VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV INFORMAČNÍCH SYSTÉMŮ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF INFORMATION SYSTEMS

STROJOVÉ UČENÍ V ÚLOZE PREDIKCE VLIVU AMINOKYSELINOVÝCH MUTACÍ NA STABILITU PROTEINU

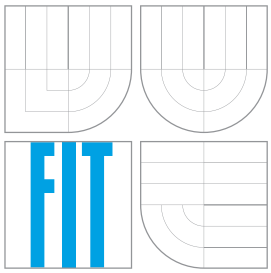
DIPLOMOVÁ PRÁCE
MASTER'S THESIS

AUTOR PRÁCE
AUTHOR

Bc. FRANTIŠEK MALINKA

BRNO 2014

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV INFORMAČNÍCH SYSTÉMŮ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF INFORMATION SYSTEMS

STROJOVÉ UČENÍ V ÚLOZE PREDIKCE VLIVU AMINOKYSELINOVÝCH MUTACÍ NA STABILITU PROTEINU

PREDICTION OF PROTEIN STABILITY UPON MUTATIONS USING MACHINE LEARNING

DIPLOMOVÁ PRÁCE
MASTER'S THESIS

AUTOR PRÁCE — Bc. FRANTIŠEK MALINKA
AUTHOR

VEDOUCÍ PRÁCE — Ing. JAROSLAV BENDL
SUPERVISOR

BRNO 2014

Abstrakt

Tato práce popisuje nový přístup k predikci vlivu aminokyselinových mutací na změnu stability proteinu. Cílem je vytvořit nový meta-nástroj, který kombinuje výstupy osmi vybraných nástrojů, díky čemuž je schopen svoji predikční schopnost zlepšit. Pro nalezení optimálního konsenzu mezi těmito nástroji je použito různých metod strojového učení. Ze všech testovaných metod strojového učení dosahuje KStar nejvyšší úspěšnosti predikce na trénovacím datasetu tvořeného experimentálně ověřenými mutacemi z databáze ProTherm. Právě z tohoto důvodu je KStar vybrán jako optimální predikční technika. Pro prokázání korektnosti výsledků tohoto meta-nástroje je použito testovacího datasetu vytvořeného ojedinělým způsobem, a to z vícebodových mutací extrahovaných taktéž z databáze ProTherm. Jelikož nebyly vícebodové mutace použity pro natrénování žádného z integrovaných nástrojů, předpokládá se, že takovéto porovnání je objektivní. Ve výsledku se tímto přístupem podařilo pomocí metody strojového učení KStar zvýšit korelační koeficient na trénovacím datasetu o 0,130, respektive o 0,239 na datasetu testovacím oproti nejúspěšnějšímu integrovanému nástroji. Na základě zjištěných údajů je možné říci, že metody strojového učení jsou vhodnými technikami pro problémy z oblasti proteinových predikcí.

Abstract

This thesis describes a new approach to the detection of protein stability change upon amino acid mutations. The main goal is to create a new meta-tool, which combines the outputs of eight well-established prediction tools and due to suitable method of consensus making, it is able to improve the overall prediction accuracy. The optimal strategy of combination of outputs of these tools is found by using a various number of machine learning methods. From all tested machine learning methods, KStar showed the highest prediction accuracy on the training dataset compiled from experimentally validated mutations originating from ProTherm database. Due to this reason, it is chosen as an optimal prediction technique. The general prediction abilities is validated on the testing dataset composed of multi-point amino acid mutations extracted also from ProTherm database. Since the multi-point mutations were not used for training any of integrated tools, we suppose that such comparison is objective. As a result, the developed meta-tool based on KStar technique improves the correlation coefficient about 0.130 on the training dataset and 0.239 on the testing dataset, respectively (the comparison is being made against the most succesful integrated tool). Based on the obtained results, it is possible to claim that machine learning methods are suitable technique for the problems from area of protein predictions.

Klíčová slova

Predikce stability, stabilita proteinu, strojové učení, mutace proteinu, protherm.

Keywords

Stability prediction, protein stability, machine learning, protein mutation, protherm.

Citace

František Malinka: Strojové učení v úloze predikce vlivu aminokyselinových mutací na stabilitu proteinu, diplomová práce, Brno, FIT VUT v Brně, 2014

Strojové učení v úloze predikce vlivu aminokyselinových mutací na stabilitu proteinu

Prohlášení

Prohlašuji, že jsem tuto diplomovou práci vypracoval samostatně pod vedením pana Ing. Jaroslava Bendla.Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

.....
František Malinka
27. května 2014

Poděkování

Tímto bych chtěl poděkovat panu Ing. Jaroslavu Bendlovi za odborné vedení, jeho cenné rady a připomínky, které mi pomohly tuto diplomovou práci sepsat a prezentovat.

Obsah

Kapitola 1

Úvod

Proteiny jsou z chemického hlediska nejsložitější a funkčně nejdůmyslnější známé molekuly, a proto není divu, že se velká část výzkumu v bioinformatice zabývá právě jimi. Mutace jednotlivých aminokyselin mohou mít významný vliv na výslednou stabilitu proteinu. Je důležité si uvědomit, že ne všechny mutace musejí vést ke stabilní molekule. Z tohoto důvodu byly vyvinuty nástroje predikující vliv aminokyselinových mutací na stabilitu proteinu.

Výsledkem této diplomové práce je návrh a vytvoření meta-nástroje, který kombinuje výstupy jednotlivých nástrojů určených pro predikci změny stability proteinu s cílem zpřesnit požadovaný výsledek vzhledem k výsledkům již existujících nástrojů.

Druhá kapitola pojednává o aminokyselinách a proteinech. Podrobněji je zde rozebrána struktura proteinu, kterou je možné rozdělit na primární, sekundární, terciální a kvarterní. Nechybí zde ani zmínka o aminokyselinách a jejich možné klasifikaci.

Třetí kapitola se zabývá problémem predikce vlivu aminokyselinové substituce na stabilitu proteinu. Konkrétně je zde popsáno rozdělení mutací aminokyselin a jednotlivé typy jsou detailněji popsány. Nastíněny jsou taktéž možné problémy při predikci stability proteinu.

V čtvrté kapitole je možné najít výčet dostupných nástrojů pro predikci stability proteinu. Vybrané nástroje jsou zde stručně popsány a klasifikovány do konkrétní skupiny nástrojů podle způsobu predikce stability. Jednotlivé metody predikce stability jsou zde taktéž rozepsány. V závěru této kapitoly jsou uvedeny metodiky a studie, zabývající se výkonností predikčních nástrojů.

Pátá kapitola je určena strojovému učení. Zde jsou popsány základní problémy, principy a metody využívané v bioinformatické praxi. Nechybí zde ani informace ohledně problémů při výběru vhodného datasetu, problému přeučení a nastínění jejich možných řešení.

Šestá kapitola je věnována implementaci meta-nástroje. Je zde popsán postup vytvoření trénovacího a testovacího datasetu, uvedeny jsou taktéž jejich základní charakteristiky.

Sedmá kapitola se zabývá testováním a experimentováním s dosaženými výsledky nad trénovacími i testovacími daty. Tyto výsledky jsou zhodnoceny a porovnány s výsledky jednotlivých predikčních nástrojů. Diskutovány jsou taktéž výsledky techniky výběru rysů.

V závěrečné kapitole je shrnuta výsledná práce s důrazem na získané výsledky. Popsán je přínos a úspěšnost řešení této práce, uvedena jsou taktéž možná vylepšení pro budoucí práci.

Kapitola 2

Proteiny

Proteiny neboli bílkoviny tvoří zhruba jednu polovinu suché hmotnosti buňky [35]. Jedná se vlastně o biopolymer tvořený jedním nebo více polypeptidovými řetězci. Polypeptidové řetězce označujeme jako polymery aminokyselin spojených navzájem peptidovými vazbami [43]. Proteiny nejsou ovšem jenom pouhými stavebními kameny, z nichž je buňka tvořena. Z [2] je patrné, že obstarávají i mnoho dalších funkcí a že proteiny lze rozdělit na:

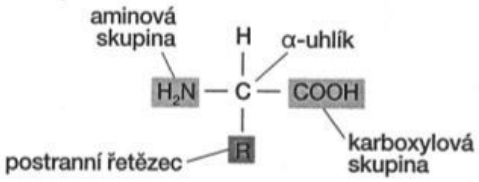
- enzymy,
- proteiny strukturní,
- transportní,
- pohybové,
- zásobní,
- signální,
- a další.

Vzhledem k jisté univerzálnosti proteinů nikoho nepřekvapí, že z chemického hlediska jsou právě proteiny nejsložitější a funkčně nejdůmyslnější známé molekuly. Velké množství funkcí, které proteiny zajišťují, je důsledkem obrovského počtu různých tvarů, kterých mohou proteiny nabývat.

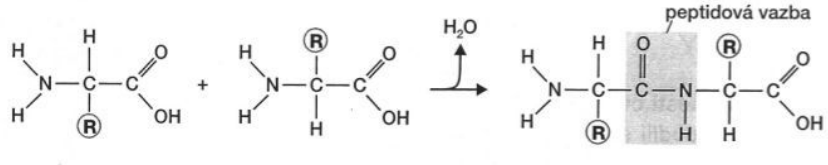
2.1 Aminokyseliny

Aminokyseliny jsou odvozeny od organických kyselin, kde na alfa uhlík je navázána karboxylová (-COOH) a aminová (-NH_2) funkční skupina. Jednotlivé aminokyseliny se od sebe liší v tzv. postranním řetězci (R), jehož podoba určuje chemické vlastnosti aminokyselin, resp. proteinů. Obecný vzorec pro tvorbu aminokyseliny je znázorněn na obrázku 2.1. Jednotlivé aminokyseliny jsou v molekule spojeny pomocí peptidové vazby, která vznikne spojením karboxylové skupiny jedné aminokyseliny s amino skupinou druhé aminokyseliny (viz obrázek 2.2). Při tvorbě této peptidové vazby se zároveň vylučuje molekula vody, což lze označit za kondenzaci.

Zřetězením více aminokyselin vzniká peptidový řetězec. Zbytky aminokyselin odstupují od osy řetězce jako tzv. postranní řetězce. Každý peptidový řetězec je na jednom konci



Obrázek 2.1: Základní obecný vzorec aminokyselin. Symbol R označuje postranní řetězec, který představuje zbytek aminokyseliny. Postranní řetězec R, karboxylová a aminová skupina jsou navázány na alfa-uhlík. [35]



Obrázek 2.2: Tvorba peptidové vazby mezi dvěma aminokyselinami. [35]

zakončen NH_2 skupinou (aminový či N konec) a na druhém COOH skupinou (karboxylový či C konec). [35]

Jak již bylo řečeno, o vlastnostech proteinů rozhoduje charakter postranních řetězců aminokyselin. Podle [43] lze aminokyseliny z hlediska fyzikálně-chemického klasifikovat takto:

- **Aminokyseliny s nepolárním zbytkem.** Do této skupiny patří všechny aminokyseliny, které mají alkylový postranní řetězec a jsou hydrofobní. Postranní řetězce se snaží shlukovat uvnitř molekuly a vyhnout se tak kontaktu s vodou, která je uvnitř buňky obklopuje. Mezi tyto aminokyseliny patří glycin, alanin, valin, leucin, izoleucin, fenylalanin, tryptofan, methionin a prolin. [2]
- **Aminokyseliny s polárním zbytkem.** Naopak aminokyseliny s polárním zbytkem se snaží zdržovat na povrchu molekuly, kde mohou vytvářet vodíkové můstky s molekulami vody a dalších polárních látek. Tyto aminokyseliny se ve vodě dobře rozpouštějí. Patří sem tyrosin, asparagin, glutamin, serin, threonin a cystein. [2]
- **Aminokyseliny s kyselým zbytkem.** Jsou to takové aminokyseliny, jejichž postranní řetězec obsahuje karboxylovou skupiny. Patří sem kyselina asparagová a kyselina glutamová. [43]
- **Aminokyseliny se zásaditým zbytkem.** Tyto aminokyseliny mají při neutrálním pH v postranním řetězci kladný náboj. Patří sem aminokyseliny histidin, arginin a lysin. [43]

Pro úplnost doplním, že dělení aminokyselin může být založeno i na struktuře jejich postranních řetězců, více lze nalézt na [43].

Jelikož je možné setkat se s více variantami zápisu konkrétní aminokyseliny, v tabulce 2.1 je uveden seznam dvaceti aminokyselin a jejich odpovídajících třípísmenných a jednopísmenných ekvivalentů.

Polární aminokyseliny			**Nepolární aminokyseliny**		
Asparagová kys.	Asp	D	Alanin	Ala	A
Glutaminová kys.	Glu	E	Glycin	Gly	G
Arginin	Arg	R	Valin	Val	V
Lysin	Lys	K	Leucin	Leu	L
Histidin	His	H	Izoleucin	Ile	I
Asparagin	Asn	N	Prolin	Pro	P
Glutamin	Gln	Q	Fenylalanin	Phe	F
Serin	Ser	S	Methionin	Met	M
Threonin	Thr	T	Tryptofan	Trp	W
Tyrosin	Tyr	Y	Cystein	Cys	C

Tabulka 2.1: Seznam 20 různých aminokyselin nacházejících se v proteinech. Vedle jména aminokyseliny je její třípísmenná i jednopísmenná zkratka. [2]

2.2 Struktura proteinové molekuly

2.2.1 Primární struktura

Primární struktura proteinu je taková struktura, která je tvořena sledem (sekvencí) jednotlivých aminokyselin v molekule. Z tohoto tvrzení vyplývá, že vlastnosti určité bílkoviny nejsou dány pouze aminokyselinovým složením, ale taktéž jejich pořadím. Tatáž množina aminokyselin může být seřazena lineárně teoreticky ve všech kombinacích. [35]

Tato struktura obsahuje informaci, podle které se tvoří sekundární, terciální a kvarterní struktura proteinu, realizuje se jejich nadmolekulární struktura a biologická funkce [43].

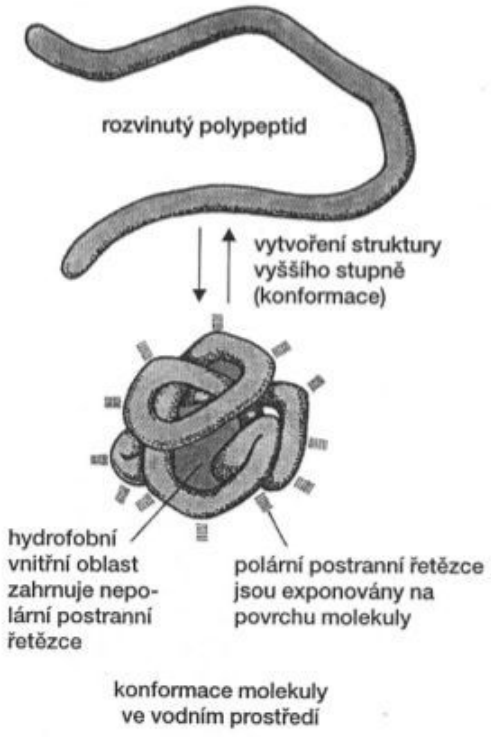
Pro zjištění primární struktury proteinu se používá technika zvaná *sekvenování*.

2.2.2 Sekundární struktura

Jelikož polypeptidový řetězec umožňuje volnou rotaci atomů, můžeme tyto řetězce označit jako velmi flexibilní. Tvar řetězce v prostoru označujeme jako konformaci proteinu. Konformace ovšem není náhodná, ale je určována silami, které působí uvnitř molekuly. Především se jedná o rozložení sil mezi aminokyselinami s polárními a nepolárními postranními řetězci. Nepolární postranní řetězce jsou přitahovány k sobě (dovnitř molekuly), kdežto polární postranní řetězce se orientují na povrch molekuly (viz obrázek 2.3). [35]

Další silou, která zde působí, jsou vodíkové můstky mezi peptidovými vazbami v řetězci, dále mezi nimi a postranními řetězci a mezi postranními řetězci navzájem [35]. Důsledkem těchto sil je to, že daný polypeptidový řetězec zaujme vždy stejnou konformaci. Změníme-li poměr těchto sil (např. denaturací), polypeptidový řetězec se vrátí zpět do původního stavu, jakmile tyto síly přestanou působit (např. renaturací).

Při bližším zkoumání struktur proteinu si lze všimnout, že obvykle obsahují dva základní modely. Prvním modelem je α-šroubovice (α-helix). α-helix je takové prostorové uspořádání,



Obrázek 2.3: Rozvinutý polypeptidový řetězec zaujímá ve vodném prostředí určitou prostorovou strukturu. Nepolární postranní řetězce se soustřeďují uvnitř molekuly, kdežto hydrofilní postranní řetězce se vyskytují na povrchu molekuly, kde interagují s molekulami vody. [35]

kde řetězec vytváří šroubovici. Tato konformace je stabilizována vodíkovými můstky mezi nad sebou ležícími peptidovými vazbami. [35]

Druhým modelem je β-struktura (β skládaný list). V β-struktuře probíhají úseky řetězce paralelně vedle sebe. Tato struktura je stabilizována vodíkovými můstky mezi sousedícími úseky. [35]

2.2.3 Terciální struktura

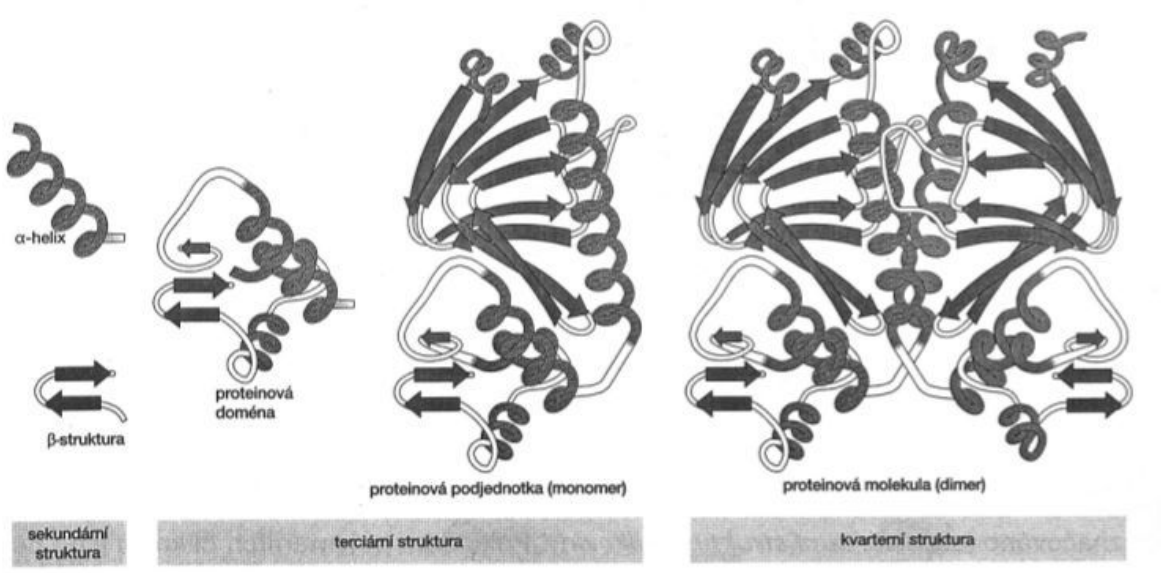
Terciální strukturou se označuje prostorové trojrozměrné uspořádání polypeptidového řetězce. Hlavním důvodem pro vytvoření terciární struktury proteinů je různost chemické povahy aminokyselinových postranních řetězců schopných tvořit nekovalentní vazby. [43]

Jelikož jsou všechny interakce udržující konformační stabilitu energeticky slabé (taktéž nekovalentní), působením vnějších faktorů dochází ke změně terciální struktury [35]. Pokud je tato změna vratná, mluvíme o vratné denaturaci, jinak ji označujeme jako nevratnou denaturaci.

2.2.4 Kvarterní struktura

Kvarterní struktura řeší uspořádání jednotlivých polypeptidových řetězců v molekule proteinu. Toto se týká ovšem jen oligomerních proteinů, tj. takových proteinů, které jsou tvořený více jak jedním polypeptidovým řetězcem. Je zajímavé, že i přestože je protein tvořen několika polypeptidovými řetězci, chová se v roztoku a v živé soustavě jako jedna molekula vyznačující se určitou biologickou funkcí. [43]

Sekundární, terciální a kvarterní strukturu lze zhlédnout na obrázku 2.4.



Obrázek 2.4: V levé části obrázku můžeme vidět sekundární strukturu proteinu (konkrétně α-helix a β-strukturu). V prostřední části je zobrazena terciální struktura s proteinovou doménou a proteinovou podjednotkou (monomerem). V pravé části se nachází proteinová molekula (dimer) řadící se do kvarterní struktury. [35]

Kapitola 3

Problém predikce vlivu aminokyselinové substituce na stabilitu proteinu

3.1 Stabilita proteinu

Stabilita proteinu je určena množinou navzájem působících a ovlivňujících se sil. Pokud protein označíme za stabilní, nachází se ve své původní složené konformaci. Na druhou stranu, pokud je protein nestabilní, dojde k jeho rozložení (denaturaci). Protein ve složené konformaci je stabilizován různými vzájemnými interakcemi jako jsou hydrofobní, elektrostatické, vodíkové vazby či van der Waaalsovi síly. V rozložené konformaci dominuje entropická a neentropická volná energie. [19]

Interakce mezi hlavním řetězcem a jeho postranními řetězci určuje všechny možné konformace, kterých protein může nabývat. Struktura výsledného proteinu je omezena také pomocí tzv. torzních úhlů. Tyto torzní úhly umožňují rotaci okolo $N - C\alpha$ a $C\alpha - C$ jednoduchých vazeb jednotlivých residuí. Důsledkem je druhý termodynamický zákon, který říká, že systémy s konstantní teplotou a tlakem najdou rovnovážný bod jako jistý kompromis mezi entalpií (H), entropií (S) a termodynamickou teplotou (T). Výsledkem je tzv. Gibbsova volná energie vyjádřená vztahem $G = H - T * S$. [26]

Pokud přihlédneme k možnostem vzniku mutací mající za následek změnu aminokyseliny, je zřejmé, že může dojít jak ke změně konformace proteinu, tak i ke změně jeho stability. Podrobnější informace o mutacích lze nalézt v kapitole 3.2.

3.1.1 Databáze ProTherm

Termodynamická data proteinů jsou velmi důležitá pro porozumění základním mechanismům proteinové stability. Z tohoto důvodu bylo během posledních desetiletí provedeno mnoho experimentů s cílem získat tato data. Výsledky těchto experimentů byly většinou publikovány v různých časopisech zabývajících se touto tématikou. Jelikož se data nevyskytovala na jednom místě, hledání konkrétních záznamů byl velký problém. Proto v roce 1998 vznikla elektronicky dostupná databáze ProTherm [25], která shromažďuje takto experimentálně získaná data. Tato databáze obsahuje termodynamická data (např. změna Gibbsovy volné energie, změna entalpie aj.), strukturní informace, měřící metody, odkazy na související literaturu nebo podmínky, ve kterých byl experiment proveden [26]. V současné době tato databáze obsahuje 25 820 záznamů [1].

Shromažďování těchto dat a zpřístupnění vědecké komunitě může pomoci vyvinout nové metody pro lepší porozumění a předpovídání stability proteinu. Tohoto faktu je využito i v této diplomové práci.

3.2 Mutace v proteinu

Jak bylo řečeno v úvodu kapitoly, stabilitu proteinu je možné ovlivnit zejména mutací jednotlivých aminokyselin.

Termínem *mutace* jsou v souvislosti s lidským genomem označovány náhlé, náhodné nebo neusměrněné změny genetického materiálu. Jsou to všechny změny genetické informace, které nejsou výsledkem segregací a rekombinací části genotypů již existujících [49].

Dle [35] mohou mutace měnit obsah genomu na třech úrovních, podle toho rozlišujeme mutace:

- genové (mění informaci nesenou genem),
- chromozomové (způsobena změnou struktury chromozomu),
- genomové (změna počtu chromozomů).

Jak již bylo zmíněno, primární struktura proteinu je určována z informací obsažených v DNA a právě DNA je místem, kde probíhají mutace, které mohou, ale také nemusí mít zásadní vliv na strukturu resp. funkci proteinu. Z tohoto důvodu se v dalších podkapitolách budeme podrobněji zmiňovat jen o mutacích genových.

3.2.1 Příčina vzniku mutací

V této podkapitole jsou popsány fyzické i chemické faktory ovlivňující vznik mutageneze (tj. procesu vzniku mutací). Genové mutace mohou vzniknout například jako chyby při replikaci DNA. Pokud se zaměříme spíše na přenos genetické informace, mutace mohou ovlivnit procesy jako transkripce či translace. Známým případem mutace je například srpkovitá anémie. Ta vzniká mutací genu pro hemoglobin, konkrétně záměnou v jeho beta-peptidickém řetězci, kde se na šesté pozici místo glutaminové kyseliny objevuje valin, který způsobuje srpkovitost červených krvinek. [49]

Mezi fyzikální faktory způsobující mutaci můžeme zařadit záření, a to jak ionizující, tak i neionizující. Stupeň poškození molekulární struktury DNA je přímo úměrný absorbované dávce záření. Mezi ionizující záření lze zařadit především rentgenové záření, neutrony, protony a elektrony o vysokém obsahu energie. Toto záření způsobuje přerušení kontinuity vlákna DNA. Mezi neionizující záření zařazujeme především záření ultrafialové, které poškozuje DNA.

Mezi chemické faktory ovlivňující strukturu DNA lze zařadit látky zvané *genotoxiny*. Těchto látek je obrovské množství a patří mezi ně například alkylační činidla, silná oxidační činidla, činidla interkalační a jiné. Některé látky ovšem nemusejí poškozovat DNA přímo, ale mohou narušovat například replikaci. [49]

3.2.2 Typy mutací

Dle [35] mezi základní typy mutací patří:

- substituce,
- inzerce,
- delece.

Všechny ostatní typy mutací jsou jenom různými variantami těchto tří zmíněných mutací. *Substituce* je záměna jednoho či několika po sobě jdoucích párů nukleotidů. *Transpozicí* se označuje změna pořadí nukleotidů nebo nukleotidových párů. *Inverze* je výměna jednoho nebo více nukleotidových párů mezi oběma vlákny DNA. Včlenění jednoho nebo více po sobě následujících nukleotidů nebo nukleotidových párů označujeme jako *inzerce*. *Delece* je pak ztráta jednoho nebo několika po sobě následujících nukleotidů či nukleotidových párů. Všechny uvedené mutace můžeme přehledně vidět v tabulce 3.1.

vlákno standardní DNA	a	b	c	d	e	f		
substituce	a	r	c	d	e	f		
transpozice	a	c	d	b	e	f		
inzerce	a	b	m	n	c	d	e	f
duplikace	a	b	b	c	d	e	f	
delece	a	b	d	e	f			
inverze	a	b			e	f		
			c	d				

Tabulka 3.1: Běžné typy genových mutací (přepracováno z [35]).

