VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV INFORMAČNÍCH SYSTÉMŮ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF INFORMATION SYSTEMS

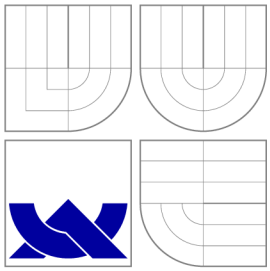
EVOLUČNÍ STRATEGIE V ÚLOZE ANOTACE FUNKCE NUKLEOTIDOVÉHO POLYMORFISMU

BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

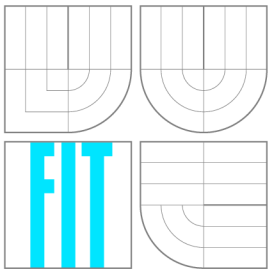
AUTOR PRÁCE
AUTHOR

ONDŘEJ ŠALANDA

BRNO 2013



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV INFORMAČNÍCH SYSTÉMŮ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF INFORMATION SYSTEMS

EVOLUČNÍ STRATEGIE V ÚLOZE ANOTACE FUNKCE NUKLEOTIDOVÉHO POLYMORFISMU

FUNCTIONAL ANNOTATION OF NUCLEOTIDE POLYMORPHISM USING EVOLUTION STRATEGY

BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

AUTOR PRÁCE — ONDŘEJ ŠALANDA
AUTHOR

VEDOUCÍ PRÁCE — Ing. JAROSLAV BENDL
SUPERVISOR

BRNO 2013

Abstrakt

Tato práce prezentuje nový přístup k predikci vlivu nukleotidového polymorfismu na funkci proteinu. Cílem je vytvořit nový metanástroj, který pomocí váhového konsensu kombinuje vlastnosti osmi již existujících nástrojů za účelem zvýšení přesnosti a univerzálnosti predikce. K nalezení vhodného rozložení vah je přistoupeno inovativně, používá se evoluční strategie. Parametry pro její spuštění jsou zjištěny experimentálně. Na závěr je uvedeno zhodnocení úspěšnosti nového nástroje a porovnání výsledků na testovacích sadách.

Abstract

This thesis brings a new approach to the prediction of the the effect of amino acid substitution. The main goal is to create a new meta-tool, which combines evaluations of eight already implemented prediction tools. The use of weighted consensus over those tools should lead to better accuracy and versatility of prediction. The novelty of developed tool lies in involving evolution strategy with experimentally defined parameters as a way to determine the best weight distribution. At the end, a complex comparison and evaluation of results is given.

Klíčová slova

Protein, mutace, jednoduchý nukleotidový polymorfismus, evoluční strategie, váhový konsensus.

Keywords

Protein, mutation, simple nucleotide polymorphism, evolution strategy, weighted consensus.

Citace

Ondřej Šalanda: Evoluční strategie v úloze anotace funkce nukleotidového polymorfismu, bakalářská práce, Brno, FIT VUT v Brně, 2013

Evoluční strategie v úloze anotace funkce nukleotidového polymorfismu

Prohlášení

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně pod vedením ing. Jaroslava Bendla a uvedl jsem všechny literární prameny, ze kterých jsem čerpal.

.....
Ondřej Šalanda
10. května 2013

Poděkování

Vřele děkuji svému vedoucímu panu ing. Jaroslavu Bendlovi za vedení a konzultace při vypracovávání této práce. Díky jeho ochotě a odborné pomoci jsem byl schopen vyřešit mnohé problémy. Děkuji rovněž MetaCentru za poskytnutí přístupu k distribuované výpočetní infrastruktuře nezbytné pro provedení dostatečně velkého množství experimentů (projekt LM2010005).

Obsah

Kapitola 1

Úvod

Proteiny a jejich funkce jsou jedním z nejpodstatnějších odvětví bioinformatického studia. Protože mutace mají signifikantní vliv na proteinovou funkci, byly vyvinuty nástroje, které umí vliv těchto mutací s různou přesností predikovat. I když existuje větší množství těchto nástrojů, stále je prostor k jejich zdokonalování.

Předmětem práce je návrh a implementace nového metanástroje (dále označnovaný jako META), který bude kombinovat výsledky osmi již existujících nástrojů tak, aby bylo dosaženo zvýšení přesnosti predikce. Metanástroj je reprezentován váhovým konsensem mezi nástroji, přičemž hlavním cílem bude najít co nejoptimálnější rozdělení vah. Pomocí strojového učení a evolučních programovacích technik, konkrétně evoluční strategie, bude takové řešení nalezeno a posléze otestováno na speciálně vytvořených testovacích datových sadách. Evoluční strategie a váhový konsensus jsou v tomto výzkumném směru ojedinělým prvkem, i proto nakonec bude provedeno komplexní srovnání a zhodnocení výsledků.

Druhá kapitola práce se zabývá proteiny, jejich složením, funkcí a strukturami, kterými jsou popsány. Nechybí ani rozbor aminokyselin, které jsou řetězcovými články proteinů, stejně jako popis vzniku proteinů transkripcí a translací gentické informace.

Ve třetí kapitole jsem se zaměřil na mutace v genetickém kódu, které způsobují změnu v primární struktuře proteinu. Provedl jsem jejich klasifikaci a soustředil se na jednobodové substituční mutace a jejich potenciální funkční vliv.

Čtvrtá kapitola je věnována možnostem a metodám pro predikci vlivu mutací, přehledu těchto metod a metrikám, kterými je lze porovnat.

V páté kapitole podrobněji rozebírám dílčí použité nástroje, metody a algoritmy, které používají, a jejich první srovnání na základě literatury.

Šestá kapitola je věnována evolučním výpočetním technikám se zaměřením na evoluční strategii a možnostem její implementace. Krátce se zmiňuji o různých parametrech, které lze v této souvislosti měnit za účelem zefektivnění práce.

V sedmé kapitole popisuji konkrétní implementaci výpočetního frameworku a budování datových sad.

V osmé kapitole pak provádím analýzu výsledků. K tomu využívám různých statistických metrik a funkcí, jejichž význam a objektivita je v této části rovněž rozebírána.

Na závěr je uvedeno krátké shrnutí práce s důrazem na získané výsledky, vyzdvižení dosažených cílů a uvedení možností vylepšení pro případnou budoucí práci.

Kapitola 2

Proteiny

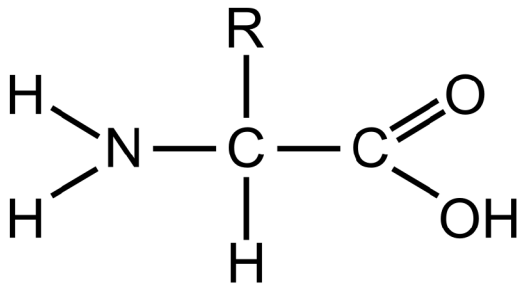
Proteiny jsou základními látkami, ze kterých se skládají všechny tkáně. Díky rozmanitosti tvarů, kterých mohou nabývat, plní poměrně velké množství funkcí, přičemž tou nejvýznamnější je funkce stavební. Zastavájí ale také další buněčné funkce, na základě kterých se v [2] dělí na:

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

Právě pro toto množství druhů a funkcí jsou proteiny předmětem studia a bádání. Schopnost analyzovat funkci proteinu, stejně jako znalost toho, jak protein některé funkce zbavit nebo naopak její účinek znásobit, jsou cíle, které jsou důležité nejen v oblasti proteinového inženýrství, ale i ve výzkumu chorob, jejich příčin a možností léčby.

2.1 Aminokyseliny

Pro stanovení funkce proteinů je nutné pochopit jejich strukturu na atomární úrovni, protože funkce reflektuje právě tvar a vnitřní uspořádání proteinu. Každý protein je řetězcem (polymerem) menších jednotek, které se nazývají aminokyseliny. Jedná se o molekuly, které se skládají vždy z karboxylové skupiny ($-COOH$), aminoskupiny (H_2N-) a vedlejšího řetězce, jehož chemické složení rozlišuje druh aminokyseliny [2]. Vedlejší řetězec (R) určuje některé důležité vlastnosti aminokyseliny, může mít vlastnosti kyselé nebo zásadité, může být polární či nepolární. Na obrázku 2.1 je vidět, že všechny jmenované složky se váží na α-uhlík a vytváří tak aminokyselinu. Ta se k další aminokyselině váže přes kovalentní peptidickou vazbu, čímž vzniká výsledný bílkovinový řetězec [2].



Obrázek 2.1: Chemická struktura aminokyseliny [2].

2.2 Struktura a funkce proteinu

Podle složitosti rozlišujeme čtyři druhy struktury proteinu, postupně primární až kvartérní. Primární strukturou rozumíme posloupnost aminokyselin v pořadí tak, jak jsou na sebe navázány bez ohledu na prostorový tvar, který takový řetězec zaujímá. Přímo z této sekvence lze poté predikovat vyšší strukturu, tedy například sekundární, kde se vyskytují především dva útvary, a to alfa šroubovice (α-helix) a beta skládaný list (β-sheet) [2].

Jak již bylo zmíněno, funkce proteinu vychází právě z jeho prostorového uspořádání. Je to proto, že svým specifickým tvarem protein umožňuje navázat se na další specifické molekuly, tzv. ligandy, které mohou zajišťovat požadovanou funkčnost [2]. Pokud by došlo ke změně primární struktury a tedy i potenciální změně vyšších struktur, které z ní vychází, mohlo by dojít k narušení vazebného místa pro určitou partnerskou molekulu, a potom by protein přestal plnit některou funkci. Důležitý je ale rovněž fakt, že změna v primární struktuře se nemusí nutně projevit narušením vazebné pozice v její blízkosti, ale díky vnitřnímu uspořádání může ke změnám dojít i hlouběji v proteinu [29].

2.3 Vznik proteinu - proteosyntéza

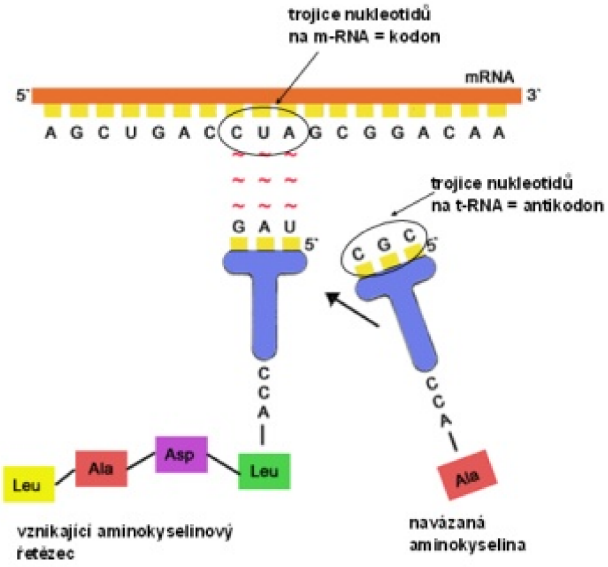
V této práci se zabývám vlivem aminokyselinových mutací na funkci proteinu, tato kapitola se věnuje vzniku proteinu a tedy procesu nabytí znalosti jeho primární struktury, ve které pak může docházet k mutacím, vedoucím k potenciálním změnám funkce.

Protein je vytvářen na základě kódující části vlákna DNA, které nese genetickou informaci o tom, v jakém pořadí se budou navazovat aminokyseliny do výsledného řetězce. Kódování je realizováno pomocí sekvence nukleotidů, označených prvními písmeny svého názvu (adenin, tymin, cytozin, guanin), přičemž každé tři po sobě jdoucí nukleotidy (tzv. triplet nukleotidů neboli kodón) kódují právě jednu aminokyselinu. Pro popis dekódování lze zavést pojem třímístného čtecího okna, které se posunuje po řetězci nukleotidů po třech místech a jeho obsah tak vždy kóduje jednu aminokyselinu. Vzhledem k tomu, že existuje 20 základních druhů aminokyselin, a celkový počet tripletů je $4^3 = 64$, jsou některé aminokyseliny kódovány více kodóny, některé kombinace tripletů nukleotidů jsou pak vyhrazeny pro konec sekvence. Přehledná tabulka je na obrázku 2.2 [20].

	U		C		A		G	
U	UUU	fenylalanin	UCU	serin	UAU	tyrosin	UGU	cystein
	UUC	fenylalanin	UCC	serin	UAC	tyrosin	UGC	cystein
	UUA	leucin	UCA	serin	UAA	**stop**	UGA	**stop**
	UUG	leucin	UCG	serin	UAG	**stop**	UGG	tryptofan
C	CUU	leucin	CCU	prolin	CAU	histidin	CGU	arginin
	CUC	leucin	CCC	prolin	CAC	histidin	CGC	arginin
	CUA	leucin	CCA	prolin	CAA	glutamin	CGA	arginin
	CUG	leucin	CCG	prolin	CAG	glutamin	CGG	arginin
A	AUU	izoleucin	ACU	treonin	AAU	asparagin	AGU	serin
	AUC	izoleucin	ACC	treonin	AAC	asparagin	AGC	serin
	AUA	izoleucin	ACA	treonin	AAA	lysin	AGA	arginin
	AUG	**metionin**	ACG	treonin	AAG	lysin	AGG	arginin
G	GUU	valin	GCU	alanin	GAU	kys.	GGU	glycin
	GUC	valin	GCC	alanin	GAC	asparagová	GGC	glycin
	GUA	valin	GCA	alanin	GAA	kys.	GGA	glycin
	GUG	valin	GCG	alanin	GAG	glutamová	GGG	glycin

Obrázek 2.2: Kódování aminokyselin pomocí kodónů mRNA [20].

Proces vzniku proteinu se nazývá proteosyntéza a probíhá uvnitř buněk na ribozómech. Začíná transkripcí informace v DNA, při které vzniká k ní komplementární vlákno RNA, kokrétně jeden její typ, a to mRNA (mediátorová). Toto vlákno tedy obsahuje onu sekvenci nukleotidů, které se po třech označují jako kodóny. Ve druhé fázi, která se nazývá translace, jsou pomocí struktury tRNA (transferová RNA) dopraveny k tomuto vláknu aminokyseliny, kdy je každá tato aminokyselina spojena s antikodónem tRNA. Antikodón je, stejně jako kodón, trojice nukleotidů. Translací, neboli překladem, rozumíme navázání kodónů na antikodóny pricipem komplementarity, kdy se nukleotid vždy váže na svůj protějšek. Párování probíhá následovně: adenin je komplementární s uracilem (RNA obsahuje uracil místo tyminu), cytozin je komplementární s guaninem. Jakmile dojde k propojení struktur mRNA a tRNA, aminokyseliny se odpoutají od antikodónů a vytváří protein. Celý proces proteosyntézy a ukázka dekódování je na obrázku 2.3 [20].



Obrázek 2.3: Průběh proteosyntézy, konkrétně translace [20].

