VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
FACULTY OF INFORMATION TECHNOLOGY

ÚSTAV INFORMAČNÍCH SYSTÉMŮ
DEPARTMENT OF INFORMATION SYSTEMS

PREDIKCE VLIVU MUTACE NA ROZPUSTNOST PROTEINŮ

PREDICTION OF THE EFFECT OF MUTATION ON PROTEIN SOLUBILITY

DIPLOMOVÁ PRÁCE
MASTER'S THESIS

AUTOR PRÁCE — **Bc. JAN VELECKÝ**
AUTHOR

VEDOUCÍ PRÁCE — **Ing. JIŘÍ HON**
SUPERVISOR

BRNO 2020

Vysoké učení technické v Brně
Fakulta informačních technologií

Ústav informačních systémů (UIFS) Akademický rok 2019/2020

Zadání diplomové práce

Student: **Velecký Jan, Bc.**
Program: Informační technologie Obor: Bioinformatika a biocomputing
Název: **Predikce vlivu mutace na rozpustnost proteinů**
Prediction of the Effect of Mutation on Protein Solubility
Kategorie: Bioinformatika

Zadání:

1. Seznamte se s problematikou rozpustnosti proteinů a existujícími predikčními nástroji.
2. Vytvořte trénovací a testovací datovou sadu vlivu mutace na rozpustnost proteinů na základě dostupných dat.
3. Navrhněte nástroj pro predikci vlivu mutace na rozpustnost proteinů a proveďte jeho implementaci. K predikci využijte strukturu proteinu a jeho povrchový elektrostatický potenciál.
4. Činnost nástroje zhodnoťte na testovací datové sadě.
5. Zhodnoťte dosažené výsledky a diskutujte možnosti dalšího pokračování projektu.

Literatura:

- CARBALLO-AMADOR, M. Alejandro, Edward A. MCKENZIE, Alan J. DICKSON a Jim WARWICKER. Surface Patches on Recombinant Erythropoietin Predict Protein Solubility: Engineering Proteins to Minimise Aggregation. BMC Biotechnology. 2019, 19(1)
- WRENBECK, Emily E., Matthew A. BEDEWITZ, Justin R. KLESMITH, Syeda NOSHIN, Cornelius S. BARRY a Timothy A. WHITEHEAD. An Automated Data-Driven Pipeline for Improving Heterologous Enzyme Expression. ACS Synthetic Biology. 2019, 8(3), 474-481
- Dále dle pokynů vedoucího.

Při obhajobě semestrální části projektu je požadováno:

- Splnění bodů 1 a 2 zadání.

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

Vedoucí práce: **Hon Jiří, Ing.**
Vedoucí ústavu: Kolář Dušan, doc. Dr. Ing.
Datum zadání: 1. listopadu 2019
Datum odevzdání: 20. května 2020
Datum schválení: 22. října 2019

Abstrakt

Cílem práce je vytvoření prediktoru změny rozpustnosti mutovaného proteinu ze znalosti jeho původní 3D struktury. Predikce rozpustnosti proteinů je část bioinformatiky, která dosud není uspokojivě vyřešena. Přičemž především predikci ze 3D struktury není věnována náležitá pozornost.

Práce obsahuje souhrn relevantních znalostí o proteinech, jejich rozpustnosti a stávajících predikčních nástrojích. Princip navrženého prediktoru je inspirován článkem Surface Patches, a práce si tak dává za cíl také ověřit výsledky v něm dosažené. Navržená predikce funguje podle změn oblastí kladného elektrického potenciálu nad povrchem proteinu.

Nástroj byl úspěšně implementován a byla provedena celá řada výpočetně náročných experimentů. Z nich vyplynulo, že tímto způsobem lze elektrický potenciál, a tedy i prediktor, použít úspěšně jen u omezené množiny proteinů. Metoda použitá v článku navíc koreluje s mnohem jednodušší proměnou celkového náboje proteinu.

Abstract

The goal of the thesis is to create a predictor of the effect of a mutation on protein solubility given its initial 3D structure. Protein solubility prediction is a bioinformatics problem which is still considered unsolved. Especially a prediction using a 3D structure has not gained much attention yet.

A relevant knowledge about proteins, protein solubility and existing predictors is included in the text. The principle of the designed predictor is inspired by the Surface Patches article and therefore it also aims to validate the results achieved by its authors. The designed tool uses changes of positive regions of the electric potential above the protein's surface to make a prediction.

The tool has been successfully implemented and series of computationally expensive experiments have been performed. It was shown that the electric potential, hence the predictor itself too, can be successfully used just for a limited set of proteins. On top of that, the method used in the article correlates with a much simpler variable – the protein's net charge.