3.2.3 Nukleotidový polymorfismus

Všichni lidé, s výjimkou identických sourozenců, mají unikátní DNA sekvenci. Při porovnání jedinců, kteří nebyli v příbuzenském vztahu, se zjistilo, že se genom těchto jedinců liší zhruba o 0,1%. Většina těchto odlišností je způsobena právě nukleotidovými polymorfismy, konkrétně jednobodovým polymorfismem označovaným SNP (*Single-nucleotide polymorphism*) [26]. Odhaduje se, že více jak 93% lidských genů obsahuje alespoň nějaký SNP, z toho přibližně 98% genů je ve vzdálenosti do 5000 párů bází od SNP. [10]

SNP lze tedy chápat jako genetickou variabilitu mezi jedinci v populaci, kde dochází k substituci, inzerci nebo deleci pouze u jednoho páru bází. Příkladem budiž již zmíněná srpkovitá anémie. [26]

Pokud se podíváme na tabulku 3.2, která znázorňuje kódování aminokyselin pomocí kodonů mRNA, zjistíme, že určitá aminokyselina může být kódována různými kodony. Z tohoto faktu vyplývá, že při mutaci nemusí vždy dojít ke změně aminokyseliny a s tím související změně primární struktury příslušného proteinu.

Dle [35], [26] lze SNP rozdělit na:

- synonymní (tichou) mutace, které nezpůsobí záměnu aminokyseliny na dané pozici,
- nesynonymní mutace, kde vznikají kodony určující jinou aminokyselinu,
- nesmyslné (nonsense) mutace, kde vznikají ukončovací kodony, čímž dojde ke zkrácení polypeptidových řetězců.

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

Tabulka 3.2: Kódování aminokyselin pomocí kodonů mRNA. [35]

3.2.4 Důsledky mutací strukturních genů

Jelikož mutace mohou ve svém důsledku měnit primární strukturu proteinů, je zřejmé, že tyto mutace také mohou vyvolávat podstatné změny metabolických i jiných procesů v buňce (dokonce mohou způsobit i její zánik). Je taktéž zajímavé, že pokud dojde k záměně aminokyseliny v místě nevýznamném pro biologickou funkci proteinu, mutace se ve fenotypu neprojeví. Pokud dojde k záměně aminokyseliny v aktivním či vazebném místě proteinu, funkce proteinu se změní nebo se daný protein stane nefunkčním.

Delece nebo inzerce nukleotidů způsobí změnu čtecího rámce, jehož důsledkem je vždy syntéza nefunkčního proteinu.

Fenotypové projevy mutací strukturních genů jsou různé podle změny stupně genového produktu. Může se jednat o změnu kinetiky enzymu či vazebných schopností proteinu nebo o absenci určité metabolické funkce. [35]

Kapitola 4

Nástroje pro predikci stability proteinu

V průběhu posledního desetiletí bylo vyvinuto několik metod k určení vlivu aminokyselinových mutací na stabilitu proteinu. Většina z těchto výpočetních metod je primárně založena na výpočtu změny volné energie ($\Delta\Delta$G). Některé z těchto metod používají pro výpočet $\Delta\Delta$G energetické funkce, které popisují interakce mezi jednotlivými residui. Jiné nástroje mohou používat metod strojového učení (*machine learning approaches*), kde k natrénování modelu využívají databáze s experimentálně naměřenými hodnotami změn po provedení mutací.

Způsob určení energetických funkcí můžeme rozdělit dle [23] na přístupy založené na:

- fyzikálním potenciálu (*physical potential approaches*),
- statistickém potenciálu (*statistical potential approaches*) a
- empirickém potenciálu (*empirical potential approaches*).

Výpočty $\Delta\Delta$G založené na fyzikálním potenciálu simulují rozložení sil mezi jednotlivými atomy (tj. analýza sil). Z tohoto důvodu je tato metoda predikce stability výpočetně náročnější než metody strojového učení. Výpočet statistického potenciálu využívá analýzy různých vlastností extrahovaných z datasetu proteinových struktur (statistické analýzy prostředí, substitučních frekvencí či korelací přilehlých residuí nalezených experimentálně v proteinových strukturách). Při výpočtu energetické funkce je empirický potenciál určen kombinací váhovaných fyzikálních a statistických energetických výrazů [23]. Některé přístupy taktéž mohou kombinovat výhody statistické analýzy a metod strojového učení, respektive neuronových sítí. V některých případech se vyskytují hybridní přístupy založené na fyzikálním a statistickém potenciálu [39].

Dále můžeme predikční nástroje rozdělit dle způsobu práce s proteinovými záznamy (strukturami) na

- strukturní a
- sekvenční.

Predikční nástroje využívající 3D struktury proteinu vyžadují ke svému chodu soubory ve formátu PDB (*Protein data bank*) [6], které jsou volně on-line dostupné[1]. K nevýhodám tohoto přístupu patří právě závislost na PDB souborech obsahujících potřebné strukturní informace. Zdrojem dat bývá experimentální měření metodami NMR a X-ray krystalografií. [6]

Nástroje využívající sekvenčního přístupu vyžadují pouze sekvenci aminokyselin daného proteinu. V tomto případě zde odpadá přítomnost chyb, kdy experimentální měřící metody (NMR a X-ray krystalografie) nejsou schopny zaznamenat určité pozice atomů, jak se tomu děje v některých PDB záznamech, které tak znemožňují predikci vlivu mutace na stabilitu proteinu na daných atomových souřadnicích. Na druhou stranu tímto přístupem ztrácíme informaci o prostorovém uspořádání atomů proteinu.

Níže uvedené predikční nástroje byly vybrány takovým způsobem, aby byla pokryta co možná nejširší škála způsobů a metod jak predikovat stabilitu proteinu a bylo tím dosaženo co možná nejvyšší míry univerzálnosti výsledného meta-nástroje.

4.1 AUTO-MUTE

AUTO-MUTE je kolekcí tří nástrojů ($\Delta\Delta$G, $\Delta\Delta G^{H_2O}$ a ΔT_m) sloužících pro predikci vlivu aminokyselinových mutací na stabilitu proteinu. V tomto textu se budeme zabývat nástrojem označeným $\Delta\Delta$G, který predikuje vliv jednobodových mutací na stabilitu proteinu s ohledem na tepelnou denaturaci.

Predikční modely tohoto nástroje byly trénovány na mírně upravených záznamech získaných z databáze ProTherm (blíže popsáno v [8]). Původní dataset obsahoval 1948 jednobodových mutací z celkem 58 proteinových struktur, které se zároveň vyskytovaly v databázi PDB. Po různých úpravách (např. odstranění proteinových struktur, které neobsahovaly kompletní informace o 3D struktuře proteinu), dataset obsahoval 1925 jednobodových mutací v 55 proteinových strukturách.

Poskytnuty jsou dva klasifikační modely (pouze pro predikci znaménka $\Delta\Delta$G) a dva regresní modely (predikce hodnoty $\Delta\Delta$G). U klasifikačních metod lze použít *Random Forest* (RF) a *Support Vector Machine* (SVM), regresní metody nabízejí možnost volby mezi *Tree Regression* (REPTree) a *SVM regression* (SVMreg). Výběr mezi těmito modely je ponechán na uživateli, podrobnější informace lze nalézt na [31].

K povinným vstupním parametrům patří: PDB ID (jednoznačný čtyřpísmenný identifikátor proteinové struktury v PDB databázi), proteinový řetězec, mutace (ve formátu původní residuum, pozice mutace, nahrazené residuum), teplota (v rozsahu 0°C až 100°C) a pH (v rozsahu 0 -log[H+] až 14 -log[H+]).

Výsledný efekt mutace je určen na základě hodnoty $\Delta\Delta$G. Pokud je splněna podmínka $\Delta\Delta G > 0$ kcal/mol, jde o stabilizující mutaci, jinak je mutace označena za destabilizující. K dalším výstupům nástroje patří například i predikce sekundární struktury. Samotný nástroj umožňuje predikovat až pět mutací současně.

4.2 SDM

Site Directed Mutator (SDM) je on-line nástroj založený na výpočtu statistického potenciálu energetické funkce vyvinutý Christopherem M. Tophamem [44] k predikci efektu

[1] http://www.pdb.org

jednobodových mutací na stabilitu proteinu. SDM používá specifické prostředí aminokyselinových substitučních frekvencí v rámci homologních proteinových rodin k výpočtu tzv. *stability skóre*. Tento typ výpočtu lze považovat za analogii ke změně volné energie mezi divokým typem (z anglického překladu *wild-type*) a mutovaným proteinem [47]. Další informace ohledně principu výpočtu predikce stability proteinu lze nalézt na [44].

Nástroj k predikci využívá strukturních informací, proto je nutné zadat PDB ID nebo je možné nahrát vlastní PDB soubor. Dále je nutné určit proteinový řetězec, pozici mutovaného residua a samotné mutované residuum. Nástroj neumožňuje zadat původní residuum na zvolené pozici. Tato vlastnost se zvláště při použití automatického zpracování ukázala jako nevýhodná, a to vzhledem k faktu, že některé PDB soubory neobsahují kompletní posloupnost atomů a může tak dojít k chybnému určení mutovaného místa. Typicky se jedná o problém na začátcích a koncích řetězce, kde vlivem použité experimentální metody nemusí být daná aminokyselina uvedena a může tak dojít k nekonzistenci mezi pozicemi aminokyselin v záznamu SEQRES a atomovými souřadnicemi. Kvůli absenci kontroly ekvivalence můžeme v těchto případech predikovat stabilitu proteinu na jiné pozici, než bylo původně požadováno.

K zajímavým vlastnostem tohoto nástroje patří, že kromě predikce stability proteinu předpovídá i možnost onemocnění. Mutovaná pozice je zároveň ukázána v Jmol appletu, kde jsou jednotlivá residua obarvena podle jejich chemických vlastností.

4.3 CUPSAT

Cologne University Protein Stability Analysis Tool (CUPSAT) je webový nástroj sloužící k analýze a predikci změn stability proteinu způsobené jednobodovými aminokyselinovými mutacemi. Nástroj k výpočtu $\Delta\Delta G$ používá potenciálu specifických strukturních atomů a potenciálu torzních úhlů. CUPSAT, jako jediný z vybraných predikčních nástrojů, lze zařadit do kategorie nástrojů, které pro výpočet energetické funkce používají přístupu založeném na empirickém potenciálu.

Požadované vstupní parametry jsou PDB ID, pozice mutace v aminokyselinovém řetězci a původní (přirozená) aminokyselina na zadané pozici. Dále je možné určit experimentální metodu, kde má uživatel na výběr ze dvou možností *Thermal* a *Denaturants*. Při výběru mezi těmito dvěma metodami byly brány v úvahy údaje obsažené v databázi ProTherm. Pokud jako metoda denaturace nebyla v záznamu databáze ProTherm uvedena metoda *Thermal*, byla vybrána experimentální metoda *Denaturants*, v jiném případě byla vybrána metoda *Thermal*.

Pro zadané vstupní parametry nástroj predikuje celkový efekt na stabilitu proteinu (stabilní/destabilní), torzní úhly (favourable/unfavourable) a konkrétní hodnotu $\Delta\Delta G$. Kladné hodnoty $\Delta\Delta G$ jsou zde brány jako stabilizující, záporné jako destabilizující.

Ačkoliv autoři ve svém článku [36] slibují aktualizaci lokálního PDB repozitáře přibližně jednou měsíčně, u některých proteinových struktur obsažených v databázi PDB nelze stabilitu predikovat. Tento problém lze řešit ručním nahráním PDB souboru do lokálního repozitáře nástroje. [36]

4.4 I-Mutant3.0

Autoři tohoto nástroje použili na rozdíl od všech zmíněných nástrojů třístavovou klasifikaci. Dle [9] se v použitém datasetu vyskytovalo okolo 32% hodnot $\Delta\Delta G$, které byly blízké nule

(v intervalu -0,5 až 0,5 kcal/mol). Hodnoty v tomto rozsahu ovšem nemusejí být určeny přesně (způsobeno například chybou měření či přesností měřící metody) a je možné, že vliv mutace bude špatně klasifikován. Z tohoto důvodu autoři použili již zmíněnou třístavovou klasifikaci, kde destabilizující mutace musí splňovat podmínku $\Delta\Delta G < -1{,}0$ kcal/mol, stabilizující mutace $\Delta\Delta G > 1{,}0$ kcal/mol a neutrální mutace $-1{,}0 \leq \Delta\Delta G \geq 1{,}0$ kcal/mol.

I-Mutant3.0 je nástroj využívající metod strojového učení, konkrétně metody Support Vector Machine (SVM). Autoři vytvořili dvě verze tohoto programu, v první verzi je predikce založena na strukturní analýze, druhá verze využívá sekvenční analýzu.

Trénovací dataset pro sekvenční verzi I-Mutant3.0 je tvořen 1623 různými jednobodovými mutacemi obsaženými v 58 různých proteinech. Pro strukturní verzi trénovacího datasetu bylo vybráno 1576 různých mutací z celkem 55 proteinů. Aplikováním termodynamické reverzibility (předpokládáme, že reverzní mutace způsobuje stejnou změnu $\Delta\Delta G$ jako mutace původní) na každou mutaci byl počet mutací pro sekvenční dataset zvýšen na 3246, pro strukturní dataset 3152 mutací.

Kromě predikce efektu mutace a jejím $\Delta\Delta G$ je výstupem tohoto nástroje RSA (*Relative Solvent Accessible Area*) a index spolehlivosti (*Reliability index*) v intervalu 1-9.

4.5 iPTREE-STAB

iPTREE-STAB je on-line nástroj umožňující predikci celkového efektu na stabilitu proteinu (stabilní/nestabilní) a predikci změny stability proteinu ($\Delta\Delta G$) v závislosti na jednobodových mutacích aminokyselinového řetězce. Pro výpočet je použita sekvence aminokyselin, proto na rozdíl od nástrojů využívajících strukturních vlastností proteinu není nutné vkládat PDB soubor. Rozhodování o stabilitě proteinu je ponecháno na metodách strojového učení, konkrétně na jednoduchém rozhodovacím stromu. Autoři v [22] uvádějí, že pro natrénování rozhodovacího stromu bylo použito celkem 1859 neredundantních záznamů jednobodových mutací, které byly získány z databáze ProTherm.

Jako jediný z uvedených nástrojů, iPTREE-STAB neumožňuje určit pozici, na které dojde k mutaci. Místo toho se používá jednoduchého principu, kdy nástroj analyzuje pouze aminokyseliny v okolí vyšetřovaného (mutovaného) residua. Před i za požadovaným residuem je nutné zadat tři předcházející/následující aminokyseliny. Mimo tyto určující údaje je nutné vyplnit i pH a teplotu.

Jelikož se jedná o nástroj využívající metod strojového učení, výpočet predikce je v tomto případě velmi rychlý.

4.6 mCSM

Nástroj mCSM (mutation Cutoff Scanning Matrix) používá nově navržený přístup výpočtu změny stability proteinu blíže popsaný v [37]. Na rozdíl od ostatních přístupů, tento využívá graf založený na signaturách. Pro pochopení toho, jakou roli mají mutace v onemocnění, autoři umožnili ohodnotit nejen proteinovou stabilitu, ale také interakce mezi proteinem-proteinem a proteinem-nukleovou kyselinou. Prostředí residuí může být reprezentováno grafy, kde uzly jsou atomy a hrany jsou fyzikálně-chemické interakce mezi nimi. Z těchto grafů může vzniknout strukturální signatura, která je vytvořena extrahováním a sumarizováním vzdálenostních vzorů. Tato signatura je poté použita jako objekt pro trénování prediktivních modelů.

Výpočet je možné uskutečnit pomocí webového rozhraní, a to třemi způsoby nazvanými Single mutation, Mutation list a Systematic. Single mutation poskytuje stejný přístup, jaký jsme viděli u předcházejících nástrojů. V tomto případě je nutné nahrát PDB soubor, určit mutovaný řetězec a konkretizovat mutaci její pozicí, wild-typem a mutantem. Systematic se chová obdobně - jen s tím rozdílem, že predikce stability je vypočítána pro všech 19 zbývajících aminokyselin. Mutation list poskytuje možnost vytvoření konfiguračního souboru, ve kterém může být uvedeno více mutací vztahující se k jednomu proteinu, resp. PDB souboru. Tento postup je výhodný zejména pro větší počet zpracovávaných mutací nebo pro automatizované skripty.

Kladné hodnoty $\Delta\Delta G$ vyjadřují stabilizující mutace, naopak hodnoty záporné destabilizující mutace.

Výstupem je snadno zpracovatelný textový soubor, který kromě predikované $\Delta\Delta G$ obsahuje i RSA (Relative Solvent Accessibility). Pokud je ovšem v konfiguračním souboru uvedena nekorektní mutace, nejsou v tomto konkrétním souboru provedeny žádné predikce.

4.7 PoPMuSiC

PoPMuSiC-2.1 je webový server predikující změnu termodynamické stability způsobenou jednobodovými mutacemi proteinů. Predikce je založena na lineární kombinaci statistických potenciálů, jejichž koeficienty závisejí na *solvent accessibility*[2] mutovaných residuí. Dle [14] je predikce vyjádřena lineární kombinací právě třinácti statistických potenciálů. Predikční model obsahuje celkem 64 parametrů, jejichž hodnoty jsou upraveny pomocí neuronových sítí se snahou o minimalizaci střední kvadratické odchylky.

Tento predikční nástroj, jako jediný, požaduje pro svůj chod registraci uživatele. Výhoda tohoto požadavku je v tom, že všechny výsledky v minulosti vypočítaných úloh jsou uživateli volně dostupné.

Rozhraní tohoto nástroje je podobně rozčleněné jako v případě mCSM. Výpočty je možné provádět ve třech režimech Single, Systematic a File. Režim Single slouží pro ohodnocení jedné mutace určené pomocí proteinového řetězce, wild-typem a mutantem. PDB strukturu je možné identifikovat pomocí PDB ID nebo tento záznam nahrát na server. Systematic vypočítá $\Delta\Delta G$ pro všechny zaznamenané pozice aminokyselin v zadané PDB struktuře, a to pro všech devatenáct možných variant mutací. V tomto režimu je taktéž možné zobrazit graf, ve kterém je vynesen na každé pozici součet záporných predikcí $\Delta\Delta G$. Struktura α-helix je obarvena červenou barvou, β-struktura modře a ostatní struktury (turns a coils) jsou zelené. V režimu File je možné pro konkrétní PDB strukturu vytvořit konfigurační soubor obsahující požadované mutace. Tento přístup je velmi rychlý a na rozdíl od nástroje mCSM se při výskytu chybné mutace výpočet nepřeruší.

Na rozdíl od zmíněných nástrojů, PoPMuSiC pro stabilizující mutace vrací zápornou hodnotu $\Delta\Delta G$, pro destabilizující mutace pak hodnotu kladnou. Aby se při práci se všemi nástroji používalo stejné notace, byla hodnota predikovaná tímto nástrojem převrácena na kladnou pro stabilizující, na zápornou pro destabilizující mutaci.

4.8 Porovnání a shrnutí

Všechny nástroje a jejich zařazení do jednotlivých skupin uvedených v úvodu této kapitoly lze přehledně nalézt v tabulce 4.1. Snahou bylo vybrat takové predikční nástroje,

[2] Povrchová plocha biomolekuly, která je dostupná rozpouštědlu.

které by pokryly co možná nejvíce možných metod a postupů pro výpočet predikce stability proteinu. Tímto způsobem jsme schopni markantně zvýšit celkovou velikost prostoru řešitelných mutací v závislosti na zadaném vstupu. Výsledný prostor řešitelných mutací je dán sjednocením prostorů řešitelných mutací jednotlivých nástrojů.

Nástroje	**Způsob výpočtu**	**Algoritmus**	**Typ dat**
AUTO-MUTE [31]	strojové učení	random forest, SVM, REPTree, SVMreg	strukturní
SDM [47]	energetické funkce	statistický potenciál	strukturní
CUPSAT [36]	energetické funkce	empirický potenciál	strukturní
I-Mutant3.0 [9]	strojové učení	SVM	strukturní, sekvenční
iPTREE-STAB [22]	strojové učení	rozhodovací strom	sekvenční
mCSM [37]	energetické funkce	statistický potenciál	strukturní
PoPMuSiC [14]	energetické funkce	statistický potenciál	strukturní

Tabulka 4.1: Přehled nástrojů a jejich metodologií výpočtu.

Zároveň zde byla i snaha použít nástroje, jejichž doba predikce je přibližně stejná. Celková doba běhu vytvořeného meta-nástroje je totiž vždy dána časem nejpomalejšího predikčního nástroje. Z tohoto důvodu tudíž není příliš vhodné použít nástroje s diametrálně odlišnými dobami běhu, přijmeme-li předpoklad, že výsledné váhové ohodnocení jednotlivých nástrojů nebude diametrálně odlišné. V tabulce 4.2 lze nalézt informace o potřebném čase pro výpočet jedné mutace, omezení počtu mutací pro vstupy jednotlivých nástrojů a také nechybí popis jejich omezení.

Nástroje	**Čas výpočtu**	**Vstup**	**Omezení**
AUTO-MUTE [31]	$<$ 5 min	1-5 mutací	neumožňuje nahrání vlastní struktury
SDM [47]	$<$ 1 min	1 mutace	chybí kontrola původní aminokyseliny (wild-type)
CUPSAT [36]	$<$ 1 s	1 mutace	neaktualizovaný lokální PDB repozitář, chybí kontrola původní aminokyseliny (wild-type)
I-Mutant3.0 [9]	$<$ 1 min	1 mutace	neumožňuje nahrání vlastní struktury
iPTREE-STAB [22]	$<$ 1 min	1 mutace	není možnost určit pozici mutace
mCSM [37]	$<$ 1 min	lib. počet	neumožňuje zadat PDB ID
PoPMuSiC [14]	$<$ 1 min	lib. počet	nutnost registrace

Tabulka 4.2: Tabulka udává přibližný čas výpočtu jedné mutace, počet mutací, které je možné dát na vstup nástroje (libovolný počet mutací se vztahuje k jedné proteinové struktuře) a popis omezení jednotlivých nástrojů.

4.9 Výsledky predikčních nástrojů

Tato kapitola se bude zabývat výsledky jednotlivých predikčních nástrojů. Poznatky budou čerpány ze studií [23], [39] a [11], dosažené výsledky budou diskutovány.

4.9.1 Metodika porovnání nástrojů

Pro základní pochopení statistických veličin je nutné definovat pojmy uvedené v [4]. Kvalitu predikce lze popsat parametry jako přesnost (*accuracy*), specificita (*specificity*), senzitivita (*sensitivity*) a také pomocí Matthewsova korelačního koeficientu (*MCC*). Zatímco senzitivita je pravděpodobnost správné predikce pozitivního případu, specificita je definována jako pravděpodobnost, že hodnota pozitivní predikce je správná. [4]

Vztah pro výpočet přesnosti predikce je definován níže. TP (true positive) v tomto případě značí počet výskytů pravdivě pozitivních (reálně stabilizující mutace označena jako stabilizující), FP (false positive) falešně pozitivních (reálně destabilizující mutace je označena jako stabilizující), TN (true negative) pravdivě negativních (reálně destabilizující mutace je označena jako destabilizující) a FN (false negative) falešně negativních (reálně stabilizující mutace označena jako destabilizující). Matthewsův korelační koeficient dosahuje hodnot v rozmezí -1 až 1. Hodnota $MCC = 1$ označuje nejlepší možnou predikci, zatímco $MCC = -1$ indikuje nejhorší možnou predikci (někdy označováno antikorelace). Pro hodnotu $MCC = 0$ není zjistitelná žádná lineární závislost (predikce je výsledkem náhody). [4]

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{4.1}$$

$$Specificity = \frac{TN}{TN + FP} \tag{4.2}$$

$$Sensitivity = \frac{TP}{TP + FN} \tag{4.3}$$

$$MCC = \frac{TP \times TN - FN \times FP}{\sqrt{(TP + FN)(TP + FP)(TN + FN)(TN + FP)}} \tag{4.4}$$

4.9.2 Výsledky jednotlivých studií

Pro porovnání jednotlivých predikčních nástrojů lze použít závěry ze studie [23], která porovnávala výkonnost 11 online dostupných nástrojů. Mezi tyto nástroje patří CUPSAT [36], Dmutant [48], FoldX [20], I-Mutant2.0 [8], I-Mutant3.0 (strukturní i sekvenční verze) [9], MultiMutate [15], MUpro [12], SCide [17], Scpred [16] a SRide [28].

Pro testování přesnosti predikce jednotlivých nástrojů byla použita databáze ProTherm s experimentálně zjištěnými hodnotami ΔΔG. Mutace v intervalu ΔΔG mezi 0,5 a -0,5 kcal/mol byly klasifikovány jako neutrální mutace (neměnící stabilitu proteinu), jelikož průměrná hodnota maximální experimentální chyby se dle [24] pohybuje okolo $\pm 0,48$ kcal/mol (chyba měření by mohla ovlivnit klasifikaci do třídy stabilizující/destabilizující).

Výsledný testovací dataset obsahoval 1784 neduplicitních mutací z celkově 80 proteinů, kde 931 mutací bylo destabilizujících (ΔΔG $\geq$ 0,5 kcal/mol), 222 stabilizujících (ΔΔG $\leq$ -0,5 kcal/mol) a 631 mutací bylo neutrálních (0,5 kcal/mol > ΔΔG $\geq$ -0,5 kcal/mol). Znaménko hodnoty ΔΔG bylo v této studii převráceno oproti hodnotám v databázi ProTherm. Velikost trénovacích datasetů pro jednotlivé nástroje byla proměnná, a to z toho důvodu, že některé nástroje používaly pro natrénování svého predikčního modelu část záznamů z databáze ProTherm a výsledky by v tomto případě byly zkreslené (nadhodnocené). Z tohoto důvodu byly vybrány každému nástroji pro testování pouze ty záznamy, které se v databázi ProTherm zveřejnily až po jejich vydání. Velikosti datasetů jsou přehledně znázorněny v tabulce 4.3.