Kapitola 3

Mutace a jejich vliv na funkci proteinu

Mutací rozumíme změnu v genetické informaci uložené v DNA jedince [21]. Některé takové změny jsou přirozené a mají velký význam v evolučních teoriích, například v Darwinově teorii o vzniku a vývoji druhů, protože se jedná o přirozenou reakci na změnu prostředí. Jiné mutace mohou způsobit lehká i závažná onemocnění, označovaná jako genetické choroby. Tyto aspekty tedy jen podtrhují význam a důležitost výzkumu v této oblasti.

3.1 Dělení mutací

Následující část se tedy zabývá mutacemi, jakožto možnou příčinou změny funkce proteinů. Mutací je obecně více typů podle toho, jak ovlivňují primární strukturu proteinu. V praktické části práce nejsou uvažovány všechny typy, vysvětlení, proč jsou některé mutace předmětem zkoumání a jiné nikoliv, je uvedeno přímo v rámci rozdělení níže.

V předchozích kapitolách již bylo zmíněno, že funkce proteinu úzce souvisí s jeho primární strukturou a že tato struktura vzniká rozkódováním informace v DNA. A právě genetická informace DNA je místem, kde dochází k mutaci, která se projeví obecně jako zásah do sekvence nukleotidů, který může mít za následek změnu posloupnosti aminokyselin. Důležitý je fakt, že ne každá změna v DNA se projeví na primární struktuře. Je to způsobeno tím, že většina úseků DNA je tzv. nekódující, tedy že není předlohou pro vznik žádného proteinu. Mutace v takové oblasti se pochopitelně navenek nijak neprojeví, proto se zabýváme výhradně mutacemi v kódujících částech. Tyto úseky DNA jsou ale navzdory své důležitosti ve značné menšině, uvádí se, že pouhých 1,5 % délky lidské DNA je pokryto kódujícími úseky [20], [21].

Dalším z hledisek pro dělení mutací je způsob a rozsah změn, které mutace provede v kódující části DNA. Podle [20] pak rozlišujeme:

- inzerci,
- deleci,
- substituci.

Z těchto tří typů ve své práci neuvažuji inzerce a delece. Inzercí rozumíme vložení jednoho nebo více nukleotidů na určité místo v DNA kódu. Nyní však potenciálně dochází k problému, protože aminokyseliny jsou kódovány triplety nukleotidů, a tedy vložením takového

počtu nukleotidů, který není dělitelný třemi, dochází k posunu třímístného čtecího okna, kterým se DNA dekóduje. Nedochází tak k pouhému vložení nových aminokyselin, ale ke změně celého řetězce, případně i k jeho zkrácení, protože může být dekódován terminační kodón. Stejný problém může nastat u delecí. Efekty mutací, způsobených inzercemi nebo delecemi nukleotidů jsou, zvláště pokud se jedná o vložení či odstranění většího počtu nukleotidů, poměrně snadno odhalitelné a vysvětlitelné [29]. Proto se v této práci zaměřuji na jednobodové substituční mutace, které pro klasifikaci vyžadují detailnější a specifičtější přístup.

Posledním typem mutací jsou tedy substituce, při nichž dochází k záměně jednoho nebo více nukleotidů, nedochází tedy ke změně délky řetězce a mutace ovlivňuje pouze tu aminokyselinu, v jejímž kodónu provedla změnu.

3.2 Nukleotidový polymorfismus

Polymorfismus označuje změnu (mutaci) nějakého znaku jedinců, která má ale určitou patrnou četnost v populaci, v literatuře [29] se často uvádí údaj o relativní četnosti s hodnotou alespoň 1 %. Ve své práci se právě zaměřuji na polymorfismy a to konkrétně jednobodové, označované zkratkou SNP.

3.2.1 Nesynonymní SNP

Omezení na jednobodové mutace však stále ještě není konečné. Podle obrázku 2.2 je zřejmé, že některé aminokyseliny mohou být zakódovány více kodóny, proto je nutné podotknout, že ne každá SNP mutace musí způsobit záměnu aminokyseliny na dané pozici a tedy nemusí se promítnout jako změna primární struktury korespondujícího proteinu. Podle [21], [20] mají SNP ještě několik variant, a to:

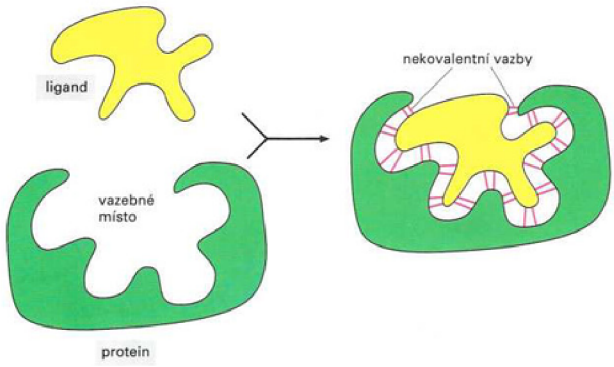
- synonymní (silent) mutace, při kterých změna nukleotidu nezpůsobuje záměnu aminokyseliny na dané pozici,
- nesmyslné (nonsense) mutace, kdy nový triplet nukleotidů místo aminokyseliny kóduje ukončovací kodón,
- nesynonymní (missense) mutace, které změní triplet nukleotidů tak, že kóduje jinou aminokyselinu.

V lidské DNA se vyskytuje přibližně 10 milionů SNP, z nichž nesynonymních je 67 000 až 200 000. Je tedy zřejmé, že nesynonymní SNP varianty (označované nsSNP) jsou poměrně vzácné, přesto však se velké množství genetických onemocnění připisuje právě jim, přičemž u některých chorob je dokonce známa přesná pozice a znění mutace, která tuto chorobu způsobuje. Jednou z takových chorob je například srpkovitá anémie, onemocnění červených krvinek. [8] Cílem této práce je zpřesnit predikci vlivu aminokyselinových substitucí, je tedy zřejmé, že se budu zabývat právě nsSNP, které jsou přímou příčinou těchto substitucí [21].

3.3 Vliv aminokyselinových substitucí na protein

Znalost kompletního lidského genomu přispěla k růstu důležitosti studia mutací a jejich možného dopadu na lidské zdraví. Velmi rychle se rozvinul výzkum, který objasňoval spojitost mezi genotypem (souborem všech genů) a genetickými chorobami [29], [11]. V tomto ohledu je zkoumání vlivu nsSNP esenciální pro další pochopení těchto spojitostí.

Aminokyseliny díky svému specifickému složení přímo ovlivňují tvar proteinu. V důsledku nsSNP dojde ke změně aminokyseliny na určité pozici proteinového řetězce a tedy i ke změně výsledného tvaru proteinu. Z obrázku 3.1 lze vyvodit, že změna může vést k tomu, že protein již v daném místě není schopen vázat partnerské molekuly (tzv. ligandy), které před efektem mutace díky svému specifickému tvaru vázány být mohly. Znamená to tedy, že protein ztratil některou ze svých funkcí, případně došlo k jejímu omezení. Pokud v důsledku mutace k takové změně funkce dojde, je tato mutace označena jako škodlivá (angl. deleterious). Pokud naopak protein nezmění svůj tvar natolik, že by pozbyl možnosti vázat původní ligandy, označuje se mutace jako neutrální (angl. benign). Nicméně, označení škodlivé mutace je třeba chápat v širším kontextu, protože protein například v důsledku ztráty některé své funkce může přestat působit škodlivě na organismus, označení škodlivosti mutace se tedy spíše vztahuje k proteinu samotnému, pro lepší pochopení by bylo správnější interpretovat škodlivost jako změnu funkce.



Obrázek 3.1: Vazba ligandu na vazebné místo proteinu [2].

3.3.1 Výskyt mutací

Některé studie ukazují, že přirozená mutace má velkou pravděpodobnost výskytu v takové části DNA, která není evolucí pozměňována, zatímco úseky, které během evoluce dosáhly většího počtu změn, se ukazují jako mutačně nečinné [21].

3.3.2 Projevy mutací

Aminokyselinová substituce může být přímou příčinou projevu fenotypu chorob. K tomu dochází, pokud je mutací zasaženo místo velmi důležité pro funkci proteinu, například úsek v kalatytické části enzymu nebo, jak už bylo zmíněno, úsek zajišťující kooperaci s partnerskými molekulami. Na druhou stranu může ale mutace způsobit rovněž zesílení funkce proteinu, případně snadnější návaznost na partnerské molekuly [29]. Většina modifikací je ale považována za škodlivé, a proto jsou takové změny eliminovány v rámci evolučního vývoje, zatímco prospěšné mutace se mohou do populace prosadit a ustálit, čímž přispívají k diferenciaci organismů [21].

Mimo přímých dopadů na funkci proteinu mohou substituce vést ke změně struktur proteinu, mohou například způsobit nadměrný výskyt β-listů na úkor α-šroubovic v sekundární struktuře, způsobit celkovou nestabilitu a následný rozpad proteinu nebo naopak jeho smrštění. I takto malá změna v primární struktuře, jakou je záměna jedné aminokyseliny, může mít tedy zcela fatální následky, v kontrastu s tímto faktem lze ovšem najít řadu

případů, kdy po provedení inzerce nebo delece poměrně značné části řetězece vykazoval protein jen nepatrné změny funkce [29].

Dalším případem projevu nsSNP mohou být pretranslační změny, k nimž se řadí změna stability vlákna mRNA nebo zpomalení translační fáze proteosyntézy. Tyto jevy se však vyskytují velmi zřídka [29].

Kapitola 4

Predikce vlivu mutace na funkci proteinu

V této kapitole budou diskutovány aspekty, které nejvíce napomáhají přesné predikci vlivu aminokyselinových substitucí na funkci proteinu. Z bioinfomatického hlediska je vhodné využít sofistikovaných metod, které šetří čas i prostředky, jenž by musely být vynaloženy při laboratorním zkoumání.

4.1 Bioinfomatika

Bioinformatika je věda, zabývající se shromažďováním, uchováváním, organizací a analýzou biologických dat. K tomuto účelu vyvíjí řadu metod, které napomáhájí manipulovat s daty tak, aby bylo možné snadné zacházení a hledání informací. Hlavní náplní práce v oblasti informatiky je vytváření nástrojů, které mohou poskytovat biologické znalosti.

Uplatnění nachází zejména v experimentální molekulární biologii, kde se hojně využívá k vizualizaci a analýze signálů a obrazových dat. V genetické biologii, kam spadá i obsah této práce, je hlavní oblastí přínosu problematika zarovnávání sekvencí a anotace genomu a jeho pozorovaných mutací. Bioinformatické nástroje napomáhají porovnávání genetických a genomických dat a pochopení vývojových aspektů molekulární biologie. Velkým přínosem je také vytváření rozsáhlých databází biologických dat, které umožňují rychlý souběžný přístup [12].

4.1.1 Bioinformatický přístup k predikci

V souvislosti s anotací funkce nsSNP jsou bioinformatické metody využívány při několika krocích. Jako počáteční bod při predikci se používají databáze mutací, které slouží jako zdroje dat. V těchto databázích se nacházejí informace o fenotypických projevech mutací společně s informacemi o korespondujícím genu a proteinu.

Analýza sekvencí (sequence analysis) poskytuje informace o tom, které úseky genů a posléze proteinů, jsou během evoluce zachovávány beze změn, a tedy mají velký funkční význam, který by byl případnou mutací velmi pravděpodobně porušen. Existuje již celá řada prediktorů, založených na analyzování sekvencí na základě dalších biochemických vlastností proteinu, například agregační tendence (aggregation)a stability (stability) [29].

Pokud je k dispozici experimentálně ověřená struktura proteinu, lze posunout mutační analýzu na strukturální úroveň, čímž se výsledky predikcí stávají věrohodnější, neboť vychází nejen ze sekvence, ale i z vyšších struktur proteinu. Mutace mohou být v takovém

případě do struktury modelovány a roli nového úseku sekvence je pak snadnější analyzovat. Lze totiž pozorovat, jak tento úsek do proteinu zapadá a jak ovlivňuje jeho důležitá vazebná místa [29].

Až dosud bylo zvykem pro predikci použít jeden nebo několik málo principů. Nyní je ale výzkum zaměřen na kombinování většího množství metod, tak, aby výsledky byly co nejpřesnější. To je i případ praktické části této práce, kdy je kombinováno celkem osm dostupných bioinformatických nástrojů pro dosažení co nejspolehlivějších výsledků na širokém spektru proteinů a mutací.

4.2 Přehled metod analýzy mutací

Pro úplnost nyní uvádím přehled metod, které jsou dnes používány pro anotaci funkce nukleotidového polymorfismu. Jak již bylo zmíněno, tyto metody lze mezi sebou kombinovat za účelem dosažení vyšší spolehlivosti predikce. V další kapitole potom bude následovat rozbor použitých bioinfomatických nástrojů, které právě těchto metod a principů využívají.

4.2.1 Databáze

Databáze slouží jako základ pro bioinformatický výzkum v oblasti mutací a strukturálního základu chorob. Příkladem těchto databází jsou CMDBs (Central mutation databases), z nichž nejvýznamnější jsou HGMD (Human Gene Mutation Database) [25] a OMIM (Online Mendelian Inheritace in Man) [13]. V těchto databázích jsou k nalezení informace z literatury, zaměřené na variace v lidských genech a jejich vliv.

V databázi UniProtKB/Swissprot [3] se nacházejí manuálně ohodnocené proteinové složky, které obsahují seznamy známých variant sekvencí. Mimo těchto lze najít ještě další databáze s různým zaměřením, takže datový základ pro predikci je ze strany databází velmi široký [29].

4.2.2 Konzervovanost sekvencí (Conservation analysis)

Tato metoda využívá již zmíněného faktu, že totiž evoluce zanechává některé úseky genů nezměněné ve svém průběhu. Je nasnadě, že tyto úseky mají velký význam a zastávají typicky velmi důležité funkce. Pokud dojde v takovém místě genu k nsSNP polymorfismu, velmi často také dojde k poruše funkce proteinu. A naopak, úseky genu, které podlehly během evoluce určitým modifikacím, vykazují mnohem menší potenciál svých mutací mít patogenní účinek. Je to způsobeno tím, že škodlivé mutace ovlivňují chemicko-fyzikální vlastnosti a tvar proteinu mnohem radikálněji než mutace, které během evoluce způsobují odlišnost jednotlivých druhů.

Pro anotaci vlivu nsSNP je tedy nutné znát úroveň evoluční konzervovanosti různých částí proteinu. Na základě těchto znalostí lze pak dokonce posoudit možný účel daného úseku či jeho funkci, případně zjistit, které aminokyseliny lze v daném místě zaměnit za jiné bez negativního dopadu.

K analýze konzervovanosti sekvencí je nutné použít mechanismus, který dovede podobné sekvence zarovnat a zařadit analyzovaný řetězec do proteinové rodiny, podle jejíž charakteristiky je rozhodnuto o funkční anotaci. K tomuto účelu vznikla celá řada metod, které s různou spolehlivostí sekvence zařazují. Klasické metody, například ClustalW [16], podávají kvalitní výsledky na sekvencích, které se příliš neliší, zato však nefungují tak spolehlivě,

pokud diference přesáhne určitou mez. Každá metoda má své silné a slabé stránky, přičemž neexistuje žádná bezchybná metoda [29].

