   69   51 ]
 [   49   55 1845   32 ]
 [   23   52   50 1861]]

Individual family accuracy score:
0: 0.937, 1: 0.926, 2: 0.931, 3: 0.937

HIDDEN LAYER SIZE: 65
BATCH SIZE: 1000
PATIENCE: 3
```

Obr. 5.5: Príklad testového výstupu pri trénovaní štyroch tried.

Výpis obsahuje celkový výsledok presnosti a Mathewovho korelačného koeficientu klasifikátora. Nasleduje klasifikačná správa, ktorá ukazuje výsledky zhodnosti, recallu, f1-skóre a počet prvkov v jednotlivých triedach. Nasledujú priemerné výsledky a priemerné váhové výsledky rodín vzhľadom na metriky zhodnosti, recallu a f1-skóre. Pre bližšiu vizualizáciu výsledkov je vo výstupe aj matica zámeny a výsledky presnosti jednotlivých tried. Posled-

ných pár riadkov je určených pre použité parametre v danom klasifikátori. UML diagram všetkých tried aplikácie je uvedený na obrázku 5.6.



Obr. 5.6: UML diagram prepojenia jednotlivých modulov.

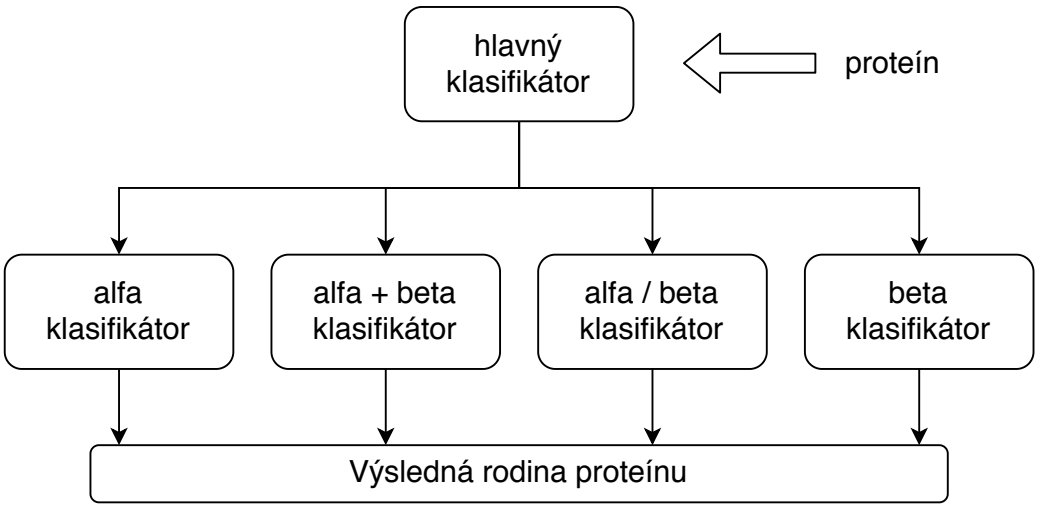
5.3 Hierarchický nástroj na klasifikáciu proteínov

Na to, aby sme vedeli nájsť rodinu každému novému proteínu, musíme byť schopní v rozumnej časovej aj priestorovej zložitosti klasifikovať čo najväčší počet rodín. Pri klasifikácii

desiatok tisícov dostupných rodín jedným klasifikátorom, by však do veľkej miery utrpela kvalita výsledku, a zároveň by sa zvýšila časová náročnosť trénovania.

Jedným z možných riešení škálovania počtu rodín je práve hierarchická klasifikácia, ktorá bola použitá v nástroji popísanom v sekcii 4.4.2. Ide o spoluprácu viacerých klasifikátorov, ktoré postupne klasifikujú proteín do štrukturálnych tried, foldov, superrodín a nakoniec rodín. V mojej implementácii som použil dvojúrovňovú klasifikáciu do štrukturálnych tried a následne priamo do rodín. Vývojový diagram implementovaného nástroja vidíme na obrázku 5.7. Hlavný klasifikátor použije vybranú metódu extrakcie príznakov na vstupný proteín a klasifikuje ho do štrukturálnej triedy. Po tejto klasifikácii sa proteín zdeleguje do konkrétneho klasifikátora pre jednotlivé štrukturálne triedy, kde sa znovu prevedie na vektor príznakov vybranou metódou.

Práca s týmto nástrojom je jednoduchá. Vstupom je súbor s proteínmi vo FASTA formáte. Výstupom je výpis na štandardný výstup, ktorý obsahuje sekvenciu testovaného proteínu, klasifikovanú štrukturálnu triedu a konkrétnu rodinu. Okrem toho je pri každej klasifikácii percentuálny údaj, ktorý udáva istotu klasifikátora pri rozhodnutí. Príklad výstupu hierarchického nástroja je ukázaný na obrázku 5.8. Konkrétne metódy na extrakciu príznakov a algoritmy strojového učenia ktoré sa použili v tomto nástroji, budú vybrané v ďalšej kapitole.



Obr. 5.7: Diagram výsledného nástroja, vhodného na klasifikáciu väčšieho množstva rodín proteínov.

```
Tested protein sequence: ACDYGHADDA
Structural class prediction: Alpha proteins (97.27% certainty).
Predicted family: MmgE_PrpD - 15.family (98.28% certainty).
```

Obr. 5.8: Príklad výstupu hierarchického nástroja.

Kapitola 6

Experimenty

V tejto kapitole sú prezentované výsledky mojich experimentov, ako aj experimentov výskumníkov, ktorí uviedli merania metód extrakcie v článkoch z ktorých čerpám. Experimenty súvisia predovšetkým s klasifikovaním proteínov do proteínových rodín. Vytvorené klasifikačné modely sú hodnotené využitím metrík uvedených v sekcii 3.5. Kapitola taktiež obsahuje popis výberu dátových sád, ktoré som využíval pri experimentoch, ako aj výber klasifikátorov pre hierarchický nástroj na škálovanie počtu klasifikovaných rodín proteínov. Parametre modelov strojového učenia boli vo väčšine prípadov získané pomocou mriežkového prehľadávania (angl. *grid search*) v kombinácii s krížovou validáciou na dátových sadách uvedených pri každom experimente. Použité parametre všetkých experimentov sú uvedené v prílohe B. Všetky prezentované hodnoty sú priemerné hodnoty viacerých experimentov s rovnakými nastaveniami parametrov algoritmov strojového učenia.

6.1 Výber dátovej sady

Na účely diplomovej práce som vytvoril dve dátové sady. Prvá je využitá v experimentoch, ktoré slúžia na užší výber metód na extrakciu príznakov a druhá v experimentoch, slúžiacich na výber klasifikátorov do spomínaného hierarchického nástroja. Rodiny obidvoch dátových sád som získal z databázy proteínových rodín Pfam [12]. Pri výbere som uplatňoval dve základné kritériá. Prvé kritérium sa týkalo množstva dostupných proteínov v danej rodine, ktoré muselo dosahovať aspoň tisíc prvkov. Keďže sa v databáze Pfam vyskytujú nielen rodiny proteínov ale aj rodiny domén proteínov, druhým kritériom bolo vyberať čisto rodiny proteínov. Okrem toho budem v práci používať aj dátové sady z článkov na overenie naimplementovaných metód extrakcie príznakov.

6.1.1 Dátové sady použité na overenie naimplementovaných metód extrakcie príznakov

Na overenie implementácie metód extrakcie príznakov používam dve dátové sady. Prvá je dostupná na stiahnutie pri článku [24], v ktorom sa výskumníci snažia čo najpresnejšie klasifikovať rezidenty endoplazmatického retikula, ktoré hrajú dôležitú úlohu vo veľkom množstve bunkových procesov, ako napríklad proteínová syntéza alebo posttranslačné spracovanie novo syntetizovaných proteínov. Stručná charakteristika tejto dátovej sady je uvedená v tabuľke 6.1. Túto dátovú sadu použijem na overenie implementácie aminokyselinovej kompozícii, dvojíc a rozdelenej aminokyselinovej kompozícii.

Proteíny	Trénovacia dátová sada	Testovacia dátová sada
Rezidenty endoplazmatického retikula	124	65
Rezidenty ne-endoplazmatického retikula	150	2900

Tabuľka 6.1: Distribúcia rezidentných a nerezidentných proteínov ER v rôznych súboroch údajov použitá pri overovaní mojej implementácie aminokyselinovej kompozície, dvojíc a rozdelenej aminokyselinovej kompozície s referenčným článkom [24].

Rodiny proteínov	Trénovacia dátová sada	Testovacia dátová sada
receptory spojené s G proteínmi (GPCR)	185	180
cyklické di-AMP receptory (ne-GPCR)	185	180

Tabuľka 6.2: Dátová sada proteínov v rôznych súboroch údajov použitá pri overovaní mojej implementácie metód globálneho kódovania, vzdialených párov a fyzikálno-chemických dvojíc s referenčným článkom [32].

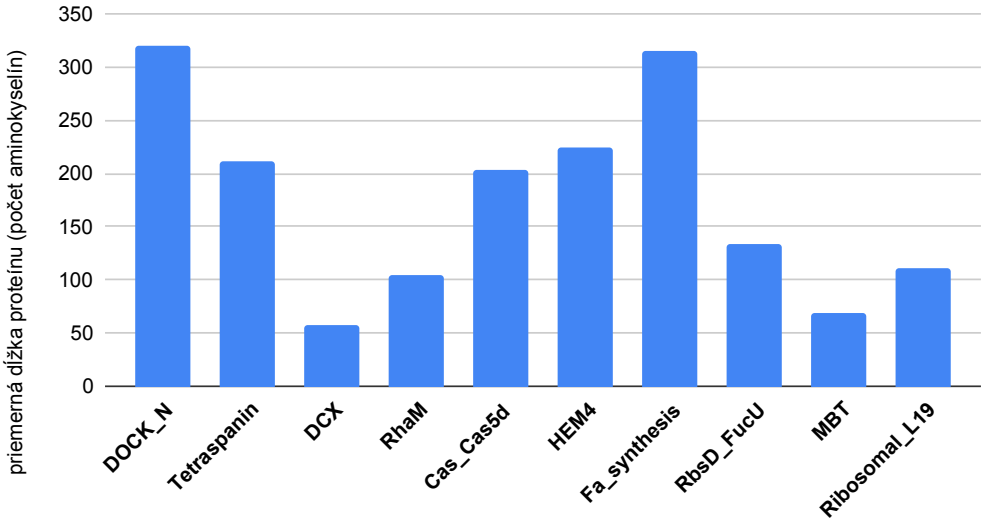
Druhú dátovú sadu som vytvoril na základe informácií z článku [32], v ktorom výskumníci pracujú s dátovou sadou ktorej stručná charakteristika je uvedená v tabuľke 6.2. Výskumníci sa snažia odlíšiť receptory spojené s G proteínmi, ktoré sa nachádzajú v eukaryotoch a hrajú dôležitú úlohu pri bunkovej adhézii, pohyblivosti, cytoskeletálnej remodelácii a prenose membrán, oproti ostatným receptorom. V článku nie je presne uvedené ktoré nespojené receptory s G proteínmi použili, tak som vybral Cyklické di-AMP receptory (PF06153). Túto dátovú sadu použijem na overenie implementácie metód globálneho kódovania, vzdialených párov a fyzikálno-chemických dvojíc. Obidve dátové sady sú v priloženom DVD nosiči.

6.1.2 Dátová sada použitá na experimenty pre výber metód extrakcie príznakov

Dátová sada použitá na experimenty pre výber metód extrakcie príznakov (ďalej len *základná dátová sada*), obsahuje desať rodín proteínov, ktoré môžeme vidieť v tabuľke 6.3. Proteínový strom týchto rodín sa nachádza v prílohe C.2. Vyberal som rodiny z celej škály databázy, pričom *alfa proteíny* s *beta proteínmi* sú zastúpené dvoma rodinami a *alfa a beta proteíny (alfa + beta)* i *alfa a beta proteíny (alfa / beta)* troma rodinami. Priemerné dĺžky aminokyselinových sekvencií v jednotlivých rodinách je ukázaná na obrázku 6.1. Jednotlivé rodiny sú:

1. **DOCK alfa-špirála (DOCK_N):** [12] táto rodina sa nachádza v blízkosti dedikátora N-konca cytokinézy, medzi variantami domény SH3 a C2.
2. **Tetraspanin:** [19] je rodinou membránových proteínov nachádzajúcich sa vo všetkých mnohobunkových eukaryotoch. Sú označované ako proteíny transmembránovej superrodiny 4 (TM4SF). Jedna z ich hlavných úloh je pripevňovať viac proteínov do jednej oblasti bunkovej membrány.
3. **Doublecortin (DCX):** [7] je rodina proteínov, ktoré sú exprimované prekurzorovými bunkami neurónov a nezrelými neurónmi v embryonálnych a dospelých kortikálnych štruktúrach. Používa sa ako ukazovateľ neurogenézy.

4. **RhaM:** [45] táto rodina obsahuje L-ramnózovú mutarotázu, čo je glykozylhydroláza, ktorá prevádza monosacharid L-ramnopyranózu z alfa na beta stereoizomér.
5. **Cas Cas5d:** [12] je jedna z rodín, ktoré sa objavujú iba v spojení s opakovaniami CRISPR (*Clustered, Regularly Interspaced Short Palindromic Repeats*).
6. **HEM4:** [43] je rodina proteínov, ktorá sa podieľa na metabolizme cyklickej tetrapyrrolovej zlúčeniny porfyrínu. Slúži na premenu hydroxymetylbilanu na uroporfyrinogén.
7. **FA syntéza:** [59] rodina proteínov, ktoré kódujú niekoľko kľúčových biosyntetických enzýmov mastných kyselín.
8. **RbsD FucU:** [13] rbsD je vysokoafinitný transportný proteín D-ribózy pyranázy. Katalyzuje konverziu medzi beta-alofuranózou a beta-alopyranózou. FucU je L-fukózová mutarotáza, ktorá sa podieľa na anomérnej premene L-fukózy.
9. **MBT opakovanie:** [5] nachádza sa v mnohých jadrových proteínoch ako napríklad midleg Q9VHA0. Funkcia zatiaľ nie je známa.
10. **Ribozomálny proteín L19:** [12] je rodina proteínov z veľkej ribozomálnej superrodiny. Môžu hrať úlohu v štruktúre a funkcii väzbového miesta aminoacylu v tRNA.



Obr. 6.1: Priemerná dĺžka aminokyselinových sekvencií proteínov v jednotlivých rodinách základnej dátovej sady použitej na výber metód extrakcie príznakov.

6.1.3 Dátová sada použitá na výber klasifikátorov do hierarchického nástroja

Dátová sada použitá na výber klasifikátorov do hierarchického nástroja (ďalej len *hlavná dátová sada*), je nadmnožinou základnej dátovej sady. Obsahuje osemdesiat rodín proteínov, ktoré sú uvedené v prílohe C.2 spolu s ich proteínovými stromami. Z každej štrukturálnej triedy je vybratých dvadsať proteínových rodín, pričom niektoré sú z hľadiska štruktúry podobnejšie ako iné.