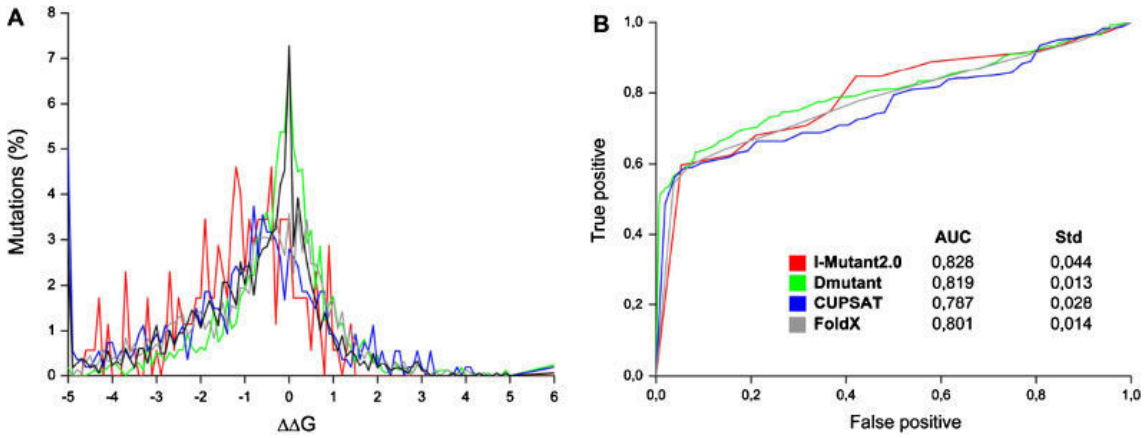
Celkově nejlepších výsledků dosahovaly I-Mutant3.0 (strukturní verze), Dmutant a FoldX. Přesnost těchto nástrojů kolísala od hodnot 0,54 do 0,64. Nejlepší senzitivitu vykazoval nástroj MUpro (0,74), hodnoty senzitivity pro I-Mutant2.0 a CUPSAT byly jen nepatrně menší (0,71 a 0,69). Nejvyšší specificitu zaznamenal nástroj SRide (0,95). Hodnoty Matthewsova korelačního koeficientu byly ovšem nízké pro všechny predikční nástroje. Nejlepšího korelačního koeficientu dosáhl nástroj I-Mutant3.0 (strukturní verze), jeho hodnota se pohybovala okolo 0,27. Naopak nejhoršího korelačního koeficientu (-0,39) dosáhl nástroj MUpro.

V tabulce 4.3 lze nalézt dosažené výsledky pro vybrané predikční nástroje. Kompletní výsledky všech nástrojů lze nalézt v [23].

Parametry	**CUPSAT**	**I-Mutant3.0 (strukturní)**	**I-Mutant3.0 (sekvenční)**
velikost datasetu	536	115	115
přesnost	0,50	0,64	0.52
specificita	0,30	0,63	0.39
senzitivita	0,69	0,64	0.66
MCC	-0,01	0,27	0.05

Tabulka 4.3: Vybrané výsledky z [23] pro nástroj CUPSAT a I-Mutant3.0 ve strukturní i sekvenční verzi.

Obrázek 4.1 zobrazuje graf distribuce predikovaných a experimentálně naměřených $\Delta\Delta G$ hodnot, které jsou vyjádřeny normální distribuční křivkou. Hodnoty predikované pomocí nástrojů I-Mutant2.0 a CUPSAT jsou vychýlené směrem k negativním hodnotám (hodnoty značící destabilizaci), zatímco u nástroje Dmutant směřují spíše ke kladným hodnotám, ačkoliv nejvyšší vrchol jeho křivky je pro $\Delta\Delta G = 0$. Distribuční křivka pro FoldX neobsahuje jasně čitelný vrchol, větší množství $\Delta\Delta G$ hodnot je menších než -4 kcal/mol.

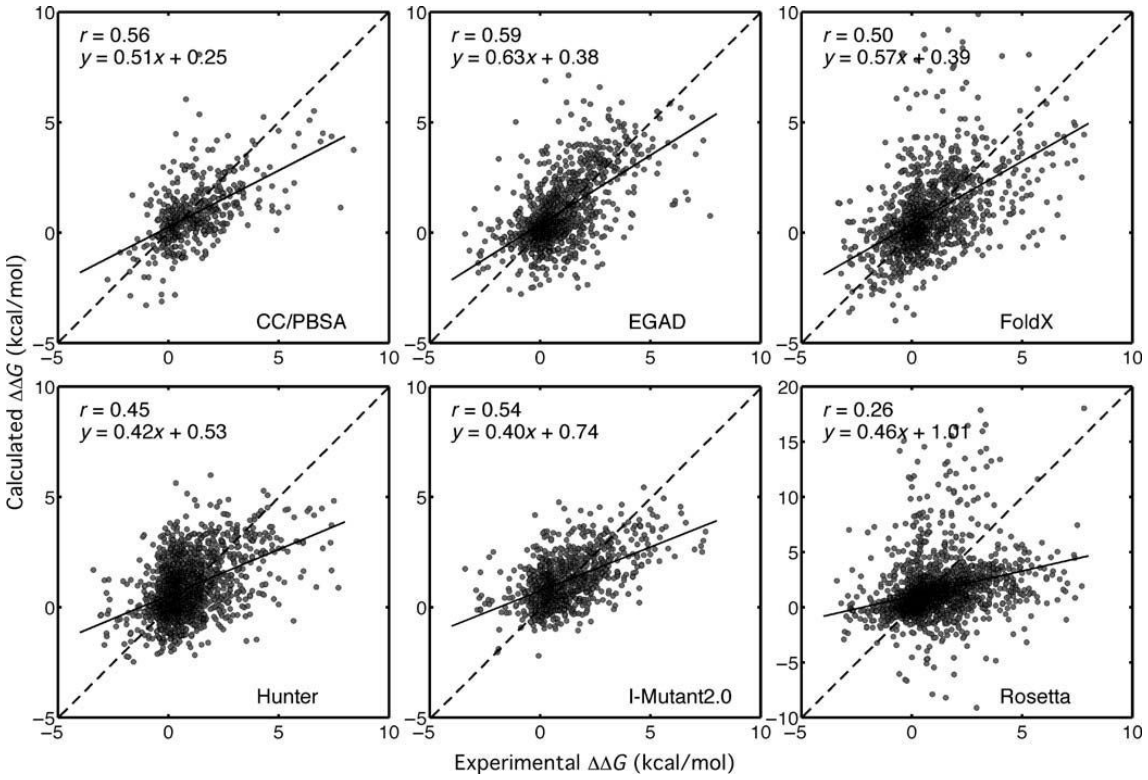


Obrázek 4.1: **A:** Distribuce predikovaných a experimentálně naměřených $\Delta\Delta G$ hodnot. Jako predikční nástroje byly použity I-Mutant2.0 (červená), Dmutant (zelená), CUPSAT (modrá), FoldX (šedá). Experimentální hodnota $\Delta\Delta G$ je znázorněna černou barvou. **B:** ROC křivka znázorňující úspěšnost nástrojů FoldX, I-Mutant2.0, Dmutant a CUPSAT. Zobrazeny jsou taktéž hodnoty AUC a standardní odchylky odvozené od ploch pod jednotlivými křivkami. Barevné označení nástrojů je zaznačeno na obrázku. [23]

Ve výsledcích této studie nebyly zahrnuty predikční nástroje jako PoPMuSiC, ERIS, iPTREE-STAB, AUTO-MUTE a jiné. PoPMuSiC nebyl zařazen z toho důvodu, že během psaní studie [23] nebyla dostupná stabilní verze tohoto nástroje (stabilní verze byla vydána až po dokončení studie). Rozhraní nástroje ERIS dle autorů neumožňuje dávkové zpracování, což znemožnilo její zařazení. iPTREE-STAB používá metodu rozhodovacího stromu, není zde ovšem možné přesně určit pozici či proteinovou strukturu. Nástroj AUTO-MUTE obsahoval pouze 28 případů, které nebyly použity pro natrénování jeho trénovacího datasetu. Pro statistickou analýzu je toto číslo příliš malé. Pro těchto 28 případů byl nástroj AUTO-MUTE schopen správně predikovat 6 případů (21%).

Studie [39] porovnává celkem 6 odlišných nástrojů pro predikci změny stability proteinu. Mezi tyto nástroje patří CC/PBSA [5], EGAD [38], FoldX [20], I-Mutant2.0 [8], Rosetta [42] a Hunter. Pro ohodnocení přesnosti predikce byl použit dataset obsahující 2156 jednobodových mutací, které nebyly použity pro trénování u jednotlivých nástrojů. Korelační koeficient mezi experimentální a predikovanou hodnotou ΔΔG byl v rozmezí 0,59 pro nejlepší a 0,26 pro nejhorší nástroj. Všechny predikční nástroje vykazují správný trend v predikci svých výsledků (celkový efekt na stability proteinu), ve větší míře ovšem selhávají při poskytování přesných hodnot.

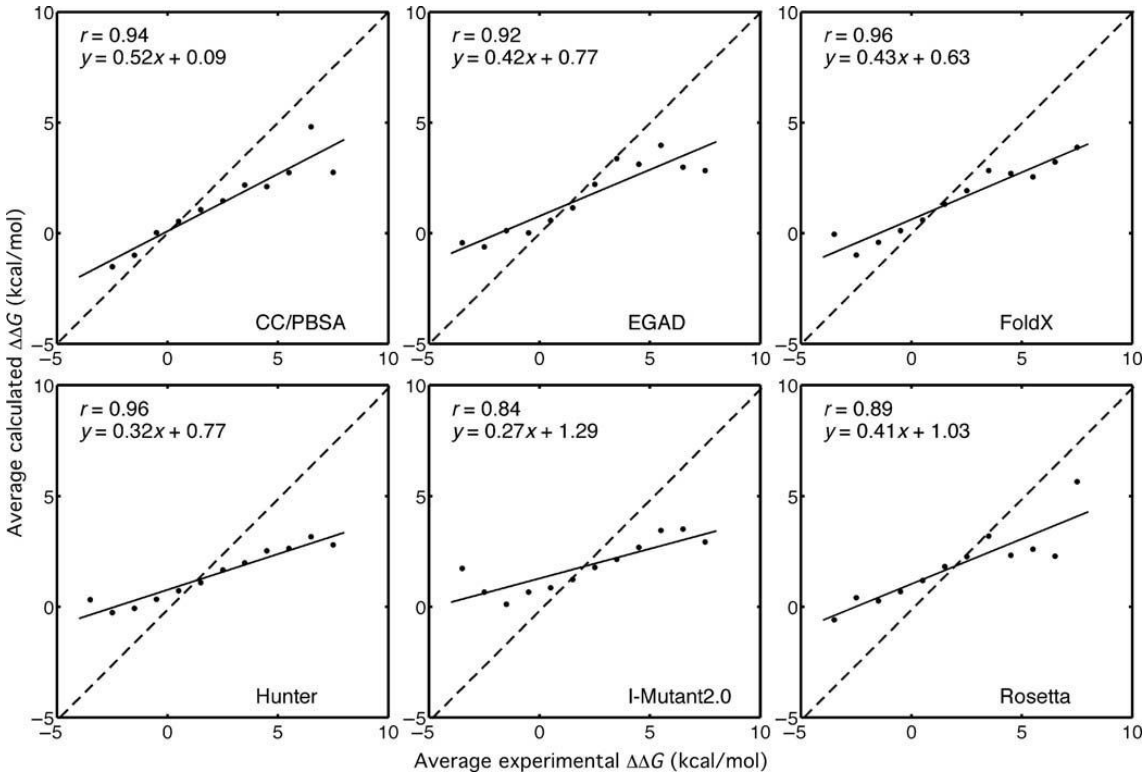
Na obrázku 4.2 lze vidět distribuce experimentálních a predikovaných ΔΔG hodnot pro jednotlivé nástroje. Na každém z uvedených grafů jsou na horizontální ose vyneseny



Obrázek 4.2: Porovnání různých nástrojů pro predikci změny stability. Každý nástroj byl testován na mutacích, které nebyly obsaženy v jejich trénovacích sadách. Na každém grafu v jeho horním rohu je zaznačen korelační koeficient (r) a rovnice regresní přímky (y). Plnou čarou je vyjádřena regresní přímka vypočtená z bodů na grafu. [39]

hodnoty experimentální ΔΔG, na vertikální ose je to získaná (predikovaná) hodnota ΔΔG. Přerušovaná čára s předpisem $y = x$ znázorňuje ideální polohu jednotlivých bodů. Plnou čarou je vyjádřena regresní přímka vypočtená z bodů grafu. Čím více regresní přímka překrývá přerušovanou přímku, tím je výsledek přesnější. Na každém grafu je v horním rohu zaznačen korelační koeficient (r) a rovnice regresní přímky (y).

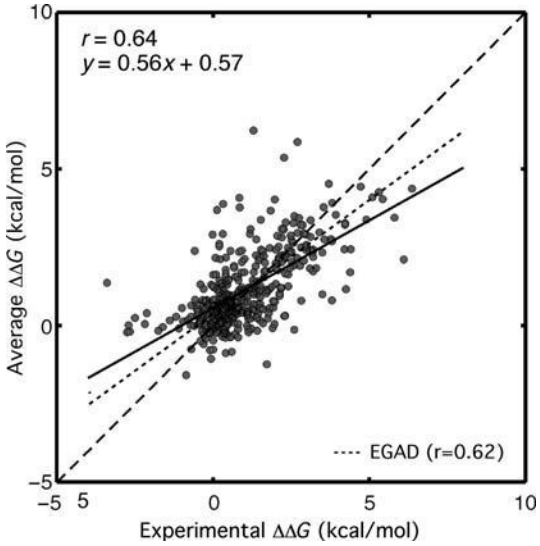
Na obrázku 4.3 je taktéž znázorněna distribuce experimentálních a predikovaných ΔΔG hodnot pro jednotlivé nástroje jak tomu bylo na obrázku 4.2. V tomto případě bylo ovšem použito metody tzv. *binning*, kde došlo k rozdělení daného prostoru na 12 intervalů a v každém intervalu byly jednotlivé výsledků zprůměrovány. Výsledkem je tedy jeden bod reprezentující hodnoty v určitém intervalu.



Obrázek 4.3: Porovnání různých nástrojů pro predikci změny stability s využitím metody *binning* pro 12 intervalů. Každý nástroj byl testován na mutacích, které nebyly obsaženy v jejich trénovacích sadách. Na každém grafu v jeho horním rohu je zaznačen korelační koeficient (r) a rovnice regresní přímky (y). Plnou čarou je vyjádřena regresní přímka vypočtená z bodů na grafu. [39]

Autoři této studie se taktéž zaměřili na kombinování výsledků různých metod s předpokladem, že dosáhnou lepšího výsledku. Celkově bylo vytvořeno 57 různých kombinací dvou a více nástrojů, kde výsledky těchto kombinací byly zprůměrovány. Ve výsledku ovšem došli k závěru, že kombinací různých metod nedojde k signifikantnímu zlepšení predikční přesnosti v porovnání s použitím jediného. Toto tvrzení je podloženo výsledkem zobrazeným na obrázku 4.4. Tento graf znázorňuje výsledek kombinování nástrojů s cílem zlepšit predikci ΔΔG. Výsledky nástrojů EGAD, I-Mutant2.0 a Rosetta byly zprůměrovány a zaneseny do grafu oproti experimentálně zjištěným hodnotám ΔΔG. Tečkovanou čarou je

znázorněna regresní přímka pro výsledky samotného nástroje EGAD. Lze si také všimnout, že korelační koeficient pro zprůměrované výsledky těchto nástrojů dosahuje hodnoty 0,64, pro samotný EGAD potom 0,62. Jak již bylo zmíněno výše, kombinováním (průměrováním) různých nástrojů nebylo dosaženo velkého zlepšení. [39]



Obrázek 4.4: Graf znázorňující výsledky kombinování nástrojů pro zlepšení predikce ΔΔG. EGAD, I-Mutant2.0 a Rosetta byly použity pro predikování ΔΔG na datasetu o 407 mutacích. Průměr těchto tří nástrojů byl vypočítán pro každou mutaci a zanesen do grafu. Tyto zprůměrované výsledky byly porovnány na stejném datasetu se samotným nástrojem EGAD (tečkovaná přímka). [39]

Další zajímavá studie [11] porovnává celkem 5 predikčních nástrojů, kterými jsou I-Mutant2.0, AUTO-MUTE, MUpro, PoPMuSiC a CUPSAT. Pro I-Mutant2.0 byla použita jeho sekvenční (I-Mutant_SEQ) i strukturní verze (I-Mutant_PDB). Pro nástroj AUTO-MUTE byly dostupné čtyři predikční modely, autoři této studie zvolili pro porovnání model využívající *random forest* (AUTO-MUTE_RF) a *support vector machine* (AUTO-MUTE_SVM). MUpro využívá modelu *support vector machine*, kde pro svoji predikci primárně používá sekvenčních informací. Tento nástroj umožňuje predikovat pouze celkový efekt na stabilitu proteinu (stabilní/nestabilní).

Pro porovnání výkonnosti jednotlivých nástrojů bylo použito dvou odlišných datasetů. Tyto datasety byly vytvořeny z databáze ProTherm. První dataset (S1948) byl použit při konstrukci I-Mutant2.0 a obsahuje 1948 mutací z celkem 58 proteinů. Druhý dataset (S2648) byl použit při trénování PoPMuSiC a obsahuje 2648 mutací z celkem 119 proteinů. V datasetu S1948 se nachází množství mutací se stejným PDB ID a stejnými hodnotami ΔΔG (mírně odlišné byly jen hodnoty pH a teploty). Těchto 637 redundantních záznamů bylo odstraněno, zbývajících 1311 mutací vytvořilo nový dataset pojmenovaný M1311. Dataset S2648 sdílel celkem 815 mutací s datasetem M1311, pro dosažení vzájemné nezávislosti těchto datasetů byly tyto mutace odstraněny. Celkově tedy druhý dataset obsahoval 1820 mutací a byl pojmenován M1820. Sloučením datasetů M1311 a M1820 vznikl třetí dataset s označením M3131.

V tabulce 4.4 jsou zobrazeny výsledky uvedených predikčních nástrojů pro dataset M1311. Matthewsův korelační koeficient se v tomto případě pohybuje v rozmezí od 0,341 pro CUPSAT do 0,906 pro nástroj AUTO-MUTE s predikčním modelem *random forest*.

Nástroj	**Specificita**	**Senzitivita**	**Přesnost**	**MCC**
I-Mutant_PDB	0,922	0,555	0,800	0,530
I-Mutant_SEQ	0,973	0,702	0.883	0,734
AUTO-MUTE_RF	0,991	0,893	0.958	0,906
AUTO-MUTE_SVM	0,975	0,772	0.907	0,789
MUpro_SVM	0,956	0,775	0.896	0,761
PoPMuSiC	0,941	0,313	0.724	0,341
CUPSAT	0,823	0,579	0.742	0,411
Průměr	**0,984**	**0,737**	**0.902**	**0,779**

Tabulka 4.4: Porovnání výsledků predikčních nástrojů pro dataset M1311. [11]

V tabulce 4.5 jsou zobrazeny výsledky uvedených predikčních nástrojů pro dataset M1820. Matthewsův korelační koeficient se zde pohybuje v rozmezí od 0,072 pro AUTO-MUTE s predikčním modelem *support vector machine* do 0,352 pro nástroj PoPMuSiC.

Nástroj	**Specificita**	**Senzitivita**	**Přesnost**	**MCC**
I-Mutant_PDB	0,906	0,198	0,670	0,148
I-Mutant_SEQ	0,899	0,212	0.670	0,155
AUTO-MUTE_RF	0,985	0,129	0.700	0,234
AUTO-MUTE_SVM	0,965	0,067	0.666	0,072
MUpro_SVM	0,885	0,276	0.682	0,206
PoPMuSiC	0,952	0,303	0.736	0,352
CUPSAT	0,757	0,370	0.628	0,133
Průměr	**0,984**	**0,113**	**0.693**	**0,212**

Tabulka 4.5: Porovnání výsledků predikčních nástrojů pro dataset M1820. [11]

Tabulka 4.6 obsahuje výsledky jednotlivých predikčních nástrojů pro dataset M3131 vzniklý sloučením dvou předcházejících datasetů. Matthewsův korelační koeficient se pohybuje v rozmezí od 0,261 pro CUPSAT do 0,615 pro nástroj AUTO-MUTE s predikčním modelem *random forest*.

Celkově nejlepších výsledků dosáhl nástroj AUTO-MUTE s predikčním modelem *random forest*. Je však nutné podotknout, že právě u tohoto nástroje byl trénovací dataset vytvořen z databáze ProTherm. Takto dobrý výsledek může být tedy způsoben neadekvátním použitím modelu a nemusí obecně korespondovat s výsledky na nezávislém datasetu.

Nástroj	**Specificita**	**Senzitivita**	**Přesnost**	**MCC**
I-Mutant_PDB	0,377	0,916	0,736	0,357
I-Mutant_SEQ	0,457	0,934	0.775	0,464
AUTO-MUTE_RF	0,511	0,989	0.829	0,615
AUTO-MUTE_SVM	0,420	0,969	0.786	0,499
MUpro_SVM	0,526	0,908	0.780	0,480
PoPMuSiC	0,308	0,945	0.733	0,348
CUPSAT	0,474	0,780	0.678	0,261
Průměr	**0,425**	**0,980**	**0.795**	**0,527**

Tabulka 4.6: Porovnání výsledků predikčních nástrojů pro dataset M3131.

Kapitola 5

Strojové učení

Strojové učení je v dnešní době chápáno jako disciplína umělé inteligence. Její základní technikou je prohledávání stavového prostoru. K charakteristickým rysům patří využívání znalostí, práce se symbolickými či strukturovanými proměnnými či aplikace moderních poznatků z oboru nestandardních logik. Typicky se v těchto úlohách hledají zajímavé souvislosti či průběhy pozorovaných jevů, které lze považovat za charakteristické. Nejtypičtější aplikací strojového učení je pomoc při získávání znalostí pro expertní systémy, kde bylo dosaženo výrazných úspěchů v podobě zkrácení doby nutné pro tvorbu a ladění báze znalostí. Další uplatnění strojového učení je například při porozumění přirozenému jazyku, v počítačovém vidění nebo právě v bioinformatice.

Dá se říci, že strojové učení patří mezi nejstarší disciplíny matematické informatiky. Proto se již od padesátých let hledají způsoby, jak tvorbu programů zautomatizovat. Strojové učení založené na umělé inteligenci je jednou z metod této automatizace. [29]

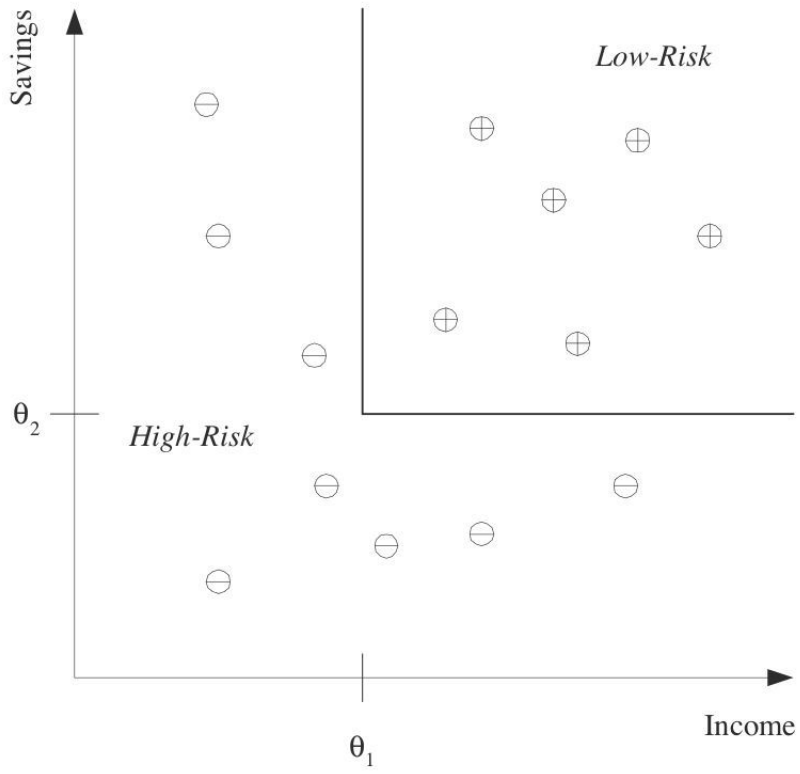
Dle [3] lze rozdělit algoritmy strojového učení na

- klasifikaci,
- regresi a
- hledání asociací.

Klasifikace, resp. klasifikační problém je takový problém, který řeší přiřazení tříd objektům. Typickou úlohou udávanou jako příklad klasifikace je určení rizikovosti půjčky. O jednotlivých zákaznících jsou uchovávány všechny relevantní informace ovlivňující schopnost splácet půjčku (příjem, úspory, povolání, věk atd.). Cílem je najít asociace mezi zákazníkovými atributy a rizikem nesplacení. Toto je klasický příklad klasifikačního problému pro dvě třídy (nízká a vysoká rizikovost půjčky). Vstupem jsou tedy informace o zákazníkovi, výstupem jsou tyto dvě třídy (vysoká/nízká rizikovost). Po natrénování modelu může být klasifikační pravidlo pro tuto úlohu například ve tvaru

IF prijem $> \Theta_1$ AND uspory $> \Theta_2$
THEN nizka rizikovost ELSE vysoka rizikovost.

Na obrázku 5.1 je znázorněn příklad rozdělení prostoru možných řešení. Horizontální osa reprezentuje velikost příjmu, vertikální osa znázorňuje velikost úspor. Označené body Θ_1 a Θ_2 určují hranice rozdělení prostoru. Kružnice zde reprezentují datové instance, znaménkem + jsou označeny instance patřící do třídy nízkorizikových půjček, znaménku - patří třída vysokorizikových půjček. Plnou čarou je znázorněno rozdělení těchto tříd v prostoru.