4.2.3 Agregace (Aggregation)

Nativní nebo strukturně narušené proteiny mají tendenci seskupovat se v agregáty. Tyto agregáty se vyznačují zvýšenou úrovní β-struktury, jsou v naprosté většině škodlivé, dokonce jsou zahrnuty mezi příčinami některých závažných chorob, například diabetes typu II, Alzheimerova a Parkinsonova choroba. Polymorfismy nsSNP mohou změnit některé vlastnosti proteinu tak, že bude více náchylný k vytvoření agregátu, přičemž bylo zjištěno, že k tomu, aby byl protein takto modifikován, stačí i jednobodová změna primární struktury [29].

Pro predikci tendence proteinu k agregaci bylo již vytvořeno velké množství algoritmů. Princip těchto metod vychází z ohodnocení všech aminokyselin hodnotou, která určuje, jak toto residuum mění náchylnost k agregaci. Jednotlivá ohodnocení jsou stanovena a ověřena experimentálně. Predikovat lze ale také na základě sekundárních struktur a jejich dopadu na agregační tendenci nebo na základě specifických interakcí mezi dvěma proteiny vypočítaných ze statistické analýzy nativních globulárních proteinů [29].

4.2.4 Analýza strukturálních parametrů

Při nahrazení aminokyseliny jinou v důsledku nesynonymní mutace dochází samozřejmě i ke změnám fyzikálních a chemických vlastností. Substituce může vytvořit specifické strukturální uspořádání, zvláště pokud je původní aminokyselina značně menší než nová. Pokud se nové uspořádání výrazně liší od původního, je pravděpodobné, že mutace měla škodlivý efekt a funkce proteinu bude narušena.

K predikci toho, zda daná substituce způsobí nadměrné strukturální přeuspořádání, se užívá takzvané rotamerové analýzy. Principem je namodelování struktury proteinu po substituci a rotování mutovaného residua a ohodnocováním stavů podle toho, zda zapadají do struktury. Z těchto rotamerů jsou poté vybrány ty, které mají skóre nejvyšší, a tím pádem nejlépe vyhovují proteinu, a ty jsou použity k další analýze. Po jejím ukončení celkové skóre identifikuje škodlivost dané mutace. Pokud jsou k dispozici experimentáně ověřené struktury, lze tyto použít jako předlohy pro analýzu efektu mutací [29].

4.2.5 Zkoumání stability a vztahů mezi částmi řetězce

Dalším z molekulárních patogenních následků mutací je snižování stability proteinu a narušení skládání (folding) proteinu. Obě tyto vlastnosti jsou spolu úzce spjaty. Škodlivá mutace může narušit skládání natolik, že dojde u většiny molekul k celkové poruše skládání a protein potom zastává zcela odlišné funkce. Co se stability týče, ta může být snížena v důsledku porušení energetické rovnováhy mezi proteinovými segmenty.

Chemické vazby v proteinu určují detailní tvar proteinu, přičemž hydrofobní vazby v proteinovém jádře mají největší vliv na jeho celkovou stabilitu. Stačí, aby v důsledku substituce došlo jen k malé odchylce v síti těchto vazeb, a dochází k destabilizaci řetězce. Na základě namodelování mutace lze pomocí různých technik strojového učení predikovat její škodlivost, využívají se například SVM (support vector machine) a neuronové sítě [4].

4.3 Srovnání metod

K predikci vlivu nsSNP na protein jsou využívány informace o sekvenci a/nebo struktuře, případně posuzování stability. Metody založené na sekvenci se snaží protein zařadit do rodiny vyhledáváním příbuzných proteinů v databázi, následně pomocí metod zarovnání sekvencí hledají evolučně konzervované úseky. Podobně pracují i metody založené na analýze struktury, ovšem v databázích hledají protein, který vykazuje nejpodobnější strukturální vlastnosti. Další metody mohou také čistě spoléhat na anotaci uvedenou v databázích Swiss-Prot a dalších.

Přímé porovnání metod je velmi složité, protože je mnoho kritérií, která mohou mít různou váhu podle situace. Navíc byly metody trénovány a testovány na odlišných datových sadách, takže výsledky nemusí být směrodatné. Je důležité podotknout, že neexistuje softwarová predikční metoda, která by byla ideální [21].

4.3.1 Pokrytí

Metody se mohou lišit v počtu substitucí, které jsou vůbec schopny analyzovat. V tomto ohledu mají značnou převahu metody vycházející ze sekvence, neboť jejich databáze jsou obsáhlejší (znalost řetězce je běžná) a snadněji se protein zařadí do své rodiny. Naopak, znalosti struktur proteinů jsou méně časté, proto strukturální metody dokáží určit pouze necelých 15 % mutací, přičemž v případě sekvenčních metod lze dosáhnout pokrytí až 81 % [21]. Proto je většina dnešních metod založena na analýze sekvence a strukturální analýzu nabízí jako volitelné kritérium. V případě použití databází anotací záleží na použití té které konkrétní databáze. Poslední studia však ukazují, že predikce na základě anotace z databází Swiss-Prot a podobných nevede ke zvýšení přesnosti predikce, ale spíše k lehkému poklesu [21].

4.3.2 FNR, FPR

Při testování metod se obvykle postupuje ve dvou fázích. Nejprve se metoda testuje na datové sadě mutací ověřených jako škodlivých. Procento mutací, které metoda označí nesprávně, tedy v tomto případě jako neutrální, se nazývá FNR (false negative rate). Ve druhé části se naopak použije datová sada neutrálních mutací a procento mutací, které byly nesprávně posouzeny jako škodlivé, se označí FPR (false positive rate). Tyto dvě metriky dokumentují chybovost metody, snažíme se tedy o jejich snižování [21]. Pokud dosáhne například metrika FNR hodnoty 0, pak můžeme říci, že metoda úspěšně odhalí všechny škodlivé mutace, žádnou nevynechá. Tato vlastnost by byla velmi žádoucí, protože pak by tato technika mohla být použita k vyloučení mutací, které jsou zcela jistě neutrální a značně tak zmenšit celkovou sadu mutací, zbylé mutace by pak mohly být analyzovány laboratorně. Míra zmenšení prohledávaného prostoru ale závisí právě na druhé metrice, tedy FPR.

4.4 Význam predikce

V souvislosti s touto prací je nejvýznamější přínos predikcí v proteinovém inženýrství. Toto odvětví využívá vlastnosti proteinů uměle za účelem dosažení konkrétních funkcí a jejich zesílení. Z tohoto důvodu jsou důležité znalosti mutací, které mohou potenciálně měnit funkci proteinu tak, aby odpovídala požadavkům. Například, protein sám o sobě váže a rozkládá konkrétní toxin ve svém okolí. Při aplikaci substituce lze dosáhnout toho, že tuto

činnost bude vykonávat mnohem rychleji a účinněji. Problém je ovšem s nalezením správné mutace na správné pozici. Laboratorní zkoumání mutací je sice přesné, ale zdlouhavé a nákladné, proto je důležité mít nástroje, které jsou schopné identifikovat potenciální pozice a substituce, které mohou splnit požadovaný účel, a až tuto množinu poté laboratorně testovat. Jedná se o značné urychlení celého procesu výzkumu a každý byť i minimální nárůst přesnosti predikce se velmi významně projevuje.

Anotace vlivu nsSNP je také důležitá z hlediska výzkumu genetických chorob a možností jejich léčby. Stejně jako v případě proteinového inženýrství je velmi esenciální co nejradikálnější zúžení množiny mutací, které by mohly s chorobami souviset. Predikční metody tedy mohou přímo odhalit příčinu Mendelovských chorob, které jsou způsobeny právě jedinou změnou v genu, lze je ale také efektivně použít při objasňování příčin komplexních chorob, právě díky již zmíněné schopnosti identifikovat množinu potenciálně škodlivých mutací [21].

Kapitola 5

Použité nástroje

V této kapitole budou rozebrány konkrétní softwarové nástroje, které byly použity v praktické části, a bude provedeno srovnání jejich přístupu a metodologií. Většina z těchto nástrojů kombinuje více metod predikce za účelem zvýšení její přesnosti. Konsensus bude vytvářen nad výstupy z celkem osmi nástrojů. Za tímto účelem byly vybrány nástroje různých principů, aby bylo dosaženo co nejvyšší míry univerzálnosti výsledného metanástroje.

5.1 MAPP

Metodika nástroje MAPP vychází z analýzy konzervovanosti, která byla popsána v předchozí kapitole. Důležitou součástí je tedy algoritmus zarovnání sekvence, díky kterému může nástroj vyhledat příbuzné sekvence. K účelu zarovnávání MAPP využívá vlastní techniku, doplněnou o fylogenetický strom. Proces predikce má potom několik fází. Nejprve jsou sekvence podrobeny analýze podobnosti a rozděleny pomocí fylogenetické struktury. Pod sebou seřazené sekvence jsou poté procházeny po jednotlivých sloupcích, kdy jsou porovnávány jednotlivé vlastnosti aminokyselin pod sebou (např. hydropatie, polarita či el. náboj). Vzniká tak matice, dokumentující fyzikálně-chemické vlastnosti. Pro každou část řetězce je pak provedeno ohodnocení, jak se tyto vlastnosti mění a o kolik se odchylují při provedení mutace, přičemž vyšší skóre dopadu implikuje zásah do funkčnosti proteinu [26].

Vstupem pro predikci je aminokyselinový řetězec a příslušná mutace. Výstupy všech nástrojů jsou rozebírány v praktické části.

5.2 nsSNPAnalyzer

Druhý nástroj, nsSNPAnalyzer, využívá vedle znalostí o sekvencích také strukturální informace. Využívá techniky strojového učení, která předpokládá vyrovnanou trénovací datovou sadu mutací, které jsou nástroji předloženy i s jejich skutečnou anotaci. Znalosti, které nástroj z této sady získá, jsou použity při predikování. Výhody a nevýhody této techniky budou shrnuty v závěru kapitoly.

Algoritmus nsSNPAnalyzeru uvažuje vyhledání příbuzných sekvencí v databázi EMBL a analýzu konzervovanosti. Zároveň jsou v databázi ASTRAL vyhledány homologní struktury se svými parametry. Výstupy obou větví jsou poté poskytnuty jednotce strojového učení, kterou je v tomto případě náhodný les (random forest). Klasifikátor náhodného lesa byl natrénován na části datové sady z databáze Swiss-Prot, sdílí tedy její případné nepřesnosti [5].

Vstupem nástroje je sekvence, mutace a volitelně i strukturální informace ve formátu PDB. Pokud je tato zadána, je přeskočen krok s vyhledáváním v databázi ASTRAL.

5.3 PANTHER

PANTHER je nástrojem, který opět vychází z analýzy konzervovanosti proteinového řetězce. Inovace jeho přístupu spočívá v použití vlastní knihovny rodin proteinů. Tyto rodiny jsou dále děleny na podrodiny tak, aby se členové těchto podrodin od sebe lišili co nejméně. Každá podrodina i rodina je reprezentována statistickým modelem, který se nazývá skrytý markovský model (HMM - Hidden Markov Model). Při analýze obdrženého řetězce se pomocí HMM zjistí, do které podrodiny patří, a podle členů této rodiny je rozhodnuto o patogenicitě mutace pro konkrétní úsek. Pokud protein více patří k HMM rodiny než k HMM kterékoli její podrodiny, vzniká nová podrodina. Výhodou tohoto přístupu je, že se knihovna i modely aktualizují s novými proteiny. Na druhou stranu, pokud PANTHER nedokáže protein zařadit do rodiny, není schopen predikovat [28].

Vstupem nástroje je vzhledem k použité technice opět reprezentace primární struktury proteinu a mutace.

5.4 PhD-SNP

PhD-SNP je založen na metodě analýzy konzervovanosti sekvencí. Predikční model navíc používá jako atribut i fyzikálními metodami zjištěnou zmenu stability po provedení mutace. K tomuto určení navic nepotřebuje znát 3D strukturu proteinu. Využívá technik strojového učení a SVM (Support Vector Machine) se čtyřmi jádry, ve které zahrnuje do predikce i prostředí, ve kterém k mutaci došlo. Do predikce tak promlouvají i sousední residua [10].

Vstupem klasifikátoru je řetězec aminokyselin a mutace.

5.5 PolyPhen-1 a PolyPhen-2

Oba tyto nástroje predikují na základě jak sekvence, tak struktury i anotace, kterou získávají z databáze Swiss-Prot. Rozdíl mezi nimi je v tom, že PolyPhen-1 nepracuje na principu strojového učení a využívá klasifikátoru založeného na empiricky sestavené tabulce pravidel [27]. Pravidla určují škodlivost mutace na základě výstupu ze sekvenční a strukturální analýzy, případně berou v potaz i anotaci z databáze. PolyPhen-2 používá strojové učení - naivní Bayesův klasifikátor. Tento klasifikátor provádí porovnání vlastností proteinu před a po mutaci a na základě nabytých znalostí ze strojového učení rozhoduje o škodlivosti.

Vstupem těchto nástrojů je vždy sekvence a mutace, strukturální informace jsou nepovinné [23], [1].

5.6 SIFT

SIFT je nástroj využívající čistě znalosti sekvence. Proces predikce nejprve pomocí vlastního zarovnávacího algoritmu odhalí rodinu, do které vstupní protein patří, a identifikuje konzervované úseky. Pokud mutace spadá do takového úseku, SIFT zkontroluje, do jaké míry se liší vlastnosti nové aminokyseliny od původní. Například pokud je hydrofobní valin nahrazován rovněž hydrofobním isoleucinem, obvykle je mutace klasifikována jako neutrální. V případě nahrazení jinou aminokyselinou, například polární, je výstupem škodlivý vliv.

Vstupem je proteinový řetězec a mutace [14].

5.7 SNAP

Posledním vybraným nástrojem je SNAP, pracující na principu strojového učení a využívající znalosti sekvence a anotace z databáze Swiss-Prot. Při klasifikaci používá natrénovanou neuronovou síť, která rozhoduje na základě atributů vypočtených ze sekvence, např. PSIC profilu, změny elementu sekundární struktury nebo změny tzv. solvent accesibility[1]. Na rozdíl od ostatních nástrojů nebyl SNAP trénován na mutacích z databází genetických chorob, ale z databází pro proteinové inženýrství (PMD). Vstupem predikce je opět sekvence a mutace [9].

5.8 Porovnání a shrnutí

Všechny nástroje a příslušné algoritmy jsou přehledně zobrazeny v tabulce 5.1. Pro účely práce byla snaha vybrat nástroje pokud možno různorodé, abychom mohli využít potenciál více metod. Na druhou stranu bylo nutné vybrat nástroje s přijatelnou schopností predikce, aby nebyl výsledný konsensus váhově příliš vychýlen směrem k některému nástroji. Další diskuse a analýza tohoto problému se nachází v praktické části.