#	Rodina	Superrodina	Fold	Štrukturálna trieda	Pfam kód
1	DOCK_N	DOCK alpha-helical	Immunoglobulin/ albumin-binding	Alfa	PF16172
2	Tetraspanin	Tetraspanin	Tetraspanin		PF00335
3	DCX	Doublecortin	Beta-Grasp	Alfa + Beta	PF03607
4	RhaM	Dimeric alpha+beta barrel	Ferredoxin		PF05336
5	Cas_Cas5d	CRISPR-associated proteins			PF09704
6	HEM4	HemD	HemD	Alfa / Beta	PF02602
7	FA_synthesis	Isopropylmalate dehydrogenase	Isopropylmalate dehydrogenase		PF02504
8	RbsD_FucU	RbsD	RbsD		PF05025
9	MBT	Tudor/ PWWP/MBT	SH3 barrel	Beta	PF02820
10	Ribosomal _L19	Translation proteins SH3			PF01245

Tabuľka 6.3: Rodiny použité ako dátová sada v experimentoch pre výber metód extrakcie príznakov.

6.2 Overenie naimplementovaných metód extrakcie príznakov

Skôr ako som začal experimentovať s vlastnými dátovými sadami, som svoju implementáciu metód overil na výsledkoch článkov iných výskumníkov. Metódy extrakcie príznakov ktoré som vybral a implementoval, sa dajú rozdeliť do dvoch základných skupín. Prvá sa zameriava čisto na frekvenčnú analýzu sekvencie proteínov, medzi ktoré patrí aminokyselinová kompozícia, rozdelená aminokyselinová kompozícia, dvojice a globálne kódovanie. Druhá využíva fyzikálno-chemické vlastnosti aminokyselín, kde patria fyzikálno-chemické dvojice a vzdialené páry.

6.2.1 Metódy frekvenčnej analýzy

Na kontrolu metód využívajúcich čisto frekvenčnú analýzu sekvencie proteínov použijem viacero zdrojov. Metódu aminokyselinovej kompozície (AK), rozdelenej aminokyselinovej kompozície (RAK) a dvojíc (D2) porovnávam s článkom [24]. V článku testujú tieto metódy na dátovej sade, ktorej stručná charakteristika je uvedená v tabuľke 6.1. Na klasifikovanie používajú algoritmus Support Vector Machines, pričom sa zamerali na metriku presnosti (accuracy) a na Mathewov korelačný koeficient (MCC). Porovnanie výsledkov výskumníkov v článku s mojou reprodukciou výsledkov môžeme vidieť v tabuľke 6.4. Výsledky ukazujú, že moja implementácia týchto metód extrakcie príznakov funguje uspokojivo.

Metódu globálneho kódovania (GE) porovnávam s článkom [32]. Výsledné hodnoty presnosti článku a mojej reprodukcie sú ukázané v tabuľke 6.5. Článok taktiež ukazuje, že presnosť jednotlivých metód je vo veľkej miere ovplyvnená výberom rodín a veľkosťou trénovacej dátovej sady. Nastavenie klasifikačnej metódy Support Vector Machines v článkoch

Metóda	presnosť (článok)	MCC (článok)	presnosť (môj výsledok)	MCC (môj výsledok)
AK	0,733	0,29	0,713	0,31
RAK	0,814	0,42	0,806	0,31
D2	0,723	0,26	0,711	0,27

Tabuľka 6.4: Výsledky uvedené v referenčnom článku s mojimi výsledkami na rovnakej dátovej sade pri použití algoritmu Support Vector Machines.

Metódy	presnosť (článok)	presnosť (môj výsledok)	MCC (môj výsledok)
GE	0,989	0,967	0,935
PCD2	0,978	0,988	0,976
VP	0,991	0,993	0,986

Tabuľka 6.5: Výsledky uvedené v referenčnom článku s mojimi výsledkami pri použití algoritmu Support Vector Machines metód globálneho kódovania, fyzikálno-chemických dvojíc a vzdialených párov.

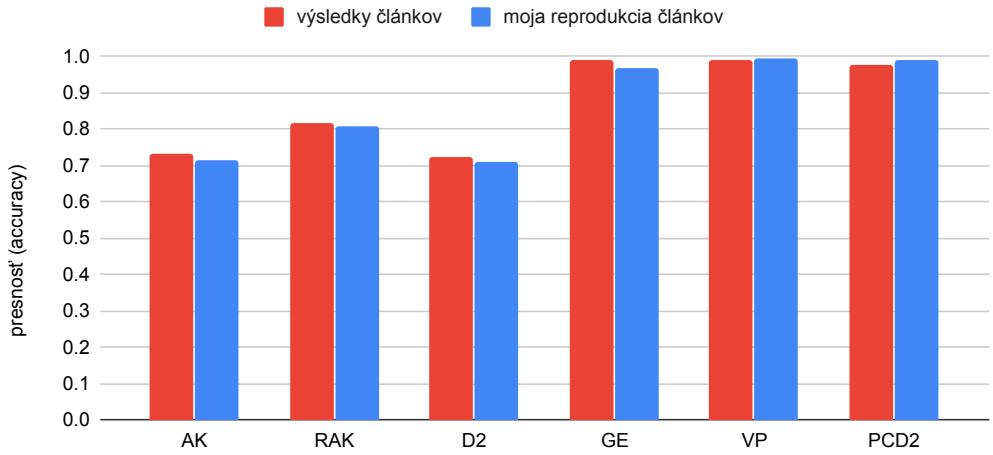
nebolo vždy uvedené, tak som musel hľadať optimálne nastavenie pomocou mriežkového prehľadávania (angl. *grid search*). Vplyv zmeny parametrov SVM bol menší, čím bližšie optimálnemu riešeniu som bol. Presnosť pri parametroch, ktoré boli veľmi vzdialené (päť násobne väčšie a viac) od optimálneho riešenia, mala pokles niekedy i v rádoch desiatok percent. I pri tejto metóde moja implementácia odpovedá presnosti článku.

6.2.2 Metódy využívajúce fyzikálno-chemické vlastnosti

Na účely experimentov tejto práce som vybral dve metódy využívajúce fyzikálno-chemické vlastnosti. Implementáciu fyzikálno-chemických dvojíc (PCD2) a vzdialených párov (VP) som kontroloval článkom [32]. Použil som rovnakú dátovú sadu ako v prípade metódy globálneho kódovania, ktorej stručnú súhrnnú charakteristiku vidíme v tabuľke 6.2. V článku sa bližšie nešpecifikovalo, ktoré fyzikálno-chemické vlastnosti výskumníci použili, tak som vybral hydrofóbnosť aminokyselín, ktorej hodnoty sú uvedené tabuľke 2.2. Vlastný výber vlastnosti spolu s výberom nespojených receptorov s G proteínmi sú hlavné dôvody drobných odchýlok mojich výsledkov oproti výsledkom z článku, ktoré sú uvedené v tabuľke 6.5. Moja implementácia metód využívajúcich fyzikálno-chemické vlastnosti je taktiež uspokojivá. Grafické porovnanie výsledkov článkov s mojou reprodukciou všetkých vybraných metód je ukázané na obrázkoch 6.2 a 6.3.

6.3 Experimenty pre výber metód extrakcie príznakov

Cieľom týchto experimentov bolo vybrať najefektívnejšie metódy pre dané rodiny proteínov a ich následné využitie pri výbere klasifikátorov do hierarchického nástroja. Ďalej o nich budem hovoriť ako o základných experimentoch. Boli tvorené na dátovej sade popísanej v sekcii 6.1.2. Z hľadiska algoritmov strojového učenia som na tieto experimenty vybral neurónové siete, Support Vector Machines a náhodný les. Vybrané metódy na extrakciu príznakov sú: aminokyselinová kompozícia (AK), rozdelená aminokyselinová kompozícia (RAK), dvojice (D2), globálne kódovanie (GE), vzdialené páry (VP) a fyzikálno-chemické dvojice (PCD2). Keďže nie všetky známe rodiny proteínov majú viac ako tisíc prvkov, základné experi-



Obr. 6.2: Graf porovnávajúci výsledky presnosti spomínaných metód z článkov, s mojou implementáciou.

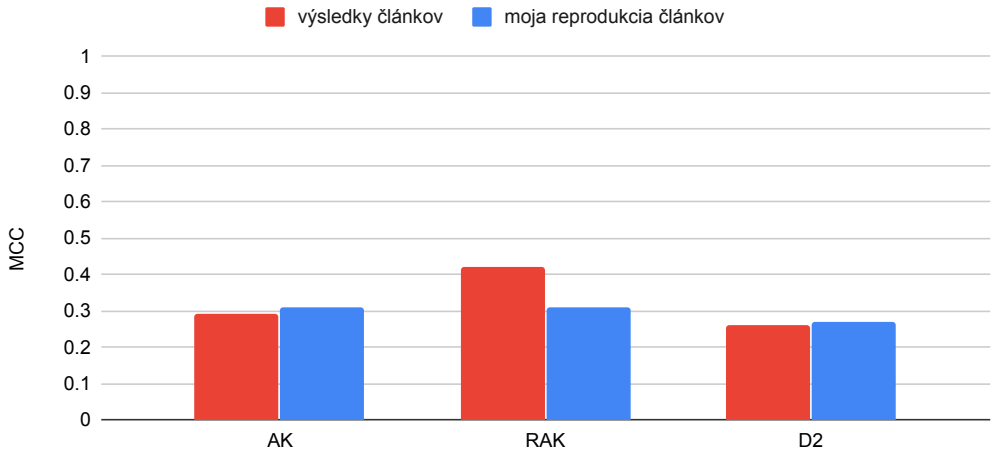
menty mali tri fázy. Prvá fáza bola zameraná na experimentovanie s kompletnou dátovou sadou, druhá s redukovanou, kde každá rodina bola zastúpená len so sto prvkami a tretia s čiastočne redukovanou, kde jedna rodina proteínov z každej štrukturálnej triedy mala sto prvkov a zvyšné boli v plnom zastúpení.

6.3.1 Experimenty s kompletnou dátovou sadou

Kompletná dátová sada sa skladá z desiatich tisícov prvkov. Pri trénovaní klasifikátorov som ju rozdelil na trénovaciu, validačnú a testovaciu časť v pomere 8:1:1. Validačná časť sa používa pri procese trénovania klasifikátora na zabránenie preučeniu (angl. *overfitting*). Testovacia časť slúži na test finálneho klasifikátora a zisk hodnôt metrík. Výsledky presnosti jednotlivých metód sú ukázané na obrázku 6.4 a v tabuľke 6.6. Na obrázku vidíme, že všetky algoritmy strojového učenia zlepšujú svoju presnosť pri náraste veľkosti vektoru príznakov popisujúceho daný proteín. Aminokyselinová kompozícia, rozdelená aminokyselinová kompozícia a globálne kódovanie majú vektor príznakov menší ako sto prvkov, dvojice majú 400, fyzikálno-chemické dvojice 800 a vzdialené páry 1200 prvkov. Túto vlastnosť si môžeme lepšie všimnúť na obrázku 6.6a, na ktorom sú metódy zoradené podľa veľkosti vektoru príznakov. Ďalej si môžeme všimnúť, že náhodný les má vo všetkých metódach horšie výsledky ako Support Vector Machines.

Metóda globálneho kódovania sa vo všetkých algoritmoch strojového učenia prepadla, čo pripisujem veľkej miere abstrakcie generovania vektoru príznakov, ktorá prestala mať takú rozlišovaciu schopnosť pri viacerých rodinách. Ďalším zaujímavým ukazovateľom je presnosť Support Vector Machines v metóde fyzikálno-chemických dvojíc, ktorá padla pod presnosť `0,5`. To môže byť spojené so šumom, na ktorý je tento algoritmus strojového učenia háklivý. Výsledný model potom klasifikoval väčšinu proteínov do jednej rodiny. Po opakovanom premiešaní vstupu bol výsledok podobný, len s tým rozdielom, že sa zmenila rodina, do ktorej model klasifikoval väčšinu proteínov.

Každý algoritmus strojového učenia reaguje rôznym spôsobom na použitie jednotlivých fyzikálno-chemických vlastností. Aby som udržal prehľadnosť, vybral som dve najlepšie fyzikálno-chemické vlastnosti pre každú kombináciu metód:

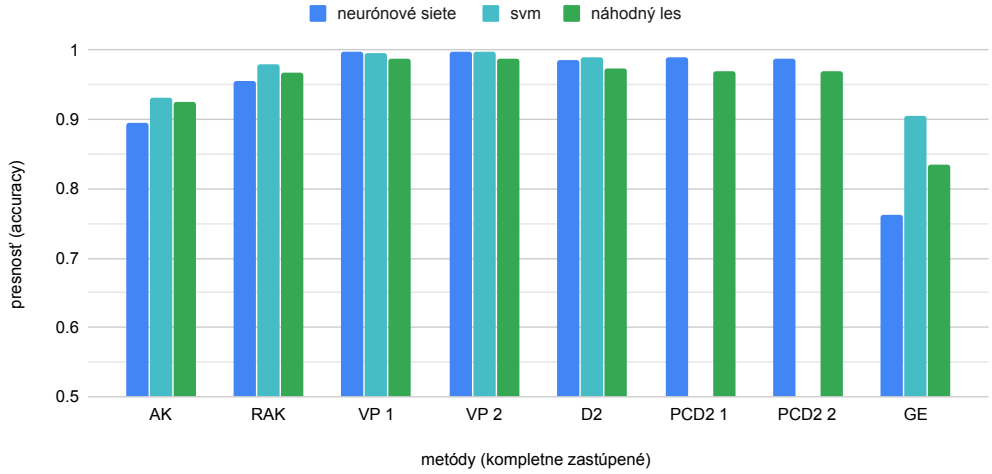


Obr. 6.3: Graf porovnávajúci výsledky Mathewovho korelačného koeficientu spomínaných metód z článkov, s mojou implementáciou.

metóda	AK	RAK	VP 1	VP 2	D2	PCD2 1	PCD2 2	GE
neurónové siete	0,895	0,956	0,998	0,997	0,986	0,99	0,987	0,762
SVM	0,931	0,979	0,996	0,997	0,989	-	-	0,905
náhodný les	0,925	0,968	0,987	0,987	0,974	0,969	0,97	0,834

Tabuľka 6.6: Výsledky presnosti jednotlivých metód pri kompletnom zastúpení rodín.