Obrázek 5.1: Příklad trénovacího datasetu, kde každá kružnice náleží jedné datové instanci. Tyto instance reprezentují vstupy zobrazené na příslušných souřadnicích, kde znaménka + či - určují příslušnost do třídy nízkorizikové resp. vysokorizikové. Plnou čarou je znázorněno oddělení těchto tříd. [3]

Regresní metody, na rozdíl od klasifikace, neurčují do jaké třídy vstupní prvek patří, ale rovnou odhadují (predikují) jeho číselnou hodnotu. Jako příklad lze uvést systém, který bude predikovat cenu ojetého automobilu. Vstupem mohou být atributy jako značka automobilu, rok výroby, počet najetých kilometrů atd. Pro jednodušší znázornění uvažme počet najetých kilometrů jako jediný atribut ovlivňující cenu automobilu. Regresní přímka poté nabývá lineární tvar

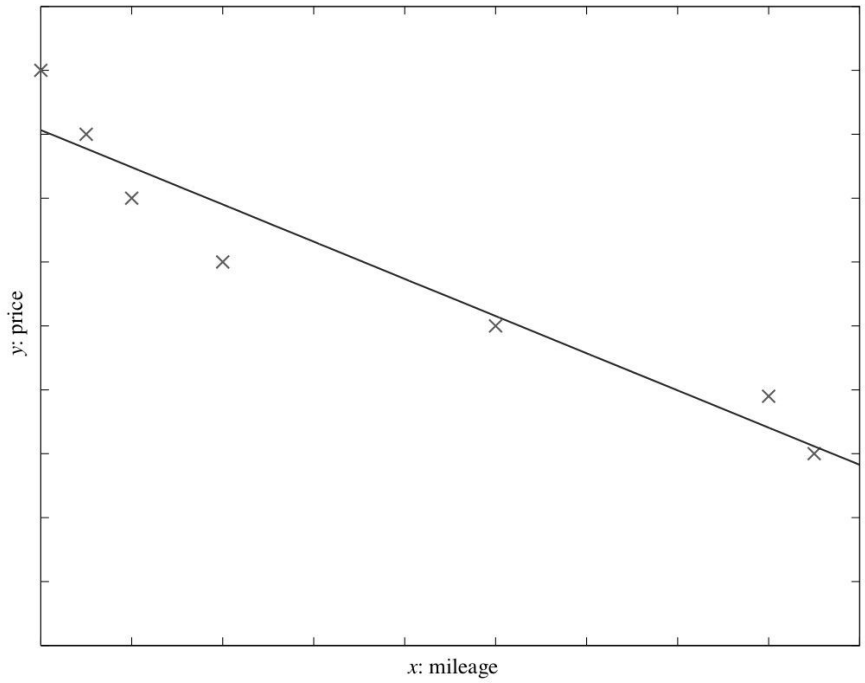
$$y = wx + w_0.$$

Na obrázku 5.2 je příklad lineárně řešitelného problému závislosti ceny automobilu na počtu ujetých kilometrů (mílích). Pokud je lineární model příliš omezující, lze využít například kvadratickou funkci

$$y = w_2x^2 + w_1x + w_0,$$

polynomiální funkci vyšších řádů či jinou nelineární funkci.

Asociační pravidla (*association rules*) jsou využívána pro hledání zajímavých asociací nebo korelací nad velkým množstvím datových položek. Nalezení zajímavých asociací nad obchodními transakčními záznamy může pomoci v procesu obchodního rozhodování, jako je



Obrázek 5.2: Příklad trénovacího datasetu pro výpočet ceny ojetého automobilu. Pro zjednodušení je uvažován pouze jeden vstupní atribut (počet najetých kilometrů), jedná se tedy o lineární model. Regresní přímka je určující predikovanou hodnotou, je dána předpisem $y = wx + w_0$. [3]

návrh katalogů, akčních nabídek nebo rozmístění zboží v obchodě. Typickým příkladem je analýza nákupního košíku. Tento proces analyzuje chování zákazníka, hledá asociace mezi zbožím, které zákazník umístí do svého nákupního košíku. Tímto lze tedy zjistit, jaké druhy zboží si zákazníci nejčastěji kupují dohromady.

Při hledání těchto asociačním pravidel nás zajímá zejména podmíněná pravděpodobnost uváděná ve formě $P(Y|X)$, kde Y je produkt podmíněný výskytem produktu X, což je produkt nebo množina produktů, u kterých víme, že je zákazník nakupuje. Uveďme například pravděpodobnost $P(limonada|oplatky) = 0,7$. Tímto výrazem definujeme, že 70 procent zákazníků, kteří si koupili oplatky taktéž koupili limonádu.

Dle [3] lze algoritmy strojového učení podle způsobu učení rozdělit na

- učení s učitelem a
- učení bez učitele.

Pro učení s učitelem je specifické to, že při fázi učení jsou kromě vstupních dat dostupná i data výstupní. Učitel je tedy schopný získat výsledky z daného modelu a porovnat je s požadovaným výstupem. Mezi algoritmy strojového učení, které je možné zařadit do této kategorie, patří klasifikace i regrese.

Naopak pro učení bez učitele je specifické to, že nejsou k dispozici data výstupní (není tedy možné výstup jednotlivých modelů strojového učení porovnat s jakýmkoli jiným výstupem). Typickým příkladem může být například technika shlukování (*clustering*), která

slouží k třídění jednotek do shluků tak, aby si objekty náležící do stejné třídy byly podobnější (podobnost určena např. pomocí vzdálenosti) než objekty z různých tříd.

5.1 Generalizační schopnost a její odhad

Určíme-li jednoznačné hodnotící hledisko úspěšnosti modelů, můžeme ho využít ke stanovení a porovnání generalizační schopnosti relevantních modelů. Při vytváření modelů strojového učení není rozhodující jejich výkonnost či přesnost nad známými daty (takovými daty, která byla využita při vytváření modelu či jejich trénování), ale nad daty neznámými (nezávislá data, která nebyla použita při trénování modelu). Právě generalizační schopnost je vztažena k výkonnosti modelu nad neznámými daty a hraje tedy důležitou roli při výběru daného modelu. [30]

Pokud se více zaměříme na počet objektů (příkladů), které máme k dispozici pro řešení daného problému, lze situaci dle [30] rozdělit na dva případy.

1. **K dispozici je dostatek objektů dostatečně reprezentujících modelovaný problém.** Pro tento příklad se výchozí množina dat standardně rozdělí na tři disjunktní podmnožiny, a to trénovací (*training set*) D_t, testovací (*testing set*) D_e a validační (*validation set*) D_v ($D = D_t \cup D_e \cup D_v$). Trénovací data jsou použita ve fázi učení jednotlivých modelů. Testovací data slouží k ověření jejich prediktivních schopností a k volbě nejlepšího kandidáta. Pokud roste během konstrukce modelu jeho složitost (velikost), je potřeba odhadnout okamžik, kdy s procesem trénování přestat, aby nedošlo k přeučení (více o této problematice lze nalézt v podkapitole 5.1.2). Z tohoto důvodu se během učení zároveň měří chyba na množině D_e (chyba na D_e zpočátku trénování klesá, později v důsledku přeučení na D_t začne stoupat). Právě v okamžiku dosažení minima chyby na D_e je ukončena konstrukce modelu. Validační data jsou použita k nezávislému odhadu generalizační síly zvoleného modelu tak, že se na nich určuje chyba predikce modelu. Validační dataset je tedy skutečně nezávislý na trénovacím procesu (na rozdíl od testovacího datasetu, který není nevychýleným odhadem generalizační schopnosti). Typicky používané rozdělení je 50 % dat pro trénovací dataset, 25 % pro testovací dataset a zbylých 25 % pro validační dataset.

2. **K dispozici není dostatek reprezentujících objektů.** V tomto případě musíme generalizační schopnost odhadovat jiným způsobem. Výše uvedené rozdělení je v tomto případě nevhodné zejména ze dvou důvodů. Data použitá pro validaci a testování snižují počet trénovacích příkladů. Zároveň počet příkladů v těchto množinách není dostatečný na to, aby byl odhad generalizační schopnosti spolehlivý. V těchto případech je možné použít metody křížové validace (*cross-validation*) nebo tzv. bootstrapping. Tyto metody využívají speciálního způsobu rozdělení dat na trénovací a testovací nebo validační.

Jak bylo uvedeno výše, pokud nemáme k dispozici dostatek reprezentujících objektů pro vytvoření, validaci a testování modelu strojového učení, lze využít metod křížové validace a bootstrapping.

- Křížová validace (*cross-validation*) je metoda založená na náhodném rozdělení příkladů

do K disjunktních množin[1]. Dataset X je tedy náhodně rozdělen do K částí ekvivalentních velikostí $(X, i = 1, \ldots, K)$. Po vygenerování těchto K částí datasetu je jedna část použita pro validaci a zbývajících $K-1$ částí je použito pro vytvoření trénovacího datasetu. Tento postup je opakován K-krát, kde při každém běhu je použit jiný dataset pro validaci. Celkový výsledek je potom zprůměrován. Vytvoření částí datasetu je tedy následující:

$$\begin{aligned} \nu_1 &= X_1 \qquad \tau_1 = X_2 \cup X_3 \cup \cdots \cup X_K \\ \nu_2 &= X_2 \qquad \tau_2 = X_1 \cup X_3 \cup \cdots \cup X_K \\ &\vdots \\ \nu_K &= X_K \qquad \tau_K = X_1 \cup X_2 \cup \cdots \cup X_{K-1}. \end{aligned}$$

K typicky nabývá hodnoty 10 či 30. Extrémním případem *K-Folf Cross-Validation* je metoda zvaná *leave-one-out*. Dataset obsahuje N instancí, ale pouze jedna instance je použita pro validaci. Pro trénování je pak požito zbývajících $N-1$ instancí. Zároveň je zřejmé, že tato metoda maximalizuje velikost trénovacích dat. Praktické využití metody *leave-one-out* je možné najít zejména v aplikacích medicínské diagnostiky. [3]

- Bootstrapping [18] je metoda založená na statistickém náhodném výběru s opakováním. Tento postup využívá toho, že většina učících algoritmů může pracovat se dvěma nebo více stejnými trénovacími instancemi a že jejich počet může ovlivnit výsledek učení. Nejčastější postup je takový, že pokud máme N příkladů, generujeme z nich trénovací data opět s N příklady na základě náhodného výběru s opakováním (navracením). Pravděpodobnost, že bude vybrána jedna instance je $\frac{1}{N}$. Pravděpodobnost, že vybrána nebude je $1 - \frac{1}{N}$. Pravděpodobnost, že daná instance nebude vybrána po N opakování, lze vyjádřit jako

$$\left(1 - \frac{1}{N}\right)^N \approx e^{-1} = 0,368.$$

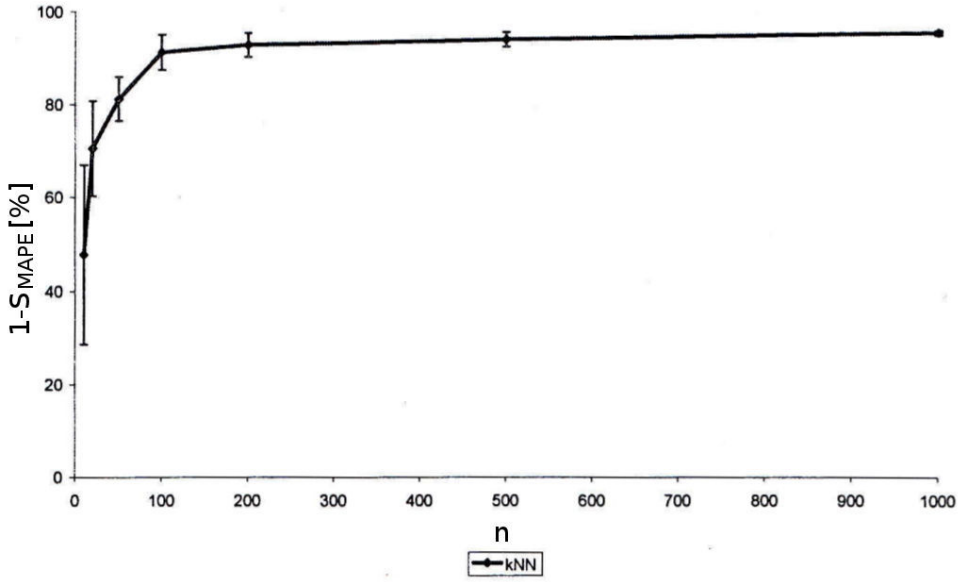
Z tohoto zápisu vyplývá, že přibližně 63,2 % instancí bude použito pro trénovací dataset. Zbylých 36,8 % dat bude použito pro testování. Celá procedura dělení na trénovací a testovací data bude několikrát opakována a výsledek jednotlivých testů bude zprůměrován. [3]

5.1.1 Křivka učení

Dalším neméně důležitým ukazatelem schopnosti modelu induktivně generalizovat daná data může být křivka učení (viz obrázek 5.3). Při učení je předpokládáno, že generalizační schopnost poroste se zvyšující se sumou zkušeností, která je zde reprezentována objemem předložených trénovacích případů. Křivku učení je možné generovat takovým způsobem, že na horizontální ose je zobrazen zvyšující se počet příkladů, na kterých se algoritmus může učit. Na ose vertikální jsou vyneseny jednotlivě zjištěné testovací chyby (odhad generalizační schopnosti). Na počátku křivky se zpravidla vyskytuje zcela neinformovaný model (s náhodnou počáteční parametrizací), na jejím konci je naopak počet trénovacích příkladů maximální, popřípadě takový, kdy už generalizační schopnost dále neroste. Pokud je zvolený model schopen s dostatečnou přesností popsat vzor chování charakterizující

[1] Lze se také setkat s pojmem *K-Fold Cross-Validation*, kde K značí počet disjunktních množin, na které je dataset rozdělen.

zpracovávaná data, pak se pro velký počet trénovacích dat musí počet chyb stále snižovat a hodnota S_{MAPE}[2] se blíží k hodnotě 0, jak je možné vidět na obrázku 5.3 (v tomto případě $1 - S_{MAPE}$). Tohoto tvaru křivky se ovšem nepodaří dosáhnout, pokud model není vhodný pro popis chování zpracovaných dat. V případě, že je celkový počet příkladů omezený či nízký, nemůžeme pracovat s konstantní množinou testovacích dat, ale příklady postupně dělíme mezi trénovací a testovací dataset. Je ovšem zřejmé, že zejména v obou krajních oblastech křivky je k dispozici pouze malý počet trénovacích (resp. testovacích) dat a pro minimalizaci statistické chyby je nutné experimenty opakovat. Z tohoto důvodu při konstrukci křivky učení často využíváme analogie křížové validace. Kromě průměrné zjištěné chyby znázorňujeme i její standardní odchylku. [30]



Obrázek 5.3: Křivka učení modelu k-nejbližších sousedů (pro $k = 3$) v úloze predikce spotřeby plynu. Data byla rozdělena na trénovací dataset obsahující 1460 příkladů (pro roky 1997-2000) a testovací dataset obsahující 365 případů pro rok 2001. Postup vytvoření křivky byl následující: Z trénovací množiny náhodně vyber n příkladů, ty skutečně použij pro trénink. Model ověř nad testovacími daty, kde hodnotícím kritériem je průměrná absolutní procentuální přesnost predikce $(1 - S_{MAPE})$. Náhodný výběr opakuj 10krát, výsledky zprůměruj, vynes průměrnou přesnost předpovědi a její standardní odchylku. (přepracováno z [30])

5.1.2 Přeučení

Přeučení (*overfitting*), někdy taktéž nazýváno přetrénování. Rostoucí počet trénovacích cyklů vede k postupnému přizpůsobování modelu trénovacím datům a také k růstu jeho

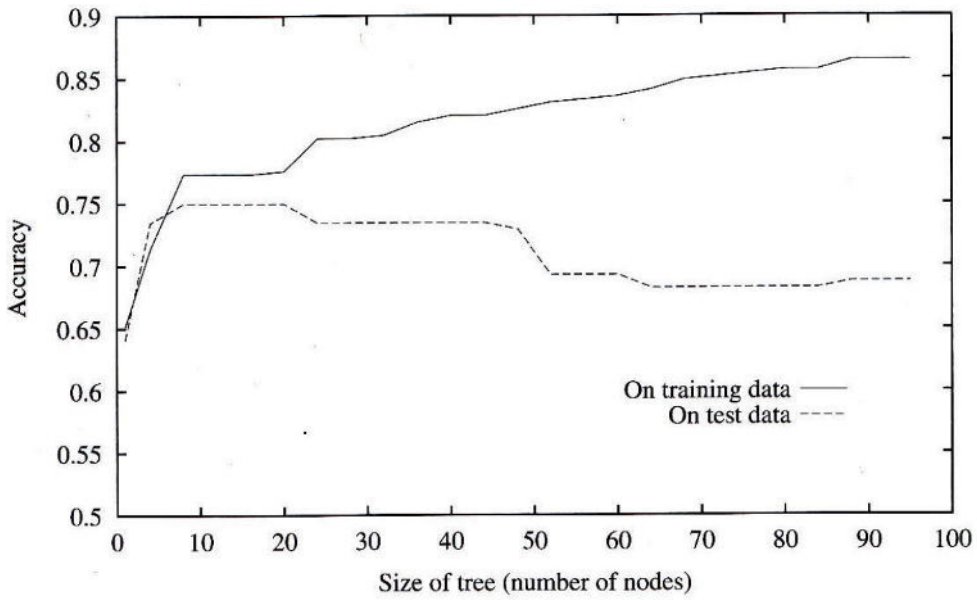
[2]Střední absolutní relativní chyba (*mean absolute precentage error, MAPE*) je využívána jako základní hodnotící funkce pro regresní modely. Definována je jako

$$S_{MAPE}(M, \theta) = \frac{1}{n} \sum_{i=1}^{n} \frac{|\tilde{y}(i, M, \theta) - y(i)|}{y(i)}.$$

složitosti. Od jistého okamžiku již ale tato složitost nepřispívá ke zlepšení generalizační schopnosti, tento efekt se nazývá přeučení modelu. [30].

> ***Definice:*** *Uvažujme hypotézu v nějakém prostoru H, kde platí $h \in H$. O h mluvíme jako o přeučení na tréninkových datech, jestliže existuje nějaká alternativní hypotéza $h' \in H$ taková, že h má menší chybu než h' oproti trénovacím datům a zároveň h' má menší chybu než h oproti celé distribuci dat.* [34]

Obrázek 5.4 ilustruje účinek přeučení u typického příkladu učení rozhodovacího stromu. V tomto případě je použit ID3[3] algoritmus aplikovaný na lékařskou úlohu se snahou určit, kteří pacienti mají diabetes. Na horizontální ose tohoto grafu jsou vyneseny jednotlivé počty uzlů rozhodovacího stromu tak, jak byl strom postupně vytvářen. Vertikální osa zobrazuje přesnost predikcí určených pomocí rozhodovacího stromu. Plnou čarou jsou znázorněny přesnosti predikcí rozhodovacího stromu na tréninkových datech, zatímco přerušovanou čarou jsou zobrazeny přesnosti naměřené na nezávislém datasetu testovacích dat (tato data nebyla součástí trénovacího datasetu). Jak je očekávané, přesnost predikce na trénovacích datech se monotónně zvyšuje s rostoucím počtem uzlů stromu. Přesnost měřená na nezávislých testovacích datech nejdříve roste, poté se již ale snižuje. Je možné si všimnout toho, že od velikosti stromu obsahujícího přibližně 25 uzlů se přesnost pro tréninková data dále zvyšuje, naopak pro data testovací přesnost klesá.



Obrázek 5.4: Přeučení pro model strojového učení využívající rozhodovacího stromu (ID3). Přesnost naměřená na rozhodovacím stromu vytvořeném na trénovacích datech je monotónně rostoucí. Pokud je měření provedeno na datech zcela nezávislých (testovacích) je přesnost nejprve rostoucí, ale od určitého bodu již lze zaznamenat klesající tendenci. [34]

Jak je ovšem možné, že daný strom dosahuje lepší přesnosti na trénovacích datech než na datech testovacích? Odpovědí na tuto otázku může být přítomnost náhodných chyb

[3]Technika ID3 se používá při konstrukci rozhodovacího stromu shora dolů. Odpovídá na otázku jaký atribut zvolit jako uzel v dané úrovni stromu.

či šumu v tréninkovém datasetu. Konkrétně pro rozhodovací stromy to znamená zanesení nesprávných rozhodovacích podmínek pro uzly blízko listů stromu.

Efekt přeučení není závažný problém jenom pro metody využívající rozhodovacích stromů, ale i pro ostatní metody strojové učení. Například v experimentální studii [33] zabývající se ID3 metodami na pěti různých úlohách obsahujících šum či nedeterministická data, bylo zjištěno, že přeučení rozhodovacího stromu sníží přesnost predikce o 10-25% u většiny definovaných problémů.

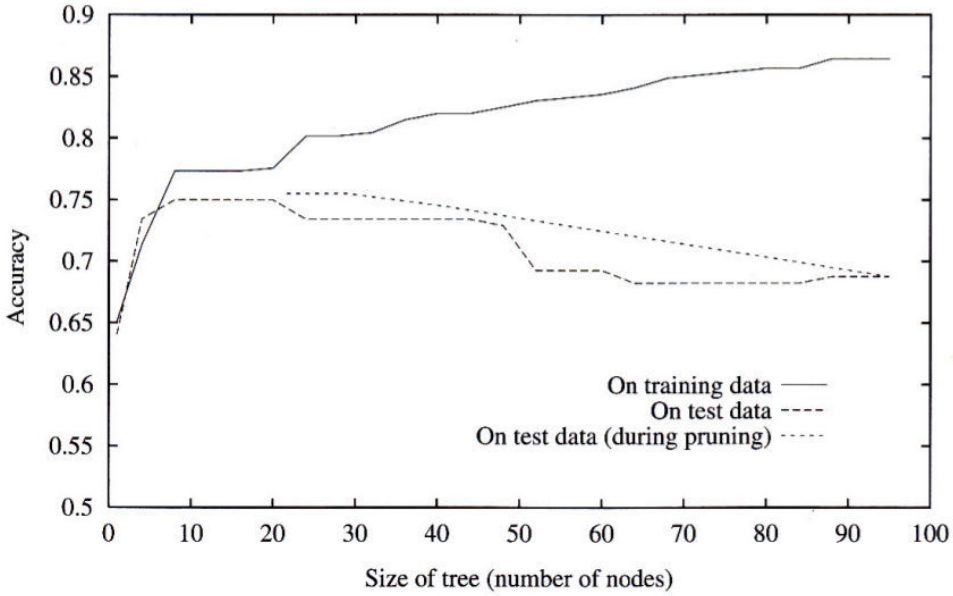
Pro problémy přeučení pro rozhodovací stromy existuje několik přístupů jak zamezit přeučení. Tyto přístupy mohou být shrnuty do dvou tříd:

- přístupy, které ukončí generování stromu dříve než by se mohl projevit efekt přeučení,
- přístupy, které dovolí přeučení na datech, ale poté použijí metody prořezání stromu (*post-prune*).

Ačkoliv se první přístup může zdát jako více přímý, v praxi se spíše osvědčil druhý přístup prořezání rozhodovacího stromu. Nevýhoda prvního přístupu je v tom, že není zcela jasné, kdy ukončit růst stromu. Bez ohledu na to, zda je výsledná velikost stromu určena pomocí prvního přístupu či použitím prořezání, klíčovou otázkou zůstává, jaké je kritérium pro stanovení správné velikosti stromu vedoucí k co nejlepším výsledkům. Dle [34] mezi hlavní přístupy lze zařadit:

- Použít nezávislý dataset, který je odlišný od trénovacích příkladů, k vyhodnocení užitečnosti prořezání stromu.
- Použít všechna dostupná data pro trénování. Rozhodnutí, zda rozšířit či prořezat konkrétní uzel, který by s největší pravděpodobností produkoval zlepšení i mimo trénovací množinu, ponechat na výsledku statistického testu. Například v [40] je použit chí-kvadrát pro testování a odhadnutí, který uzel a kdy je nutné rozšířit.
- Použít explicitní míru složitosti pro zakódování tréninkových příkladů a rozhodovacího stromu, kdy je zastaven růst stromu a je minimalizována velikost tohoto kódování. Tento přístup je založen na heuristice nazvané *Minimum Description Length principle*, podrobnější diskuzi lze nalézt v [32].

Na obrázku 5.5 lze vidět porovnání přesností modelu vytvořeného pomocí tréninkových dat, testovacích dat a pomocí prořezání stromu na testovacích datech. Jak je vidět na tomto obrázku, použitím metody prořezání stromu lze dosáhnout lepší přesnosti modelu (v některých místech dosahuje zlepšení přibližně o 0,05 oproti testovacím datům bez použití prořezání stromu).
Další příklad potvrzující snížení přesnosti predikce modelu vlivem přeučení, lze nalézt na obrázku 5.6. Tento obrázek vyjadřuje přeučení vrstvené neuronové sítě v úloze předpovědi úmrtnosti. Pro lepší porovnání jsou zde zobrazeny dvě architektury. První z nich pracuje s menším počtem neuronů ve skrytých vrstvách. Druhá architektura je složitější, přesnější, ale má větší tendenci k přeučení (toto lze jednoduše pozorovat jako rozdíly mezi výsledky trénovacích a validačních dat jednotlivých architektur). Je zřejmé, že přibližně okolo 500. cyklu učení se již přesnost na validačních datech příliš nezvětšuje, naopak dojde ke zhoršení dosažených výsledků.



Obrázek 5.5: Redukce chyb použitím techniky prořezání rozhodovacího stromu. Tento graf ukazuje stejné hodnoty křivek pro testovací a tréninková data jako graf 5.4. Na rozdíl od obrázku 5.4 jsou zde redukovány chyby pomocí prořezání stromu vytvořeného pomocí metody ID3. Porovnáme-li přesnost modelu na testovacích datech bez prořezání a s prořezáním stromu zjistíme, že v některých případech dojde ke zlepšení přesnosti o zhruba 0,05. Technika prořezání rozhodovacího stromu v tomto případě vykazuje lepších výsledků. [34]

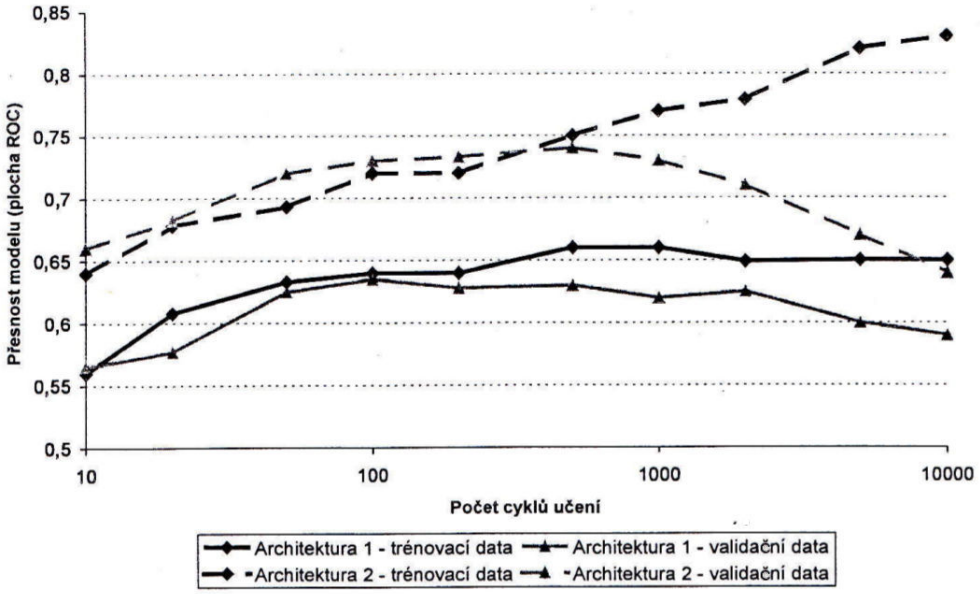
5.2 WEKA - platforma pro analýzu znalostí

WEKA (z angl. *Waikato Environment for Knowledge Analysis*) je populární balík programů strojového učení napsaný v programovacím jazyce Java, vyvinutý na University of Waikato, Nový Zéland. WEKA je svobodný software dostupný podle licence *GNU General Public License*. Platforma WEKA je široce rozšířená v akademické i komerční sféře, disponuje aktivní komunitou a byla stažena více než 1,4 milionkrát od uveřejnění na Source-Forge (od dubna 2000).