Nástroj	**Metoda predikce**	**Algoritmus**
MAPP [26]	Sekvence, fyz.-chem. vlastnosti	Zarovnání sekvencí
nsSNPAnalyzer [5]	Strukturální a funkční par.	Rozhodovací strom (random forest)
PANTHER [28]	Analýza konzervovanosti	Markovské řetězce
PhD-SNP [10]	Analýza konzervovanosti	SVM
PolyPhen-1 [23]	Sekvence i struktura, anotace Swiss-Prot	Empricky sestavená tabulka pravidel
PolyPhen-2 [1]	Sekvence i struktura, anotace Swiss-Prot	Naivní Bayesovský klasifikátor
SIFT [14]	Analýza konzervovanosti	Zarovnání sekvencí
SNAP [9]	Analýza konzervovanosti, anotace Swiss-Prot	Neuronové sítě

Tabulka 5.1: Souhrn metodologií a algoritmů nástrojů.

5.8.1 Strojové učení

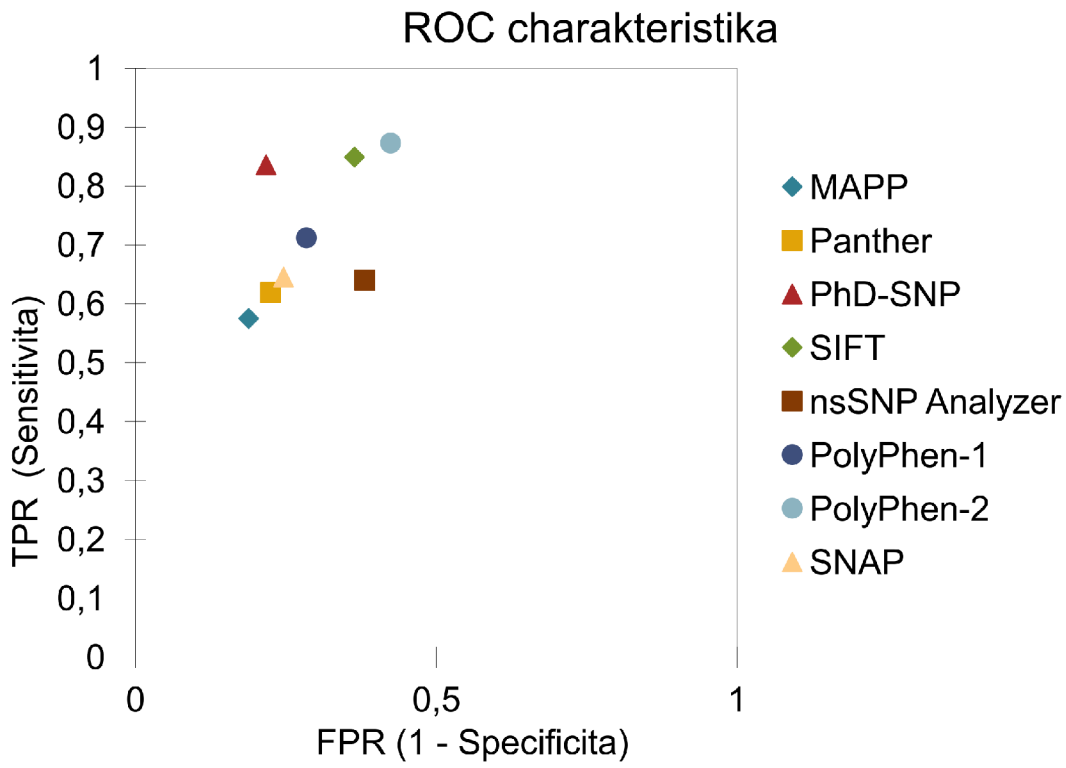
Nástroje SNAP, nsSNPAnalyzer, PolyPhen-2 a PhD-SNP pracují na principu strojového učení. Jak už bylo naznačeno, jedná se o metodu, která vytvoří klasifikátor a natrénuje jej na bázové datové sadě. Je třeba, aby tato sada byla jednak co nejširší, čímž se dosáhne potřebné univerzálnosti, ale také vyvážená z pohledu proteinů a mutací. Každý nástroj je totiž díky unikátnosti svého přístupu různě úspěšný na odlišných proteinových rodinách. Pokud by v trénovací sadě značně převládaly proteiny z nějaké rodiny, nástroj by v praxi na této rodině pracoval velmi dobře, v jiných případech by ale vykazoval výsledky horší [4]. Při aplikaci techniky strojového učení je tedy třeba se s tímto problémem vypořádat. Na

[1] velikost styčné plochy proteinu a okolí ve vazebném místě.

druhou stranu, pokud se toto podaří, pak dostává nástroj oproti dalším velkou výhodu, neboť má větší znalosti a obecně podává lepší výsledky predikce, což je vidět i na výsledcích v praktické části.

5.8.2 ROC charakteristika

ROC charakteristika je způsob porovnání, který u predikčních metod a nástrojů (ale i obecně ve statistice) umožňuje ukázat schopnost nástroje správně identifikovat neutrální, respektive škodlivé mutace. Jedná se o závislost mezi senzitivitou (TPR - true positive rate) a specificitou (TNR - true negative rate). Senzitivita udává, jaké procento z celkového počtu škodlivých mutací v datové sadě nástroj skutečně označí jako škodlivé. Specificita naopak vyjadřuje míru úspěšnosti identifikace neutrálních mutací. V literatuře [6] jsou tyto hodnoty uvedeny, ROC charakteristiky můžeme vidět na obrázku 5.1. Jedná se pouze o obecné srovnání, v praktické části se porovnáváním nástrojů zabývám znovu a na konkrétním trénovacím a testovacím datasetu.



Obrázek 5.1: ROC charakteristiky nástrojů sestavené podle komparativní studie [6].

Kapitola 6

Evoluční algoritmy

Evoluční algoritmy (EA) označují třídu stochastických výpočetních metod a matematických postupů, které využívají modelů evolučních procesů v přírodě. Prvotním zdrojem tohoto přístupu je Darwinova teorie o přírodním výběru. Tato teorie popisuje průběh vývoje organismů jako proces postupné adaptace jedinců na svoje okolí, kdy se každý jedinec přizpůsobuje různě. Z celé generace potomků pak přežívají pouze ti, kteří se adaptují nejlépe [17].

Všechny tyto metody nesou několik společných rysů. Jedná se o numerické optimalizační metody, pracující s množinou potenciálních řešení daného problému namísto soustředění se na řešení jediné [15]. V průběhu algoritmu pak dochází k postupnému vylepšování řešení až do chvíle, kdy je dosaženo optima, respektive sub-optima. Ke zlepšování řešení jsou využity různé techniky, které opět vychází z přírodních jevů, jedná se o mutace a křížení, což znamená, že stejně jako v Darwinově teorii, noví jedinci vznikají kombinacemi a mutacemi svých rodičů [15].

První myšlenka využití evolučních procesů v informatice se datuje do 60. let 20. století, k jejich reálnému využití a rozmachu ale došlo až později v důsledku růstu výpočetní síly a rychlosti strojů, které úlohy prováděly. Na konci 20. století se jasně oddělují rozličné přístupy jak evoluční procesy modelovat a jak je využít. Evoluční algoritmy se tak dělí na genetické algoritmy a genetické programování, evoluční programování a evoluční strategie [17]. Tato kapitola bude věnována převážně evoluční strategii (známé pod zkratkou ES), která byla poté použita i v praktické části.

6.1 Evoluční strategie

Prvotní idea této stochastické výpočetní metody se datuje do roku 1963, autory byli studenti Hans-Paul Schwefel a Ingo Rechenberg, kteří pracovali na optimalizaci tvaru těles za účelem snížení turbulence v tunelu. K dosažení řešení se právě inspirovali vývojem v přírodě a svůj přístup pak prezentovali jako evoluční strategii [17].

6.1.1 Použití

Evoluční strategie a obecně evoluční algoritmy jsou vhodné tam, kde nemůžeme ideální stav najít pomocí analytického řešení. Zatímco genetické algorimy lze použít na celou řadu úloh, použití ES je díky její specifičnosti v praxi spíše ojedinělé [17]. ES se používá výhradně na optimalizaci vektoru reálných parametrů tak, aby bylo dosaženo extrému hodnotící funkce (v angličtině označovanou jako *fitness*). Genetické algoritmy jsou v tomto směru obecnější,

mohou pracovat s jakýmikoli bitovými hodnotami, převod na čísla provádí až ve chvíli, kdy je to bezpodmínečně nutné.

6.1.2 Základní princip ES

ES mají více podob a druhů, které se od sebe liší přístupem a hodnotami parametrů. Základní princip je však u všech stejný, na začátku je vždy rodič, tedy vektor reálných parametrů. Ten je inicializován, v průběhu evolučního procesu jsou pak vytvářeni potomci mutací hodnot rodiče. Celý algoritmus popsán podle [18] pomocí následujícího pseudokódu:

1. Inicializuje se rodič a další proměnné (např. dosud nejlepší výsledek).
2. Vyhodnotí se koncová podmínka, pokud je splněna, algoritmus končí.
3. Vektor rodiče je n-krát mutován a vzniká generace potomků.
4. Potomci jsou ohodnoceni funkcí *fitness*.
5. Nejlepší z potomků je porovnán s rodičem, pokud vykazuje lepší vlastnosti, stává se novým rodičem.
6. Pokud je nalezené řešení lepší než dosavadní nejlepší, dojde k aktualizaci této proměnné.
7. Pokračuje se krokem 2.

Varianty ES se mohou lišit například v ukončovací podmínce, tou může být maximální doba výpočtu, konečný počet generací nebo uspokojivá hodnota *fitness* funkce aktuálně nejlepšího jedince (vektoru). Dále se mohou lišit například počtem potomků (n) v jedné generaci, případně tím, zda má rodič možnost přetrvat v populaci (v případě, že ani jeden z jeho potomků nebude lepší). Evoluční algoritmy, a tedy i ES, se pro rozlišení této podmínky rozdělují podle [15] na 2 typy:

- (μ+λ) - EA, kde je nový rodič vybírán z množiny potomků a předchozích rodičů,
- (μ,λ) - EA, kdy je nový rodič vybírán pouze z množiny potomků předchozí generace.

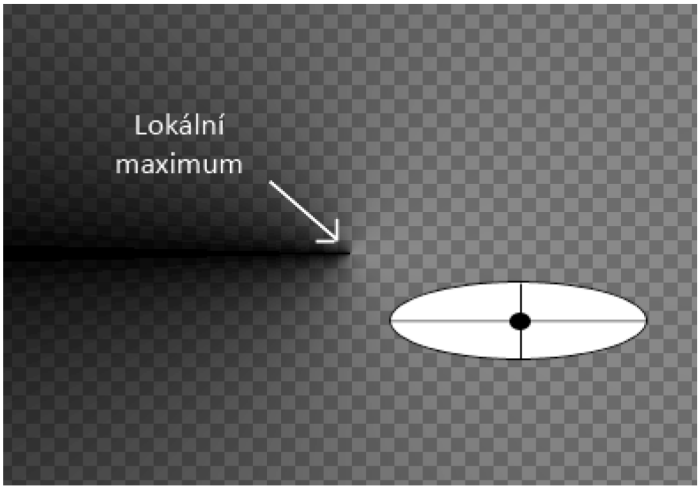
6.1.3 Reprodukce

Vznik potomka v průběhu ES se děje změnou rodičovského vektoru. Tuto změnu zajišťují evoluční operátory, ke kterým patří křížení a mutace. Varianta ES v praktické části pracuje pouze s jedním rodičem, proto se při reprodukci vychází z mutací. Jedince můžeme obecně popsat jako vektor reálných proměnných $x = (x_1, x_2, \ldots, x_n)$, kde hodnoty x_i mohou být zdola i shora ohraničené. Mutace se pak provádí obecně podle vzorce $x'_i = x_i + N(0, \sigma)$ pro $i = 1 \ldots n$, kde σ je směrodatná odchylka mutace. Tato odchylka je rovněž na začátku algoritmu inicializována a buď může mít konstantní hodnotu, nebo ji lze během procesu měnit. Jednou z možností je pravidlo jedné pětiny, které počítá procento úspěšných reprodukcí, tedy takových, kdy potomek vykazoval lepší vlastnosti než jeho předek [15]. Pokud toto procento přesáhne 20 %, odchylka se zvýší, neboť stále nacházíme lepší řešení a chceme se k výsledku dostat rychleji. Naopak, při úspěšnosti reprodukce menší než 20 % se odchylka snižuje, protože je nasnadě, že se řešení blíží k extrému.

6.2 Autoevoluce řídících parametrů

V této práci jsem použil variantu ES, kde se spolu s vektorem řešení evolučně vyvíjí i řídící parametry, konkrétně směrodatná odchylka. Takového chování je dosaženo díky parametrům učení, kterých může být různý počet. Tento způsob je účinnější než klasické pravidlo jedné pětiny, protože se výsledky mohou rychleji blížit k extrému funkce *fitness* [7], [18]. Existují tři typy autoevolučních ES:

1. Směrodatná odchylka je stejná pro všechny reálné parametry vektoru řešení, potomci jsou generováni se stejnou pravděpodobností ve všech směrech od rodiče.
2. Směrodatná odchylka se vyvíjí pro každou složku vektoru zvlášť, což umožňuje rychlejší posun řešení k hledanému extrému.
3. Kovarianční matice umožňuje ještě větší nárůst rychlosti blížení se k suboptimální hodnotě.



Obrázek 6.1: Evoluční strategie typu 2 s vyšší pravděpodobností výskytu potomků ve směru k extrému hodnotící funkce.

K nalezení optimálních vah nástrojů a tedy navržení konsensu byl použit typ 2, který je reprezentován obrázkem 6.1a. Tato varianta se vyznačuje specifickým způsobem vytváření potomků. Jsou přítomny dva parametry učení: (i) společný parametr učení τ', (ii) specifiký parametr τ pro každý cílový parametr. Jedinec je poté reprezentován jako vektor $(x_1, \ldots, x_n, \sigma_1, \ldots, \sigma_n)$, kde hodnoty x_i jsou cílové reálné parametry a σ_i směrodatné odchylky, podle kterým se mění. Parametry učení jsou vyjádřeny vztahy $\tau' = 1/\sqrt{2n}$ a $\tau = 1/\sqrt{2\sqrt{n}}$ [18]. V těle cyklu pak nejprve dochází k mutaci vektoru směrodatných odchylek a následně i k mutaci cílových parametrů, podle těchto vzorců [18]:

$$\sigma_i' = \sigma_i \cdot exp(\tau' \cdot N(0,1) + \tau \cdot N_i(0,1)) \tag{1}$$

$$x_i' = x_i + \sigma_i' \cdot N_i(0,1) \tag{2}$$

Zápis $N(0,1)$ odpovídá náhodně generovanému číslu z normálního rozdělení se střední hodnotou 0 a směrodatnou odchylkou 1. Hodnota variance je dále typicky zdola omezena, aby proces vývoje neustále pokračoval [18].