1. Vzdialené páry (VP 1)
   (a) Neurónové siete: *polarita aminokyselín*
   (b) Support Vector Machines: *hydrofóbnosť aminokyselín*
   (c) Náhodný les: *objem bočného reťazca*
2. Vzdialené páry (VP 2)
   (a) Neurónové siete: *pravdepodobnosť vzniku alfa-špirály*
   (b) Support Vector Machines: *pravdepodobnosť vzniku beta skladaného listu*
   (c) Náhodný les: *pravdepodobnosť vzniku alfa-špirály*
3. Fyzikálno-chemické dvojice (PCD2 1)
   (a) Neurónové siete: *pravdepodobnosť vzniku beta skladaného listu*
   (b) Support Vector Machines: *hydrofóbnosť aminokyselín*
   (c) Náhodný les: *hydrofóbnosť aminokyselín*
4. Fyzikálno-chemické dvojice (PCD2 2)
   (a) Neurónové siete: *membránová preferencia cytochrómu*
   (b) Support Vector Machines: *hydrofilnosť aminokyselín*
   (c) Náhodný les: *pravdepodobnosť vzniku alfa-špirály*



Obr. 6.4: Výsledky presnosti jednotlivých metód pri kompletnom zastúpení rodín.

rodina	1	2	3	4	5	6	7	8	9	10
(3) n. siete	3,6%	0,4%	**88,2%**	0,8%	2,2%	1,2%	0,8%	0,7%	1%	1,1%
(6) n. siete	1,9%	0,7%	1,5%	0,5%	3,4%	**82,9%**	3,2%	5,1%	0,7%	0,1%

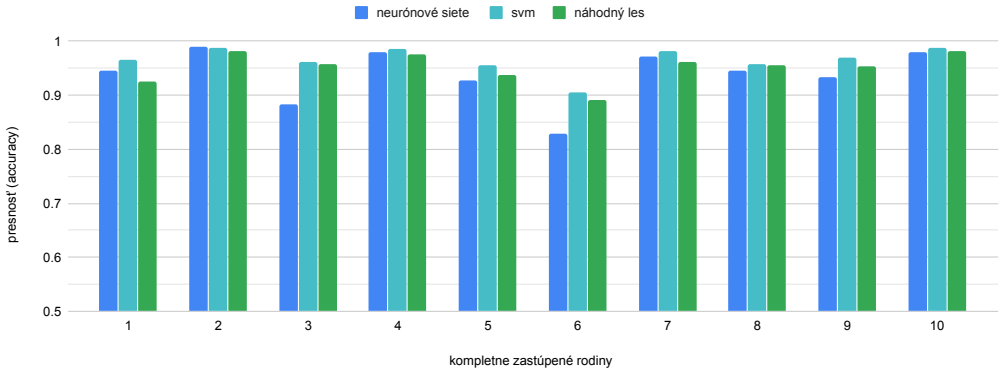
Tabuľka 6.7: Priemerná matica zámeny pre tretiu a šiestu rodinu algoritmu neurónových sietí. Použitá kompletná základná dátová sada.

Priemer výsledkov presnosti všetkých metód extrakcie príznakov jednotlivých rodín proteínov pri klasifikácii sú ukázané na obrázku 6.5. Z obrázku môžeme vidieť, že celkovo dopadla najhoršie šiesta rodina a z hľadiska neurónových sietí pomerne zle aj tretia rodina. Na matici zámeny týchto dvoch rodín ukázanej v tabuľke 6.7 si môžeme všimnúť, že veľké množstvo zlých klasifikácii pri šiestej rodine skončilo práve v siedmej a ôsmej rodine. Tieto tri rodiny majú spoločnú štrukturálnu triedu (*alfa / beta*), čo znamená, že sú si podobnejšie. Istú podobnosť vidíme taktiež medzi piatou a šiestou rodinou i napriek tomu, že patria do iných štrukturálnych tried. Pri tretej rodine sú zlé klasifikácie rovnomerne rozmiestnené, čo pripisujem malej priemernej dĺžke proteínov v tejto rodine, ktoré má za následok väčšie výkyvy hodnôt príznakov pri zmenách v aminokyselinových sekvenciách proteínov.

6.3.2 Experimenty s redukovanou dátovou sadou

V redukovanej dátovej sade sa mení hlavne veľkosť trénovacej a validačnej časti. Tie sú zmenšené na desať percent oproti kompletnej dátovej sade, pričom zvyšné proteíny sa presunuli do testovacej časti. Výsledky presnosti metód pri použití tejto dátovej sady sú uvedené v hornej časti tabuľky 6.8. Rozdiel výsledkov redukovanej dátovej sady oproti kompletnej sú ukázané v dolnej časti tabuľky 6.8. Fyzikálno-chemické vlastnosti pre vzdialené páry a fyzikálno-chemické dvojice ostali rovnaké ako sú uvedené v sekcii 6.3.1. Výsledky ukázali, že najmenej stabilné pri zmenšení dátovej sady sú metódy s menším vektorom príznakov. Najhoršie dopadla metóda globálneho kódovania, ktorej sa pri algoritme náhodného lesa znížila presnosť o viac ako 10%.

Najmenší vplyv na redukciu testovacej sady mal algoritmus neurónových sietí. Podľa očakávania sa výsledok fyzikálno-chemických dvojíc pri Support Vector Machines nezlepšil



Obr. 6.5: Priemer výsledkov presnosti všetkých metód extrakcie príznakov jednotlivých rodín proteínov pri kompletnom zastúpení.

metóda	**AK**	**RAK**	**VP 1**	**VP 2**	**D2**	**PCD2 1**	**PCD2 2**	**GE**
neurónové siete	0,863	0,936	0,982	0,97	0,952	0,956	0,956	0,756
SVM	0,883	0,941	0,984	0,987	0,96	-	-	0,82
náhodný les	0,863	0,93	0,96	0,953	0,946	0,928	0,925	0,731
rozdiel	**AK**	**RAK**	**VP 1**	**VP 2**	**D2**	**PCD2 1**	**PCD2 2**	**GE**
neurónové siete	-0,032	-0,02	-0,016	-0,027	-0,034	-0,034	-0,031	-0,006
SVM	-0,048	-0,038	-0,012	-0,01	-0,029	-	-	-0,085
náhodný les	-0,062	-0,038	-0,027	-0,034	-0,028	-0,041	-0,045	-0,103

Tabuľka 6.8: Výsledky presnosti metód, pri použití redukovanej dátovej sady v hornej časti tabuľky. Rozdiel výsledkov pri použití redukovanej dátovej sady oproti kompletnej dátovej sade v dolnej časti tabuľky.

a stále dochádzalo ku väčšinovému klasifikovaniu do jednej rodiny. Ako aj pri kompletnej dátovej sade, tak aj pri redukovanej dochádzalo ku tomu, že metódy ktoré tvorili väčší vektor príznakov boli pri klasifikácii úspešnejšie. Tento jav môžeme vidieť na obrázku 6.6b.

Rozdiely výsledkov presnosti jednotlivých rodín proteínov medzi použitím redukovanej dátovej sady a kompletnej dátovej sady sú uvedené v tabuľke 6.9. Môžeme vidieť, že najväčší pokles presnosti zaznamenala šiesta rodina, ktorá bola kritická už pri kompletnej dátovej sade. V tabuľke 6.10 sú uvedené časti matice zámeny, v ktorých vidíme podobné správanie ako pri kompletnej dátovej sade. Veľké množstvo zlých klasifikácii šiestej rodiny sa presunulo ostatným rodinám so spoločnej štrukturálnej triedy. Okrem toho vidíme veľkú podobnosť šiestej rodiny s piatou či už zlou klasifikáciou šiestej rodiny (2. a 4. riadok) ako aj piatej rodiny (3. riadok). Veľký pokles môžeme pozorovať pri tretej a deviatej rodine, čo spájam s krátkymi aminokyselinovými sekvenciami proteínov v týchto triedach. Tabuľka 6.10 nám taktiež ukazuje väčšiu podobnosť deviatej rodiny s rodinami *alfa + beta* štrukturálnej triedy.

6.3.3 Experimenty s čiastočne redukovanou dátovou sadou

V čiastočne redukovanej dátovej sade dochádza ku redukcii len pri jednej rodine z každej štrukturálnej triedy. Konkrétne sa jedná o prvú, piatu, šiestu a desiatu rodinu. Týmto sa snažím zistiť, aký vplyv na presnosť má nevyvážená dátová sada či už pri plne zastúpených rodinách ako aj pri redukovaných. Výsledky presnosti metód pri použití takto redukovanej

rodina	1	2	3	4	5	6	7	8	9	10
n. siete	-0,006	-0,014	-0,021	-0,018	-0,033	-0,02	-0,009	-0,017	-0,01	-0,019
SVM	-0,022	-0,008	-0,072	-0,023	-0,038	-0,076	-0,022	-0,031	-0,052	-0,022
n. les	-0,036	-0,005	-0,049	-0,014	-0,088	-0,175	-0,006	-0,02	-0,056	-0,026

Tabuľka 6.9: Rozdiel priemerných výsledkov presnosti všetkých metód extrakcie príznakov jednotlivých rodín redukovanej dátovej sady oproti kompletnej.

rodina	1	2	3	4	5	6	7	8	9	10
(3) SVM	3,2%	0,6%	**89%**	0,4%	2,1%	2,3%	0,4%	0,5%	0,9%	0,7%
(6) SVM	3,3%	0,6%	1,2%	0,3%	4,4%	**82,8%**	2,9%	4,2%	0,2%	0,1%
(5) n. les	2,3%	1,2%	1,4%	1,7%	**85%**	5,2%	0,5%	1,3%	1%	0,6%
(6) n. les	4,7%	2,7%	1,4%	1,1%	8,2%	**71,6%**	3,6%	5,8%	0,4%	0,4%
(9) n. les	0,4%	0,6%	2,4%	3,4%	2,1%	0,2%	0,2%	0,6%	**89,8%**	0,3%

Tabuľka 6.10: Priemerná matica zámeny pre tretiu a šiestu rodinu algoritmu svm a piatu, šiestu a deviatu rodinu algoritmu náhodný les. Použitá redukovaná základná dátová sada.

základnej dátovej sady sú uvedené v hornej časti tabuľky 6.11. Rozdiel výsledkov čiastočne redukovanej dátovej sady oproti kompletnej dátovej sade sú ukázané v dolnej časti tabuľky 6.11. Fyzikálno-chemické vlastnosti pre vzdialené páry a fyzikálno-chemické dvojice ostali rovnaké, ako sú uvedené v sekcii 6.3.1.

Vidíme, že dosiahnuté výsledky sú najviac ovplyvnené veľkosťou vektoru príznakov. V metódach, ktoré majú malé vektory príznakov sa presnosť zvýšila. To je zapríčinené tým, že pri trénovaní zanedbali rodiny s malým zastúpením a teda prakticky rozlišovali len šesť rodín, ktoré mali po tisíc prvkov. Uvedený výsledok je váhový, čo znamená, že každá rodina má vo finálnom výsledku takú váhu, ako je v testovaní zastúpená, kvôli čomu sa slabé výsledky redukovaných rodín neprejavili. I napriek malej zmene vo výsledkoch presnosti sa i pri redukovanej dátovej sade ukazuje, že metódy s väčším počtom príznakov vo vektore dopadli lepšie. To môžeme vidieť v obrázku 6.6c. Rozdiely vo výsledkoch jednotlivých rodín sú uvedené v tabuľke 6.12. Pokles presnosti pri niektorých redukovaných rodinách dosiahol až hranicu `0,4` (40%). Pri plne zastúpených rodinách môžeme vidieť mierny nárast, ktorý je rovnako spôsobený zanedbaním tých slabšie zastúpených.

Časť matice zámeny, ktorá je ukázaná v tabuľke 6.13 nám ukazuje podobnosti medzi jednotlivými rodinami. Pri piatej a šiestej rodine vidíme, že väčšina zlých klasifikácií sa udr-

metóda	AK	RAK	VP 1	VP 2	D2	PCD2 1	PCD2 2	GE
neurónové siete	0,911	0,971	0,996	0,993	0,974	0,988	0,983	0,812
SVM	0,954	0,983	0,995	0,997	0,991	-	-	0,935
náhodný les	0,954	0,978	0,98	0,98	0,978	0,947	0,964	0,89
rozdiel	**AK**	**RAK**	**VP 1**	**VP 2**	**D2**	**PCD2 1**	**PCD2 2**	**GE**
neurónové siete	+0,016	+0,015	-0,002	-0,004	-0,012	-0,002	-0,004	+0,05
SVM	+0,023	+0,004	-0,001	0	+0,002	-	-	+0,03
náhodný les	+0,029	+0,01	-0,007	-0,007	+0,004	-0,022	-0,006	+0,056

Tabuľka 6.11: Výsledky presnosti metód pri použití čiastočne redukovanej dátovej sady v hornej časti tabuľky. Rozdiel výsledkov pri použití čiastočne redukovanej dátovej sady oproti kompletnej dátovej sade v dolnej časti tabuľky.

rodina	1	2	3	4	5	6	7	8	9	10
n. siete	-0,245	-0,008	+0,071	-0,003	-0,199	-0,24	+0,011	+0,014	+0,024	-0,037
SVM	-0,082	+0,003	+0,013	+0,005	-0,028	-0,173	+0,006	+0,021	+0,004	-0,02
n. les	-0,16	+0,007	+0,032	+0,015	-0,4	-0,458	+0,013	+0,024	+0,017	-0,067

Tabuľka 6.12: Rozdiel priemerných výsledkov presnosti všetkých metód extrakcie príznakov jednotlivých rodín čiastočne redukovanej dátovej sady oproti kompletnej.

rodina	1	2	3	4	5	6	7	8	9	10
(1) n. siete	**70%**	0%	23,9%	0,3%	2,1%	0,8%	0,5%	0,8%	1,6%	0%
(1) SVM	**88,3%**	0%	5,6%	0,9%	1,3%	1,3%	1,3%	0,9%	0,4%	0%
(1) n. les	**76,7%**	3,5%	7,3%	2,3%	0,6%	0,3%	5,2%	3,2%	0,9%	0%
(5) n. siete	1,5%	0,3%	8,5%	2,6%	**72,8%**	4,4%	0,9%	0,9%	7,6%	0,6%
(5) SVM	1,6%	0,3%	1,9%	1%	**92,7%**	1,3%	0,6%	0,3%	0,3%	0%
(5) n. les	0,3%	8,8%	6,6%	12,2%	**53,6%**	1,6%	0,6%	12,5%	3,8%	0%
(6) n. siete	2,6%	3,3%	5,6%	1,7%	2,6%	**58,9%**	9,6%	12,3%	3,3%	0%
(6) SVM	0,4%	2,6%	4,8%	0,4%	2,2%	**73,2%**	6,5%	8,7%	1,3%	0%
(6) n. les	0%	13%	3,7%	4,3%	1,2%	**43,5%**	9%	23,9%	1,2%	0%
(10) n.siete	0,3%	0%	4,2%	0%	0,3%	0%	0,6%	0%	0,3%	**94,2%**
(10) SVM	0,4%	0%	1,5%	0%	1,1%	0,4%	0%	0%	0%	**96,7%**
(10) n. les	0%	0%	4,6%	0,6%	0%	0%	0,9%	1,7%	0,9%	**91,4%**

Tabuľka 6.13: Priemerná matica zámeny pre prvú, piatu, šiestu a desiatu rodinu. Použitá čiastočne redukovaná základná dátová sada.