Cílem projektu WEKA je poskytnout rozsáhlou kolekci různých algoritmů pro úlohy strojového učení a nástroje pro předzpracování dat pro vědeckou i veřejně odbornou komunitu. Umožňuje uživatelům rychle vyzkoušet a porovnat různé techniky strojového učení na vytvořeném datasetu. Modulární, rozšiřitelná architektura umožňuje sofistikované dolování dat z poskytnutých kolekcí učících algoritmů a nástrojů. Rozšíření tohoto nástroje je velmi snadné díky jednoduchému API a plugin mechanismům, které automatizují integraci nových algoritmů do WEKA pomocí grafického rozhraní. WEKA obsahuje algoritmy pro regresi, klasifikaci, shlukování, získávání asociačních pravidel a výběr atributů (rysů). O předběžný pohled na distribuci a vlastnosti dat je postaráno pomocí nástrojů pro vizualizaci, nabídnuto je taktéž velké množství dalších nástrojů pro předzpracování. [21]

Samotný nástroj WEKA lze rozdělit do čtyř různých aplikací.

- **Explorer** je hlavním grafickým uživatelským rozhraním. Toto rozhraní využívá panelů (*panel-based*), kde jednotlivé panely korespondují s daným typem úlohy.



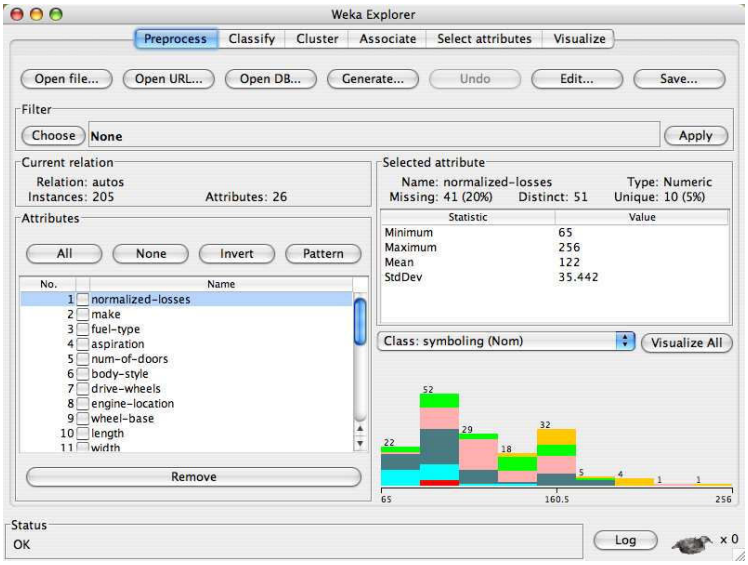
Obrázek 5.6: Přeučení vrstvené neuronové sítě v úloze předpovědi úmrtnosti. Architektura 1 pracuje s menším počtem neuronů ve skrytým vrstvách. Architektura 2 je složitější, přesnější, ale má větší tendenci k přeučení. S rostoucím počtem iterací při učení neuronové sítě dochází k postupnému zpřesňování klasifikace trénovacích dat. Je zde však viditelné, že přibližně od 500. cyklu učení se již nezvyšuje generalizační schopnost sítě a kvalita předpovědi se zhoršuje. Složitost modelu je vysoká, dochází k jevu, který nazýváme přeučení. [30]

První panel nazvaný *Preprocess* slouží pro nahrání dat do modelu a umožňuje využít nástrojů pro předzpracování (filtry). Data mohou být nahrána z databáze, souboru či URL. Podporované formáty souborů jsou ARFF (nativní formát pro nástroj WEKA), CSV, LibSVM formát a C4.5 formát.

Druhý panel s názvem *Classify* umožňuje přístup k výběru klasifikačních a regresních algoritmů. V tomto panelu je možné pracovat i s křížovou validací (možnost nastavit počet foldů), ve výchozím nastavení je použita 10-fold křížová validace. Možnost využití externího testovacího datasetu je taktéž podporována.

WEKA samozřejmě poskytuje kromě algoritmů využívající učení s učitelem i algoritmy bez učitele. Ve třetím panelu je možné najít algoritmy shlukovací, ve čtvrtém panelu pak metody pro hledání asociačních pravidel. V panelu *Cluster* je povoleno uživatelům využívat shlukovací algoritmy na datech nahraných v panelu *Preprocess*. Samozřejmostí jsou jednoduché statistické výstupy hodnotící výkonnost shlukovacích algoritmů.

Pravděpodobně jedna z nejdůležitějších úloh praktického dolování dat je identifikace atributů (rysů), které se největší měrou podílejí na úspěšnosti predikce. V nástroji WEKA je výběr těchto rysů (*feature selection*) umístěn v panelu *Select attributes*. Vzhledem k faktu, že je možné kombinovat různé prohledávací metody s odlišnými evaluačními kritérii, je zde důležité ponechat širokou škálu možných kandidátních technik. Robustnost výběru atributů může být validována skrze přístupy založené na



Obrázek 5.7: Uživatelské rozhraní programu WEKA Explorer. [21]

křížové validaci. V mnoha praktických aplikacích vizualizace dat poskytuje důležité poznatky. Tyto poznatky mohou dokonce vést k tomu, že je možné se dále vyhnout analýze pomocí strojového učení a dolování dat. Pokud toto není možné, může vizualizace posloužit například pro výběr vhodného algoritmu. Možnost vizualizace je možné najít v posledním panelu nazvaném *Visualize*, který obsahuje jednotlivé barevně odlišené bodové grafy.

- **Experimenter**. Toto rozhraní je navrženo tak, aby co nejvíce usnadnilo porovnávání výkonností predikčních algoritmů založených na různých hodnotících kritériích, které jsou k dispozici ve WEKA. Experimenty je možné provádět na vícero algoritmech, které běží na vícero datasetech (například opakovaná křížová validace). Experimenty mohou být rovněž distribuovány na různých výpočetních uzlech na síti pro snížení výpočetního zatížení. Výsledky experimentu je možné uložit ve formě XML či v binární formě.

- **KnowledgeFlow**. Některé algoritmy strojového učení nebyly implementovány přímo do prostředí *Explorer*, ale jejich inkrementální povaha (tj. taková povaha, kde algoritmy lze rozdělit do posloupnosti jednotlivých operací) byla vložena do grafického uživatelského rozhraní nazvaného *KnowledgeFlow*. Většinu úloh, které je možné řešit v prostředí *Explorer*, lze spustit i v *KnowledgeFlow*. Toto prostředí nabízí celkem osm panelů, kde každý panel obsahuje jemu příslušné moduly (uzly), které je možné umístit na pracovní plochu. Tyto moduly mohou být formou vazeb mezi sebou pospojovány a vytvořit tak funkční tok dat. Samozřejmostí jsou nástroje pro ohodnocení i pro vizualizaci dat. Propojení jednotlivých modulů je konfigurovatelné a pro pozdější použití je možné danou konfiguraci uložit.

- **Simple CLI** je jednoduché konzolové prostředí pro nástroj WEKA pomocí něhož je možné jednoduše vytvářet sady příkazů (např. vytvoření či ohodnocení modelu). Pomocí této konzole je možné ovládat program WEKA bez znalostí vyšších programovacích jazyků.

Jak již bylo zmíněno, nástroj WEKA umožňuje práci s velkým množstvím algoritmů pro klasifikaci, regresi, shlukování nebo analýzu asociačních pravidel. V tabulce 5.1 jsou uvedeny jednotlivé třídy algoritmů, do kterých jsou konkrétní zástupci přiřazeni na základě způsobu klasifikace/regrese. Podrobné vysvětlení jednotlivých metod lze nalézt na [46].

Třída algoritmu	**Příklady jednotlivých algoritmů**
Bayes	AODE, BayesNet, NaiveBayes, NaiveBayesSimple
Functions	SMO, LinearRegression, MultilayerPerceptron
Lazy	IB1, IBK, KStar, LBR, LWL
Meta	Bagging, Random SubSpace, GridSearch, Vote, Stacking
Mi	MIBoost, MDD, MINND, MISVM, MIWrapper, MILR
Misc	VFI, HyperPipes
Rules	DecisionTable, M5Rules, ZeroR, JRip, ConjunctiveRule
Trees	M5P, J48, RandomForest, REPTree, ID3, ADTree

Tabulka 5.1: Uvedeny jsou jednotlivé třídy algoritmů, do kterých jsou konkrétní zástupci přiřazeni na základě způsobu klasifikace/regrese. Toto rozdělení platí pro pro verzi WEKA 3.6.10. [46]

5.2.1 KStar

V této podkapitole bude podrobněji rozebrána metoda strojového učení KStar, která dosáhla nejlepšího výsledku na trénovacím datasetu (viz kapitola 7.2).

KStar patří do kategorie *lazy learning* metod. Obecně lze o skupině těchto metod říci, že uchovávají tréninkové instance (data) a nedělají žádnou reálnou práci až do okamžiku, kdy je vznesen požadavek (na rozdíl od *Eager learning*). KStar je metoda využívající principu nejbližšího souseda se zobecněnou vzdálenostní funkcí založenou na transformacích. [46]

Použití entropie jako míry vzdálenosti má několik výhod. Mezi ně patří například konzistentní přístupy k symbolickým atributům, reálným hodnotám atributů a chybějícím hodnotám.

Samotná klasifikace je založena na podobnosti, kde vycházíme z předpokladu, že podobné instance budou mít podobné výsledky klasifikace. Otázka ovšem leží na definici "podobné instance"a "podobné výsledky klasifikací". Odpovědí je vzdálenostní funkce, která určuje, jak si jsou navzájem dvě instance podobné, a klasifikační funkce, která specifikuje podobnost instancí oproti výsledku klasifikace nových instancí.

Entropie a míra vzdálenosti

Tento přístup výpočtu na základě vzdálenosti mezi dvěma instancemi je motivován teorií informací. Přirozenou intuicí lze definovat vzdálenost dvou instancí jako složitou transformaci jedné instance na druhou. Výpočet této složitosti je možné rozdělit do dvou základních kroků. Prvním krokem je vytvoření konečné množiny transformací, která mapuje instance na instance definované. Program dále transformuje jednu instanci (a) na jinou (b) vytvořením konečné sekvence transformací začínající v a a končící v b.

V návaznosti na teorii složitosti jsou programy (sekvence) tvořeny bez prefixů připojením ukončovacího symbolu ke každému řetězci. Obvyklou definicí složitosti programu (definováno jako Kolmogorovova složitost v [27]) je délka nejkratšího řetězce reprezentující

daný program. Kolmogorovova vzdálenost mezi dvěma instancemi může být definována jako vzdálenost nejkratšího řetězce spojující tyto dvě instance. Tento přístup je zaměřen na jedinou transformaci (tu nejkratší) z množiny mnoha možných transformací. Výsledkem je taková vzdálenostní míra, která je velmi citlivá na malé změny v prostoru instancí. KStar se s tímto problémem snaží vypořádat pomocí součtu přes všechny možné transformace mezi dvěma instancemi.

Specifikace KStar

Nechť $\mathbf{I}$ (možno nekonečná) množina instancí a $\mathbf{T}$ je konečná množina transformací na $\mathbf{I}$. Je definováno zobrazení t, kde pro každé $t \in \mathbf{T}$ zobrazuje instance na instance $t : \mathbf{I} \to \mathbf{I}$. $\mathbf{T}$ obsahuje rozlišujícího člena σ (symbol pro zastavení), který doplňuje zobrazení o zobrazení samo na sebe (reflexivita, tj. $\sigma(a) = a$).

Nechť $\mathbf{P}$ je množina všech prefixových kódů z $\mathbf{T}^*$, které jsou ukončeny σ. Prvky $\mathbf{T}^*$ (a taktéž z $\mathbf{P}$) jsou jednoznačně definovány transformací na $\mathbf{I}$:

$$\bar{t}(a) = t_n(t_{n-1}(\dots t_1(a)\dots)), \quad kde \;\; \bar{t} = t_1, \dots, t_n. \tag{5.1}$$

Pravděpodobnostní funkce p je definována na $\mathbf{T}^*$ a musí splňovat následující vlastnosti:

$$0 \leq \frac{p(\bar{t}u)}{p(\bar{t})} \leq 1 \tag{5.2}$$

$$\sum_u p(\bar{t}u) = p(\bar{t}) \tag{5.3}$$

$$p(\Lambda) = 1. \tag{5.4}$$

V důsledku tohoto splňuje následující:

$$\sum_{\bar{t} \in P} p(\bar{t}) = 1. \tag{5.5}$$

Pravděpodobnostní funkce P^* je definována jako pravděpodobnost všech cest z instance a do instance b:

$$P^*(b|a) = \sum_{\bar{t} \in P : \bar{t}(a) = b} p(\bar{t}). \tag{5.6}$$

Snadno se ukáže, že P^* splňuje následující vlastnosti:

$$\sum_b P^*(b|a) = 1 \tag{5.7}$$

$$0 \leq P^*(b|a) \leq 1. \tag{5.8}$$

Funkce KStar (K^*) je definována jako

$$K^*(b|a) = -log_2 P^*(b|a). \tag{5.9}$$

K^* není striktně vzdálenostní funkce. Například pro $K^*(a|a)$ je obecně nenulový a zároveň tato funkce (jak je zdůrazněno | notací) je nesymetrická. Přesto následující vlastnosti jsou prokazatelné:

$$K^*(b|a) \geq 0 \tag{5.10}$$

$$K^*(c|b) + K^*(b|a) \geq K^*(c|a). \tag{5.11}$$

KStar algoritmus

Pro implementaci tohoto klasifikátoru používajícího entropickou míru vzdálenosti popsanou výše, je nutné vhodně zvolit parametry x_0, s a způsob použití hodnot vrácených mírou vzdálenosti.

Pro každou dimenzi je nutné určit parametry x_0 (pro reálné atributy) a s (pro symbolické atributy). Chování vzdálenostní míry při změně těchto parametrů je zajímavé. Uvažujme pravděpodobnostní funkci pro symbolické atributy při změnách s. Při hodnotě s blížící se 1, kdy instance obsahují dva odlišné symboly, bude pravděpodobnost transformace nízká, zatímco instance se stejnými symboly bude mít vysokou pravděpodobnost transformace. Z tohoto důvodu bude vzdálenostní funkce vykazovat chování podobné technice nejbližšího souseda (*nearest neighbour*). Pokud se s blíží hodnotě 0, pravděpodobnost transformace přímo ukazuje pravděpodobnostní distribuci symbolů. Zvýhodňuje tedy symboly, které jsou frekventovanější. Toto chování je velmi podobné výchozím pravidlům pro mnoho technik strojového učení, které jednoduše určí tu nejpravděpodobnější klasifikaci. S tím, jak se mění hodnota s, dochází k plynulé změně mezi těmito dvěma extrémy. Vzdálenostní míra pro reálné hodnoty atributů vykazuje stejné vlastnosti. Pokud x_0 je malé hodnoty, pravděpodobnost se velmi rychle snižuje se vzrůstající vzdáleností. Tato funkce je tedy podobná taktéž technikám využívajících nejbližšího souseda. Na druhou stranu, když je x_0 vysoké číslo, skoro všechny instance budou mít stejnou transformaci a velmi podobnou váhou.

V obou těchto případech můžeme uvažovat o počtu těchto instancí, které jsou zahrnuty v rámci pravděpodobnostního rozdělení pohybujícího se od extrému 1 (distribuce jako nejbližší soused) k druhému extrému N, kdy mají všechny instance stejnou váhu.

Efektivní počet instancí může být spočítán pro jakoukoliv funkce P^* použitím následujícího výrazu:

$$n_0 \leq \frac{\left(\sum_b P^*(b|a)\right)^2}{\sum_b P^*(b|a)^2} \leq N. \tag{5.12}$$

N v tomto případě značí celkový počet trénovacích instancí a n_0 počet instancí v nejmenší vzdálenosti od a. Algoritmus KStar určí hodnotu pro x_0 (nebo pro s) výběrem čísla mezi hodnotami n_0 a N, které musí zohledňovat výraz výše. Z tohoto vyplývá, že pokud bude vybrána hodnota n_0 bude uplatněn algoritmus nejbližšího souseda, pokud bude vybrána hodnota N, instance budou mít stejné váhy. Pro lepší přehlednost je specifikován nový parametr b (*blending parametr*), který může dosahovat hodnot v rozsahu $b = 0\%$ (pro n_0) až $b = 100\%$ pro N se středními hodnotami lineárně interpolovanými.

Výsledky KStar

K získání přehledu o tom, jak dobře algoritmus KStar funguje v praxi, byla provedena klasifikace na několika datasetech. Tyto datasety byly pořízeny z *UCI Machine Learning Database Repository*.

Jednotlivé datasety byly rozděleny následovně: 2/3 pro trénování modelu a zbylá 1/3 pro testování. Toto rozdělení dat bylo provedeno celkem 25krát pro každý dataset. Celkově byly ohodnoceny všechny datasety pro všech 25 různých rozdělení a jejich výsledky byly zprůměrovány. Výsledky těchto běhů jsou ukázány v tabulce 5.2, kde jsou zvýrazněny nejvyšší dosažené přesnosti predikce pro každý dataset. U metody C4.5 bylo použito prořezání stromu (*P-Tree*) a rozhodovacích pravidel (*Rules*).

Dataset	C4.5 (P-Tree)	C4.5 (Rules)	FOIL	1R	1B1	KStar (b=20)	KStar (b=best)
BC	**70,7**	68,8	54,3	67,5	66.1	68,6	**70,8**
CH	**99,2**	**99,2**	29,3	64,9	89,6	93,2	93,3
GL	66,0	64,8	50,0	52,1	67,8	**72,4**	**73,9**
G2	72,9	74,2	64,4	69,0	76,4	**82,3**	**82,7**
HD	75,7	**77,6**	64,2	73,8	75,5	75,0	**82,2**
HE	68,7	79,5	66,6	78,4	**80,8**	80,4	**83,8**
HO	76,1	**81,7**	62,5	**81,7**	77,4	76,2	79,2
HY	91,3	**99,2**	98,2	97,8	97,7	98,5	98,6
IR	94,3	94,3	89,8	92,3	**95,3**	94,9	**95,3**
LA	72,2	84,2	65,3	76,4	84,2	**90,9**	**92,0**
LY	74,8	75,8	66,2	72,7	80,9	**82,2**	**82,6**
SE	75,4	**97,8**	95,8	95,1	93,8	95,2	95,7
SO	-	-	96,3	79,2	**99,8**	**99,8**	**99,8**
VO	91,9	94,8	87,6	**95,4**	91,9	93,0	93,2
V1	83,4	89,8	77,4	87,3	87,3	**90,5**	**90,5**

Tabulka 5.2: Přesnost klasifikace pro různé datasety. [13]

Jak je vidět, algoritmus KStar funguje velmi dobře na široké škále všech modelů. Téměř ve všech případech je lepší než ostatní algoritmy (konkrétně v šesti případech z patnácti).

Kapitola 6

Implementace

Praktickou část této diplomové práce je možné rozdělit do posloupnosti několika hlavních kroků. Jelikož se jedná o přístup využívající strojového učení, bylo nutné nejdříve vytvořit trénovací dataset. Pro tyto účely lze použít volně dostupnou databázi ProTherm obsahující experimentálně získaná termodynamická data proteinů a jejich mutací (popis databáze lze nalézt v kapitole 3.1.1). Tato databáze byla kvůli předpokládanému častému dotazování nad obsaženými daty převedena do MySQL databáze. Samotnému kroku převedení samozřejmě předcházela analýza dat a návrh vhodných relačních tabulek. Vypracovány byly také postupy opravující chybné, či jinak poškozené záznamy z této databáze.

Druhým krokem byl výběr vhodných predikčních nástrojů, mezi kterými se v pozdější fázi predikce hodnoty stability hledal konsenzus (konsenzuální funkce). Po určení těchto nástrojů bylo nutné vytvořit vlastní platformu automatizovaných skriptů, které systematickým dotazováním získávaly ohodnocené mutace, výsledky pro jednotlivé nástroje byly ukládány do příslušných relačních tabulek databáze MySQL.

Po získání všech relevantních dat bylo nutné aplikovat metody strojového učení. Aby bylo dosaženo co nejlepšího výsledku, testováno bylo celkem 28 modelů podporujících regresi. V tomto případě bylo využito nástroje WEKA, jehož popis lze nalézt v kapitole 5.2.

Zároveň po získání dat z jednotlivých modelů a jejich výsledků byla snaha o zlepšení dosažené přesnosti predikce. Zkoumán byl taktéž vliv přetrénování pro jednotlivé metody strojového učení na použitém trénovacím datasetu. Více se o těchto výsledcích lze dočíst v kapitole 7.

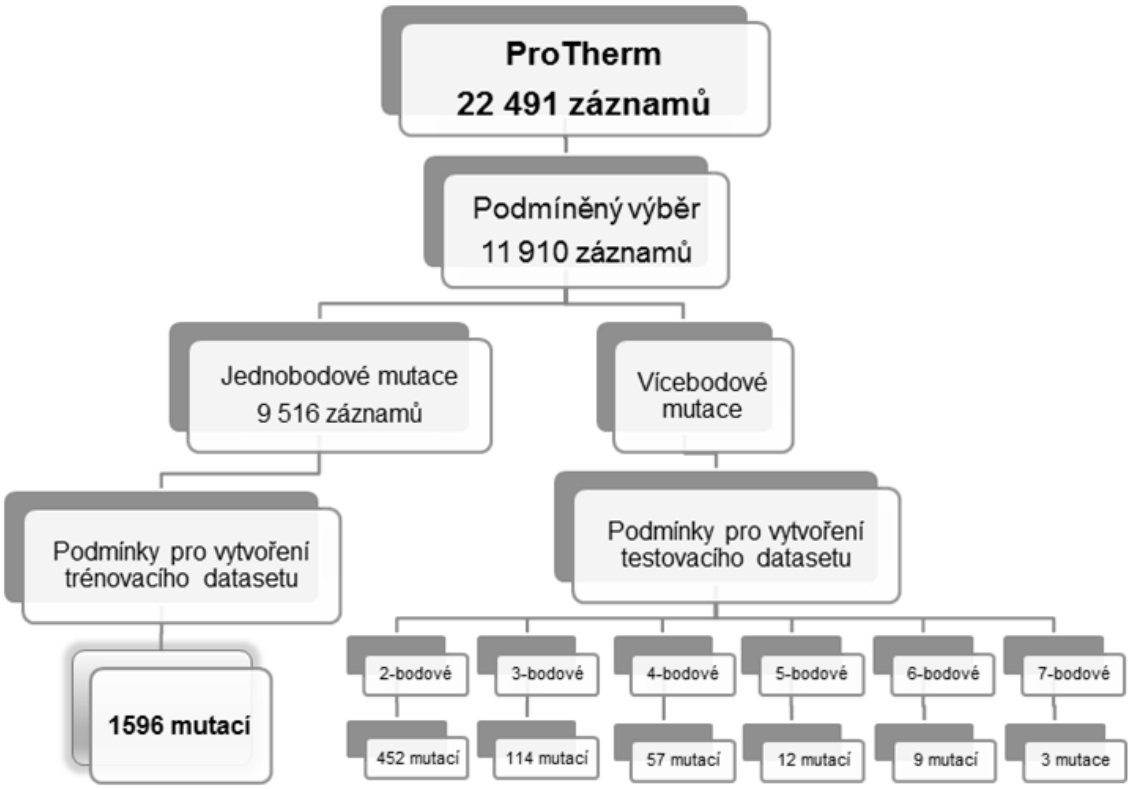
Pro veškeré vytvořené skripty byl použit skriptovací jazyk Perl, který je plně přenositelný a je ho možné použít jak na platformě Microsoft Windows, tak například na platformě Unix.

6.1 Použité datové sady

V této podkapitole budou popsány jednotlivé datové sady. Jelikož kvalita trénovacího datasetu je jeden z klíčových parametrů ovlivňující kvalitu či přesnosti predikované hodnoty, byl na výběr jednotlivých mutací kladen velký důraz. Zároveň byl vytvořen také testovací dataset, který měl i přes použitou 10-fold křížovou validaci ukázat, s jakou přesností je schopen daný model predikovat hodnoty na nezávislém datasetu (tj. dataset obsahující mutace, které nebyly použity při trénování modelu). Taktéž je z výsledků dosažených na testovacím datasetu možno posoudit, jakou roli zde hraje přeučení.

6.1.1 Trénovací dataset

Jak již bylo zmíněno, pro trénovací dataset byly zvoleny záznamy pocházející z databáze ProTherm, kde jednotlivé databázové položky byly pro jednodušší dotazování převedeny do databáze MySQL. Celkově sice databáze ProTherm obsahovala 22 491 záznamů, pro zpracování však bylo vybráno pouze 11 910 záznamů vyhovujících stanoveným kritériím (omezující byl například požadavek na existenci proteinové struktury v některé veřejně přístupné databázi). Zároveň došlo k rozpoznání jednobodových a vícebodových mutací a tyto mutace lze v databázi rozlišit skrze specifickou hodnotu odpovídajícího atributu. Při převodu dat do relační databáze byl kladen důraz na korektnost atributů vztahujících se k mutacím a jejich příslušným pozicím. Opravnými algoritmy bylo tímto získáno 986 záznamů, které by jinak skončily neúspěšnou predikcí stability (došlo například k přepočtu pozice mutace).



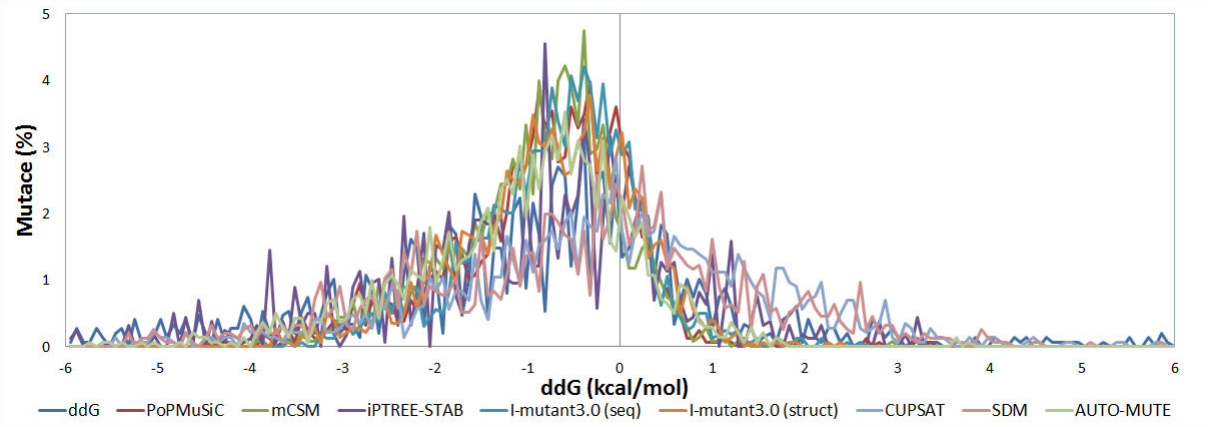
Obrázek 6.1: Posloupnost úkonů vedoucí k vytvoření trénovacího datasetu. U podmíněného výběru byla nutná podmínka existence proteinové struktury. U podmínky vedoucí k vytvoření trénovacího datasetu bylo nutné mít specifikované $\Delta\Delta G$. Pokud byly experimentální podmínky u jednotlivých záznamů stejné, došlo k zprůměrování $\Delta\Delta G$ hodnot, jinak byl vybrán záznam s pH nejblíže fyziologické hodnotě 7 a zároveň teplota byla menší nebo rovna hodnotě 50° C.