6.3 Parametry ES

ES se vyznačuje nativní prací s vektorem reálných čísel, proto je tato metoda vhodná pro nalezení váhového konsensu mezi zvolenými nástroji. Navzdory tomuto svému specifickému přístupu lze nastavovat poměrně velké množství parametrů. Vzorec pro mutaci je možné modifikovat například nastavením jiné střední hodnoty nebo zvolením zcela jiného pravděpodobnostního rozdělení, ačkoli toto není příliš obvyklé. Do další generace může být vybrán buď nejlepší z nových potomků nezávisle na ohodnocení rodiče, nebo i stávající rodič v případě, že se ani jeden z potomků dané generace neadaptoval lépe. Počáteční konfigurace rodičovského vektoru dále nemusí být náhodná, můžeme ji nastavovat téměř podle libosti, stejně jako počáteční vektor směrodatných odchylek pro jednotlivé cílové parametry. Stěžejním faktorem pro úspěšné nalezení optima je rovněž počet generací, počet potomků v generaci a velikost populace. A z jistého pohledu nejdůležitějším parametrem je minimální standardní odchylka, která zaručuje pokračování hledání, čímž může pomoci zabránit uváznutí v lokálním extrému.

Vzhledem k množství parametrů a ke schopnosti ES nalézt pouze suboptimální řešení je doporučeno opakovat pokusy o nalezení vícekrát v řadě i s různými parametry, aby byla prohledána co největší část stavového prostoru.

Kapitola 7

Implementace

V praktické části práce jsem se zabýval navržením a implementací konsensuální funkce nad existujícími bioinformatickými nástroji pro predikci vlivu nsSNP. Konsensu může být dosaženo více způsoby, metanástroj, který jsem realizoval, využívá váhový přístup, který umožňuje upřednostnit některé použité nástroje před jinými podle přesnosti jejich predikce na širokém spektru mutací.

K implementaci byla využita technika strojového učení na dostupném trénovacím datasetu. Součástí praktické části bylo dolování dat za účelem vybudování univerzálního testovacího datasetu, na kterém bude výsledný metanástroj ohodnocen a diskutován.

Nejpodstatnější částí implementace je nalezení optimálního rozložení vah jednotlivých nástrojů. S využitím evoluční strategie a opakováním experimentů jsem hledal řešení, které vykazovalo nejvyšší přesnost na trénovací datové sadě. Použití ES v této konkrétní úloze je myšlenka ojedinělá a její výsledky by mohly predikci podstatně zefektivnit. Pro ověření výkonnosti nástroje byla provedena 10-fold křížová validace (10-fold cross validation).

Programy jsou napsány ve skriptovacím jazyce Python 3, implementace probíhala na operačním systému Microsoft Windows, skripty jsou však plně přenositelné. Vzhledem k velké výpočetní náročnosti experimentálních fází bylo rovněž využito externího výpočetního střediska MetaCentrum s unixovými platformami.

7.1 Použité datové sady

Princip strojového učení předpokládá trénovací dataset správně vyhodnocených predikcí, podle kterého bude program postupně nastavovat váhy. Po natrénování musí následovat fáze testování, která nový klasifikátor META ohodnotí na testovacích datech. Trénovací množina dat musí být co nejširší, aby pokrývala maximální procento stavového prostoru, a také co nejrozmanitější tak, aby program dokázal vyhodnotit různé typy vstupních hodnot s podobnou přesností. Žádoucí je rovněž i vyrovnaný poměr velikostí množin škodlivých a neutrálních mutací v datové sadě. Na testovací data nejsou kladeny takové nároky, ale pokud mají být výsledky objektivní, je dobré použít sadu, která se nepřekrývá s trénovací množinou, a rovněž zajistit maximální rozmanitost, případně použít více testovacích sad a výsledky porovnat [30].

7.1.1 Trénovací dataset

V současné době existuje pouze jeden obecně přijatelný dataset pro predikci efektu SNP, nazývá se VariBench. Jeho využití v oblasti vývoje nástrojů je poměrně široké, v této práci

byla použita jeho podmnožina, které neobsahuje mutace asociované s výskytem rakoviny. Trénovací sada pak obsahuje celkem 14 286 škodlivých a 17 339 neutrálních mutací a je budována tak, aby zajistila potřebnou diverzitu sekvencí a mutací i odstranění redundance mezi jednotlivými položkami. Veškeré údaje, které VariBench poskytuje, jsou experimentálně ověřeny, a tedy anotace funkcí jednotlivých položek (mutací) lze považovat za správné a směrodatné [19].

7.1.2 Testovací datasety

Pro účely testování navrženého konsensu byl použit reprezentativní výběr mutací z databáze PMD (Protein Mutant Database), dostupný na webových stránkách VariBench. Dále jsem použil také nově vytvořený dataset z patentové aplikace popisující vliv mutací serinové proteázy v organismu Bacilus subtilis (dataset dále nazývám Bacilus). Cílem této aplikace je výzkum vedoucí ke zlepšení vlastností proteinu z hlediska proteinového inženýrství. Jedná se o databázi, obsahující mj. proteinový řetězec (konkrétně serinová proteáza) a tabulku mutací, provedených na tomto řetězci, přičemž na každé pozici byly postupně zaměněny všechny přípustné aminokyseliny (tzv. saturační mutageneze) a efekt byl laboratorně ověřen.

```
            1         10        20          30        40          50
BPN'        AQSVPYGVSQIKAPALHSQGYTGSNVKVAVIDSGIDSSHPDLKVAGGASM
GCI-P036    AQSVPWGISRVQAPAAHNRGLTGSGVKVAVLDTGIS:THPDLNIRGGASF
GCI-P037    AQSVPWGISRVQAPAAHNRGLTGSGVKVAVLDTGIS:THPDLNIRGGASF
```

Obrázek 7.1: Část zdrojové sekvence pro dataset Bacilus [24].

Performance Index (PI) Values for Variants of GCI-P036 for Various Tested Properties.

POSITION (BPN' Numbering)	GG36 Variant (BPN' Numbering)	TCA Assay PI	CS-38 Microswatch Assay PI	PI BMI pH 8 32° C.	LAS-EDTA PI	AAPF Assay PI
1	A001C	0.93	0.87	1.14	0.62	1.11
1	A001E	1.25	0.94	1.00	1.08	1.34
1	A001F	1.15	1.18	1.01	0.53	1.24

Obrázek 7.2: Ukázka dat v patentu pro vytvoření testovacího datasetu [24].

Při vytváření datové sady Bacilus jsem nejprve pomocí programu ABBYY FineReader vyextrahoval sekvenci z patentové aplikace dostupné ve formátu PDF. Ze stejného dokumentu jsem poté z tabulky doloval záznamy o jednotlivých mutacích a podle hodnot ve sloupci AAPF Assay PI jsem je zařazoval mezi neutrální, respektive škodlivé. Jako rozhodovací práh byla na základě informací z [24] stanovena hodnota PI indexu 0.5, kdy škodlivé mutace mají hodnotu menší a neutrální naopak vyšší než tato prahová hodnota. Hlavička tabulky a první dolované hodnoty jsou pro ilustraci zobrazeny na obrázku 7.2. První sloupec udává pozici v sekvenci, na které k mutaci došlo, ve druhém sloupci je zakódována

mutace i s pozicí, přičemž první písmeno označuje původní aminokyselinu a písmeno na konci označuje mutací dosazenou aminokyselinu. Součástí dolování byla i skriptová kontrola, zda se původní aminokyselina na dané pozici v sekvenci skutečně nachází a nesouhlasné případy tak byly z výsledné sady ihned eliminovány. Další sloupce nejsou pro účely této práce důležité, s výjimkou posledního (AAPF Assay PI), podle něhož usuzujeme o reálné anotaci mutace způsobem popsaným výše.

Datová sada Bacilus obshauje celkem 1 664 škodlivých a 2 535 neutrálních SNP, vybraná podmnožina PMD pak 877 škodlivých a 821 funkčně neutrálních mutací.

7.1.3 Ohodnocení datových sad

Po výběru datových sad bylo provedeno jejich ohodnocení jednotlivými nástroji. Získali jsme tak ke každému řádku (mutaci) vedle reálné anotace také údaje o tom, jaký vliv jí predikují jednotlivé nástroje a do jaké míry svému úsudku věří. V tabulce 7.1 je část prvního řádku ohodnoceného datasetu VariBench, v druhých třech sloupcích se nachází výstupní data nástroje MAPP (hodnoty UNKNOWN, 0, trueú, sémantika těchto sloupců se poté opakuje pro zbývajících 7 nástrojů.

A010805	G74A	DELETERIOUS	UNKNOWN	0	true	...

Tabulka 7.1: Řádek s mutací v ohodnoceném datasetu.

Jednotlivé položky v řádku označují tyto hodnoty: První sloupec je identifikátor sekvence, ve které k mutaci došlo. Druhá hodnota je kód mutace s wild-type aminokyselinou, pozicí a substituovaným reziduem. Ve třetím sloupci je experimentálně ověřená anotace vlivu této mutace, kterou lze považovat za pravdivou a směrodatnou. Další sloupce se po třech vždy vztahují k jednomu nástroji, po řadě MAPP, nsSNPAnalyzer, PANTHER, PhD-SNP, PolyPhen-1, PolyPhen-2, SIFT a SNAP. Trojice hodnot pak po řadě označuje predikovanou anotaci (hodnoty BENIGN - neutrální, DETERIOUS - škodlivá a UNKNOWN - neznámá), míru důvěryhodnosti výsledku (confidence score, číslo v intervalu $< 0, 1 >$, kde vyšší hodnota znamená vyšší věrohodnost predikce) a validitu predikce (hodnoty true a false). Hodnoty confidence score poskytují samy nástroje v rámci výstupu predikce, ovšem jen některé z nich produkují hodnoty z intervalu $< 0, 1 >$. Proto se v rámci fáze ohodnocení datových sad provedla i transformace Confidence score do cílového intervalu, aby byly všechny výstupy nástrojů unifikované a snadněji zpracovatelné.

7.2 Nalezení konsensuální funkce

Jádrem praktické části je implementace evoluční strategie 2. typu podle [18] za účelem nalezení vhodné distribuce vah mezi nástroje a dosažení co nejvyšší přesnosti predikcí na trénovací datové sadě. V kapitole 6 bylo již uvedeno, že inicializace, průběh, rychlost a výsledky výpočtu evoluční strategie jsou značně ovlivněny větším množstvím parametrů. Doporučuje se tedy opakovat experimenty vícekrát a měnit i hodnoty parametrů. Na druhou stranu, výpočetní náročnost jednotlivých pokusů nedovoluje testování všech možných kombinací. Navíc, k úspěšnému výsledku se lze pravděpodobně dostat pomocí různých počátečních konfigurací. To zamená, že jednou z možností je také to, že zvolíme počáteční parametry co nejvhodnějším způsobem podle aktuálních znalostí a pouze provedeme dostatečný počet pokusů. Alternativou je naopak spuštění evoluční strategie s co nejvyšším

počtem kombinací atributů, ale nižší četností pokusů na jednu takovou kombinaci. V této práci jsem experimentoval s několika konfiguracemi, se kterými jsem provedl velké množství pokusů.

7.2.1 Funkce ohodnocení

V průběhu ES je nutné mnohokrát porovnávat různá rozložení vah a identifikovat to nejvhodnější z nich. K tomu slouží funkce *fitness*, která ohodnotí aktuální řešení a vrátí jeho procentuální úspěšnost. V kapitole 7.1.3 jsou popsány výstupní hodnoty predikce jednotlivých nástrojů, které musí být zkombinovány k dosažení konečného výsledku.

Mějme vektor vah w s položkami indexovanými $i = 0..7$, tedy pro každý nástroj jedna váha. Dále, v trénovacím datasetu mějme obecně n mutací, indexovaných $j = 0..n-1$. Hodnoty confidence score bude představovat proměnná c_{ij}, predikovanou anotaci uložme do proměnné a_{ij}. Hodnoty anotace jsou však vyjádřeny slovně, matematický model vyžaduje jejich transformaci do číselné podoby, neutrální (BENIGN) mutace budou mít hodnotu $a = -1$, škodlivé (DELETERIOUS) pak $a = 1$. Pro každou mutaci pak *fitness* funkce ohodnotí vektor vah pomocí následujícího vzorce:

$$P_j = \left(\sum_{i=0}^{7} w_i \cdot c_{ij} \cdot a_{ij} \right) / \left(\sum_{i=0}^{7} w_i \cdot c_{ij} \right), \quad j = 0..n-1 \tag{3}$$

Výsledná hodnota P_j pak obsahuje predikci konsensuláního nástroje pro daný řádek, číselně vyjádřenou reálným číslem z intervalu $< -1, 1 >$. V případě, že platí $P_j < 0$, pak nový nástroj vyhodnotil mutaci jako neutrální, v opačném případě jako škodlivou. Následně pak dojde k porovnání s reálnou funkční anotací a v případě shody byl vektor vah na tomto řádku úspěšný. Funkce *fitness* pak vrací procentuální úspěšnost vektoru na celém testovacím datasetu, tedy podíl správně stanovených řádků a celkového počtu mutací.

7.2.2 Mutace

Mutační krok byl implementován podle rozboru v kapitole 6.2. V jistých krocích výpočtu je třeba generovat náhodná čísla z normálního pravděpodobnostního rozdělení. K tomuto účelu jsou hojně využívány kongruentní generátory pseudonáhodných čísel, v této práci jsem dal přednost funkci z modulu jazyka Python, která generátor sama implementuje.

7.2.3 Váhový vektor

ES byla zvolena pro svou podporu manipulace s vektory reálných čísel. Váhová hodnota, kterou položky vektoru představují, však má ze sémantického hlediska jistá omezení. Ve většině případů, které manipulují s váhovými koeficienty, je hodnota váhy shora omezena prahem 1.0 . Dolní hranicí je pak zpravidla 0.0, protože ne všechny případy sémanticky umožňují zpracovat záporné váhy. Problém, který je v této práci řešen, v podstatě umožňuje akceptovat i záporné váhy, ovšem vzhledem k názornosti a lepší schopnosti interpretace jsem se rozhodl, že akceptovány budou pouze kladné hodnoty vah. Vygenerované vektory potomků tedy musí být před ohodnocením transformovány tak, aby všechny jejich položky splňovaly podmínky dané intervalem $< 0, 1 >$. Transformaci můžeme zapsat vztahy:

$$w_i = w_i / \left(\sum_{j=0}^{7} |w_j| \right) \tag{4}$$

$$w_i = \begin{cases} w_i & w_i > 0 \\ |w_i| & jinak \end{cases} \tag{5}$$

Program umožňuje pomocí přepínače úpravu danou vztahem (5) vynechat a pracovat tak i s vahami z intervalu $< -1, 1 >$.

7.3 Křížová validace datové sady

Jakkoli mohou vyjít výsledky testů na testovacích sadách, existuje ještě jeden ukazatel, ke kterému lze přihlédnout při hodnocení klasifikátoru. Křížová validace (X-fold cross validation) je metoda, kterou lze zjistit, jak bude vytvořený model reagovat na nezávislé datové sady. V této práci jsem implementoval nejčastější variantu, a to 10-fold křížovou validaci.