žala v ich štrukturálnych triedach. Pre piatu je to (*alfa+beta*) a pre šiestu (*alfa/beta*). Prvá rodina sa podľa výsledkov najviac podobá na tretiu rodinu, čo môžeme potvrdiť maticami zámen v tabuľkách 6.7 a 6.10, v ktorých vidíme podobnosť tretej rodiny s prvou.

Desiata rodina je dobre rozlíšiteľná od ostatných, čo ukazujú nie len jej dobré výsledky i pri zníženom počte prvkov, ale aj veľmi nízke hodnoty pri chybách klasifikácie ostatných rodín. Malá podobnosť sa dá spozorovať len s treťou rodinou. Zaujímavý poznatok je vidno v tom, že i napriek podobnosti prvej a desiatej rodiny s treťou rodinou sa neukázala žiadna väčšia podobnosť medzi prvou a desiatou rodinou. Vzťah medzi podobnosťami rodín teda nie je tranzitívny.

6.3.4 Zhrnutie základných experimentov

V základných experimentoch vidíme, že pri metódach s malým vektorom príznakov najlepšie dopadol algoritmus Support Vector Machines a najhoršie neurónové siete. Tieto metódy extrakcie majú väčší problém s nevyváženou dátovou sadou, pri ktorej často dochádza ku zanedbávaniu rodín s malým zastúpením. Metódam s väčším počtom príznakov sa lepšie darilo klasifikovať rodiny s malým zastúpením vďaka lepšej rozlíšiteľnosti príznakov. Pri tých sa najlepšie umiestnili práve neurónové siete. Porovnávaním použitých troch algoritmov strojového učenia môžeme usúdiť, že najmenej úspešný pri klasifikovaní proteínov vybranými metódami extrakcie príznakov je náhodný les. Na obrázkoch 6.7 vidíme časy trénovania jednotlivých algoritmov strojového učenia na jednotlivých metódach, ktoré sú zoradené podľa veľkosti vektoru príznakov. Pri algoritme Support Vector Machines si môžeme všimnúť, že nárast vektoru príznakov sa do veľkej miery odzrkadlil na trénovacom čase. To je spôsobené jeho časovou zložitosťou, ktorá sa pohybuje medzi $\mathcal{O}(n^2 * f)$ až $\mathcal{O}(n^3 * f)$, v závislosti od efektivity cache pamäte, kde f je počet príznakov vo vektore [41]. Zaujímavá

situácia nastala pri algoritme neurónových sietí, kde trénovanie redukovanej dátovej sady zabralo viac času ako trénovanie kompletnej dátovej sady na všetkých metódach extrakcie príznakov.

Do hlavných experimentov som vybral štyri metódy. Rozdelenú aminokyselinovú kompozíciu (RAK) kvôli jej najlepším výsledkom s malým vektorom príznakov, dvojice (D2) kvôli najlepším výsledkom spomedzi metód, ktoré nevyužívajú fyzikálno-chemické vlastnosti a vzdialené páry (VP 1, VP 2), kvôli najlepším výsledkom spomedzi všetkých metód.

6.4 Experimenty pre výber klasifikátorov do hierarchického nástroja

Experimenty pre výber klasifikátorov do hierarchického nástroja (ďalej len *hlavné experimenty*), boli tvorené ná dátovej sade popísanej v sekcii 6.1.3. Cieľom týchto experimentov bolo vybrať najefektívnejšie metódy extrakcie príznakov a metódy strojového učenia pre jednotlivé klasifikátory nástroja, popísaného v sekcii 5.3. Hlavné experimenty mali dve fázy. Prvá fáza bola zameraná na experimentovanie s kompletnou dátovou sadou. Kvôli zložitosti algoritmu Support Vector Machines som však nebol schopný dostať výsledok trénovania na kompletnej dátovej sade v rozumnom čase. Aby som vedel porovnať všetky tri algoritmy strojového učenia, musel som teda pracovať nie len s kompletnou dátovou sadou ale aj redukovanou, kde každá rodina bola zastúpená s desatinou proteínov z pôvodnej dátovej sady.

6.4.1 Výber hlavného klasifikátora

Hlavný klasifikátor má za úlohu priradiť proteínu správnu štrukturálnu triedu. Vstupom pri trénovaní je spomínaných osemdesiat tisíc proteínov, rozdelených do štyroch tried. Z tabuľky 6.14 vidíme, že pri kompletnej dátovej sade sú až na metódu rozdelenej aminokyselinovej kompozícii neurónové siete podstatne lepšie ako náhodný les. Ďalej si môžeme všimnúť podobné správanie neurónových sietí ako v základných testoch. Čím väčší vektor príznakov mali, tým boli presnejšie. Taktiež vidíme, že pri metódach RAK a D2 bol pokles presnosti pri použití redukovanej dátovej sady oproti kompletnej dátovej sade najvyšší. Z dosiahnutých výsledkov teda vyplýva, že najpresnejšia je metóda vzdialených párov s využitím polarity aminokyselín (VP 1).

Zaujímavý pohľad na podobnosť jednotlivých štrukturálnych tried nám poskytujú matice zámen, uvedené v tabuľke 6.15. Najviac rozlíšiteľná je *alfa* štrukturálna trieda, spolu s *beta* štrukturálnou triedou. Tie majú najvyššie správne pozitívne percento.

Najmenej správne pozitívnych prípadov má *alfa+beta*. To pripisujem väčšiemu rozptýleniu po priestore príznakov, kvôli čomu dochádza ku častej zámene za iné štrukturálne triedy. Celkovo vidíme veľkú koreláciu medzi *alfa+beta* a *alfa/beta* štrukturálnymi triedami, čo potvrdzuje aj podobnosť medzi piatou a šiestou rodinou zo základných testov.

6.4.2 Výber ostatných klasifikátorov

Ostatné klasifikátory z nástroja majú za úlohu priradiť proteín do konkrétnej rodiny danej štrukturálnej triedy. Vstupom pri trénovaní každého klasifikátora je teda dvadsať tisíc proteínov zo štrukturálnej triedy, ktorej rodiny klasifikujeme. Najlepšie výsledky boli zistené pri *alfa+beta* štrukturálnej triede, čo je v súlade zo zistením z tabuliek zámen 6.15. Keďže sú viac rozptýlené po priestore príznakov, sú taktiež viac rozlíšiteľné pri rodinách proteínov.

všetky proteíny	**presnosť**				**MCC**			
metódy	**RAK**	**VP 1**	**VP 2**	**D2**	**RAK**	**VP 1**	**VP 2**	**D2**
neurónové siete	0,884	0,958	0,943	0,922	0,845	0,944	0,935	0,91
náhodný les	0,898	0,912	0,914	0,897	0,866	0,881	0,885	0,864
metódy (10%)	**RAK**	**VP 1**	**VP 2**	**D2**	**RAK**	**VP 1**	**VP 2**	**D2**
neurónové siete	0,775	0,903	0,902	0,788	0,701	0,867	0,867	0,718
SVM	0,81	0,9	0,897	0,803	0,747	0,875	0,865	0,736
náhodný les	0,806	0,823	0,829	0,805	0,743	0,766	0,775	0,742

Tabuľka 6.14: Výsledok presnosti a Mathewovho korelačného koeficientu pri plnom zastúpení dátovej sady v hornej časti tabuľky a pri redukovanom zastúpení na 10% v dolnej časti tabuľky. Dátová sada sa skladá zo všetkých proteínov osemdesiatich rodín.

RAK	**alfa**	**a+b**	**a/b**	**beta**	**D2**	**alfa**	**a+b**	**a/b**	**beta**
alfa	**85,5%**	4,6%	7%	2,9%	**alfa**	**84%**	4,7%	8,4%	2,9%
a+b	6,8%	**75,4%**	10,7%	7,1%	**a+b**	7,6%	**71,9%**	11,8%	8,6%
a/b	9,4%	9,6%	**75,8%**	5,2%	**a/b**	10%	6,9%	**78,4%**	4,7%
beta	3,3%	5,4%	6,8%	**84,5%**	**beta**	3,6%	6,1%	6,3%	**83,9%**
VP 1	**alfa**	**a+b**	**a/b**	**beta**	**VP 2**	**alfa**	**a+b**	**a/b**	**beta**
alfa	**90,9%**	2,9%	4,1%	2,1%	**alfa**	**90,6%**	3%	4,6%	1,8%
a+b	4,1%	**82,4%**	7,5%	6%	**a+b**	4,4%	**82,5%**	7,3%	5,8%
a/b	6,4%	4,5%	**85,9%**	3,2%	**a/b**	6,4%	4,6%	**86,2%**	2,8%
beta	1,7%	3,8%	3,9%	**90,6%**	**beta**	1,9%	3,9%	4%	**90,2%**

Tabuľka 6.15: Matice zámen klasifikácie proteínov do jednotlivých štrukturálnych tried za použitia redukovanej dátovej sady, obsahujúcej osemdesiat rodín proteínov.

Najhoršie dopadli klasifikátory *alfa/beta* štrukturálnej triedy. To je spôsobené veľkou podobnosťou rodín, čo nám dosvedčujú aj výsledky základných experimentov. Z dosiahnutých výsledkov vidíme, že najpresnejšie metódy pre jednotlivé štrukturálne triedy sú:

- **alfa klasifikátor:** vzdialené páry s využitím polarity aminokyselín (VP 1). Tabuľku výsledkov vidíme v 6.16. Na trénovanie sa využijú neurónové siete.
- **alfa+beta klasifikátor:** vzdialené páry s využitím pravdepodobnosti vzniku alfa-špirály (VP 2). Tabuľku výsledkov vidíme v 6.17. Na trénovanie sa využijú neurónové siete.
- **alfa/beta klasifikátor:** vzdialené páry s využitím pravdepodobnosti vzniku alfa-špirály (VP 2). Tabuľku výsledkov vidíme v 6.18. Na trénovanie sa využijú neurónové siete.
- **beta klasifikátor:** vzdialené páry s využitím polarity aminokyselín (VP 1). Tabuľku výsledkov vidíme v 6.19. Na trénovanie sa využijú neurónové siete.

alfa proteíny	presnosť				MCC			
metódy	**RAK**	**VP 1**	**VP 2**	**D2**	**RAK**	**VP 1**	**VP 2**	**D2**
neurónové siete	0,92	0,989	0,979	0,958	0,92	0,977	0,973	0,957
náhodný les	0,933	0,958	0,958	0,947	0,93	0,955	0,956	0,944
metódy (10%)	**RAK**	**VP 1**	**VP 2**	**D2**	**RAK**	**VP 1**	**VP 2**	**D2**
neurónové siete	0,868	0,972	0,963	0,89	0,861	0,97	0,962	0,89
SVM	0,937	0,961	0,963	0,952	0,934	0,96	0,962	0,949
náhodný les	0,887	0,936	0,924	0,917	0,882	0,933	0,92	0,914

Tabuľka 6.16: Výsledok presnosti a Mathewovho korelačného koeficientu pri plnom zastúpení dátovej sady v hornej časti tabuľky a pri redukovanom zastúpení na 10% v dolnej časti tabuľky. Dátová sada sa skladá z dvadsiatich rodín *alfa* štrukturálnej triedy.

alfa + beta proteíny	presnosť				MCC			
metódy	**RAK**	**VP 1**	**VP 2**	**D2**	**RAK**	**VP 1**	**VP 2**	**D2**
neurónové siete	0,928	0,985	0,993	0,96	0,936	0,985	0,982	0,958
náhodný les	0,938	0,969	0,965	0,959	0,935	0,968	0,964	0,957
metódy (10%)	**RAK**	**VP 1**	**VP 2**	**D2**	**RAK**	**VP 1**	**VP 2**	**D2**
neurónové siete	0,887	0,975	0,98	0,908	0,881	0,977	0,978	0,904
SVM	0,946	0,967	0,973	0,957	0,94	0,986	0,972	0,955
náhodný les	0,893	0,947	0,948	0,928	0,887	0,944	0,946	0,925

Tabuľka 6.17: Výsledok presnosti a Mathewovho korelačného koeficientu pri plnom zastúpení dátovej sady v hornej časti tabuľky a pri redukovanom zastúpení na 10% v dolnej časti tabuľky. Dátová sada sa skladá z dvadsiatich rodín *alfa+beta* štrukturálnej triedy.

alfa / beta proteíny	presnosť				MCC			
metódy	**RAK**	**VP 1**	**VP 2**	**D2**	**RAK**	**VP 1**	**VP 2**	**D2**
neurónové siete	0,905	0,977	0,986	0,935	0,91	0,98	0,975	0,931
náhodný les	0,889	0,95	0,947	0,929	0,884	0,948	0,945	0,926
metódy (10%)	**RAK**	**VP 1**	**VP 2**	**D2**	**RAK**	**VP 1**	**VP 2**	**D2**
neurónové siete	0,807	0,969	0,935	0,864	0,796	0,969	0,943	0,856
SVM	0,904	0,956	0,958	0,929	0,9	0,95	0,95	0,926
náhodný les	0,824	0,9	0,9	0,862	0,815	0,895	0,895	0,856

Tabuľka 6.18: Výsledok presnosti a Mathewovho korelačného koeficientu pri plnom zastúpení dátovej sady v hornej časti tabuľky a pri redukovanom zastúpení na 10% v dolnej časti tabuľky. Dátová sada sa skladá z dvadsiatich rodín *alfa/beta* štrukturálnej triedy.

6.5 Výsledky hierarchického nástroja

Na zistenie celkovej presnosti hierarchického klasifikátora som použil rodiny z hlavnej dátovej sady. Z jednotlivých rodín som vybral 300 proteínov, ktoré neboli zastúpené v trénovacej ani testovacej dátovej sade. V prípade, že rodina neobsahovala 1300 proteínov a teda ne-

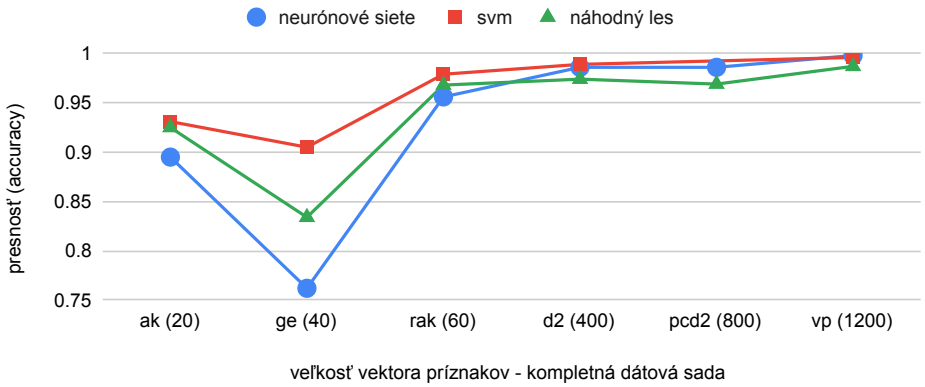
beta proteíny	**presnosť**				**MCC**			
metódy	**RAK**	**VP 1**	**VP 2**	**D2**	**RAK**	**VP 1**	**VP 2**	**D2**
neurónové siete	0,928	0,994	0,983	0,96	0,926	0,983	0,98	0,958
náhodný les	0,924	0,968	0,964	0,948	0,921	0,966	0,963	0,946
metódy (10%)	**RAK**	**VP 1**	**VP 2**	**D2**	**RAK**	**VP 1**	**VP 2**	**D2**
neurónové siete	0,877	0,97	0,977	0,897	0,872	0,985	0,975	0,894
SVM	0,939	0,969	0,973	0,962	0,935	0,97	0,97	0,96
náhodný les	0,889	0,952	0,95	0,917	0,884	0,949	0,947	0,913

Tabuľka 6.19: Výsledok presnosti a Mathewovho korelačného koeficientu pri plnom zastúpení dátovej sady v hornej časti tabuľky a pri redukovanom zastúpení na 10% v dolnej časti tabuľky. Dátová sada sa skladá z dvadsiatich rodín *beta* štrukturálnej triedy.