Pro vytvoření trénovacího datasetu byly brány v potaz pouze jednobodové mutace, touto selekcí tak byl daný prostor snížen na 9 662 záznamů. Na tyto záznamy byly aplikovány následující podmínky výběru. Záznamy nesměly obsahovat nevyplněnou $\Delta\Delta G$. Po-

kud existuje mutace s více než jedním záznamem a jsou-li experimentální podmínky stejné, byl vložen do datasetu pouze jeden záznam se zprůměrovanou hodnotou ΔΔG. Pokud jsou experimentální podmínky odlišné, byl vložen do datasetu pouze záznam, který měl atribut pH nejbližší fyziologické hodnotě 7 a zároveň byl atribut t značící teplotu menší nebo roven hodnotě 50° C.

Po splnění podmínek výběru dataset obsahoval 1596 záznamů, z toho u 179 případů došlo ke zprůměrování hodnoty ΔΔG a v důsledku rozdílných experimentálních podmínek bylo eliminováno 75 záznamů. Výsledný dataset byl vygenerován ve formátu ARFF, který je nativní pro platformu WEKA a byl použit k testování metod strojového učení. Na obrázku 6.1 je přehledně znázorněn vývoj návrhu trénovacího datasetu.

Obrázek 6.2 zobrazuje graf distribuce predikovaných a experimentálně naměřených ΔΔG hodnot, které jsou vyjádřeny normální distribuční křivkou. Z tohoto grafu lze vyčíst, že v použitém datasetu většina aminokyselinových mutací způsobuje destabilizaci proteinu, extrémní stavy stabilizace/destabilizace se vyskytují velmi zřídka.

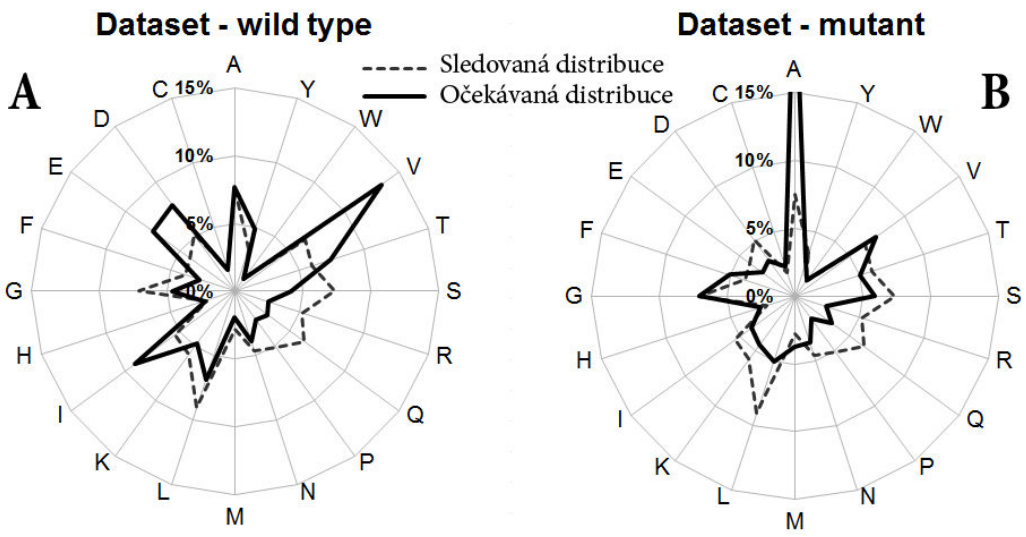


Obrázek 6.2: Distribuce predikovaných a experimentálně zjištěných ΔΔG hodnot. Mezi testované nástroje patří: AUTO-MUTE, SDM, CUPSAT, I-Mutant3.0 (strukturní verze), I-Mutant3.0 (sekvenční verze), iPTREE-STAB, mCSM a PoPMuSiC.

Na obrázku 6.3 je znázorněno sledované a očekávané zastoupení aminokyselin v trénovacím datasetu. Očekávané zastoupení aminokyselin je odvozeno z frekvence jejich výskytu v databázi OWL [7] vytvořené sloučením databází Uniprot/SwissProt, PIR, GenBank a NRL-3D (s odstraněním redundance mezi záznamy).

V levé části (obrázek 6.3A) je možné pozorovat procentuální zastoupení pro mutace původní, v pravé části (obrázek 6.3B) pak pro mutace mutantního typu. Procentuální zastoupení alaninu pro mutantní typ dosahuje hodnoty 25 %. Takto vysoká hodnota je způsobena použitím experimentální metody *alanin scanning*. Kvůli zachování měřítka a s tím spojenou možnost lepšího vizuálního porovnání s grafem původních mutací, nebyl v tomto grafu zobrazen vrchol distribuce pro aminokyselinu alanin, jelikož by vzhledem ke stejné velikosti grafů muselo dojít právě ke změně měřítka, což by znemožnilo snadné porovnání těchto grafů.

Obrázek 6.4 znázorňuje jednotlivé aminokyseliny a jejich náchylnost k destabilizaci vyjádřenou v procentech. Jednotlivé řádky a sloupce jsou popsány pomocí jednopísmenných zkratek aminokyselin (viz tabulka 2.1). Řádky v tomto případě popisují původní amino-



Obrázek 6.3: Sledované a očekávané zastoupení aminokyselin v trénovacím datasetu. Graf (A) vyjadřuje zastoupení aminokyselin pro aminokyseliny původní, graf (B) pro aminokyseliny mutantního typu.

kyselinu, sloupce pak mutantní typ aminokyseliny. Průsečíky řádků a sloupců vyjadřují poměrné zastoupení mutací, které vedou k destabilizaci proteinu. Pro trénovací dataset například platí, že obsahuje zhruba 67 % destabilizujících mutací pro mutaci vedoucí z alaninu na cystein. Již z obrázku 6.2 je zřejmé, že většina mutací je destabilizujících, tudíž bude tabulka obsahovat většinu čísel blížících se k hodnotě 1.

6.1.2 Testovací dataset

Hlavní podmínkou pro vytvoření objektivního testovacího datasetu je jeho nezávislost na trénovacích datech. V tomto případě by bylo možné tato nezávislá data získat například z jiných termodynamických databází nebo dolováním potřebných dat z vydaných patentů zaměřených na saturační mutagenezi enzymů používaných s průmyslovým použitím. Ovšem mnohem zajímavějším a ojedinělým přístupem by bylo použít zbývajících vícebodových mutací z databáze ProTherm a přistupovat k nim jako k posloupnosti na sebe navazujících jednobodových mutací. Před samotným testováním tohoto nového přístupu se naskytla otázka, zda tento postup bude korelovat ke správným výsledkům. Intuitivně lze totiž předpokládat, že vliv jednotlivých mutací na výslednou stabilitu proteinu nebude aditivní, tj. že vícebodovou mutaci je nutné popsat složitěji než jako součet efektu jednobodových mutací [41, 45]. Jelikož však jiné modely nejsou dostatečně prozkoumány, byl nakonec použit právě aditivní přístup. Na druhou stranu všem predikčním nástrojům byl předložen stejný dataset, predikční nástroje tedy měly stejné podmínky pro predikci a nebyly mezi sebou nijak zvýhodněny. Jak se ukázalo během experimentů (viz kapitola 7.2.2), korelační koeficienty jednotlivých nástrojů byly v tomto případě podobné hodnotám dosaženým na trénovacím datasetu. Tímto je tedy možné ukázat, že ačkoliv tento přístup není úplně přesný, pro účel zjištění přesnosti predikčních nástrojů na nezávislých datech je použitelný.

	A	C	D	E	F	G	H	I	K	L	M	N	P	Q	R	S	T	V	W	Y
A	-	0,67	0,75	1	0,67	0,93	1	0,4	0,83	0,75	0,67	0,67	0,63	1	0,67	0,91	0,7	0,67	1	1
C	0,75	-	-	-	-	1	-	1	-	1	1	-	-	-	-	0,75	0,5	1	-	-
D	0,61	0,5	-	0,25	0,33	0,67	0,33	0,33	0,55	0,33	0,5	0,56	1	0,5	0	0,38	0,25	0,5	0,5	0
E	0,81	0,33	1	-	0,33	0,6	0,33	0	0,73	0,33	0,33	1	0,33	0,67	0,5	0,6	0,75	0,5	0,5	0,5
F	0,88	-	-	-	-	1	1	-	1	1	1	-	-	-	-	1	1	1	0	0,43
G	0,69	1	1	0	1	-	1	-	1	1	-	0,67	1	1	0,8	0,71	1	0,57	1	-
H	0,71	0	1	0,5	-	0,75	-	-	1	0	-	1	0	0,67	1	0,5	1	-	1	0,2
I	1	0,67	1	1	1	0,89	-	-	-	0,79	0,93	0	1	-	-	1	0,9	0,89	1	1
K	0,65	-	1	0,75	0	0,86	1	0,5	-	-	0,4	0,25	0,5	0,78	0,5	-	-	0	1	0
L	0,98	1	-	1	0,67	1	0,5	1	1	1	0,89	1	0,5	1	0,5	1	1	0,76	0	0
M	1	-	-	-	0,33	-	-	0,33	1	0,67	-	-	-	-	0	-	1	0,67	-	1
N	0,79	-	0,6	0	0	0,75	1	0	0,75	0	0	-	-	1	-	0,6	1	0	-	-
P	0,92	-	-	-	-	1	-	-	-	0,5	-	1	-	-	1	0,8	-	1	1	1
Q	0,67	0	-	0,67	-	0,71	1	0	0	0,2	1	1	1	-	0,5	1	1	1	-	0,5
R	0,83	1	-	0,33	-	0,5	0,75	-	0,88	1	1	-	-	0,75	-	1	-	-	-	-
S	0,8	1	0,5	0	0	1	0,5	0	0	0,5	0	0,67	1	0,5	0	-	0,33	0,4	0	0,5
T	0,95	0,8	0,71	0,8	0,5	1	0,75	0,5	0	0,67	0	0,67	1	0,75	0,33	0,64	0	0,68	0	0
V	0,9	1	0,5	0,5	1	0,94	1	0,56	1	0,53	0,64	0,83	0,5	1	0,75	1	1	-	-	1
W	-	-	-	-	1	-	1	-	-	1	-	-	-	-	-	-	-	-	-	1
Y	1	0,33	0,5	-	0,76	1	0,5	-	0	0	-	1	1	0,5	1	1	-	0	0,5	-

Obrázek 6.4: Vyjádřeno poměrné zastoupení mutací trénovacího datasetu vedoucí k destabilizaci. Jednotlivé řádky a sloupce jsou popsány pomocí jednopísmenných zkratek aminokyselin. Řádky popisují původní aminokyselinu, sloupce pak aminokyselinu mutantního typu. Průsečíky jednotlivých řádků a sloupců vyjadřují v jakém poměru jsou obsaženy destabilizující mutace v trénovacím datasetu. Barevným odstínem je vyjádřen daný poměr, kde bílou jsou označeny stabilizující mutace, naopak nejtmavším odstínem modré jsou označeny mutace, které jsou všechny destabilizující. Pro šedou barvu platí, že mutace dané kombinace není v datasetu obsažena.

6.2 Vybrané predikční nástroje

Celkově bylo vybráno 8 predikčních nástrojů (AUTO-MUTE, SDM, CUPSAT, I-Mutant3.0 strukturní verze, I-Mutant3.0 sekvenční verze, iPTREE-STAB, mCSM a PoPMuSiC). Vlastnosti jednotlivých nástrojů jsou přehledně popsány v kapitole 4, kde je popsán i důvod výběru těchto nástrojů. Ve zmíněné kapitole taktéž nechybí srovnání vybraných i zde neuvedených nástrojů nezávislými studiemi.

Všechny použité predikční nástroje používají webové rozhraní, proto bylo možné vyvinout modulární výpočetní platformu ve skriptovacím jazyce Perl zajišťující všechny potřebné operace.

V tabulce 6.1 jsou pro jednotlivé predikční nástroje vypsány URL adresy jejich webových rozhraní. Během psaní této práce bylo autory nástroje PoPMuSiC kompletně změněno webové rozhraní, z tohoto důvodu je v tabulce 6.1 uveden odkaz na starou verzi tohoto nástroje. Zároveň s touto změnou došlo i ke změněně domény, v současnosti lze nástroj PoPMuSiC nalézt na adrese http://dezyme.com/.

Nástroje	**URL pro rozhraní**
AUTO-MUTE	http://proteins.gmu.edu/automute/
SDM	http://mordred.bioc.cam.ac.uk/sdm/sdm.php
CUPSAT	http://cupsat.tu-bs.de/
I-Mutant3.0	http://gpcr2.biocomp.unibo.it/cgi/predictors/I-Mutant3.0/I-Mutant3.0.cgi
iPTREE-STAB	http://210.60.98.19/IPTREEr/iptree.htm
mCSM	http://bleoberis.bioc.cam.ac.uk/mcsm
PoPMuSiC	http://babylone.ulb.ac.be/old_popmusic

Tabulka 6.1: Přehled nástrojů a URL pro přístup k jejich rozhraním. Při psaní této kapitoly došlo ke změně rozhraní u nástroje PoPMuSiC. Nově lze tento predikční nástroj nalézt na adrese http://dezyme.com/.

Kapitola 7

Experimenty a výsledky

Tato kapitola se věnuje podrobným výsledkům vybraných predikčních nástrojů na trénovacím i testovacím datasetu. Co se týče výsledků strojového učení, je zde rozebráno 7 nejlepších reprezentantů z jednotlivých tříd strojového učení (viz 5.1) z celkového počtu 28 algoritmů podporujících regresi. Zkoumán byl též vliv přeučení na přesnost predikovaného výsledku.

7.1 Výsledky vybraných predikčních nástrojů na trénovacím datasetu

Tabulka 7.1 obsahuje korelační koeficienty a počty mutací pro vybrané predikční nástroje. Počty predikovaných stabilizujících mutací jsou v tomto ohledu nižší než počty destabilizujících mutací. Takovýto výrazný rozdíl mezi počty stabilizujících a destabilizujících mutací je možné předpovídat již z distribuce predikovaných $\Delta\Delta G$ hodnot z obrázku 6.2. Ve skutečnosti trénovací dataset obsahoval 419 stabilizujících mutací a 1177 destabilizujících. Zajímavá je taktéž otázka, kolik mutací je každý z uvedených nástrojů schopen predikovat.

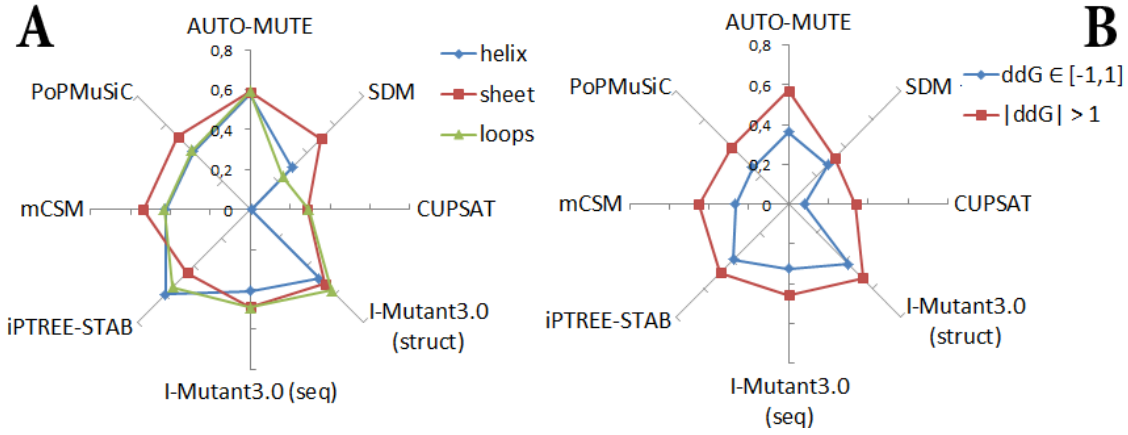
	AUTO-MUTE	SDM	CUPSAT	I-Mutant3.0 (strukturní)	I-Mutant3.0 (sekvenční)	iPTREE-STAB	mCSM	PoPMuSiC
Stab. mutace	218	627	690	273	277	378	159	235
Destab. mutace	1173	928	817	1157	1310	1216	1190	1341
Celkem	1393	1556	1510	1435	1594	1594	1349	1581
Korelační koef.	**0,583**	**0,362**	**0,177**	**0,529**	**0,464**	**0,504**	**0,488**	**0,462**

Tabulka 7.1: Korelační koeficienty pro predikční nástroje testované na trénovacím datasetu obsahujícím celkem 1596 mutací.

Vytvořený dataset obsahoval celkem 1596 mutací, nejobecnější schopnost predikce prokázal nástroj iPTREE-STAB a I-Mutant3.0 v sekvenční verzi, které byly schopni vypočítat 1594 mutací. Z tohoto pohledu byl nejhorší nástroj mCSM, který bych schopen predikovat 1349 mutací.

Vzájemné porovnání dosažených výsledků je lépe viditelné z obrázku 7.1. Na obrázku 7.1A jsou vyneseny jednotlivé korelační koeficienty pro vybrané nástroje. V tomto grafu byly korelační koeficienty vypočítány separovaně podle sekundární struktury proteinové molekuly. Modrou čarou jsou označeny struktury α-helix, červeně jsou β-sheet a zeleně jsou struktury označené jako *loops* (otočky a smyčky). Z těchto graficky prezentovaných výsledků

vyplývá, že výrazně nejhorších výsledků při predikci $\Delta\Delta G$ hodnot α-helixu dosahuje nástroj CUPSAT. Nástroj SDM vykazoval zhoršenou predikční schopnost u struktur α-helix a *loops*. U zbývajících nástrojů se nevyskytly výraznější odchylky. Na obrázku 7.1B byly korelační koeficienty vypočítány zvlášť pro intervaly $\Delta\Delta G \in [-1, 1]$ a $|\Delta\Delta G| > 1$. K povšimnutí stojí také fakt, že výsledky tohoto grafu korespondují s teoretickými úvahami popsanými v kapitole 4.9.2, kde se předpokládá, že v tomto intervalu bude docházet ke zhoršené predikci už vzhledem k faktu, že i menší chyba u mutace s experimentálně ověřeným vlivem blízkým nule může způsobit převrácení klasifikace mutace ze stabilizující na destabilizující či naopak.



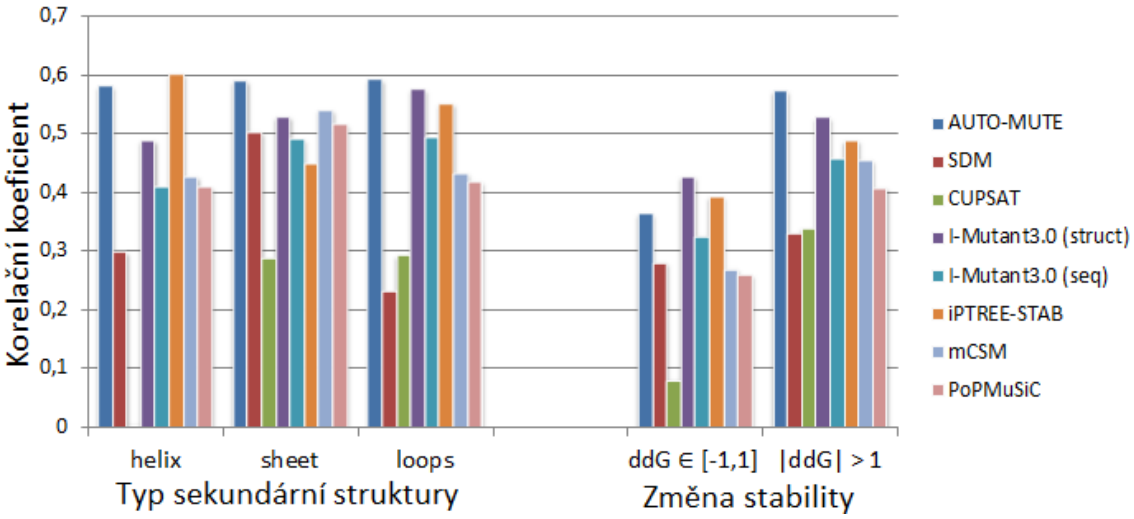
Obrázek 7.1: Graf (A) vyjadřuje dosažené korelační koeficienty vybraných nástrojů pro jednotlivé typy sekundární struktury proteinu. Modře jsou označeny korelační koeficienty pro sekundární strukturu α-helix, červeně pro β-sheet a zeleně jsou označeny *loops*. Graf (B) znázorňuje korelační koeficienty vypočítané zvlášť pro interval $\Delta\Delta G \in [-1, 1]$ (modrá barva) a $|\Delta\Delta G| > 1$ (červená barva).

Rozdíl v kvalitě přesnosti predikce $\Delta\Delta G$ hodnot v intervalu $|\Delta\Delta G| > 1$ a $[-1, 1]$ je lépe vidět na obrázku 7.2.

Zajímavá je i statistika zobrazená na obrázku 7.3. Tento obrázek vyjadřuje počet mutací pro každou z variant mutací původního typu a mutantního typu aminokyseliny. Nejvyšší počet záznamů (celkem 59 případů) dosahuje mutace z valinu na alanin. Naopak pouze 18 výskytů je pro tryptofan (W) na pozici původní aminokyseliny (v tabulce je toto znázorněno součtem hodnot řádku pro tryptofan). Taktéž pro pozici na mutantním typu dosahuje tryptofan pouze 23 výskytů, což je i v tomto případě nejméně. Fakt, že mutace z/na tryptofan se v datasetu vyskytuje nejméně, je důsledkem složitosti a rozměrnosti této aminokyseliny.

7.2 Výsledky metod strojového učení na trénovacím datasetu

Pro ohodnocení trénovacího datasetu bylo napočítáno celkem 28 metod strojového učení podporujících regresi. Pro tento úkol bylo použito platformy WEKA určenou pro analýzu znalostí (viz kapitola 5.2). V tabulce 7.2 je zobrazeno 7 modelů s nejvyššími korelačními koeficienty. Je možné zde najít zástupce různých tříd algoritmů strojového učení, uveďme



Obrázek 7.2: Výpočet korelačního koeficientu dle typu sekundární struktury (α-helix, β-sheet, *loops*) a dle hodnoty změny stability.

například lazy learning (KStar), Support Vector Machine (LibSVM Linear kernel), rozhodovací stromy (M5P) nebo klasifikaci založenou na pravidlech (M5Rules).

	Majority	Gaussian Processes	LibSVM Linear kernel	KStar	M5Rules	M5P	Bagging (REPtree)	Random SubSpace
Stab. mutace	312	368	346	416	358	360	301	211
Destab. mutace	1283	1203	1250	1179	1238	1236	1295	1385
Korelační koef.	**0,475**	**0,642**	**0,579**	**0,713**	**0,656**	**0,678**	**0,678**	**0,663**

Tabulka 7.2: Korelační koeficienty pro vybrané metody strojového učení.

Základní otázka, kterou je nutné si položit, zní, zda jsou vůbec metody strojového učení vhodné pro tento typ úlohy. Jako základní měřítko pro vyhodnocení takové úvahy může posloužit jednoduchý konsenzuální přístup založený na výpočtu aritmetického průměru vybraných nástrojů (v tabulce 7.2 označeno jako Majority). Z uvedené tabulky je zřejmé, že Majority dosahuje korelačního výsledku 0,475, kdežto nejhorší výsledek pro vybrané algoritmy strojového učení je 0,579 pro SVM model (implementace LibSVM s lineárním kernelem). Pokud výsledky Majority porovnáme se zprůměrovanými korelačními koeficienty jednotlivých predikčních nástrojů získanými z tabulky 7.1 (po zprůměrování 0,446), dojdeme k závěru, že prosté průměrování v tomto případě zlepšuje predikční schopnosti pouze zanedbatelně. Mimo jiné stejný závěr je publikován v článku [39]. Z tohoto faktu tedy vyplývá, že metody strojového učení jsou pro tento typ úloh velmi vhodné.

7.2.1 Porovnání výsledků predikčních nástrojů a přístupů strojového učení

Nejlepšího výsledku z množiny 8 existujících predikčních nástrojů dosáhl nástroj AUTO-MUTE. Jeho korelační koeficient se pohyboval na hodnotě 0,583 pro trénovací dataset. Druhý nejlepší nástroj byl I-Mutant3.0 ve strukturní verzi, který zaostal oproti AUTO-

	A	C	D	E	F	G	H	I	K	L	M	N	P	Q	R	S	T	V	W	Y	Celkem
A	0	3	4	4	3	14	3	5	6	8	6	3	8	3	3	22	10	12	2	3	122
C	8	0	0	0	0	1	0	1	0	1	1	0	0	0	0	8	2	5	0	0	27
D	28	2	0	8	3	9	9	3	11	3	2	18	4	4	4	8	4	2	2	2	126
E	21	3	5	0	3	5	3	2	26	6	3	5	3	18	2	5	4	2	2	2	120
F	17	0	0	0	0	2	1	0	1	6	1	0	0	0	0	1	1	3	4	7	44
G	36	1	4	2	1	0	1	0	1	1	0	3	1	2	5	7	1	7	1	0	74
H	7	1	2	2	0	4	0	0	1	3	0	3	1	3	1	2	1	0	1	5	37
I	40	3	1	1	7	9	0	0	0	14	15	1	1	0	0	3	10	36	2	2	145
K	17	0	2	12	2	7	3	2	0	0	5	4	2	9	6	0	0	3	1	1	76
L	42	3	0	2	3	5	2	7	1	1	9	1	2	1	4	1	6	17	1	2	110
M	9	0	0	0	3	0	0	3	3	6	0	0	0	0	1	0	2	3	0	1	31
N	19	0	15	1	1	4	3	3	4	1	2	0	0	1	0	5	1	1	0	0	61
P	25	0	0	0	0	4	0	0	0	2	0	1	0	0	1	5	0	1	1	1	41
Q	15	1	0	3	0	7	1	1	2	5	1	1	2	0	2	2	1	1	0	2	47
R	12	1	0	3	0	4	4	0	8	1	2	0	0	4	0	2	0	0	0	0	41
S	20	2	6	2	1	4	2	2	2	2	1	3	1	2	2	0	6	5	1	2	66
T	21	5	7	5	2	10	4	8	1	3	1	6	3	4	3	14	1	19	1	1	119
V	59	9	4	2	6	18	3	27	3	15	11	6	4	1	4	7	30	0	0	5	214
W	0	0	0	0	8	0	2	0	0	2	0	0	0	0	0	0	0	0	0	6	18
Y	11	3	2	0	38	6	2	0	1	1	0	2	1	2	1	2	0	1	4	0	77
Celkem	407	37	52	47	81	113	43	64	71	81	60	57	33	54	39	94	80	118	23	42	

Obrázek 7.3: Vyjádření počtu mutací pro každou z variant mutací původního typu a mutantního typu aminokyseliny. Řádky zde označují mutace původní aminokyseliny, sloupce označují mutace mutantní aminokyseliny. Barevným odstínem je vyjádřen počet mutací, kde bílou je označen nulový výskyt, naopak nejtmavším odstínem modré je označeno nejvyšší číslo vyskytující se v tabulce.