Principem této metody je rozdělení trénovací datové sady na 10 stejně velkých částí, označovaných anglicky fold. Nyní je provedeno 10 nezávislých pokusů, kdy je vždy 9 z 10 foldů je sloučeno v trénovací sadu a zbylá podmnožina je označena za testovací. Na trénovací podmnožině jsem v našem problému provedl 10 pokusů o nalezení konsensu pomocí evoluční strategie a vybral nejlepší řešení. Toto váhové rozložení bylo poté testováno na zbylé desetině původního datasetu. Takto se celý proces opakoval 10 krát, vždy s jiným testovacím foldem. Nakonec byly výsledky testování statisticky zpracovány a výsledkem je hodnota křížové validace, která udává přesnost, jakou by měl klasifikátor dosahovat na nových, nezávislých datech.

Úskalím této metody je samozřejmě trénovací sada, která musí být co nejrozsáhlejší a nejrozmanitější, aby pokrývala co největší část stavového prostoru. Dále je důležité při vytváření podmnožin postupovat tak, aby se od sebe jednotlivé foldy lišily co nejméně. Vzhledem k seřazení datasetu VariBench podle zdrojů a typů sekvencí jsem rozdělení prováděl postupným rovnoměrným rozřazováním řádků do jednotlivých foldů, takže každých 10 po sobě jdoucích mutací ve zdrojovém datasetu se ve výsledku nacházelo v různém foldu.

Bohužel, v současné době není dostupná tak rozsáhlá datová sada, která by pokryla celý stavový prostor. I proto jsou její výsledky považovány za nadhodnocené, neboť mezi testovací a trénovací podmnožinou existuje podobnost, takže přesnost dosahuje vyšších čísel. Tyto výsledky je tedy nutné konfrontovat s úspěšností klasifikátoru na testovacích sadách, v rámci této práce jsem použil dvě, rozebrané v kapitole 7.1.2. Testovací množiny by měly mít jiný charakter dat, v našem případě pocházejí testovací mutace z databází pro proteinové inženýrství, zatímco VariBench obsahuje mutace spojované s výskytem genetických chorob.

Kapitola 8

Experimenty a výsledky

Tato kapitola je věnována reálným výsledkům, kterých jsem dosáhl experimentováním s parametry a dalšími úpravami. Na závěr je uvedeno krátké shrnutí a přehledné porovnání nejdůležitějších výsledných hodnot pro objektivní posouzení přesnosti a přínosu nového klasifikátoru META.

8.1 Parametry ES

Během experimentování s evoluční strategií jsem vyzkoušel větší množství inicializačních hodnot, typů selekcí a ukončovacích podmínek. S každou konfigurací jsem provedl 120-150 pokusů o nalezení konsensu, přičemž aktuálně nejlepšího řešení jsem dosáhl s parametry a vlastnostmi, které jsou shrnuty v tabulce 8.1. Křížová validace na trénovací datové sadě poté proběhla se stejnými parametry.

Mutace	náhodná čísla z gaussovského rozdělení pravděpodobnosti
Selekce	$(\mu+\lambda)$
Výběr rodiče	rodič přežívá, pokud nemá žádného vhodnějšího potomka
Velikost populace	1
Počet generací	60
Počet potomků v generaci	25
Počáteční σ odchylka	0.4
Minimální σ odchylka	0.1
Počáteční hodnota vah	0.5 pro všechny nástroje
Transformace vah	všechny váhy kladné

Tabulka 8.1: Experimentálně zjištěné parametry ES

Nalezený konsensus je popsán vektorem vah, zobrazeným v tabulce 8.2. Na sadě VariBench je nejúspěšnějším nástrojem PhD-SNP, proto mu také odpovídá nejvyšší reálná váha. Nízké váhy u několika dalších nástrojů nemusí znamenat, že nástroj pracuje špatně, pouze mu byla trénováním přiřazena nižší hodnota a stává se tak vyvažujícím elementem, který rovněž promlouvá do cílové predikce.

MAPP	0.066
nsSNPAnalyzer	0.020
PANTHER	0.019
PhD-SNP	0.757
PolyPhen-1	0.258
PolyPhen-2	0.302
SIFT	0.040
SNAP	0.039

Tabulka 8.2: Rozdělení vah mezi dílčí nástroje.

8.2 Výkonnost META nástroje na trénovací sadě

Na obrázku A.1 v příloze je možné vidět přímé porovnání přesnosti predikce samotných nástrojů s nově vyvinutým konsensuálním metanástrojem nazvaným META. Mimo relativní přesnosti Accuracy jsou doplněny i další statistické metriky, nicméně jako funkce *fitness* byla použita právě hodnota Normalized Accuracy, popsaná v kapitole 7.2.1.

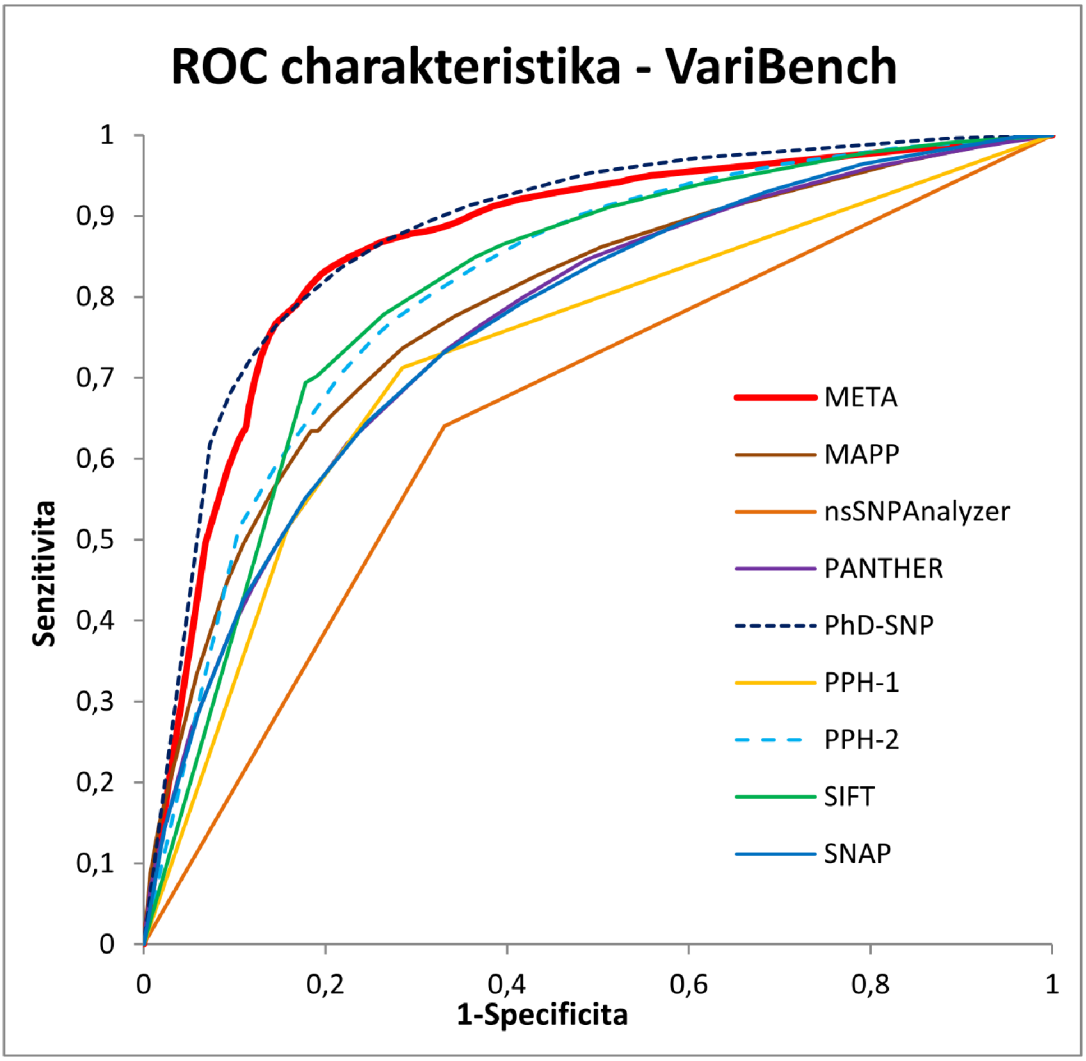
Hodnoty v řádku Cases + značí celkový počet reálně škodlivých mutací, které daný nástroj byl schopen vyhodnotit, v řádku Cases - se jedná naopak o mutace neutrální. Další zdrojové hodnoty jsou tyto:

- TP (true positive) - počet správně klasifikovaných škodlivých mutací,
- TN (true negative) - počet správně klasifikovaných neutrálních mutací,
- FP (false positive) - počet reálně neutrálních mutací klasifikátorem označených za škodlivé,
- FN (false negative) - počet reálně škodlivých mutací klasifikátorem označených za neutrální.

Z těchto údajů odvozujeme další, které napoví více o úspěšnosti testovaného nástroje:

- Senzitivita (TPR - true positive rate) - procento správně vyhodnocených mutací z množiny škodlivých mutací,
- Specificita (TNR - true negative rate) - procento správně vyhodnocených mutací z množiny neutrálních mutací,
- FNR a FPR - míry chybovosti, již vysvětleny v kapitole 4.3.2,
- Normalized Accuracy - aritmetický průměr TPR a TNR, objektivní charakteristika, neboť odráží úspěšnost na škodlivých i neutrálních podmnožinách,
- MCC (Matthewův korelační koeficient) - rovněž udává přesnost klasifikátoru, může být i směrodatnější než metrika Accuracy, protože reflektuje i různé kardinality podmnožin škodlivých a neutrálních mutací. Lze jej vyjádřit pomocí vztahu [22]:

$$MCC = \frac{TPR \cdot TNR - FPR \cdot FNR}{\sqrt{(TPR + FPR) \cdot (TPR + FNR) \cdot (FPR + TNR) \cdot (FNR + TNR)}} \quad (6)$$



Obrázek 8.1: ROC charakteristiky nástrojů včetně nového klasifikátoru na sadě VariBench.

META	MAPP	nsSNPAnalyzer	PANTHER	PhD-SNP	PPH-1	PPH-2	SIFT	SNAP
0,864	0,786	0,655	0,764	0,879	0,731	0,812	0,808	0,764

Obrázek 8.2: Hodnoty ploch pod ROC křivkou (metrika AUC) na sadě VariBench.

Z obrázku A.1 v příloze lze vyčíst, že nový nástroj META se podařilo natrénovat tak, že na trénovací sadě má lepší úspěšnost než kterýkoli z dílčích nástrojů. První předpoklad pro vytvoření úspěšnějšího klasifikátoru je tedy beze zbytku naplněn. Dále bylo zjištěno, že na této datové sadě lze přesnost META nástroje ještě zvýšit, ovšem za cenu toho, že nebude schopen vyhodnotit všechny mutace. Jedná se o dva předposlední sloupce uvozené hlavičkou s podmínkou pro výsledné skóre. Tato idea vychází z předpokladu, že pokud je výsledek predikce konsensuálního nástroje (hodnota z intervalu <-1,1>) blízko prahové hodnotě 0, pak mu nelze přisoudit potřebnou věrohodnost a vliv příslušné mutace nelze

spolehlivě vyhodnotit. Hodnota Accuracy pak při požadavku na |score|>0.25 je o 2,4 % vyšší než nejlepší z nástrojů na VariBench sadě, PhD-SNP.

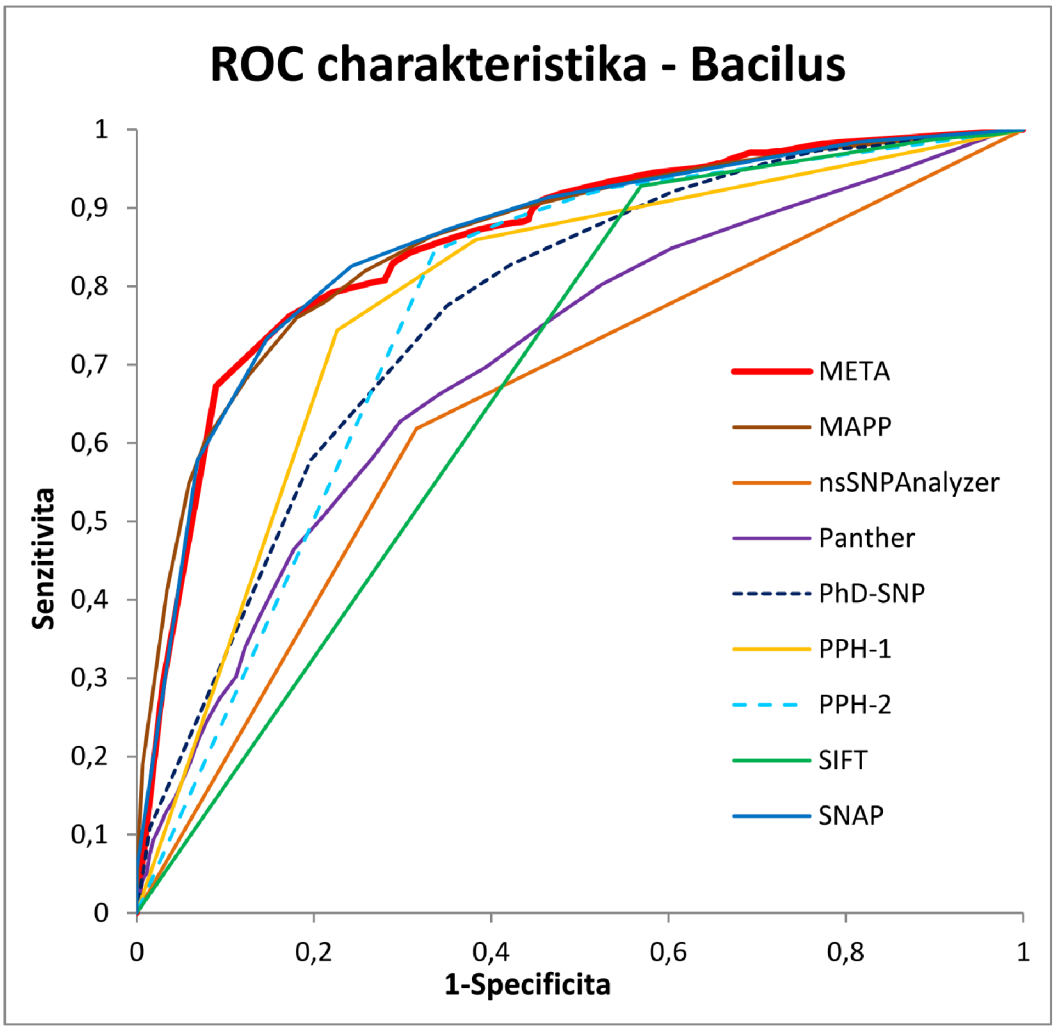
Obrázek 8.1 ukazuje ROC charakteristiku nástrojů v porovnání s novým klasifikátorem. Údaje pro vytvoření grafu byly získány postupným posouváním rozhodovacího prahu pro hodnotu P_j ze vztahu (3) v kapitole 7.2.1 od hodnoty -1 přes 0 k 1. Pouhým vizuálním srovnáním lze říci, že nástroj META na VariBenchi vykazuje podobnou přesnost jako nástroj PhD-SNP. K objektivnějšímu srovnání se proto používá metrika AUC (area under curve - plocha pod křivkou), která zahrnuje i rozdíly v přesnosti predikce na jednotlivých prazích a poskytuje komplexní srovnání. Hodnoty AUC pro VariBench jsou k dispozici na obrázku 8.2. Nástroj PhD-SNP díky lepší přesnosti při rozhodovacích prazích jiných než 0.0 dosahuje nejvyšší hodnoty AUC. Jak je vidět, ostatní nástroje na VariBenchi zdaleka nedosahují tak vysokých čísel, což odráží i rozdělení vah.