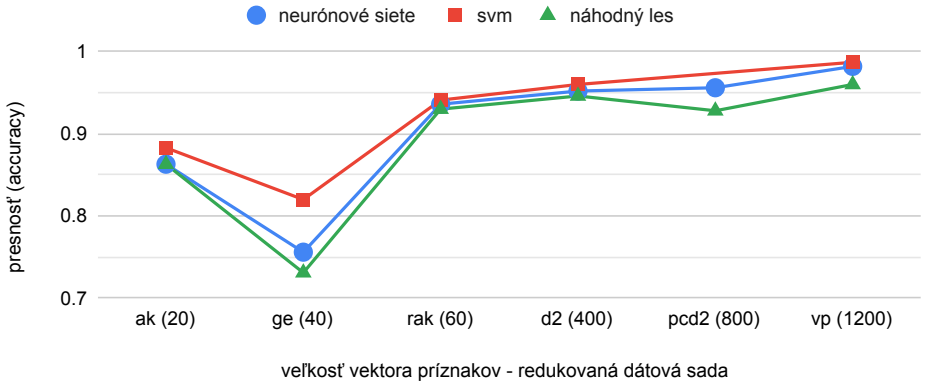
	alfa	alfa + beta	alfa / beta	beta	presnosť klasifikovania do jednotlivých rodín
alfa	**95,1%**	1,1%	3,1%	0,6%	**alfa: 0,943 (94,3%)**
alfa + beta	1%	**94,7%**	2,7%	1,6%	**alfa + beta: 0,944 (94,4%)**
alfa / beta	1,4%	1,6%	**95,8%**	1,2%	**alfa / beta: 0,947 (94,7%)**
beta	0,3%	1,1%	1%	**97,6%**	**beta: 0,971 (97,1%)**
celkový výsledok presnosti hierarchického nástroja: 0,95125 (95,125%)					

Tabuľka 6.20: Výsledná matica zámeny a celkový výsledok presnosti hierarchického nástroja.

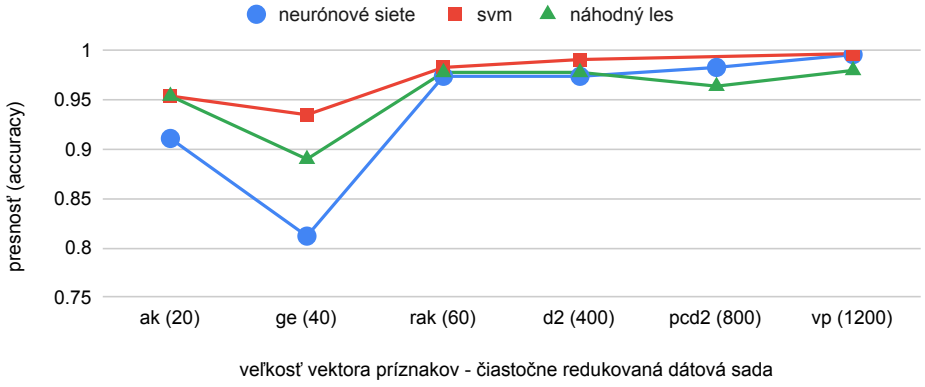
bolo možné vybrať úplne nové proteíny, som vybral náhodných 300 z celej rodiny. Výsledná matica zámeny hlavného klasifikátora v nástroji a celková presnosť nástroja je uvedená v tabuľke 6.20. Výsledky ukazujú, že nástroj dokáže klasifikovať osemdesiat spomínaných rodín v 95,125% prípadoch. V porovnaní so spomínanými nástrojmi dopadol o 0,9% horšie ako SECLAF, ktorý je jeden z najnovších vedeckých nástrojov. Zároveň však dopadol výrazne lepšie ako GPCRTree, ktorý používa rovnaké hierarchické usporiadanie klasifikátorov, kde sa výsledná presnosť pre dve úrovne klasifikácie pohybuje okolo 81,5%. To môžeme pripísať menšiemu počtu klasifikovaných rodín jedným klasifikátorom, lepšej metóde na extrakciu príznakov ako aj použitiu efektívnejšieho algoritmu strojového učenia.



(a) Presnosť vzhľadom na veľkosť vektoru príznakov uvedenú v zátvorkách pri jednotlivých použitých metódach, pri použití kompletnej dátovej sady.

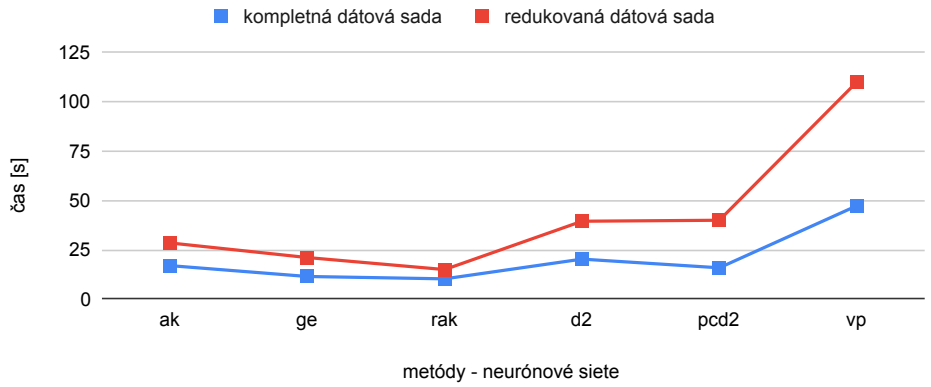


(b) Presnosť vzhľadom na veľkosť vektoru príznakov uvedenú v zátvorkách pri jednotlivých použitých metódach, pri použití redukovanej dátovej sady.

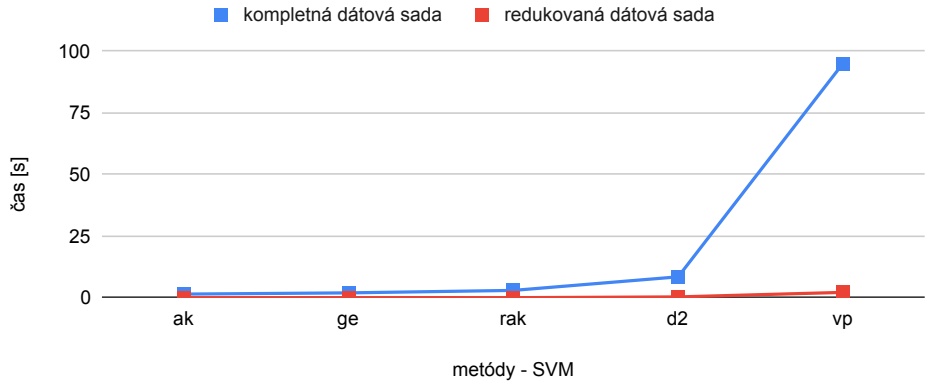


(c) Presnosť vzhľadom na veľkosť vektoru príznakov uvedenú v zátvorkách pri jednotlivých použitých metódach, pri použití čiastočne redukovanej dátovej sady.

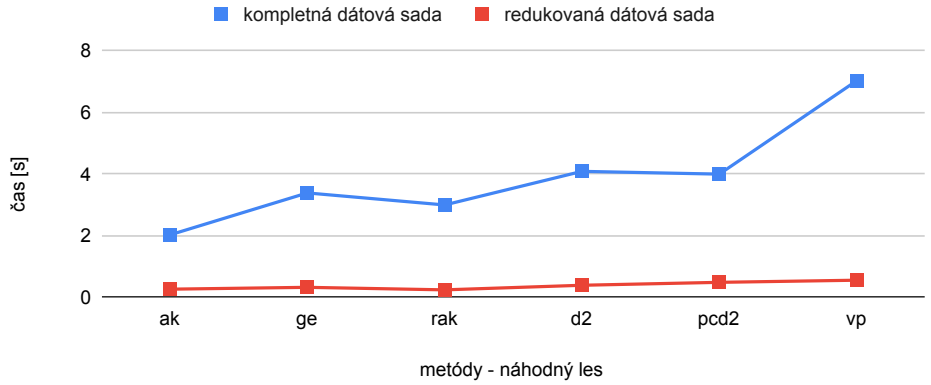
Obr. 6.6: Grafy presnosti použitých metód vzhľadom na veľkosti vektoru príznakov. Na grafe (a) sú zobrazené výsledky kompletnej dátovej sady, na (b) výsledky redukovanej dátovej sady a (c) výsledky čiastočne redukovanej dátovej sady.



(a) Čas trénovania jednotlivých metód za použitia algoritmu neurónových sietí pri kompletnej a redukovanej dátovej sade.



(b) Čas trénovania jednotlivých metód za použitia algoritmu Support Vector Machines pri kompletnej a redukovanej dátovej sade.



(c) Čas trénovania jednotlivých metód za použitia algoritmu náhodného lesa pri kompletnej a redukovanej dátovej sade.

Obr. 6.7: Časy trénovania jednotlivých metód použitých v práci. Na grafe (a) vidíme časy trénovania neurónových sietí, na grafe (b) časy trénovania Support Vector Machines a na (c) časy trénovania náhodného lesa.

Kapitola 7

Záver

Klasifikácia proteínov je veľmi dôležitou súčasťou modernej medicíny. Dôvodov, prečo je dobré sa touto disciplínou zaoberať je viacero. Používa sa pri výrobe liekov, diagnostikovaní rôznych klinických chorôb ako aj pri vzniku personalizovanej zdravotnej starostlivosti. Aby sme vedeli pracovať s proteínmi na komplexnej úrovni, musíme byť schopní určiť ich funkciu presne a čo najrýchlejšie. Na určenie funkcie proteínu nám vo väčšine prípadov stačí poznať rodinu proteínov do ktorej patrí. Hlavným cieľom práce teda bolo vybrať vhodný klasifikačný model proteínov, ktorý by vedel čo najpresnejšie a najrýchlejšie klasifikovať proteíny do rodín.

V práci bolo predstavených niekoľko existujúcich nástrojov klasifikácie proteínov. Medzi ne patrí GPCRTree, ktorý ako jeden z prvých využil hierarchický prístup klasifikácie a nástroj SECLAF, ktorý bol v roku 2018 hodnotený ako jeden z najpresnejších nástrojov v obore. Pri výbere výsledného klasifikačného modelu práce som sa inšpiroval práve nástrojom GPCRTree, kvôli dobrej škálovateľnosti počtu rodín s možnosťou paralelného trénovania klasifikátorov. Zvolil som dvojúrovňovú hierarchickú klasifikáciu, kde prvá úroveň klasifikuje proteíny do štrukturálnych tried a druhá úroveň do konkrétnych rodín v rámci štrukturálnej triedy vybranej v prvej úrovni. Výsledný nástroj implementujúci zvolený model teda obsahuje päť klasifikátorov.

Prvým krokom nájdenia vhodných klasifikátorov do hierarchického nástroja bol výber algoritmov strojového učenia a metód extrakcie príznakov z proteínov. Z metód na extrakciu príznakov som implementoval aminokyselinovú kompozíciu, rozdelenú aminokyselinovú kompozíciu, dvojice, vzdialené páry, globálne kódovanie a fyzikálno-chemické dvojice. Pri algoritmoch strojového učenia som vybral neurónové siete, náhodný les a Support Vector Machines. Implementáciu všetkých vybraných metód som skontroloval na dátových sadách použitých v článkoch z ktorých som čerpal.

Dôležitou časťou práce je popis tvorby vlastnej trénovacej a testovacej dátovej sady, použitej v experimentoch. Rodiny proteínov som čerpal zo svetovej databázy Pfam, pričom pri výbere konkrétnych rodín som sa riadil dvoma základnými kritériami. Prvé kritérium sa týkalo množstva dostupných proteínov v danej rodine, ktoré muselo dosahovať aspoň tisíc prvkov. Keďže sa v databáze Pfam vyskytujú nielen rodiny proteínov ale aj rodiny domén proteínov, druhým kritériom bolo vyberať čisto rodiny proteínov. Aby som mal rovnako zastúpené všetky štrukturálne triedy, vybral som z každej dvadsať rodín. Informácie o hierarchickom usporiadaní proteínov som čerpal z databázy SCOP (*Structural Classification of Proteins*). Výsledná dátová sada teda obsahovala osemdesiat rodín s 80 000 konkrétnymi proteínmi.

Experimenty som rozdelil do dvoch fáz. Prvá slúžila na zúženie počtu metód extrakcie príznakov. Na získanie výsledkov bola použitá podmnožina vytvorenej dátovej sady o veľkosti desiatich rodín z dôvodu veľkého množstva kombinácii algoritmov strojového učenia s metódami extrakcie. Pri použití osemdesiatich rodín by tieto experimenty viedli ku značnému časovému predĺženiu vyhodnotenia, ako aj ku nižšej flexibilite skúmania rôznych kombinácii parametrov strojového učenia. Experimenty som tvoril na dobre zastúpenej vyváženej dátovej sade (tisíc prvkov v každej rodine), ako aj na slabo zastúpenej vyváženej dátovej sade (sto prvkov v každej rodine) a na nevyváženej dátovej sade. Experimenty ukázali, že pri metódach s malým vektorom príznakov najlepšie dopadol algoritmus Support Vector Machines a najhoršie neurónové siete. Tieto metódy extrakcie mali taktiež väčší problém s nevyváženou dátovou sadou, pri ktorej často dochádzalo ku zanedbávaniu rodín s malým zastúpením. Pri metódach s väčším počtom príznakov najlepšie dopadol algoritmus neurónových sietí. Náhodný les dopadol v skoro všetkých prípadoch najhoršie, čo je pravdepodobne dôvod, prečo ho v už existujúcich nástrojoch na klasifikovanie proteínov vidno najmenej.