MUTE o hodnotu 0,054. Nejhorším nástrojem v této skupině predikčních nástrojů byl CUPSAT dosahující hodnoty korelačního koeficientu 0,177.

Z metod strojového učení se nejlépe umístil KStar s korelačním koeficientem 0,713. Na druhém místě se umístily metody M5P a Bagging, které obě zaostaly o shodnou hodnotu 0,035.

Dosažené zlepšení je přehledně viditelné v tabulce 7.3, kde KStar dosahuje korelačního koeficientu 0,713, kdežto AUTO-MUTE 0,583. Celkové zlepšení na trénovacím datasetu je 0,130.

Korelační koeficent	**AUTO-MUTE**	**KStar**
	0,583	0,713

Tabulka 7.3: Porovnání nejlepších výsledků pro predikční nástroje a metody strojového učení.

Na základě nejlepšího výkonu na vytvořeném trénovacím datasetu byl KStar zvolen jako nejvhodnější klasifikátor pro tento typ úlohy. Právě z tohoto důvodu byl použit i pro evaluaci testovacího datasetu.

Ovšem i přes fakt, že byla pro metody strojového učení použita 10-fold křížová validace, nelze tyto výsledky označit za relevantní, a to již z toho důvodu, že daný model byl testován na datech, která byla použita pro natrénování tohoto modelu. Z tohoto důvodu bylo nutné vytvořit nezávislý testovací dataset a nechat model ohodnotit i tyto nezávislá data.

7.2.2 Nezávislý dataset vícebodových mutací

Jak již bylo popsáno v kapitole 6.1.2, pro tvorbu nezávislého datasetu bylo použito vícebodových mutací, ke kterým se přistupovalo jako k posloupnosti mutací jednobodových. V tabulce 7.4 jsou znázorněny počty záznamů k-bodových mutací testovacího datasetu.

K-bodové mutace	Počet záznamů
2	452
3	114
4	57
5	12
6	9
7	3

Tabulka 7.4: Počty záznamů pro k-bodové mutace testovacího datasetu.

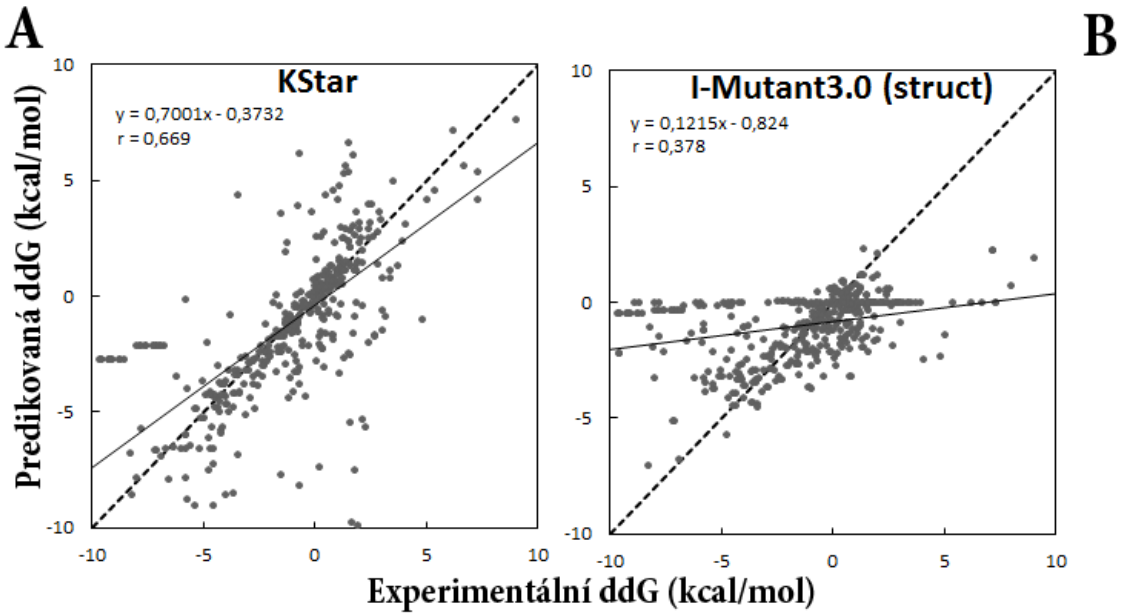
Na obrázku 7.4 je vyjádřena hodnota predikovaná vůči experimentální hodnotě $\Delta\Delta G$ 2-bodových mutací pro metodu KStar (obrázek 7.4A) a predikční nástroj I-Mutant3.0 ve strukturní verzi (obrázek 7.4B). Korelační koeficient a rovnice regresní přímky je zobrazena v levém rohu. U KStar je patrné, že regresní přímka je blíže předpisu $y = x$ (na obrázku značeno přerušovanou čarou), což odpovídá přesnější predikci. Podobný je obrázek 7.5, kde oproti obrázku 7.4 byla použita metoda binning, kde došlo k rozdělení spojitého prostoru na 12 intervalů a hodnoty z jednotlivých intervalů byly zprůměrovány.

Výsledky testů na nezávislých datech jsou ukázány v tabulce 7.5. Pro celkové srovnání vlivu k-bodových mutací bylo použito váženého průměru, a to z toho důvodu, že 5-bodové, 6-bodové a 7-bodové mutace obsahují velmi málo záznamů a docházelo by tak k velkému zkreslení, jelikož se výsledky pro tyto záznamy pohybují spíše v krajních hodnotách intervalu -1 až 1.

K-bodové mutace	AUTO-MUTE	SDM	CUPSAT	I-Mutant3.0 (strukturní)	I-Mutant3.0 (sekvenční)	iPTREE-STAB	mCSM	PoPMuSiC	KStar
2	0,301	0,394	0,103	0,378	0,306	0,364	0,068	0,412	0,669
3	0,343	0,330	0,092	0,646	0,620	0,060	0,353	0,390	0,855
4	0,519	0,569	0,619	0,779	0,816	0,272	0,514	0,637	0,715
5	0,319	-0,718	-0,254	0,249	0,419	0,630	0,022	0,149	0,287
6	0,798	0,321	0,421	0,573	0,612	0,954	0,584	0,910	0,896
7	0,992	-0,987	0,987	1	1	1	0,987	0,987	0,971
Vážený průměr	**0,338**	**0,370**	**0,148**	**0,464**	**0,416**	**0,318**	**0,168**	**0,433**	**0,703**

Tabulka 7.5: Korelační koeficienty pro jednotlivé nástroje nezávislého datasetu vícebodových mutací.

Ve výsledku tedy metoda strojového učení dosáhla korelačního koeficientu 0,703. Z existujících predikčních nástrojů nejlépe dopadl I-Mutant3.0 ve strukturní verzi s korelačním koeficientem 0,464. Celkové zlepšení na nezávislém datasetu vícebodových mutací je 0,239.



Obrázek 7.4: Vyjádření predikované hodnoty vůči experimentální hodnotě ΔΔG 2-bodových mutací pro metodu KStar (A) a nástroje I-Mutant3.0 ve strukturní verzi (B). Korelační koeficient (r) a rovnice regresní přímky (y) jsou zobrazeny v levém horním rohu.

7.2.3 Výběr rysů

Výběr rysů[1] (*feature selection*) je technika, která obecně umožňuje zlepšit úspěšnost modelů. Vychází z toho, že z daného vektoru rysů vybere pouze rysy, které kladně ovlivňují výsledek. V kontextu problému konsenzuální predikce stability to znamená, že mohou existovat takové predikční nástroje, které nějakým způsobem negativně ovlivňují schopnost správné predikce stability proteinu. Pokud bychom takové nástroje z vektoru vyřadili, mohli bychom dostat přesnější výsledky (vyšší korelační koeficient).

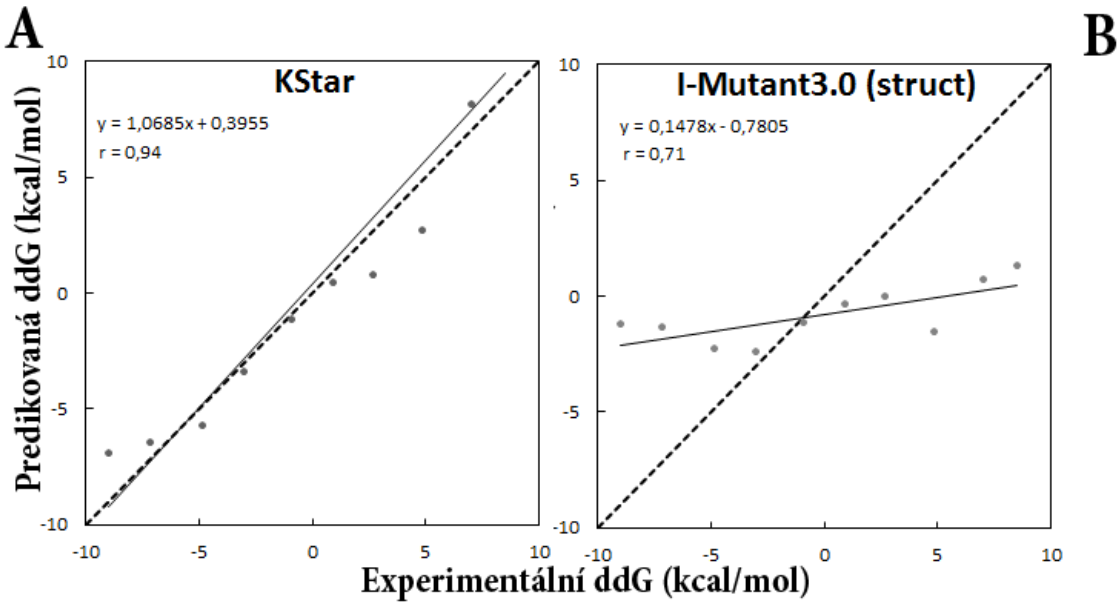
K určení relevantních rysů bylo použito vyhledávacích metod (*Search methods*) integrovaných do nástroje WEKA. Tyto vyhledávací metody obecně prohledávají prostor rysů a hledají v něm vhodnou podmnožinu. Vybrané metody jsou uvedeny níže.

- *BestFirst* používa algoritmus greedy hill climbing s principem backtracking, kde je možné určit kolik po sobě jdoucích uzlů, jenž nevedou ke zlepšení, musí být procházeno než dojde v algoritmu k navrácení. Může být použito dopředné vyhledávání (vychází se z prázdné množiny rysů), zpětné (vychází se z úplné množiny rysů) nebo vyhledávání může začít z libovolného bodu množiny rysů. [46]

- *GreedyStepwise* stejně jako BestFirst umožňuje dopředné i zpětné vyhledávání. Narozdíl od něj nepoužívá backtracking, ale ukončuje prohledávání jakmile je přidáním či odebráním nejlepšího zbývajícího rysu sníženo ohodnocení dle dané metriky. [46]

- *RandomSearch* hledá náhodným způsobem nejlepší množinu rysů. [46]

[1] Někdy je možné se setkat i s označením výběr atributů.



Obrázek 7.5: Využitím metody binning byla vyjádřena predikovaná hodnota vůči experimentální hodnotě ΔΔG 2-bodových mutací pro metodu KStar (A) a nástroje I-Mutant3.0 ve strukturní verzi (B). Korelační koeficient (r) a rovnice regresní přímky (y) jsou zobrazeny v levém horním rohu.

Původní vektor rysů je zobrazen na obrázku 7.6. Tento vektor obsahuje všech 8 predikčních nástrojů, teplotu, pH a ddG (experimentálně zjištěná hodnota ΔΔG).

temp	ph	AUTO-MUTE	SDM	CUPSAT	I-Mutant3.0 (strukturní)	I-Mutant3.0 (sekvenční)	iPTREE-STAB	mCSM	PoPMuSiC	ddG

Tabulka 7.6: Původní vektor rysů.

První zredukovaný vektor byl určen metodou *BestFirst* a je zobrazen v tabulce 7.7, obsahuje celkem 5 rysů. Určující rysy jsou ph, ddG a nástroje AUTO-MUTE, SDM a I-Mutant3.0 ve strukturní verzi. Pokud použijeme tento vektor rysů k vytvoření nového modelu pomocí strojového učení, docházíme k závěrům, že tento nově vytvořený vektor nezlepší korelační koeficient. Metoda KStar v tomto případě dosáhla výsledku 0,648, což je oproti původní hodnotě 0,713 zhoršení. Ze všech testovaných metod byla v tomto případě nejlepší metoda Bagging s hodnotou 0,668. I zde ovšem došlo ke zhoršení korelačního koeficientu, neboť na původním vektoru dosahovala metoda Bagging hodnoty 0,678.

ph	AUTO-MUTE	SDM	I-Mutant3.0 (strukturní)	ddG

Tabulka 7.7: Vytvořený vektor obsahující 5 rysů. Vybrány byly metodou *BestFirst*.

Druhá možnost redukce, zobrazená na obrázku 7.8, zohlednila více rysů. K atributům ph a ddG se zařadily nástroje AUTO-MUTE, I-Mutant3.0 ve strukturní verzi, iPTREE-STAB a PoPMuSiC. Ani v tomto případě nedošlo ke zlepšení schopnosti predikce. KStar dosáhl hodnoty korelačního koeficientu 0,657 a zároveň to byla i nejlepší metoda dosahující nejvyššího korelačního koeficientu.

ph	AUTO-MUTE	CUPSAT	I-Mutant3.0 (strukturní)	iPTREE-STAB	PoPMuSiC	ddG

Tabulka 7.8: Vytvořený vektor obsahující 7 rysů. Vybrány byly metodou *GreedyStepwise*.

Třetí vektor rysů, zobrazený na obrázku 7.9, obsahuje největší množství položek. Jsou to rysy ph, ddG a nástroje AUTO-MUTE, CUPSAT, I-Mutant3.0 ve strukturní verzi, iPTREE-STAB, mCSM a PoPMuSiC. Poslední možnost taktéž nevedla ke zlepšení. KStar dosáhl výsledku 0,655. Jako v druhém případě byla metoda KStar nejúspěšnější z celkového počtu 28 metod strojového učení, ovšem i zde bez výrazného zlepšení.

ph	AUTO-MUTE	CUPSAT	I-Mutant3.0 (strukturní)	iPTREE-STAB	mCSM	PoPMuSiC	ddG

Tabulka 7.9: Vytvořený vektor obsahující 8 rysů. Vybrány byly metodou *RandomSearch*.

Závěrem lze tedy říci, že i přes snahu zlepšit korelační koeficient pomocí techniky výběru rysů, nevedl tento experiment ke zlepšení predikční schopnosti. Tato technika je tedy pro tento konkrétní typ úlohy nevhodná.

Kapitola 8

Závěr

Cílem této práce bylo vytvořit nástroj kombinující výstupy vybraných nástrojů určených pro ohodnocení vlivu aminokyselinových mutací na stabilitu proteinu.

Prvním krokem byl výběr z existujících predikčních nástrojů. Zde byl kladen největší důraz na různorodost technik, jelikož vhodný výběr predikčních nástrojů rozšiřuje univerzálnost vytvořeného meta-nástroje. Pro vybrané nástroje byly poté vytvořeny sady automatizovaných skriptů pro řízení dávkových výpočtů predikcí stabilit.

Dalším krokem bylo vybudování trénovacího datasetu jednobodových aminokyselinových mutací na základě databáze *ProTherm*. Pro objektivní zhodnocení dosažených výsledků bylo posléze nutné vybudovat nezávislý testovací dataset, který neobsahoval data z datasetu trénovacího. Tento nezávislý dataset byl vytvořen zcela inovativním způsobem, a to z vícebodových mutací obsažených v databázi ProTherm, kde se k jednotlivým vícebodovým mutacím přistupovalo jako k posloupnosti mutací jednobodových.

Aby bylo dosaženo co největší přesnosti predikce změny stability proteinu, bylo pomocí nástroje WEKA ohodnoceno 28 různých metod strojového učení podporujících regresi. Nejlepší metoda KStar dosahovala na testovacím datasetu korelačního koeficientu 0,713, kdežto korelační koeficient nejlepšího integrovaného nástroje byl 0,583. Podobného výsledku dosáhla metoda KStar i na nezávislém datasetu vícebodových mutací, kde korelační koeficient dosáhl hodnoty 0,703. Nejlepší integrovaný nástroj, I-Mutant3.0 ve strukturní verzi, dosáhl na tomto datasetu výsledku 0,464. KStar tedy zpřesnil predikční schopnost, ve smyslu korelačního koeficientu, na trénovacím datasetu o 0,130, respektive o 0,239 na datasetu testovacím.

Další výhoda implementovaného konsenzuálního přístupu je v tom, že vytvořený meta-nástroj zvládne predikovat hodnotu vždy, když alespoň jeden z existujících nástrojů dokáže zadanou mutaci vyhodnotit.

Pro další zpřesnění byla použita technika výběru rysů (konkrétně *GreedyStepwise*, *RandomSearch* a *BestFirst*), tento postup ovšem nevedl ke zpřesnění predikovaného výsledku.

Jako návrh pro další zlepšení predikční schopnosti by bylo vhodné vytvořit nový či upravit stávající trénovací dataset tak, aby neobsahoval překryvy s trénovacími datasety integrovaných nástrojů. Takový dataset by eliminoval vliv přeučení na úrovni samotných nástrojů. Taktéž by bylo možné rozšířit množinu nástrojů o nové reprezentanty (např. Rosetta, SCide, CC/PBSA apod.) využívající jiných přístupů predikce změny stability proteinu.

Literatura

[1] Statistics of ProTherm. [Online], [cit. 2014-01-20].
URL http://www.abren.net/protherm/protherm_stat.php

[2] Alberts, B.: *Základy buněčné biologie: Úvod do molekulární biologie buňky*. Espero Publishing, druhé vydání, 1998, iSBN 80-902-9060-4.

[3] Alpaydin, E.: *Introduction to Machine Learning*. MIT Press, 2010, ISBN 978-0-262-01243-0.

[4] Baldi, P.; Brunak, S.; Chauvin, Y.; aj.: Assessing the accuracy of prediction algorithms for classification: an overview. *Bioinformatics*, ročník 16, č. 5, 2000: s. 412–424.

[5] Benedix, A.; Becker, C. M.; de Groot, B. L.; aj.: Predicting free energy changes using structural ensembles. *Nat Meth*, ročník 6, č. 1, Leden 2009: s. 3–4, ISSN 1548-7091.

[6] Berman, H. M.; Westbrook, J.; Feng, Z.; aj.: The Protein Data Bank. *Nucleic Acids Res*, ročník 28, 2000: s. 235–242.

[7] Bleasby, A. J.; Akrigg, D.; Attwood, T. K.: OWL–a non-redundant composite protein sequence database. *Nucleic Acids Research*, ročník 22, č. 17, Září 1994: s. 3574–7.

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

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

[10] Chakravarti, A.: Single nucleotide polymorphisms: ... to a future of genetic medicine. *Nature*, ročník 409, Únor 2001: s. 822–823.

[11] Chen, C.-W.; Lin, J.; Chu, Y.-W.: iStable: off-the-shelf predictor integration for predicting protein stability changes. *BMC Bioinformatics*, ročník 14 Suppl 2, 2013: str. S5, ISSN 1471-2105.

[12] Cheng, J.; Randall, A.; Baldi, P.: Prediction of protein stability changes for single-site mutations using support vector machines. *Proteins*, ročník 62, č. 4, Prosinec 2005: s. 1125–1132, ISSN 1097-0134.

[13] Cleary, J. G.; Trigg, L. E.: K*: An Instance-based Learner Using an Entropic Distance Measure. In *12th International Conference on Machine Learning*, 1995, s. 108–114.

[14] Dehouck, Y.; Kwasigroch, J. M.; Gilis, D.; aj.: PoPMuSiC 2.1 : a web server for the estimation of protein stability changes upon mutation and sequence optimality. *BMC Bioinformatics*, ročník 12, 2011: str. 151.

[15] Deutsch, C.; Krishnamoorthy, B.: Four-Body Scoring Function for Mutagenesis. *Bioinformatics*, ročník 23, č. 22, 2007: s. 3009–3015.

[16] Dosztányi, Z.; Fiser, A.; Simon, I.: Stabilization centers in proteins: identification, characterization and predictions. *J Mol Biol*, ročník 272, č. 4, Říjen 1997: s. 597–612, ISSN 0022-2836.

[17] Dosztányi, Z.; Magyar, C.; Tusnády, G. E.; aj.: SCide: Identification of Stabilization Centers in Proteins. *Bioinformatics*, ročník 19, č. 7, 2003: s. 899–900.

[18] Efron, B.; Tibshirani, R. J.: *An Introduction to the Bootstrap*. New York: Chapman & Hall, 1993.

[19] Gromiha, M.: *Protein bioinformatics: From sequence to function*. Elsevier, 2010, iSBN 978-81-312-2297-3.

[20] Guerois, R.; Nielsen, J. E. E.; Serrano, L.: Predicting changes in the stability of proteins and protein complexes: a study of more than 1000 mutations. *Journal of molecular biology*, ročník 320, č. 2, Červenec 2002: s. 369–387, ISSN 0022-2836, doi:10.1016/s0022-2836(02)00442-4.

[21] Hall, M.; Frank, E.; Holmes, G.; aj.: The WEKA data mining software: an update. *SIGKDD Explor. Newsl.*, ročník 11, č. 1, 2009: s. 10–18, ISSN 1931-0145.

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

[23] Khan, S.; Vihinen, M.: Performance of protein stability predictors. *Hum Mutat*, ročník 31, č. 6, 2010: s. 675–84, ISSN 1098-1004.

[24] Khatun, J.; Khare, S. D.; Dokholyan, N. V.: Can Contact Potentials Reliably Predict Stability of Proteins? *Journal of Molecular Biology*, ročník 336, č. 5, 2004: s. 1223 – 1238, ISSN 0022-2836.

[25] Kumar, M. D. S.; Bava, K. A.; Gromiha, M. M.; aj.: ProTherm and ProNIT: thermodynamic databases for proteins and protein-nucleic acid interactions. *Nucleic Acids Research*, ročník 34, č. Database-Issue, 2006: s. 204–206.

[26] Lesk, A. M.: *Introduction to bioinformatics*. Oxford: Oxford University Press, třetí vydání, 2008, iSBN 978-0-19-920804-3.

[27] Li, M.; Vitanyi, P.: An introduction to Kolmogorov Complexity and its Applications: Preface to the First Edition. 1997.

[28] Magyar, C.; Gromiha, M. M.; Pujadas, G.; aj.: SRide: a server for identifying stabilizing residues in proteins. *Nucleic Acids Research*, ročník 33, č. Web-Server-Issue, 2005: s. 303–305.

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

[30] Mařík, V.; Štěpánková, O.; Lažanský, J.: *Umělá inteligence*. 4, Academia, 2003, ISBN 80-200-1044-0.

[31] Masso, M.; Vaisman, I. I.: AUTO-MUTE: web-based tools for predicting stability changes in proteins due to single amino acid replacements. *Protein Eng Des Sel*, ročník 23, č. 8, 2010: s. 683–7, ISSN 1741-0134.

[32] Mehta, M.; Rissanen, J.; Agrawal, R.: MDL-based Decision Tree Pruning. AAAI Press, 1995, s. 216–221.

[33] Mingers, J.: An Empirical Comparison of Pruning Methods for Decision Tree Induction. *Machine Learning*, ročník 4, č. 2, 1989: s. 227–243, ISSN 0885-6125.

[34] Mitchell, T.: *Machine Learning*. McGraw-Hill Education, první vydání, Říjen 1997, ISBN 0-07-042807-7.

[35] Nečas, O.; kolektiv: *Obecná biologie pro lékařské fakulty*. H&H, 2000, iSBN 80-86022-46-3.

[36] Parthiban, V.; Gromiha, M. M.; Schomburg, D.: CUPSAT: prediction of protein stability upon point mutations. *Nucleic Acids Research*, ročník 34, č. Web-Server-Issue, 2006: s. 239–242.

[37] Pires, D. E. V.; Ascher, D. B.; Blundell, T. L.: mCSM: predicting the effects of mutations in proteins using graph-based signatures. *Bioinformatics*, 2013, ISSN 1367-4811.

[38] Pokala, N.; Handel, T. M.: Energy Functions for Protein Design: Adjustment with Protein-Protein Complex Affinities, Models for the Unfolded State, and Negative Design of Solubility and Specificity. *Journal of Molecular Biology*, ročník 347, č. 1, Březen 2005: s. 203–227.

[39] Potapov, V.; Cohen, M.; Schreiber, G.: Assessing computational methods for predicting protein stability upon mutation: good on average but not in the details. *Protein Engineering Design and Selection*, ročník 22, č. 9, Září 2009: s. 553–560, ISSN 1741-0134, doi:10.1093/protein/gzp030.

[40] Quinlan, J.: Induction of decision trees. *Machine Learning*, ročník 1, č. 1, 1986: s. 81–106, ISSN 0885-6125.

[41] Reetz, M. T.: The Importance of Additive and Non-Additive Mutational Effects in Protein Engineering. *Angewandte Chemie International Edition*, ročník 52, č. 10, 2013: s. 2658–2666, ISSN 1521-3773.

[42] Rohl, C. A.; Strauss, C. E. M.; Misura, K. M. S.; aj.: *Protein Structure Prediction Using Rosetta*, *Methods in Enzymology*, ročník 383. Elsevier, 2004, ISBN 9780121827885, s. 66–93.

[43] Rosypal, S.: *Úvod do molekulární biologie*. Stanislav Rosypal, třetí vydání, 1998.

[44] Topham, C. M.; Srinivasan, N.; Blundell, T. L.: Prediction of the stability of protein mutants based on structural environment-dependent amino acid substitution and propensity tables. *Protein Eng*, ročník 10, č. 1, 1997: s. 7–21, ISSN 0269-2139.