8.3 Výkonnost META nástroje na testovacích sadách

Význam statistických hodnot na obrázcích A.2 a A.3 v příloze s výsledky testů na sadách PMD a Bacilus je popsán v předchozí kapitole. Sémantika tabulky je totožná, sloupce i řádky mají stejný význam, liší se pouze hodnotami, které odpovídají příslušnému datasetu. Oproti analýze na VariBenchi chybí údaj o výsledku křížové validace, neboť tento dává smysl pouze na trénovací sadě.

8.3.1 Testování na sadě Bacilus

Podle hodnot na obrázku A.2 v příloze bychom soudili, že na sadě Bacilus nástroj META nevykazuje takovou úspěšnost predikce, nástroje SNAP a MAPP jej předstihnou. Objektivní zhodnocení výkonnosti v ROC grafu na obrázku 8.3 a AUC tabulce na obrázku 8.4 však ukazují, že ona nepřesnost je pouze lokální a posouváním rozhodovacích prahů se nový klasifikátor stává minimálně rovnocenným s nejlepšími.



Obrázek 8.3: ROC charakteristiky nástrojů včetně nového klasifikátoru na sadě Bacilus.

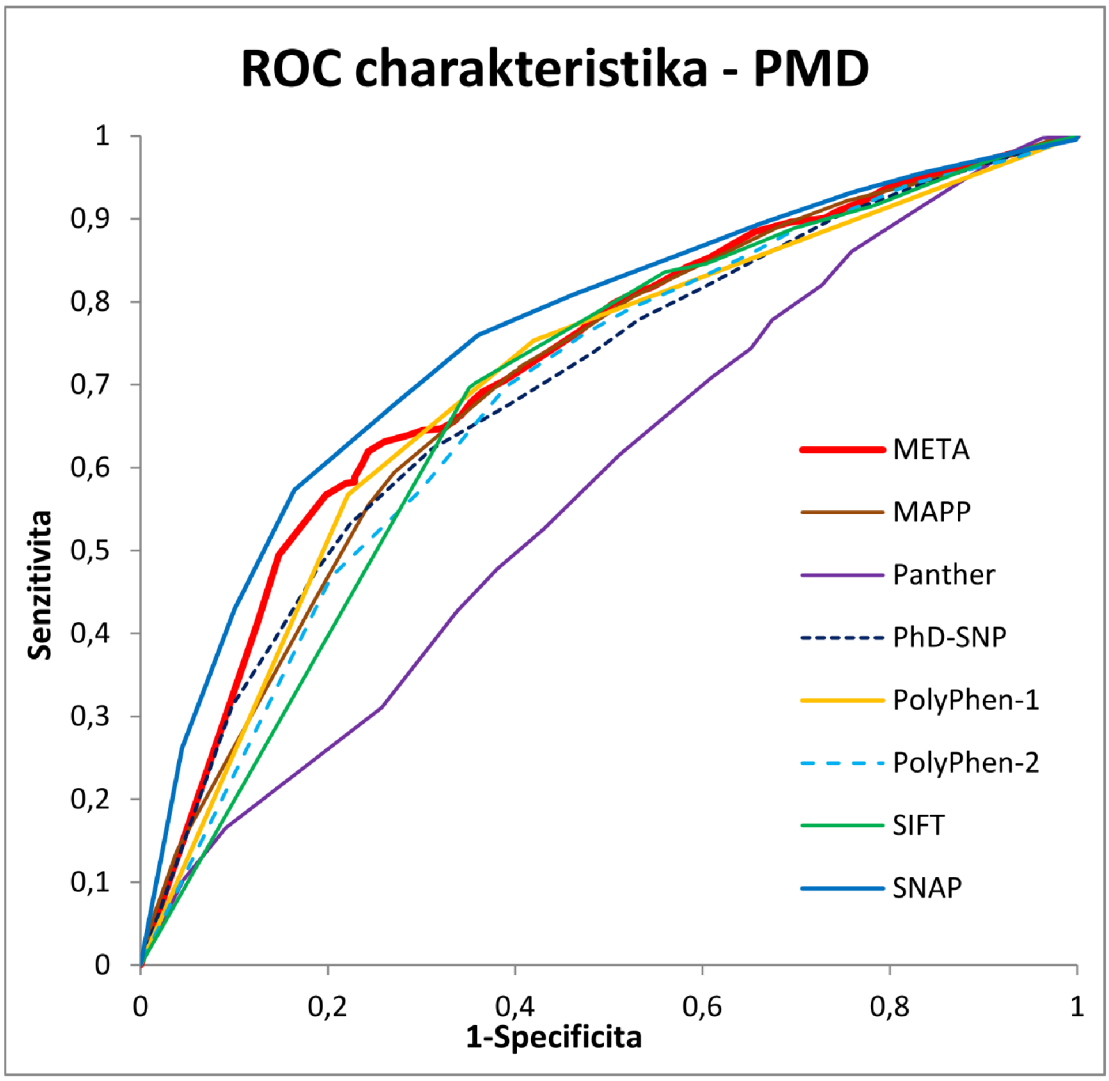
META	MAPP	nsSNPAnalyzer	PANTHER	PhD-SNP	PPH-1	PPH-2	SIFT	SNAP
0,855	0,859	0,651	0,705	0,768	0,784	0,767	0,680	0,857

Obrázek 8.4: Hodnoty ploch pod ROC křivkou (metrika AUC) na sadě Bacilus.

Sada Bacilus byla vytvořena z dat, která nebyla k dispozici pro trénování jednotlivých nástrojů, proto výsledky tohoto testování můžeme rozhodně považovat na nezávislé a do jisté míry směrodatné. Samozřejmě, výsledky nelze interpretovat jako konečné a jednoznačně správné, neboť každý z nástrojů pracuje na mírně odlišném principu a využívá jiných metod, díky čemuž se stává úspěšným jen na jistém typu mutací. Na těchto nezávislých datech byly tedy nejúspěšnější MAPP a SNAP, ovšem na jiných, rovněž nezávislých datech, mohou a typicky jsou výsledky rozdílné. Důkaz o tom podávají hodnoty metrik na druhém testovacím datasetu v následující podkapitole.

8.3.2 Testování na sadě PMD

Stejně jako v případě sady Bacilus se podle hodnot na obrázku A.3 v příloze zdá, že úspěšnost predikce META klasifikátoru značně pokulhává. V porovnání s výsledkem křížové validace v posledním sloupci tabulky A.1 je normovaná hodnota Accuracy za očekáváním. Rapidní pokles úspěšnosti je možné pozorovat u všech nástrojů, a tím tedy i u konsensu. Nejúspěšnějším nástrojem je podle obrázků 8.5 a 8.6 SNAP, který pro tuto sadu využívá, zdá se, nejefektivnějšího algoritmu a metody. Ovšem, SNAP je jeden z nástrojů, které jsou vytvářeny technikou strojového učení, je třeba se proto zaměřit na jeho trénovací sadu. SNAP byl, narozdíl od ostatních nástrojů, trénován na mutacích z databáze PMD. Je tedy zřejmé, že na podmnožině této sady bude vykazovat vyšší úspěšnost. Vzhledem k překrytí testovacího a trénovacího datasetu tak nelze tyto výsledky interpretovat jako směrodatné.



Obrázek 8.5: ROC charakteristiky nástrojů (bez nsSNPAnalyzeru) včetně nového META klasifikátoru na sadě PMD.

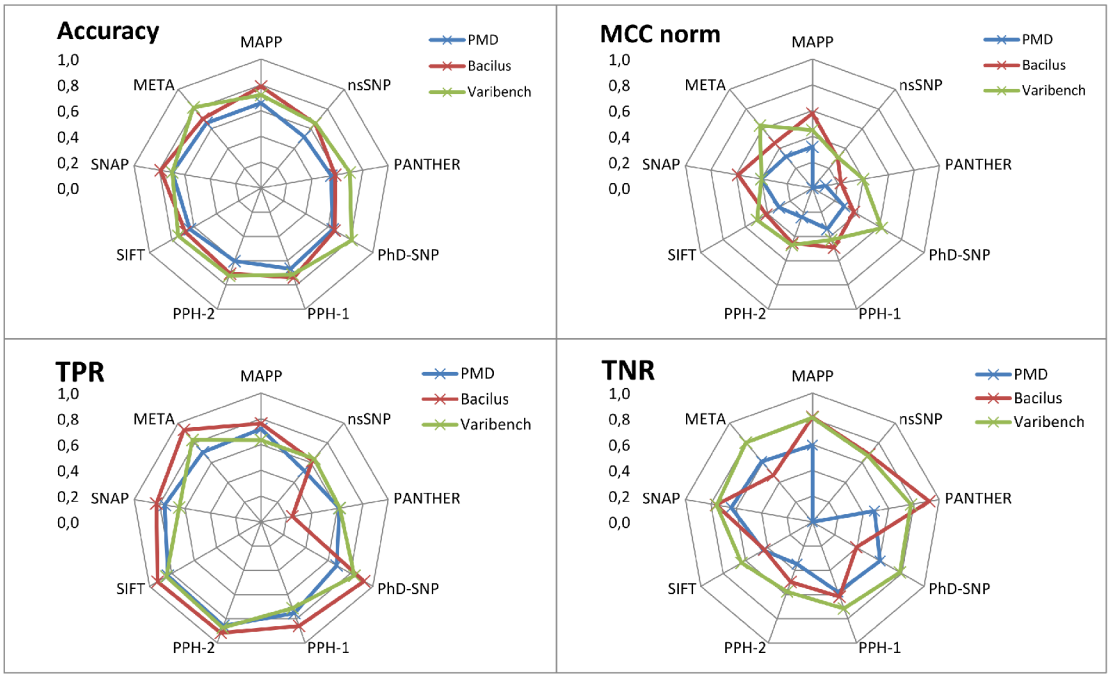
META	MAPP	PANTHER	PhD-SNP	PolyPhen-1	PolyPhen-2	SIFT	SNAP
0,723	0,704	0,577	0,696	0,702	0,687	0,686	0,759

Obrázek 8.6: Hodnoty ploch pod ROC křivkou (metrika AUC) na sadě PMD.

Při statistickém vyhodnocení výsledků a vynášením bodů do křivek ROC a posléze výpočtu hodnot AUC na obrázcích 8.5 a 8.6 jsem vypustil nástroj nsSNPAnalyzer. Údaje v tabulce A.3 totiž ukazují, že tento nástroj nebyl schopen vyhodnotit žádnou neutrální mutaci (Hodnota Cases - je rovna 0), a tedy nebylo možné stanovení dalších statistických metrik a tím pádem ani jejich vynesení do grafu. Samotná úspěšnost tohoto nástroje je v porovnání s ostatními nízká, nabízí se tedy i myšlenka úplného vyloučení nástroje z konsensu.

8.4 Analýza výsledků

Výpočetní model klasifikátoru je charakterizován některými klíčovými statistickými metrikami, proto jsem pro tento účel vytvořil přehledné grafy na obrázku 8.7, ve kterých jsou hodnoty vyneseny pro všechny nástroje a všechny testované datové sady.



Obrázek 8.7: Komplexní srovnání nástrojů a nového META klasifikátoru.

Na první pohled je zřejmé, že nástroj META na testovacích sadách nedosáhl úspěšnosti, která by převyšovala všechny dílčí nástroje. Na druhou stranu je ale nutné poukázat na hlavní výhody nového přístupu. Dílčí nástroje byly trénovány na datových sadách menšího obsahu než VariBench a tyto sady nejsou dostatečně rozmanité. Proto tyto nástroje typicky vykazují větší přesnost na množině mutací příbuzných s jejich trénovací sadou, na

jiných datech však jsou obtížně použitelné. Konsensus mezi nástroji pomáhá tyto rozdíly vyrovnat, jak můžeme konečně vidět i na samotných grafech. Na trénovací sadě byl nejúspěšnějším nástrojem PhD-SNP, proto má logicky přiděleno nejvyšší váhové ohodnocení. Na sadě Bacilus ovšem tento nástroj vykazuje jednu z nejhorších úspěšností, přesto se novému META klasifikátoru díky ostatním nenulovým vahám daří predikovat téměř o 5 % lépe.

Další výhodou konsensuálního přístupu je schopnost ohodnotit širší spektrum mutací. Stačí totiž, aby byl alespoň jeden z nástrojů schopen predikovat efekt.

Kapitola 9

Závěr

Hlavním cílem práce byl návrh a implementace nového klasifikátoru pro predikci vlivu aminokyselinových substitucí na funkci proteinu. Tento nový META klasifikátor kombinuje výsledky predikcí osmi již existujících bioinformatických nástrojů pomocí váhového konsensu.

Ke stanovení nejlepšího rozdělení vah mezi nástroji je využito evoluční strategie a strojového učení. V tomto ohledu se jedná o inovativní přístup, který se ukazuje jako efektivní. Stanovení ideálních parametrů spuštění evoluční strategie proběhlo experimentální formou a ukázalo se, že je prospěšnější spíše vícenásobně opakovat pokusy než hledat vhodnou počáteční konfiguraci. Na základě úspěšnosti vektorů vah na trénovací sadě VariBench bylo nakonec vybráno optimální rozložení vah mezi nástroje, zdokumentované v tabulce 8.2.

Ve druhé fázi byl vektor testován na nezávislých datech. K tomuto účelu byl použit reprezentativní subset databáze PMD a nově vytvořená sada, nazvaná Bacilus. Na obou sadách byl nástroj META úspěšnější, než většina dílčích nástrojů, jeho přesnost však přesto zůstala za očekáváním.

Nevýhodou nových konsenzuálních nástrojů založených na strojovém učení je použití takového trénovacího datasetu, který se obvykle pro nedostatek zdrojových dat překrývá s trénovacími sadami dílčích nástrojů. Díky tomu je natrénovaný vektor vah částečně vychýlen a nelze tedy najít skutečně optimální rozložení. Řešením by bylo vybudování nezávislé trénovací sady, která by se nepřekrývala s trénovacími sadami jednotlivých nástrojů.