Druhá fáza experimentov slúžila na konkrétny výber kombinácií algoritmov strojového učenia a metód extrakcie príznakov pre jednotlivé klasifikátory hierarchického nástroja. Na rozdiel od prvej fázy, bola použitá plná dátová sada s osemdesiatimi rodinami. Najlepšie výsledky mala metóda vzdialených párov v kombinácii s neurónovými sieťami. Táto metóda na svoj výpočet potrebuje fyzikálno-chemické vlastnosti aminokyselín, ktoré som v práci čerpal z databázy GenomeNet. Najlepšie z vybraných a prezentovaných ôsmich fyzikálno-chemických vlastností dopadla *polarita aminokyselín* a *pravdepodobnosť vzniku alfa-šprály.*

Výsledný hierarchický nástroj dosiahol presnosť klasifikácie 0,951, čo je porovnateľné s nástrojom SECLAF. Je to zároveň výrazne lepší výsledok ako má nástroj GPCRTree, ktorého výsledná presnosť je o približne 10% horšia. To môžeme pripísať menšiemu počtu klasifikovaných rodín jedným klasifikátorom, lepšej metóde extrakcie príznakov ako aj použitiu efektívnejšieho algoritmu strojového učenia.

Ďalšie pokračovanie práce by sa mohlo uberať preskúmaním viacerých metód extrakcie príznakov, najmä tých, ktoré pracujú s maticovou reprezentáciou proteínov. Zaujímavé výsledky by mohli priniesť taktiež experimenty, ktoré by kombinovali prístup viacerých metód extrakcie príznakov v jednom klasifikátori, prípadne rozšírenie hierarchického nástroja na klasifikáciu vo všetkých úrovniach hierarchie proteínov od štrukturálnych tried, foldov, superrodín až po rodiny.

Literatúra

[1] Abadi, M.; Agarwal, A.; Barham, P.; aj.: TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems. 2015, software available from tensorflow.org, [Online; navštívené 10.11.2019].
URL https://www.tensorflow.org/

[2] Andreeva, A.; Howorth, D.; Chothia, C.; aj.: SCOP2 prototype: a new approach to protein structure mining. *Nucleic Acids Research*, ročník 42, č. D1, 11 2013: s. D310–D314, ISSN 0305-1048, doi:10.1093/nar/gkt1242, [Online; navštívené 15.01.2020],
https://academic.oup.com/nar/article-pdf/42/D1/D310/3648598/gkt1242.pdf.
URL https://doi.org/10.1093/nar/gkt1242

[3] Andreeva, A.; Kulesha, E.; Gough, J.; aj.: The SCOP database in 2020: expanded classification of representative family and superfamily domains of known protein structures. *Nucleic Acids Research*, ročník 48, č. D1, 11 2019: s. D376–D382, ISSN 0305-1048, doi:10.1093/nar/gkz1064, [Online; navštívené 15.01.2020],
https://academic.oup.com/nar/article-pdf/48/D1/D376/31697305/gkz1064.pdf.
URL https://doi.org/10.1093/nar/gkz1064

[4] Bisong, E.: *Principal Component Analysis (PCA)*. 09 2019, ISBN 978-1-4842-4469-2, s. 319–324, doi:10.1007/978-1-4842-4470-8_26.

[5] Bornemann, D.; Miller, E.; Simon, J.: Expression and properties of wild-type and mutant forms of the Drosophila sex comb on midleg (SCM) repressor protein. *Genetics*, ročník 150, č. 2, October 1998: str. 675—686, ISSN 0016-6731.
URL https://europepmc.org/articles/PMC1460340

[6] Breiman, L.: Random forests, machine learning 45. *Journal of Clinical Microbiology*, ročník 2, 01 2001.

[7] Brown, J.; Couillard-Després, S.; Cooper-Kuhn, C.; aj.: Transient expression of doublecortin during adult neurogenesis. *The Journal of comparative neurology*, ročník 467, 01 2004: s. 1–10, doi:10.1002/cne.10874.

[8] Center, K. U. B.: Amino acid indices, substitution matrices and pair-wise contact potentials. 2017, [Online; navštívené 05.04.2020].
URL https://www.genome.jp/dbget/aaindex.html

[9] Cortes, C.; Vapnik, V.: Support Vector Networks. *Machine Learning*, 01 1995, doi:10.1023/A:1022627411411.

[10] Csc, D.; Warwick, K.; Kurková, V.: *The Psychological Limits of Neural Computation*. 01 1998, doi:10.1007/978-1-4471-1523-6_17.

[11] Dayhoff, M.; Schwartz, R.; Orcutt, B.: A Model of Evolutionary Change in Proteins. *A Model of Evolutionary Change in Proteins*, ročník 4, 01 1978.

[12] El-Gebali, S.; Mistry, J.; Bateman, A.; aj.: The Pfam protein families database in 2019. *Nucleic acids research*, ročník 47, 10 2018, doi:10.1093/nar/gky995.

[13] Feng, Y.; Jiao, W.; Fu, X.; aj.: Stepwise disassembly and apparent nonstepwise reassembly for the oligomeric RbsD protein. *Protein science : a publication of the Protein Society*, ročník 15, č. 6, June 2006: str. 1441—1448, ISSN 0961-8368, doi:10.1110/ps.062175806.
URL https://europepmc.org/articles/PMC2242537

[14] Gribskov, M.; McLachlan, A.; Eisenberg, D.: *Profile analysis: detection of distantly related proteins.* Proceedings of the National Academy of Sciences of the United States of America., 1987, ISBN 84(13):4355–4358.

[15] Guo, J.; Lin, Y.; Sun, Z.: A novel method for protein subcellular localization: Combining residue-couple model and SVM. 01 2005, doi:10.1142/9781860947322_0012.

[16] Hajian-Tilaki, K.: Receiver Operating Characteristic (ROC) Curve Analysis for Medical Diagnostic Test Evaluation. *Caspian journal of internal medicine*, ročník 4, 09 2013.

[17] Han, J.; Kamber, M.; Pei, J.: *Data Mining: Concepts and Techniques*. 01 2012, doi:10.1016/C2009-0-61819-5.

[18] Han, J.; Kamber, M.; Pei, J.: *Data Mining: Concepts and Techniques*. 01 2012, doi:10.1016/C2009-0-61819-5.

[19] Hemler, M.: Hemler METetraspanin functions and associated microdomains. Nat Rev Mol Cell Biol 6:801-811. *Nature reviews. Molecular cell biology*, ročník 6, 11 2005: s. 801–11, doi:10.1038/nrm1736.

[20] Javed Iqbal, M.; Faye, I.; Brahim, S.; aj.: Efficient Feature Selection and Classification of Protein Sequence Data in Bioinformatics. *TheScientificWorldJournal*, ročník 2014, 06 2014, doi:10.1155/2014/173869.

[21] Jeong, J. C.; Lin, X.; Chen, X.-W.: On Position-Specific Scoring Matrix for Protein Function Prediction. *IEEE/ACM transactions on computational biology and bioinformatics / IEEE, ACM*, ročník 8, 03 2011, doi:10.1109/TCBB.2010.93.

[22] Kawashima, S.; Kanehisa, M.: *AAindex: Amino Acid index database.* [Online; navštívené 14.11.2019].
URL https://www.ncbi.nlm.nih.gov/pmc/articles/PMC102411/

[23] Kruse, R.; Borgelt, C.; Braune, C.; aj.: *Introduction to Neural Networks*. 09 2016, ISBN 978-1-4471-7294-9, s. 9–13, doi:10.1007/978-1-4471-7296-3_2.

[24] Kumar, R.; Kumari, B.; Kumar, M.: Prediction of endoplasmic reticulum resident proteins using fragmented amino acid composition and support vector machine. *PeerJ*, ročník 5, 09 2017: str. e3561, doi:10.7717/peerj.3561.

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

[26] Li, X.; Liao, B.; Shu, Y.; aj.: Protein functional class prediction using global encoding of amino acid sequence. *Journal of theoretical biology*, ročník 261, 08 2009, doi:10.1016/j.jtbi.2009.07.017.

[27] Lodish, H.; Berk, A.; Zipursky, S.; aj.: *Molecular Cell Biology. 4th edition.* New York: W. H. Freeman, 2000, ISBN 0-7167-3136-3.

[28] Mansoori, E.; Jahromi, M.; Katebi, S.: Protein Superfamily Classification Using Fuzzy Rule-Based Classifier. *IEEE transactions on nanobioscience*, 03 2009, doi:10.1109/TNB.2008.2011901.

[29] Mcculloch, W.; Pitts, W.: A Logical Calculus of Ideas Immanent in Nervous Activity"Bulletin of Mathematical Biophysics. 1994.

[30] Moss, D.; Davies, M.; Secker, A.; aj.: GPCRTree: Online hierarchical classification of GPCR function. *BMC Research Notes*, 09 2008, doi:10.1186/1756-0500-1-67.

[31] Murphy, L.; Wallqvist, A.; Levy, R.: Simplified amino acid alphabets for protein fold recognition and implications for folding. *Protein engineering*, ročník 13, 04 2000, doi:10.1093/protein/13.3.149.

[32] Nanni, L.; Brahnam, S.; Lumini, A.: High performance set of PseAAC and sequence based descriptors for protein classification. *Journal of theoretical biology*, 09 2010, doi:10.1016/j.jtbi.2010.06.006.

[33] Nanni, L.; Brahnam, S.; Lumini, A.: Wavelet images and Chou's pseudo amino acid composition for protein classification. *Amino Acids*, ročník 43, 08 2011, doi:10.1007/s00726-011-1114-9.

[34] Nanni, L.; Lumini, A.: An ensemble of K-Local Hyperplane for predicting Protein-Protein interactions. *Bioinformatics (Oxford, England)*, ročník 22, 06 2006, doi:10.1093/bioinformatics/btl055.

[35] Nanni, L.; Lumini, A.; Brahnam, S.: *An Empirical Study of Different Approaches for Protein Classification.* [Online; navštívené 10.09.2019].
URL https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4084589/#B59

[36] Oliphant, T.: *Guide to NumPy.* 12 2006, 7 s.

[37] Olivetti, E.; Greiner, S.; Avesani, P.: Statistical independence for the evaluation of classifier-based diagnosis. *Brain Informatics*, ročník 2, 12 2014, doi:10.1007/s40708-014-0007-6.

[38] Page , G.: Introduction to Machine Learning, by Ethem Alpaydin. *Robotica*, ročník 24, 01 2006, doi:10.1017/S0263574705212584.

[39] Patel, N.; Sarraf, E.; Tsai, M.: The Curse of Dimensionality. *Anesthesiology*, ročník 129, 09 2018, doi:10.1097/ALN.0000000000002350.

[40] Pearson, W.: Selecting the Right Similarity-Scoring Matrix. *Current protocols in bioinformatics / editoral board, Andreas D. Baxevanis ... [et al.]*, ročník 43, 10 2013: s. 3.5.1–3.5.9, doi:10.1002/0471250953.bi0305s43.

[41] Pedregosa, F.; Varoquaux, G.; Gramfort, A.; aj.: Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research*, ročník 12, 2011: s. 2825–2830.

[42] Ramana, J.; Gupta, D.: LipocalinPred: A SVM-based method for prediction of lipocalins. *BMC bioinformatics*, 12 2009, doi:10.1186/1471-2105-10-445.

[43] Raux, E.; Schubert, H.; Warren, M.: Biosynthesis of cobalamin (vitamin B12): A bacterial conundrum. *Cellular and molecular life sciences : CMLS*, ročník 57, 01 2001: s. 1880–93, doi:10.1007/PL00000670.

[44] Russell; Norvig, S.: Artificial Intelligence: A Modern Approach. *Prentice Hall, Englewood Cliffs, NJ*, 01 2010.

[45] Ryu, K.-S.; Kim, J.-I.; Cho, S.-J.; aj.: Structural insights into the monosaccharide specificity of Escherichia coli rhamnose mutarotase. *Journal of molecular biology*, ročník 349, č. 1, May 2005: str. 153—162, ISSN 0022-2836, doi:10.1016/j.jmb.2005.03.047, [Online; navštívené 07.11.2019].
URL https://doi.org/10.1016/j.jmb.2005.03.047

[46] Soderberg, T.: *Organic Chemistry with a Biological Emphasis*. The LibreTexts libraries, 2019.

[47] Song, Y.-Y.; Lu, Y.: Decision tree methods: applications for classification and prediction. *Shanghai archives of psychiatry*, 04 2015, doi:10.11919/j.issn.1002-0829.215044.

[48] Szalkai, B.; Grolmusz, V.: SECLAF: A Webserver and Deep Neural Network Design Tool for Biological Sequence Classification. *Bioinformatics (Oxford, England)*, 08 2017, doi:10.1093/bioinformatics/bty116.

[49] Tantoso, E.; Li, K.-B.: AAIndexLoc: Predicting subcellular localization of proteins based on a new representation of sequences using amino acid indices. *Amino acids*, ročník 35, 08 2008, doi:10.1007/s00726-007-0616-y.

[50] Telser, A.: Molecular Biology of the Cell, 4th Edition. *Shock*, ročník 18, 09 2002, doi:10.1097/00024382-200209000-00015.

[51] Vacchina, V.; Bierla, K.; Szpunar, J.; aj.: *Quantification of SeMet and SeCys in Biological Fluids and Tissues by Liquid Chromatography Coupled to Inductively Coupled Plasma Mass Spectrometry (HPLC-ICP MS).* [Online; navštívené 15.10.2019].
URL https://www.ncbi.nlm.nih.gov/pubmed/28917043

[52] Vella, F.: Outlines of Biochemistry, 5th Edition. *Biochemical Education*, ročník 15, 06 2010: s. 217 – 217, doi:10.1016/0307-4412(87)90020-3.

[53] Vieira, S.; Pinaya, W.; Mechelli, A.: *Introduction to machine learning*. 11 2019, ISBN 978-0-12-815739-8, doi:10.1016/B978-0-12-815739-8.00001-8.

[54] Whitford, D.: *Proteins: Structure and Function*. Wiley, 2005, ISBN 978-1-118-68572-3.

[55] Wu, C.; Whitson, G.; Mclarty, J.; aj.: Protein classification artificial neural system. *Protein science : a publication of the Protein Society*, 1992, doi:10.1002/pro.5560010512.

[56] Yona, G.: *Introduction to computational proteomics*. 01 2010, 1-737 s., doi:10.1201/9781420010770.

[57] Yu, X.; Zheng, X.; Liu, T.; aj.: *Predicting subcellular location of apoptosis proteins with pseudo amino acid composition: approach from amino acid substitution matrix and auto covariance transformation. Amino Acids.* [Online; navštívené 01.02.2020]. URL https://www.ncbi.nlm.nih.gov/pubmed/21344173

[58] Zeng, Y.-h.; Guo, Y.-z.; Xiao, R.; aj.: Using the augmented Chou's pseudo amino acid composition for predicting protein submitochondria locations based on auto covariance approach. *Journal of theoretical biology*, 04 2009, doi:10.1016/j.jtbi.2009.03.028.

[59] Zhang, Y.; Cronan, J.: Transcriptional analysis of essential genes of the Escherichia coli fatty acid biosynthesis gene cluster by functional replacement with the analogous Salmonella typhimurium gene cluster. *Journal of bacteriology*, ročník 180, č. 13, July 1998: str. 3295—3303, ISSN 0021-9193.