[45] Wells, J. A.: Additivity of mutational effects in proteins. *Biochemistry*, ročník 29, č. 37, 1990: s. 8509–8517.

[46] Witten, I. H.; Frank, E.: *Data Mining: Practical Machine Learning Tools and Techniques*. Amsterdam: Morgan Kaufmann, druhé vydání, 2005, ISBN 01-208-8407-0.

[47] Worth, C. L.; Preissner, R.; Blundell, T. L.: SDM - a server for predicting effects of mutations on protein stability and malfunction. *Nucleic Acids Research*, ročník 39, č. Web-Server-Issue, 2011: s. 215–222.

[48] Zhou, H.; Zhou, Y.: Distance-scaled, finite ideal-gas reference state improves structure-derived potentials of mean force for structure selection and stability prediction. *Protein science : a publication of the Protein Society*, ročník 11, č. 11, Listopad 2002: s. 2714–2726, ISSN 0961-8368.

[49] Řehout, V.; Bláhová, B.; Čítek, J.; aj.: Základy genetiky a poradentství. Únor 2003, [Online], [cit. 2014-01-12].
URL http://www.zsf.jcu.cz/cs/katedra/katedra-klinickych-a-preklinickych-oboru/import/ucebni_texty/zaklady-genetiky-a-poradenstvi

Příloha A

Databázové schéma pro databázi Stability

Tato databáze byla vytvořena pro účely diplomové práce, jejím cílem je vytvoření meta-klasifikátoru pro predikci vlivu aminokyselinových mutací na stabilitu proteinů. Databáze obsahuje data pro trénování meta-klasifikátoru, zároveň i data pro jeho testování.

Záznamy dolované z databáze ProTherm obsahují experimentálně zjištěná data k aminokyselinovým mutacím. Hlavní tabulka *protherm* je doplněna tabulkou *protherm_mutation* samotných mutací s úzce souvisejícími informacemi. Tyto záznamy slouží pro vytvoření trénovacího datasetu jednobodových mutací.

Záznamy vícebodových mutací jsou rozlišitelné pomocí hodnoty atributu *mutation_type*. Tato část dat vícebodových mutací slouží pro vytvoření testovacího datasetu.

Databáze taktéž obsahuje nutné tabulky pro výsledky integrovaných predikčních nástrojů, kterými jsou AUTO-MUTE, SDM, CUPSAT, I-Mutant3.0 ve strukturní i sekvenční verzi, iPTREE-STAB, mCSM a PoPMuSiC.

Obrázek A.1: Diagram schématu databáze a vztahů mezi tabulkami.

protherm_related_entries (obsahuje cizí klíče pro záznamy (experimenty) vztahující se ke konkrétnímu proteinu)		
uid	int(11)	unikátní identifikátor odkazu
id_similar	int(11)	identifikátor protherm záznamu

Tabulka A.1: Tabulka protherm_related_entries.

protherm_mutation (obsahuje informace o mutacích pro jednotlivé záznamy z protherm databáze)		
uid	int(11)	unikátní identifikátor mutace
id	int(11)	identifikátor protherm záznamu
name	varchar(52)	odkaz na identifikátor části patentu z tabulky patent
mutation_wild	varchar(32)	jednopísmenná zkratka původního rezidua
mutation_mut	varchar(32)	jednopísmenná zkratka nového rezidua
mutation_pos	int(11)	celočíselná pozice mutace
mutation_pos_alt	enum	sekundární struktura mutace (helix, strand, turn, coil)
asa	float	accessible surface area

Tabulka A.2: Tabulka protherm_mutation.

protherm (obsahuje experimentálně zjištěná termodynamická data k proteinům a k jejich mutacím)		
id	int(11)	unikátní identifikátor jednotlivých záznamů
protein	varchar(128)	název proteinu
source	varchar(128)	původ proteinu
lenght	int(11)	celkový počet reziduí v proteinu
mol-weight	float	molekulová hmotnost
pir_id	varchar(32)	PIR identifikátor
swissprot_id	varchar(32)	Swissprot identifikátor
e_c_number	varchar(128)	enzyme commision number
pmd_no	varchar(32)	Protein Mutant Database accession number
pdb_wild	varchar(32)	PDB identifikátor pro proteiny před mutací
pdb_mutant	varchar(32)	PDB identifikátor pro mutované proteiny
mutated_chain	varchar(128)	řetězec obsahující mutaci
no_molecule	int(11)	počet molekul (1 = monomer, 2 = dimer, ...)
sequence_swissprot	text	sekvence aminokyselin z databáze Swissprot
swissprot_id_alias	varchar(128)	Swissprot alias identifikátor
sequence_pdb	text	sekvence aminokyselin z databáze PDB
mutation_type	int(11)	celkový počet mutací
t	float	teplota použitá při experimentu
ph	float	hodnota pH
buffer_name	varchar(128)	název použitého bufferu
buffer_conc	varchar(128)	koncentrace bufferu
ion_name_1	varchar(128)	název přidaného iontu
ion_conc_1	varchar(128)	koncentrace přidaného iontu
ion_name_2	varchar(128)	název přidaného iontu
ion_conc_2	varchar(128)	koncentrace přidaného iontu
ion_name_3	varchar(128)	název přidaného iontu
ion_conc_3	varchar(128)	koncentrace přidaného iontu
protein_conc	varchar(128)	koncentrace proteinu při experimentu
measure	varchar(128)	typ měření (fluorescenční spektroskopie, diferenční skenovaní kalorimetr, ...)
method	varchar(128)	metody denaturace (Thermal, Urea, ...)
dg_h2o	varchar(128)	Gibbsova volná energie bez odečtení vlivu denaturantu (platí pro metody používající měření denaturanty)
ddg_h2o	varchar(128)	měna Gibbsovy volné energie bez odečtení vlivu denaturantu (platí pro metody používající pro měření denaturanty)
dg	float	změna Gibbsovy volné energie
ddg	float	rozdíl změn Gibbsovy volné energie
tmv	float	thermostatic mixing valve
dtm	float	Tm(mutant) – Tm(wild) [°C]
dhvh	float	van't Hoffova entalpická změna
dhcal	float	kalorimetrická změna entalpie
m	float	závislost dG na molární koncentraci denaturantu

cm	float	koncentrace denaturátu
dcp	varchar(128)	změna tepelné kapacity denaturace
state	varchar(128)	počet přechodových stavů
reversibility	varchar(128)	reversibilní denaturace (yes, no, unknown)
activity	varchar(128)	specifická aktivita pro každou mutaci
activity_km	varchar(128)	Machaelis-Mentenova konstanta [mM]
activity_kcat	varchar(128)	Machaelis-Mentenova konstanta [1/s]
activity_kd	varchar(128)	disociační konstanta
key_words	text	klíčová slova
reference	text	odkaz na články v NCBI databázi
author	varchar(128)	jména autorů
remarks	text	komentáře
related_entries	text	seznam odkazů na jiné záznamy vztahující se k aktuálnímu proteinu
db_version	datetime	datum vložení záznamu

Tabulka A.3: Tabulka databáze protherm.

protherm_automute (obsahuje informace o mutacích pro predikční nástroj AUTO-MUTE)		
uid	int(11)	unikátní identifikátor mutace
effect	enum('INCREASING', 'DECREASING', 'NEUTRAL')	celkový efekt na stabilitu proteinu
ddg	float	predikovaná hodnota $\Delta\Delta G$
pdb_id	varchar(4)	čtyřpísmenná PDB identifikace proteinu
chain	varchar(1)	jednopísmenná zkratka proteinového řetězce
t	float	teplota
ph	float	ph
vol	float	průměrné množství simplexů (pro vertex)
st	float	střední míra simplexu (tetrahedrality)
loc	enum('S', 'U', 'B')	umístění (surface, undersurface, burried)
num	float	počet hranových kontaktů s povrchovými pozicemi
ss	enum('H', 'S', 'T', 'C')	sekundární struktura (helix, strand, coil, turn)

Tabulka A.4: Tabulka protherm_automute pro predikční nástroj AUTO-MUTE.

protherm_sdm (obsahuje informace o mutacích pro predikční nástroj SDM)		
uid	int(11)	unikátní identifikátor mutace
effect	enum('INCREASING', 'DECREASING', 'NEUTRAL')	celkový efekt na stabilitu proteinu
ddg	float	predikovaná hodnota $\Delta\Delta G$
wt_secondary_structure	varchar(30)	sekundární struktura (wild-type)
wt_solvent_accessibility_percent	float	přístupnost rozpouštědla (%)
wt_solvent_accessibility_desc	varchar(15)	popis přístupnosti
wt_sidechain_hydrogen_bond	varchar(15)	postranní vodíková vazba
mutant_secondary_structure	varchar(30)	sekundární struktura (mutant-type)
mutant_solvent_accessibility_percent	float	přístupnost rozpouštědla (%)
mutant_solvent_accessibility_desc	varchar(15)	popis přístupnosti
mutant_sidechain_hydrogen_bond	varchar(15)	postranní vodíková vazba
desc	varchar(128)	popis vlivu mutace

Tabulka A.5: Tabulka protherm_sdm pro predikční nástroj SDM.

protherm_cupsat (obsahuje informace o mutacích pro predikční nástroj CUPSAT)		
uid	int(11)	unikátní identifikátor mutace
effect	enum('INCREASING', 'DECREASING', 'NEUTRAL')	celkový efekt na stabilitu proteinu
ddg	float	predikovaná hodnota $\Delta\Delta G$
wt_ss_element	varchar(30)	typ sekundární struktury
wt_solvent_accessibility	float	přístupnost rozpouštědla (%)
wt_torsion_angle_phi	float	torzní úhly ϕ
wt_torsion_angle_psi	float	torzní úhly ψ
torsion	varchar(15)	torzní úhel (favourable/unfavourable)

Tabulka A.6: Tabulka protherm_cupsat pro predikční nástroj CUPSAT.

protherm_imutant3_struct (obsahuje informace o mutacích pro predikční nástroj I-Mutant3.0 (strukturní))		
uid	int(11)	unikátní identifikátor mutace
effect	enum('INCREASING', 'DECREASING', 'NEUTRAL')	celkový efekt na stabilitu proteinu
ddg	float	predikovaná hodnota ΔΔG
ph	float	pH
t	float	teplota
rsa	float	relative solvent accessible area
ri	float	index spolehlivost

Tabulka A.7: Tabulka protherm_imutant3_struct pro predikční nástroj I-Mutant3.0 (strukturní).

protherm_imutant3_seq (obsahuje informace o mutacích pro predikční nástroj I-Mutant3.0 (sekvenční))		
uid	int(11)	unikátní identifikátor mutace
effect	enum('INCREASING', 'DECREASING', 'NEUTRAL')	celkový efekt na stabilitu proteinu
ddg	float	predikovaná hodnota ΔΔG
ph	float	pH
t	float	teplota
rsa	float	relative solvent accessible area
ri	float	index spolehlivost

Tabulka A.8: Tabulka protherm_imutant3_seq pro predikční nástroj I-Mutant3.0 (sekvenční).

protherm_iptree (obsahuje informace o mutacích pro predikční nástroj iPTREE-STAB)		
uid	int(11)	unikátní identifikátor mutace
effect	enum('INCREASING', 'DECREASING', 'NEUTRAL')	celkový efekt na stabilitu proteinu
ddg	float	predikovaná hodnota ΔΔG
ph	float	pH
t	float	teplota

Tabulka A.9: Tabulka protherm_iptree pro predikční nástroj iPTREE-STAB.

protherm_mcsm (obsahuje informace o mutacích pro predikční nástroj mCSM)		
uid	int(11)	unikátní identifikátor mutace
effect	enum('INCREASING', 'DECREASING', 'NEUTRAL')	celkový efekt na stabilitu proteinu
ddg	float	predikovaná hodnota $\Delta\Delta$G
rsa	float	relative solvent accessible area

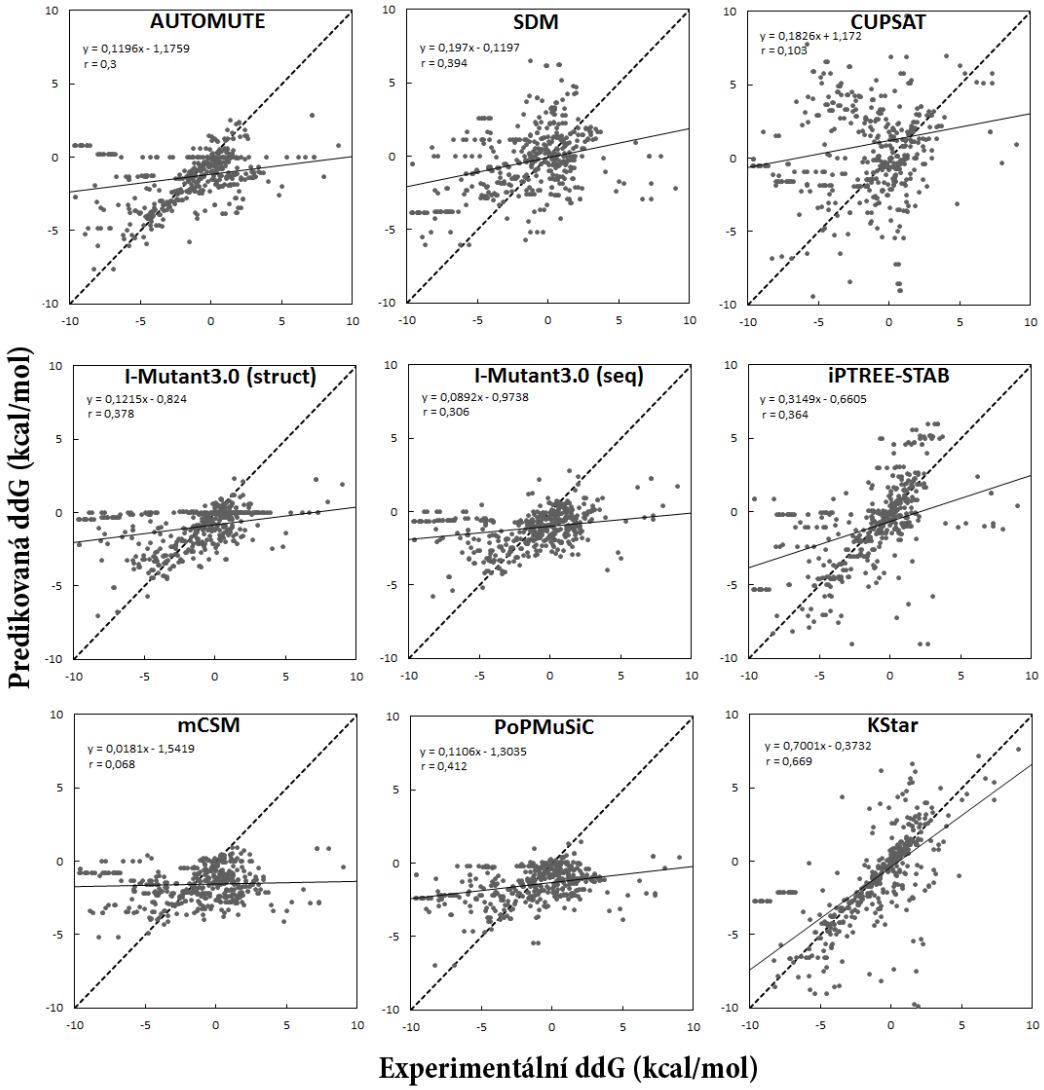
Tabulka A.10: Tabulka protherm_mcsm pro predikční nástroj mCSM.

protherm_popmusic (obsahuje informace o mutacích pro predikční nástroj PoPMuSiC)		
uid	int(11)	unikátní identifikátor mutace
effect	enum('INCREASING', 'DECREASING', 'NEUTRAL')	celkový efekt na stabilitu proteinu
ddg	float	predikovaná hodnota $\Delta\Delta$G
reliability	float	spolehlivost predikce

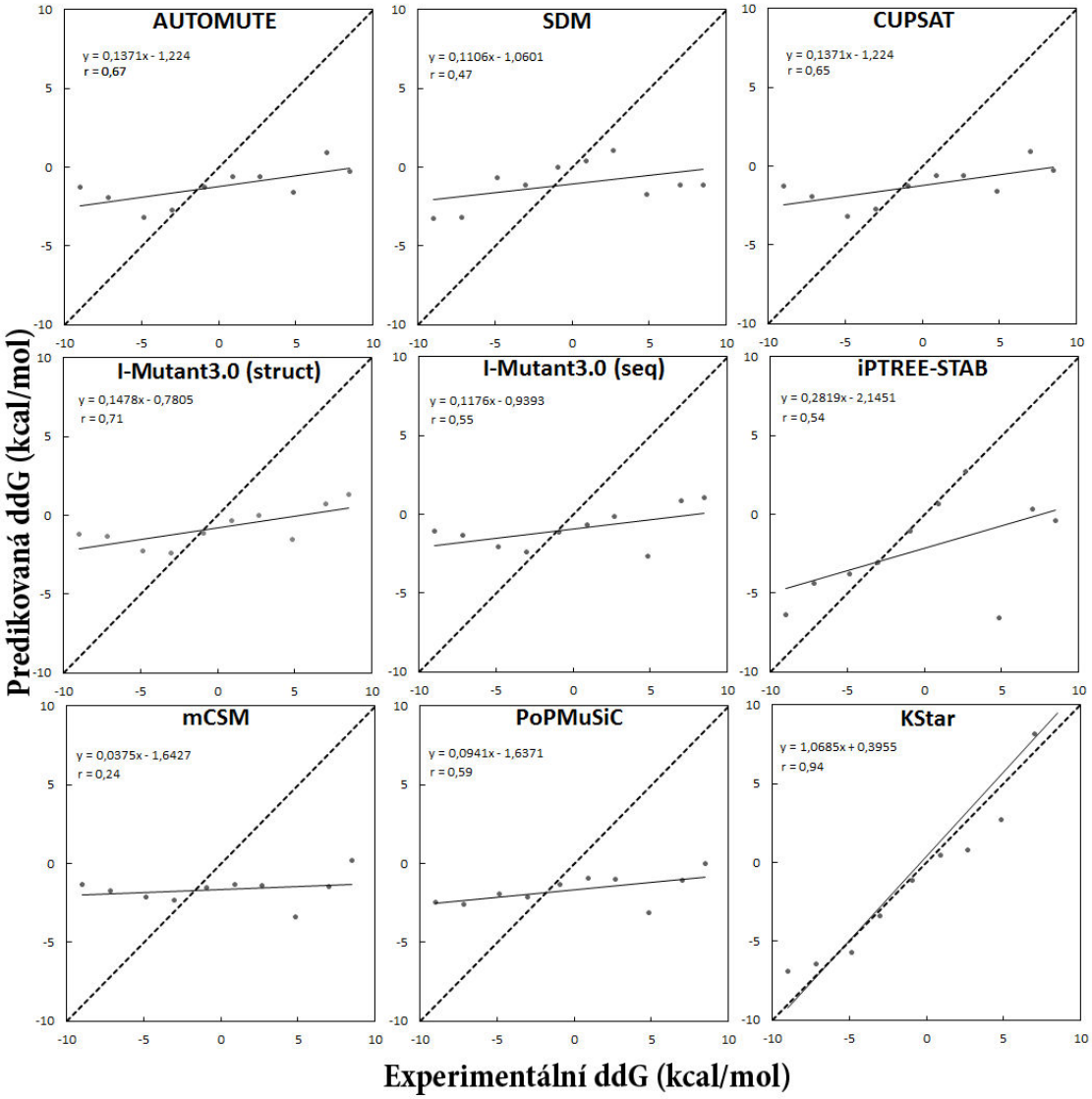
Tabulka A.11: Tabulka protherm_popmusic pro predikční nástroj PoPMuSiC.

Příloha B

Tabulky a grafy s výsledky testů



Obrázek B.1: Porovnání nástrojů pro predikci změny stability na testovacím datasetu.



Obrázek B.2: Porovnání různých nástrojů pro predikci změny stability na testovacím datasetu použitím metody binning.

	AUTO-MUTE	SDM	CUPSAT	I-Mutant3.0 (struct)
Počet stab. mutací	218	627	690	273
Počet destab. mutací	1 173	928	817	1 157
Celkem	1 393	1 556	1 510	1 435
Korelační koef.	0,58341	0,36187	0,17683	0,52861

	I-Mutant3.0 (seq)	iPTREE-STAB	mCSM	PoPMuSiC
Počet stab. mutací	277	378	159	235
Počet destab. mutací	1 310	1 216	1 190	1 341
Celkem	1 594	1 594	1 349	1 581
Korelační koef.	0,46392	0,50396	0,48773	0,46169

Obrázek B.3: Výsledky vybraných nástrojů pro trénovací dataset.

Functions					
Metoda	GaussianProcesses	LeastMedSq	LinearRegression	MultilayerPerceptron	RBFNetwork
Počet stab. mutací	368	390	373	322	373
Počet destab. mutací	1 228	1 203	1 223	1 274	1 223
Korelační koef.	**0.6423**	**0.575**	**0.6147**	**0.5161**	**0.6147**

SVM					
Metoda	SMOreg	LibSVM Linear	LibSVM Polynomial	LibSVM Radial	LibSVM Sigmoid
Počet stab. mutací	350	346	610	319	151
Počet destab. mutací	1 245	1 250	986	1 266	1 445
Korelační koef.	**0.5792**	**0.5793**	**0.0215**	**0.5767**	**-0.0949**

Meta								
Metoda	AdditiveRegression	Bagging	CVParameterSelection	MultiScheme	RandomSubSpace	RegressionBy Discretization	StackingC	Vote
Počet stab. mutací	411	301	0	0	211	183	0	0
Počet destab. mutací	1 185	1 295	1 596	1 596	1 385	1 413	1 596	1 596
Korelační koef.	**0.5095**	**0.6782**	**-0.0607**	**-0.0607**	**0.6629**	**0.6035**	**-0.0607**	**-0.0607**

Obrázek B.4: Výsledky metod strojového učení pro trénovací dataset.

Majority	Lazy learning			
Metoda	unweighted	Neighbour joining (IBk)	KStar	LWL
Počet stab. mutací	312	370	416	2
Počet destab. mutací	1 283	1 226	1 179	1 594
Korelační koef.	0,47513	0.6211	0.7131	0.4457

Rules					Trees		
Metoda	ConjunctiveRule	DecisionTable	M5Rules	ZeroR	Decision Stump	M5P	REP Tree
Počet stab. mutací	0	405	358	0	0.3999	360	360
Počet destab. mutací	1 596	1 185	1 238	1 596	1 596	1 236	1 236
Korelační koef.	0.4188	0.5815	0.6563	-0.0607	0.3999	0.6782	0.5241

Obrázek B.5: Výsledky metod strojového učení pro trénovací dataset.

K-bodové mutace	AUTO-MUTE		SDM		CUPSAT		I-Mutant3.0 (struct)	
	DDG	RMSE	DDG	RMSE	DDG	RMSE	DDG	RMSE
2	0,30093	4,31765	0,39369	4,3418	0,10268	9,12301	0,37776	4,22133
3	0,34349	5,83642	0,32994	5,20142	0,09221	13,1269	0,64614	4,98514
4	0,5193	9,01559	0,56902	6,0338	0,61886	7,02083	0,77928	7,59021
5	0,31871	5,44538	-0,71806	5,61833	-0,25352	6,96611	0,24875	5,14274
6	0,79763	3,35535	0,32146	4,45539	0,42105	3,79215	0,57322	2,27705
7	0,99203	1,47392	-0,98691	3,41909	0,98691	2,93568	0,99987	1,44809

K-bodové mutace	I-Mutant3.0 (seq)		iPTREE-STAB		mCSM		PoPMuSiC	
	DDG	RMSE	DDG	RMSE	DDG	RMSE	DDG	RMSE
2	0,30606	4,31611	0,3635	4,78196	0,0682	4,56901	0,41205	4,14855
3	0,62009	5,16904	0,06015	6,14896	0,35268	6,07223	0,3898	5,77303
4	0,81629	7,47399	0,27164	9,27612	0,51396	9,70439	0,63724	8,73313
5	0,41918	4,57053	0,62963	3,93166	0,02169	6,99005	0,1494	5,85045
6	0,61227	2,51264	0,95358	3,02379	0,58406	4,87492	0,91001	2,6054
7	0,99999	2,10753	1	2,78908	0,98691	5,22282	0,98691	3,0062

Obrázek B.6: Výsledky vybraných nástrojů pro testovací dataset vícebodových mutací.

K-bodové mutace	Kstar		GaussianProcesses		M5Rules		M5P	
	DDG	RMSE	DDG	RMSE	DDG	RMSE	DDG	RMSE
2	0,66924	3,74449	0,40012	4,3419	0,37644	4,56737	0,36585	4,60822
3	0,85489	3,64879	0,45136	5,52847	0,43804	5,60778	0,452	5,47146
4	0,71459	6,25778	0,66016	7,76031	0,65383	7,77934	0,64997	7,86047
5	0,28748	5,44136	0,628	4,5748	0,33411	6,92716	0,27581	6,80472
6	0,89592	1,34376	0,93927	3,72822	0,85422	3,9451	0,87167	4,01728
7	0,97069	0,51007	0,99974	3,14907	0,99881	3,27461	0,99914	3,38127

K-bodové mutace	Bagging		RandomSubSpace		SVM linear	
	DDG	RMSE	DDG	RMSE	DDG	RMSE
2	0,48123	3,98324	0,40905	4,12084	0,38093	4,42973
3	0,58403	5,10447	0,6341	5,30558	0,2531	5,8052
4	0,69665	7,74223	0,74555	7,95237	0,48182	8,66449
5	0,4108	5,77866	0,33314	6,18253	0,54452	4,54619
6	0,88759	3,35032	0,85612	3,7771	0,95413	2,78342
7	0,99987	2,71005	0,99855	2,3071	0,99966	2,04097

Obrázek B.7: Výsledky metod strojového učení pro testovací dataset vícebodových mutací.

Příloha C

Obsah CD

/doc	diplomová práce ve formátu pdf a LaTeX
/feature_selection	výsledné korelační koeficienty pro metodu výběru rysů
/grafy	tabulky a grafy použité v této práci
/run	příklady spouštěcích skriptů použitých při výpočtech na MetaCentru
/skripty/machine_learning	skript pro vytvoření/ohodnocení všech použitých modelů strojového učení
/skripty/protherm	skripty pro převod databáze ProTherm do relačních tabulek
/skripty/stability	skript pro získání výsledků ohodnocených mutací vybraných nástrojů
/test	skripty potřebné pro vytvoření uvedených tabulek a grafů
/weka/testing_dataset	výsledky pro testovací dataset vícebodových mutací
/weka/training_dataset	výsledky a vytvořené modely pro trénovací dataset
/zdroj	schéma SQL databáze včetně všech dat