Hlavní výhodou vyvinutého META nástroje je stabilita poskytovaných výsledků. Z provedených experimentů shrnutých na obrázku 8.7 je zřejmé, že oproti integrovaným nástrojům vykazuje menší výkyvy ve výkonnosti na jednotlivých datových sadách. Dalším kladným rysem je fakt, že díky integraci více nástrojů zvládne predikovat škodlivost i pro mutace, pro které část nástrojů selhává.

Literatura

[1] Adzhubei, I. A.; Schmidt, S.; Peshkin, L.; aj.: A method and server for predicting damaging missense mutations. *Nature Methods*, ročník 7, 2010: s. 248–249.

[2] Alberts, B.; Bray, D.; Johnson, A.; aj.: *Základy buněčné biologie*. Garland Publishing, 1998, iSBN 80-902906-2-0.

[3] Apweiler, R.; Martin, M. J.; O'onovan, C.; aj.: Reorganizing the protein space at the Universal Protein Resource (UniProt). *Nucleic Acids Res.*, ročník 40, Jan 2012: s. 71–75.

[4] Baldi, P.; Brunak, S.: *Bioinformatics: The Machine Learning Approach, Second Edition (Adaptive Computation and Machine Learning)*. A Bradford Book, 2001, iSBN 026202506X.

[5] Bao, L.; Zhou, M.; Cui, Y.: nsSNPAnalyzer: identifying disease-associated nonsynonymous single nucleotide polymorphisms. *Nucleic Acid Research*, ročník 33, 2005: s. 480–482.

[6] Bendl, J.; Zendulka, J.: Integration System for Functional Annotation of Single Nucleotide Polymorphism. *ElectroScope*, ročník 2012, č. 5, 2012: str. 5, ISSN 1802-4564.

[7] Beyer, H. G.: Toward a theory of evolution strategies: Self-adaptation. *Evol. Comput.*, ročník 3, 1995: s. 311–347.

[8] Branden, C.; Tooze, J.: *Introduction to Protein Structure*. Garland Publishing, 1998, iSBN 0815323050.

[9] Bromberg, Y.; Rost, B.: SNAP: predict effect of non-synonymous polymorphisms on function. *Nucleic Acid Research*, ročník 35, 2007: s. 3823–3835.

[10] Cappriotti, E.; Fariselli, P.; Calabrese, R.; aj.: Predicting protein stability changes from sequences using support vector machines. *Nucleic Acid Research*, ročník 21, 2005: s. 54–58.

[11] Capriotti, E.; Nehrt, N. L.; Kann, M. G.; aj.: Bioinformatics for personal genome interpretation. *Brief. Bioinformatics*, ročník 13, č. 4, Jul 2012: s. 495–512.

[12] Cvrčková, F.: Jak se čtou genomy: bioinformatika jakožto obor na pomezí biologie a exaktních věd. *Pokroky matematiky, fyziky a astronomie*, ročník 51, 2006: s. 288–300.

[13] Hamosh, A.; Scott, A. F.; Amberger, J. S.; aj.: Online Mendelian Inheritance in Man (OMIM), a knowledgebase of human genes and genetic disorders. *Nucleic Acids Research*, ročník 33, Jan 2005: s. 514–517.

[14] Kumar, P.; Henikoff, S.; Ng, P. C.: Predicting the effects of coding non-synonymous variants on protein function using the SIFT algorithm. *Nat Protoc*, ročník 4, č. 7, 2009: s. 1073–1081.

[15] Kvasnička, V.; Pospíchal, J.; Tiňo, P.: *Evolučné algoritmy*. STU Bratislava, 2000, iSBN 80-227-1377-5.

[16] Larkin, M. A.; Blackshields, G.; Brown, N. P.; aj.: Clustal W and Clustal X version 2.0. *Bioinformatics*, ročník 23, č. 21, Nov 2007: s. 2947–2948.

[17] Mařík, V.; Štěpánková, O.; Lažanský, J.; aj.: *Umělá inteligence (3)*. Academia, 2001, iSBN 80-200-0472-6.

[18] Meyer-Nieberg, S.; Beyer, H. G.: Self-Adaptation in Evolutionary Algorithms. *Studies in Computational Intelligence*, ročník 54, 2007: s. 47–75.

[19] Nair, P. S.; Vihinen, M.: VariBench: a benchmark database for variations. *Human Mutation*, ročník 34, č. 1, Jan 2013: s. 42–49.

[20] Nečas, O.; aj.: *Obecná biologie pro lékařské fakulty*. H&H, 2000, ISBN 80-86022-46-3.

[21] Ng, P. C.; Henikoff, S.: Predicting the effects of amino acid substitutions on protein function. *Annual Review of Genomics Human Genetics*, ročník 7, 2006: s. 61–80.

[22] Powers, D. M. W.: Evaluation: From Precision, Recall and F-measure to ROC, Informedness, Markedness and Correlation. *Machine Learning Technologies*, ročník 2, 2011: s. 37–63.

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

[24] Souter, F. P.; Magennis, J. E.; Ward, S. G.; aj.: Compositions and methods comprising serine protease variants. 2013.

[25] Stenson, P. D.; Ball, E. V.; Mort, M.; aj.: Human Gene Mutation Database (HGMD): 2003 update. *Human Mutation*, ročník 21, č. 6, 2003: s. 577–581.

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

[27] Sunyaev, S.; Ramensky, V.; Koch, I.; aj.: Prediction of deleterious human alleles. *Hum. Mol. Genet.*, ročník 10, č. 6, Mar 2001: s. 591–597.

[28] Thomas, P. D.; Campbell, M. J.; Kejariwal, A.; aj.: PANTHER: A Library of Protein Families and Subfamilies Indexed by Function. *Genome Research*, ročník 13, 2003: s. 2129–2141.

[29] Thusberg, J.; Vihinen, M.: Pathogenic or not? And if so, then how? Studying the effects of missense mutations using bioinformatics methods. *Human Mutation*, ročník 30, 2009: s. 703–714.

[30] Vihinen, M.: How to evaluate performance of prediction methods? Measures and their interpretation in variation effect analysis. *BMC Genomics*, ročník 13 Suppl 4, 2012.

Příloha A

Tabulky s výsledky testů

	Nástroje								Konsensuální META nástroj			
	MAPP	nsSNP	PANTHER	PhD-SNP	PPH-1	PPH-2	SIFT	SNAP	all	|score|>0.1	|score|>0.25	cross val.
TP	8682	4450	4739	11934	10162	12476	10259	9171	11843	11603	11438	N/A
TPR	0,634	0,640	0,619	0,836	0,712	0,873	0,849	0,645	0,829	0,843	0,858	N/A
FN	5007	2499	2920	2344	4106	1807	1826	5039	2442	2157	1889	N/A
FNR	0,366	0,360	0,381	0,164	0,288	0,127	0,151	0,355	0,171	0,157	0,142	N/A
TN	12250	1231	9285	13366	12201	9971	8655	12964	13930	13438	12809	N/A
TNR	0,808	0,669	0,776	0,783	0,716	0,576	0,636	0,754	0,803	0,810	0,809	N/A
FP	2906	608	2674	3694	4851	7352	4962	4228	3409	3154	3026	N/A
FPR	0,192	0,331	0,224	0,217	0,284	0,424	0,364	0,246	0,197	0,190	0,191	N/A
Cases +	13689	6949	7659	14278	14268	14283	12085	14210	14285	13760	13327	N/A
Cases -	15156	1839	11959	17060	17052	17323	13617	17192	17339	16592	15835	N/A
Accuracy	0,726	0,646	0,715	0,807	0,714	0,710	0,736	0,705	0,815	0,825	0,831	N/A
Accuracy norm	**0,721**	**0,655**	**0,698**	**0,810**	**0,714**	**0,725**	**0,742**	**0,700**	**0,816**	**0,827**	**0,834**	**0,818**
MCC	0,451	0,255	0,398	0,617	0,426	0,462	0,492	0,402	0,630	0,651	0,665	N/A
MCC norm	**0,449**	**0,310**	**0,400**	**0,620**	**0,428**	**0,470**	**0,496**	**0,402**	**0,633**	**0,654**	**0,668**	N/A

Obrázek A.1: Úplné výsledky testů na trénovací sadě VariBench.

	Nástroje								Konsensuální META nástroj			
	MAPP	nsSNP	PANTHER	PhD-SNP	PPH-1	PPH-2	SIFT	SNAP	all	|score|>0.1	|score|>0.25	cross val.
TP	1270	992	263	1533	1431	1526	1545	1375	1552	1516	1485	N/A
TPR	0,763	0,618	0,246	0,921	0,859	0,917	0,928	0,826	0,932	0,949	0,952	N/A
FN	394	612	808	131	234	139	120	290	113	82	75	N/A
FNR	0,237	0,382	0,754	0,079	0,141	0,083	0,072	0,174	0,068	0,051	0,048	N/A
TN	2064	1544	1941	1000	1565	1257	1092	1920	1196	939	887	N/A
TNR	0,814	0,684	0,920	0,394	0,617	0,496	0,431	0,758	0,472	0,447	0,441	N/A
FP	471	712	168	1535	970	1278	1443	614	1339	1161	1124	N/A
FPR	0,186	0,316	0,080	0,606	0,383	0,504	0,569	0,242	0,528	0,553	0,559	N/A
Cases +	1664	1604	1071	1664	1665	1665	1665	1665	1665	1598	1560	N/A
Cases -	2535	2256	2109	2535	2535	2535	2535	2534	2535	2100	2011	N/A
Accuracy	0,794	0,657	0,693	0,603	0,713	0,663	0,628	0,785	0,654	0,664	0,664	N/A
Accuracy norm	**0,789**	**0,651**	**0,583**	**0,658**	**0,738**	**0,706**	**0,679**	**0,792**	**0,702**	**0,698**	**0,696**	**N/A**
MCC	0,573	0,301	0,229	0,348	0,471	0,428	0,387	0,572	0,427	0,439	0,439	N/A
MCC norm	**0,578**	**0,304**	**0,225**	**0,371**	**0,491**	**0,455**	**0,413**	**0,585**	**0,455**	**0,458**	**0,457**	**N/A**

Obrázek A.2: Úplné výsledky testů na testovací sadě Bacilus.

	Nástroje								Konsensuální META nástroj			
	MAPP	nsSNP	PANTHER	PhD-SNP	PPH-1	PPH-2	SIFT	SNAP	all	|score|>0.1	|score|>0.25	cross val.
TP	620	84	216	588	654	750	677	663	617	594	578	N/A
TPR	0,721	0,519	0,615	0,676	0,753	0,858	0,836	0,759	0,705	0,727	0,751	N/A
FN	240	78	135	282	214	124	133	210	258	223	192	N/A
FNR	0,279	0,481	0,385	0,324	0,247	0,142	0,164	0,241	0,295	0,273	0,249	N/A
TN	479	0	237	493	475	283	252	522	501	456	422	N/A
TNR	0,596	0,000	0,489	0,606	0,580	0,349	0,440	0,640	0,612	0,612	0,604	N/A
FP	325	0	248	320	344	529	321	294	318	289	277	N/A
FPR	0,404	0,000	0,511	0,394	0,420	0,651	0,560	0,360	0,388	0,388	0,396	N/A
Cases +	860	162	351	870	868	874	810	873	875	817	770	N/A
Cases -	804	0	485	813	819	812	573	816	819	745	699	N/A
Accuracy	0,660	0,519	0,542	0,642	0,669	0,613	0,672	0,702	0,660	0,672	0,681	N/A
Accuracy norm	**0,658**	**0,519**	**0,552**	**0,641**	**0,667**	**0,603**	**0,638**	**0,700**	**0,658**	**0,670**	**0,677**	**N/A**
MCC	0,319	0,000	0,103	0,283	0,339	0,241	0,303	0,403	0,318	0,342	0,359	N/A
MCC norm	**0,319**	**0,000**	**0,105**	**0,283**	**0,339**	**0,240**	**0,300**	**0,402**	**0,318**	**0,341**	**0,358**	**N/A**

Obrázek A.3: Úplné výsledky testů na testovací sadě PMD.

Příloha B

Obsah CD

/bacilus

V této složce se nachází zdrojový soubor *01.csv* s tabulkou zdrojových mutací pro dataset Bacilus. V souboru *01_seq.docx* je pak k těmto mutacím proteinová sekvence. Skript *01.py* kontroluje pozici mutací a rozřazuje je do skupin na neutrální, škodlivé a nedefinované.

/cross_validace

Tato složka obsahuje ohodnocený trénovací dataset VariBench a skript, který na této sadě provádí křížovou validaci.

/dataset

Tato složka obsahuje všechny tři používané ohodnocené datasety, VariBench, Bacilus i PMD.

/doc

V této složce se nachází úplná programová dokumentace.

/evolucni_strategie

Tato složka obsahuje ohodnocený dataset VariBench a skript *es.py*, který nalezne s pomocí evoluční strategie nejúspěšnější vektor vah. V souboru *vektor_vah* se nachází vektor, nalezený při experimentování a testovaný v rámci kapitoly 8.

/tabulky_a_grafy

V této složce se nachází soubor *vysledek.xslx* s výslednými tabulkami a grafy.

/testy

Složka obsahující skripty pro analýzu výsledků a sběr dat pro vytvoření tabulek a grafů. Nachází se zde rovněž všechny ohodnocené datasety a podsložka */data*, do které jsou statistické hodnoty uloženy.

/latex

Složka obsahuje zdrojové soubory pro vytvoření tohoto dokumentu včetně obrázků.

Příloha C

Návod ke spuštění

Skript *es.py* při spuštění bez parametrů vypíše nápovědu. Pro spuštění evoluční strategie je nutné zadat šest parametrů:

python es.py generaci potomku sigmaini sigmamin kladnevahy pokusu

Parametr	Význam a hodnota
generaci	počet generací ES, kladné celé číslo
potomku	počet potomků v generaci, kladné celé číslo
sigmaini	počáteční hodnota odchylky při mutaci, reálné číslo z <0,1>
sigmamin	minimální hodnota odchylky při mutaci, reálné číslo z <0,1>
kladvevahy	1 pro váhy v intervalu <0,1>, 0 pro váhy z <-1,1>
pokusu	počet nezávislých pokusů o nalezení řešení během jednoho spuštění

Výsledky jsou pak uloženy do souborů *vysX*, kde *X* je pořadové číslo pokusu.

Skript *cross_validace.py* je spuštěn bez parametrů a výsledek křížové validace (10 hodnot Accuracy pod sebou) zapisuje do souboru *tenfold_varibench.txt*.

python cross_validace.py

Skript *01.py* je spuštěn bez parametrů a mutace rozřazuje do souborů *01_skodlive*, *01_neutralni* a *01_nedef*.

python 01.py

Skripty ve složce */testy* jsou rovněž spuštěny bez parametrů, případná změna vektoru vah se musí provést ručně. Výsledky zapisují do textových souborů do složky *testy/data*.