URL https://europepmc.org/articles/PMC107281

Príloha A

Obsah priloženého DVD

Na priloženom DVD sa nachádzajú nasledovné súbory a zložky:

- **xdekre00-DP.pdf** text diplomovej práce vo formáte PDF.
- **tex/** zdrojové kódy vo formáte LaTeX.
- **src/** zdrojové kódy implementovaných programov.
  - **main_program/** zdrojové kódy programu na extrakciu príznakov.
    - **README.md** stručný návod na použitie.
    - **sample_train/** vzorka štyroch rodín proteínov na vyskúšanie extrakcie príznakov a trénovanie modelu.
    - **sample_classify/** vzorka proteínov na vyskúšanie vytrénovaneho modelu.
  - **hierarchical_tool/** zdrojové kódy hierarchického nástroja.
    - **models/** klasifikačné modely použité v hierarchickom nástroji.
    - **README.md** stručný návod na použitie.
    - **sample_classify/** vzorka proteínov na vyskúšanie hierarchického nástroja.
- **research_datasets/** dátová sada použitá v referenčných článkoch.
  - **AK_D2_RAK_dataset/** dátová sada použitá na overenie implementácie aminokyselinovej kompozície, dvojíc a rozdelenej aminokyselinovej kompozície.
  - **GE_PCD2_VP_dataset/** dátová sada použitá na overenie implementácie globálneho kódovania, fyzikálno-chemických dvojíc a vzdialených párov.
- **my_datasets/** dátové sady použité v mojich experimentoch vo FASTA formáte.
  - **database_10/** dátová sada s desiatimi rodinami stiahnutými z databázy Pfam použitá na zúženie výberu metód extrakcie príznakov.
    - **full/** súbory všetkých rodín v plnej veľkosti.
    - **scaled_to_1000/** súbory všetkých rodín so zastúpením 1000 proteínov.
  - **database_80/** dátová sada s osemdesiatimi rodinami stiahnutými z databázy Pfam použitá na výber klasifikátorov do hierarchického nástroja.
    - **alpha/** rodiny alfa štrukturálnej triedy.
      - **full/** súbory rodín v plnej veľkosti.

    - **scaled_to_1000/** súbory rodín so zastúpením 1000 proteínov.
- **alpha-beta/** rodiny alfa štrukturálnej triedy.
    - **full/** súbory rodín v plnej veľkosti.
    - **scaled_to_1000/** súbory rodín so zastúpením 1000 proteínov.
- **alpha+beta/** rodiny alfa štrukturálnej triedy.
    - **full/** súbory rodín v plnej veľkosti.
    - **scaled_to_1000/** súbory rodín so zastúpením 1000 proteínov.
- **beta/** rodiny alfa štrukturálnej triedy.
    - **full/** súbory rodín v plnej veľkosti.
    - **scaled_to_1000/** súbory rodín so zastúpením 1000 proteínov.

Príloha B

Parametre algoritmov strojového učenia použité pri experimentoch

Jednotlivé parametre znamenajú:

- **hidden layer nodes:** počet neurónov v každej skrytej vrstve neurónovej siete
- **batch size:** veľkosť dávky pri optimalizácii učenia neurónovej siete
- **patience:** počet po sebe idúcich iterácii, v ktorých sa môže zhoršiť stratová funkcia neurónovej siete

	hidden layer nodes	**batch size**	**patience**
AK	65	1000	3
RAK	70	1000	3
VP	65	1000	2
D2	65	1000	3
PCD2	60	1000	2
GE	50	2000	5

Tabuľka B.1: Parametre použité pri trénovaní vybraných metód extrakcie príznakov algoritmom neurónových sietí.

- **C:** cena nesprávnej klasifikácie prvkov trénovacej dátovej sady algoritmu Support Vector Machines
- **gamma:** vplyv jedného trénovacieho prvku na výsledný model klasifikátora

	C	**gamma**		**C**	**gamma**
AK	80	2,5	D2	30	1,5
RAK	10	1,5	PCD2	-	-
VP	30	1	GE	90	3

Tabuľka B.2: Parametre použité pri trénovaní vybraných metód extrakcie príznakov algoritmom Support Vector Machines.

- **max depth:** maximálna hĺbka jednotlivých stromov
- **min samples split:** minimálne množstvo prvkov v uzle stromu aby bolo možné uzol rozdeliť
- **min samples leaf:** minimálne množstvo prvkov, ktoré môže byť v liste stromu

	max depth	**min samples split**	**min samples leaf**
všetky použité metódy extrakcie kompletná dátová sada	None	10	10
všetky použité metódy extrakcie redukovaná dátová sada	None	2	1

Tabuľka B.3: Parametre použité pri trénovaní vybraných metód extrakcie príznakov algoritmom náhodného lesa.

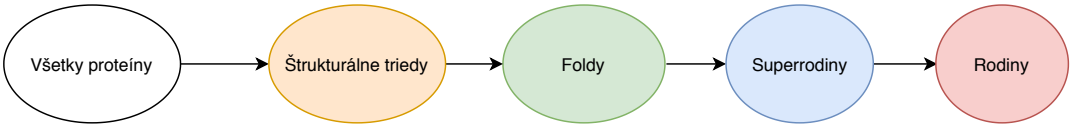
Príloha C

Dátové sady použité v diplomovej práci

C.1 Dátová sada použitá na experimenty pre výber metód extrakcie príznakov

#	**Rodina**	**Superrodina**	**Fold**	**Štrukturálna trieda**	**Pfam kód**
1	DOCK_N	DOCK alpha-helical	Immunoglobulin/ albumin-binding	Alfa	PF16172
2	Tetraspanin	Tetraspanin	Tetraspanin		PF00335
3	DCX	Doublecortin	Beta-Grasp	Alfa + Beta	PF03607
4	RhaM	Dimeric alpha+beta barrel	Ferredoxin		PF05336
5	Cas_Cas5d	CRISPR-associated proteins			PF09704
6	HEM4	HemD	HemD	Alfa / Beta	PF02602
7	FA_synthesis	Isopropylmalate dehydrogenase	Isopropylmalate dehydrogenase		PF02504
8	RbsD_FucU	RbsD	RbsD		PF05025
9	MBT	Tudor/ PWWP/MBT	SH3 barrel	Beta	PF02820
10	Ribosomal _L19	Translation proteins SH3			PF01245

Tabuľka C.1: Rodiny použité ako dátová sada v experimentoch pre výber metód extrakcie príznakov.



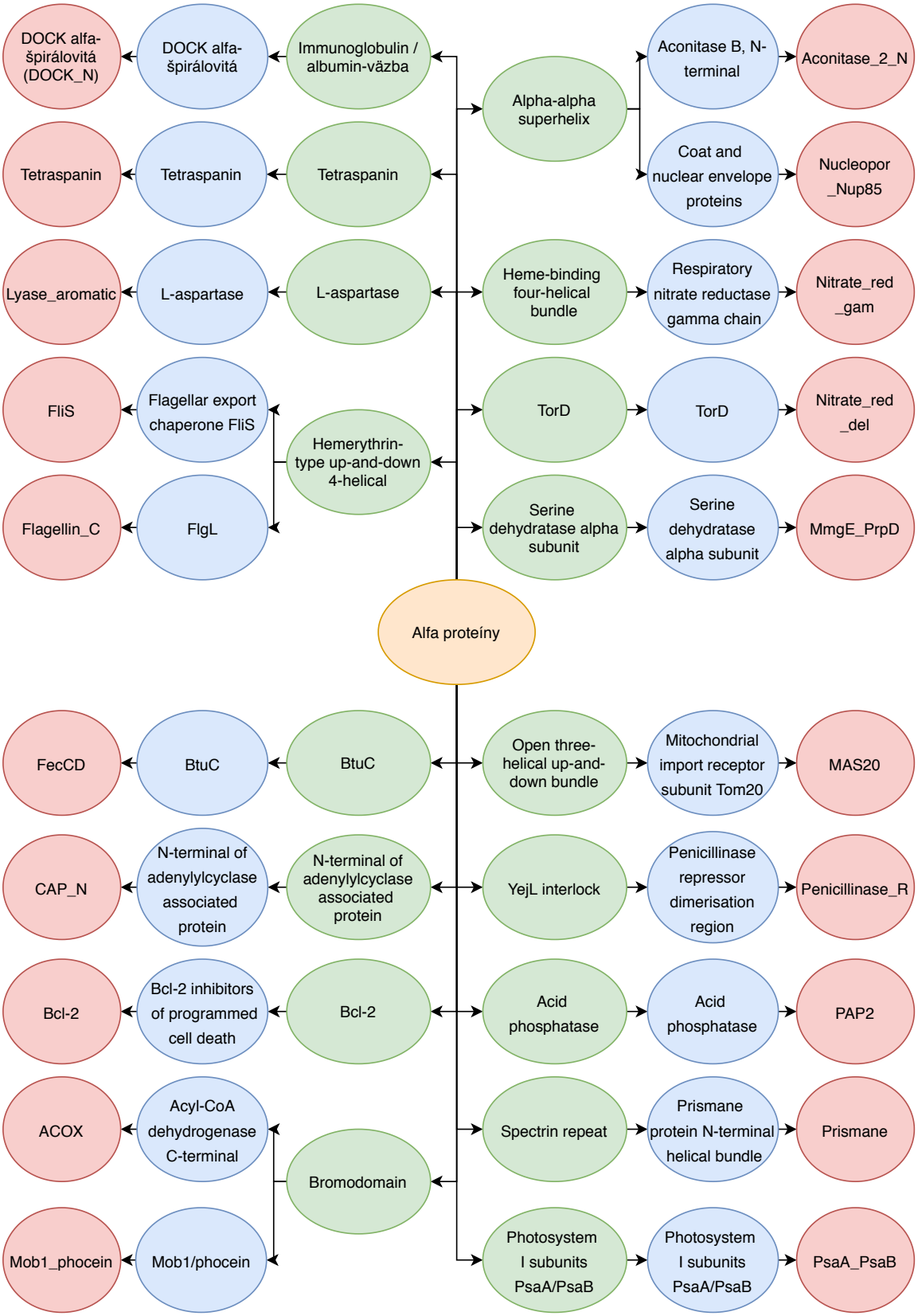
Obr. C.1: Stromová hierarchia proteínov.

Obr. C.2: Strom rodín proteínov použitých ako dátová sada v experimentoch pre výber metód extrakcie príznakov.

C.2 Dátová sada použitá na výber klasifikátorov do hierarchického nástroja

#	Rodina	Superrodina	Fold	Pfam kód
1	DOCK_N	DOCK alpha-helical	Immunoglobulin/ albumin-binding	PF16172
2	Tetraspanin	Tetraspanin	Tetraspanin	PF00335
3	Lyase_aromatic	L-aspartase	L-aspartase	PF00221
4	FliS	Flagellar export chaperone FliS	Hemerythrin-type up-and-down 4-helical	PF02561
5	Flagellin_C	FlgL		PF00700
6	FecCD	BtuC	BtuC	PF01032
7	CAP_N	N-terminal of adenylylcyclase associated protein	N-terminal of adenylylcyclase associated protein	PF01213
8	Bcl-2	Bcl-2 inhibitors of programmed cell death	Bcl-2	PF00452
9	ACOX	Acyl-CoA dehydrogenase C-terminal	Bromodomain	PF01756
10	Mob1_phocein	Mob1/phocein		PF03637
11	Aconitase_2_N	Aconitase B, N-terminal	Alpha-alpha superhelix	PF06434
12	Nucleopor_Nup85	Coat and nuclear envelope proteins		PF07575
13	Nitrate_red_gam	Respiratory nitrate reductase 1 gamma chain	Heme-binding four-helical bundle	PF02665
14	Nitrate_red_del	TorD	TorD	PF02613
15	MmgE_PrpD	Serine dehydratase alpha subunit	Serine dehydratase alpha subunit	PF03972
16	MAS20	Mitochondrial import receptor subunit Tom20	Open three-helical up-and-down bundle	PF02064
17	Penicillinase_R	Penicillinase repressor dimerisation region	YejL interlock	PF03965
18	PAP2	Acid phosphatase/ Vanadium-dependent haloperoxidase	Acid phosphatase/ Vanadium-dependent haloperoxidase	PF01569
19	Prismane	Prismane protein N-terminal helical bundle	Spectrin repeat	PF03063
20	PsaA_PsaB	Photosystem I subunits PsaA/PsaB	Photosystem I subunits PsaA/PsaB	PF00223

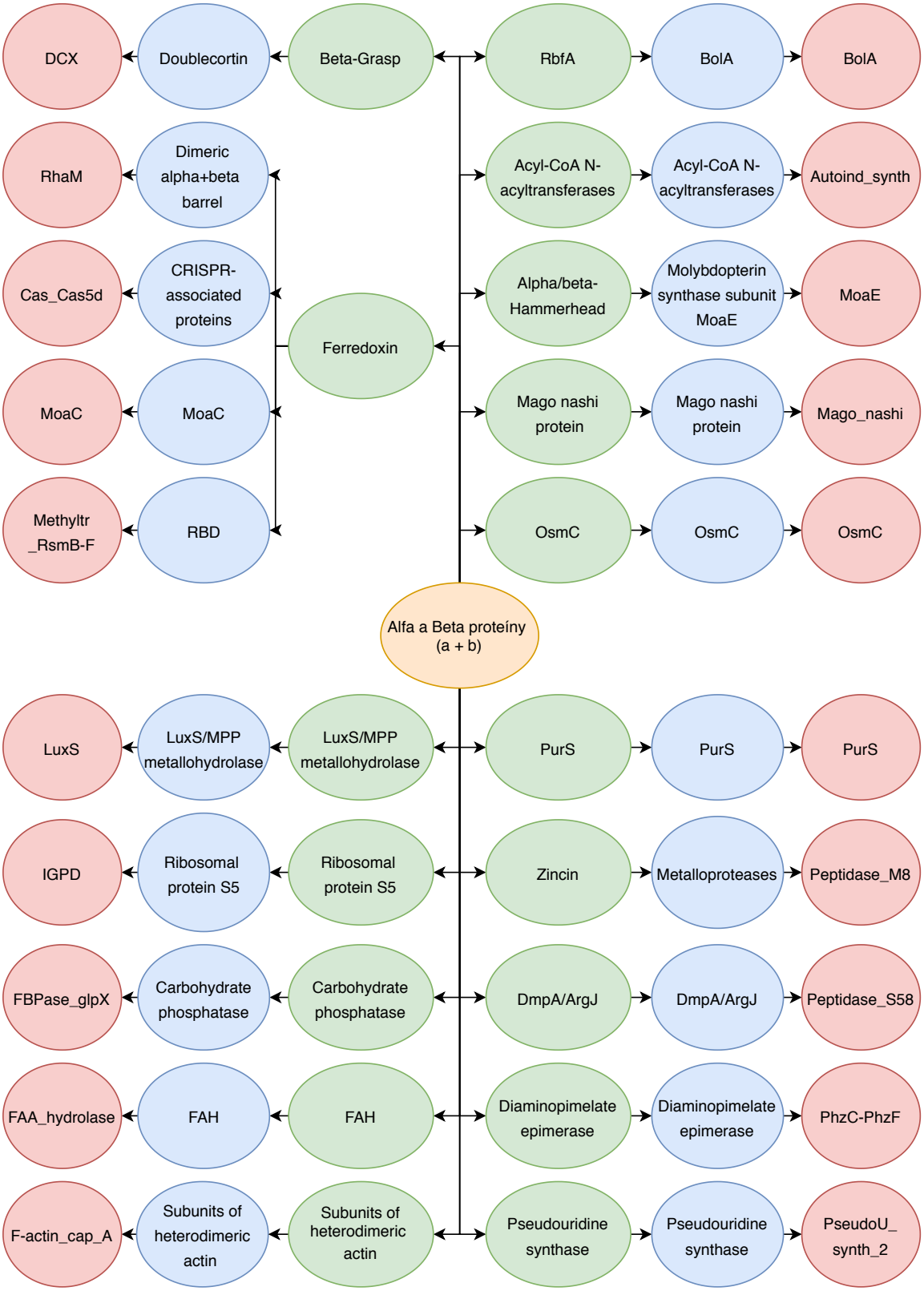
Tabuľka C.2: Rodiny alfa štrukturálnej triedy použité ako dátová sada v experimentoch na výber klasifikátorov do hierarchického nástroja.



Obr. C.3: Strom rodín alfa štrukturálnej triedy použitých ako dátová sada v experimentoch na výber klasifikátorov do hierarchického nástroja.

#	Rodina	Superrodina	Fold	Pfam kód
1	DCX	Doublecortin	Beta-Grasp	PF03607
2	rhaM	Dimeric alpha+beta barrel	Ferredoxin	PF05336
3	Cas_Cas5d	CRISPR-associated proteins		PF09704
4	MoaC	Molybdenum cofactor biosynthesis protein C, MoaC		PF01967
5	Methyltr_RsmB-F	RBD		PF01189
6	LuxS	LuxS/MPP metallohydrolase	LuxS/MPP metallohydrolase	PF02664
7	IGPD	Ribosomal protein S5	Ribosomal protein S5	PF00475
8	FBPase_glpX	Carbohydrate phosphatase	Carbohydrate phosphatase	PF03320
9	FAA_hydrolase	FAH	FAH	PF01557
10	F-actin_cap_A	Subunits of heterodimeric actin filament capping protein Capz	Subunits of heterodimeric actin filament capping protein Capz	PF01267
11	BolA	BolA	RbfA	PF01722
12	Autoind_synth	Acyl-CoA N-acyltransferases	Acyl-CoA N-acyltransferases	PF00765
13	MoaE	Molybdopterin synthase subunit MoaE	Alpha/beta-Hammerhead	PF02391
14	Mago_nashi	Mago nashi protein	Mago nashi protein	PF02792
15	OsmC	OsmC	OsmC	PF02566
16	PurS	PurS	PurS	PF02700
17	Peptidase_M8	Metalloproteases	Zincin	PF01457
18	Peptidase_S58	DmpA/ArgJ	DmpA/ArgJ	PF03576
19	PhzC-PhzF	Diaminopimelate epimerase	Diaminopimelate epimerase	PF02567
20	PseudoU_synth_2	Pseudouridine synthase	Pseudouridine synthase	PF00849

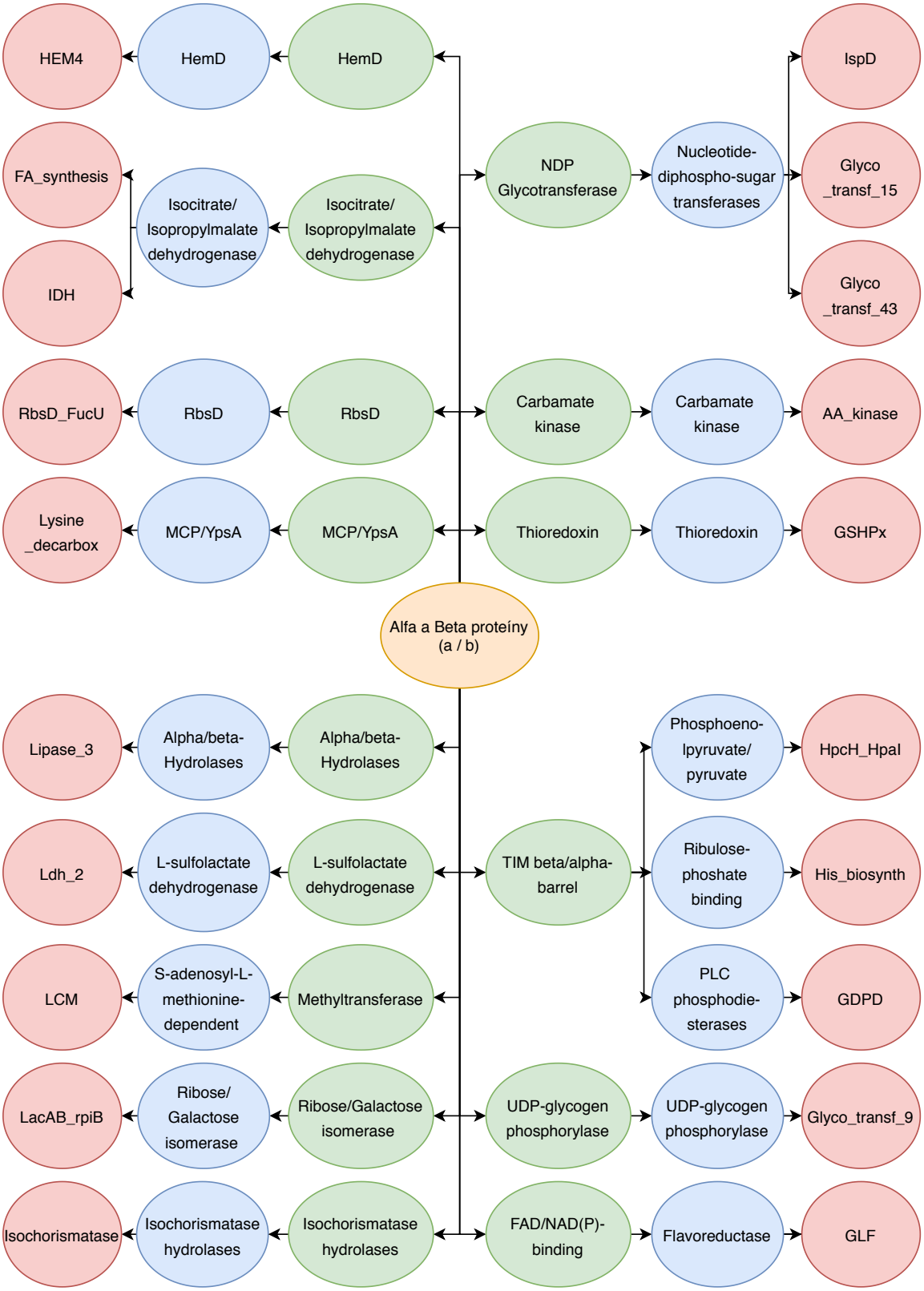
Tabuľka C.3: Rodiny alfa+beta štrukturálnej triedy použité ako dátová sada v experimentoch na výber klasifikátorov do hierarchického nástroja.



Obr. C.4: Strom rodín alfa+beta štrukturálnej triedy použitých ako dátová sada v experimentoch na výber klasifikátorov do hierarchického nástroja.

#	**Rodina**	**Superrodina**	**Fold**	**Pfam kód**
1	HEM4	HemD	HemD	PF02602
2	FA_synthesis	Isopropylmalate dehydrogenase	Isopropylmalate dehydrogenase	PF02504
3	IDH			PF03971
4	RbsD_FucU	RbsD	RbsD	PF05025
5	Lysine_decarbox	MCP/YpsA	MCP/YpsA	PF03641
6	Lipase_3	alpha/beta-Hydrolases	alpha/beta-Hydrolases	PF01764
7	Ldh_2	L-sulfolactate dehydrogenase	L-sulfolactate dehydrogenase	PF02615
8	LCM	S-adenosyl-L-methionine-dependent methyltransferases	Methyltransferase	PF04072
9	LacAB_rpiB	Ribose/Galactose isomerase RpiB/AlsB	Ribose/Galactose isomerase RpiB/AlsB	PF02502
10	IspD	Nucleotide-diphospho-sugar transferases	NDP Glycotransferase	PF01128
11	Glyco_transf_15			PF01793
12	Glyco_transf_43			PF03360
13	Isochorismatase	Isochorismatase hydrolases	Isochorismatase hydrolases	PF00857
14	HpcH_HpaI	Phosphoenolpyruvate /pyruvate	TIM beta/ alpha-barrel	PF03328
15	His_biosynth	Ribulose-phoshate binding		PF00977
16	GDPD	PLC phosphodiesterases		PF03009
17	AA_kinase	Carbamate kinase	Carbamate kinase	PF00696
18	GSHPx	Thioredoxin	Thioredoxin	PF00255
19	Glyco_transf_9	UDP-Glycosyltransferase /glycogen phosphorylase	UDP-Glycosyltransferase /glycogen phosphorylase	PF01075
20	GLF	Flavoreductase	FAD/NAD(P)-binding	PF03275

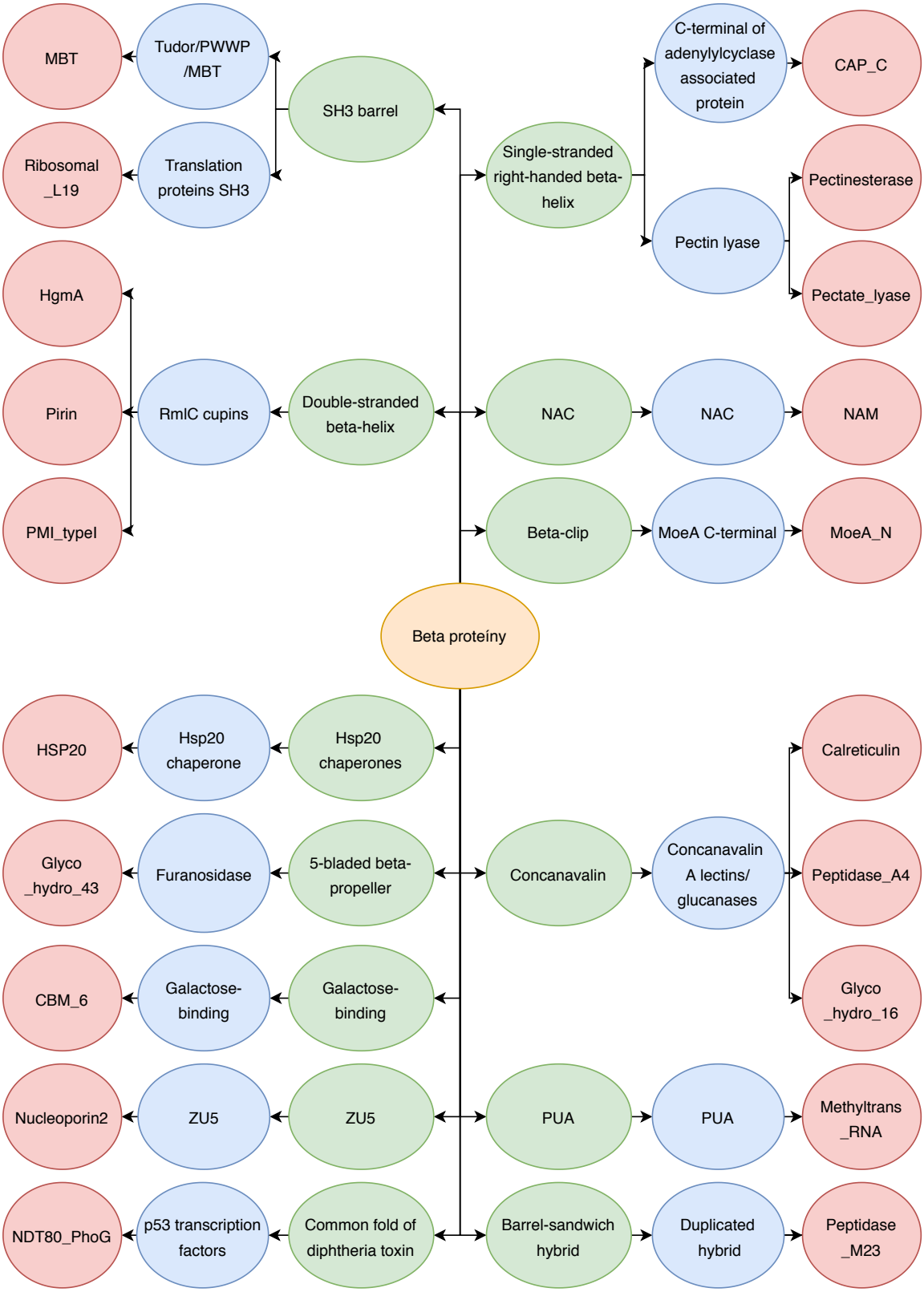
Tabuľka C.4: Rodiny alfa/beta štrukturálnej triedy použité ako dátová sada v experimentoch na výber klasifikátorov do hierarchického nástroja.



Obr. C.5: Strom rodín alfa/beta štrukturálnej triedy použitých ako dátová sada v experimentoch na výber klasifikátorov do hierarchického nástroja.

#	Rodina	Superrodina	Fold	Pfam kód
1	MBT	Tudor/PWWP /MBT	SH3 barrel	PF02820
2	Ribosomal_L19	Translation proteins SH3		PF01245
3	HSP20	Hsp20 chaperone	Hsp20 chaperones	PF00011
4	HgmA	RmlC cupins	Double-stranded beta-helix	PF04209
5	Pirin			PF02678
6	PMI_typeI			PF01238
7	Glyco_hydro_43	Furanosidase	5-bladed beta-propeller	PF04616
8	CBM_6	galactose-binding	Galactose-binding	PF03422
9	CAP_C	C-terminal of adenylylcyclase associated protein	Single-stranded right-handed beta-helix	PF08603
10	Pectinesterase	Pectin lyase		PF01095
11	Pectate_lyase			PF03211
12	Calreticulin	Concanavalin A lectins/ glucanases	Concanavalin	PF00262
13	Peptidase_A4			PF01828
14	Glyco_hydro_16			PF00722
15	Nucleoporin2	ZU5	ZU5	PF04096
16	NDT80_PhoG	p53-like transcription factors	Common fold of diphtheria toxin /transcription factors/cytochrome	PF05224
17	NAM	NAC	NAC	PF02365
18	MoeA_N	MoeA C-terminal	Beta-clip	PF03453
19	Methyltrans_RNA	PUA	PUA	PF04452
20	Peptidase_M23	Duplicated hybrid	Barrel-sandwich hybrid	PF01551

Tabuľka C.5: Rodiny beta štrukturálnej triedy použité ako dátová sada v experimentoch na výber klasifikátorov do hierarchického nástroja.



Obr. C.6: Strom rodín beta štrukturálnej triedy použitých ako dátová sada v experimentoch na výber klasifikátorov do hierarchického nástroja.