Klíčová slova

rozpustnost proteinů, změna rozpustnosti, predikce, rozpustnost, Surface Patches, povrchový potenciál, OptSolMut, Whitehead, dataset, PDB, predikce ze struktury, mutace, Modeller, FoldX, elektrický potenciál, agregace, kladná oblast

Keywords

protein solubility, change of solubility, prediction, solubility, Surface Patches, surface potential, OptSolMut, Whitehead, dataset, PDB, prediction by structure, mutation, Modeller, FoldX, electrostatic potential, aggregation, positive region

Citace

VELECKÝ, Jan. *Predikce vlivu mutace na rozpustnost proteinů.* Brno, 2020. Diplomová práce. Vysoké učení technické v Brně, Fakulta informačních technologií. Vedoucí práce Ing. Jiří Hon

Predikce vlivu mutace na rozpustnost proteinů

Prohlášení

Prohlašuji, že jsem tuto diplomovou práci vypracoval samostatně pod vedením Ing. Jiřího Hona. Uvedl jsem všechny literární prameny, publikace a další zdroje, ze kterých jsem čerpal.

.....
Jan Velecký
3. června 2020

Poděkování

Rád bych tímto poděkoval svému vedoucímu, Ing. Jiřímu Honovi, za odborné vedení a poskytnuté rady při psaní této práce.

Výpočetní zdroje byly dostupné díky projektu „e-Infrastruktura CZ“ (e-INFRA LM2018140) v rámci programu Velké infrastruktury pro výzkum, vývoj a inovace.

Obsah

Kapitola 1

Úvod

Proteiny jsou biologické, z aminokyselin skládající se, makromolekuly, ze kterých jsou postaveny všechny živé organismy. Plní stavební funkci – např. kolagen v pokožce nebo ve vlasech. Ale neslouží v organismech jen jako stavební materiál, mají i další, aktivnější, funkce. Kupříkladu plní pohybovou funkci. Tandem proteinů aktin a myosin umožňuje svalové stahy. Další plní transportní funkci. Příkladem budiž hemoglobin, který v krevním řečišti transportuje z plic molekuly kyslíku do tkání. Jiné proteiny ostatní makromolekuly a procesy řídí, regulují či katalyzují – jsou to enzymy, hormony a další. V lidském těle se nachází desetitisíce takových proteinů!

Mnoho nemocí a jiných obtíží je způsobeno právě nedostatečnou, přebytečnou, či špatnou produkcí proteinů v buňkách jedince. Jsou to problémy, které se dají řešit medikamentózní terapií – tedy podáváním léku pacientovi. Léčiva se dnes obvykle vyrábějí chemicky ve zkumavkách. Je-li ale léčivo nějaký protein, mnohem častěji se využívá biologická syntéza daného proteinu v cizorodém organismu. Jinými slovy, přinutíme bakterii, či jiný organismus vyrábět pro nás lidský protein, který se potom může podávat pacientům. Většinou se jedná o nějaký hormon, nebo enzym. První protein uměle vyrobený člověkem byl hormon inzulín, který udržuje stabilní hladinu krevního cukru a podává se pacientům trpícím cukrovkou. Nyní se obvykle vyrábí v bakterii známé jako *E. coli*. Jiný případ je hormon erytropoetin, který reguluje výrobu červených krvinek. Podává se jako medikament při krevní nedostatečnosti a jindy jako doping.

Ačkoliv je výroba proteinů již běžná činnost farmaceutických firem, její zlevnění je problematické. Protein se musí produkovat v buňce v malém množství, jinak se naráží na limity jeho rozpustnosti. Při vyšší koncentraci se začne shlukovat do agregátů či lepit na jiné molekuly anebo má poškozenou funkci. Tím se sníží jeho využitelnost nebo ho je obtížné extrahovat. To se nyní řeší různými chemickými/fyziologickými postupy – přidávání solí, manipulace s teplotou roztoku, různé metody odstředění...

Nyní se však pozornost upírá k jinému přístupu. Od roku 2009 vznikají první nástroje predikující rozpustnost proteinů z posloupnosti jejich aminokyselin. Účel těchto nástrojů je použití pro návrh takových změn v proteinech, které jejich rozpustnost vylepší a funkčnost nepoškodí. A doposud v této oblasti neexistují uspokojivé metody a nástroje.

Tato práce se zaměřuje na prověření a implementaci jedné z nejnovějších metod pro predikci změny rozpustnosti nazvanou Surface Patches, která se zaměřuje na vnější elektrické pole proteinu. Vznikl nástroj, jenž by měl predikovat změnu rozpustnosti mutací pozměněného proteinu podle jeho posloupnosti aminokyselin a 3D struktury.

Text práce je strukturován do 7 kapitol. V následující, druhé kapitole je obsažen teoretický úvod do molekulární biologie, resp. do proteinů a jejich struktury. Třetí kapitola

rozebírá přímo problém, jímž se tato práce zabývá především, a to je problematika rozpustnosti. A popisuje i některé existující prediktory rozpustnosti, datové sady a hlavně samotnou metodu Surface Patches. Čtvrtá je návrhem samotného prediktoru podle již uvedené metody. Pátá popisuje ovládání výsledného prediktoru, rozebírá jeho implementaci, v neposlední řadě také implementaci experimentů s tímto nástrojem a nakonec postup vytváření ilustrací k této práci. Experimentování s touto metodou a vytvořeným prediktorem jsou v kapitole šesté spolu s analýzou získaných dat a prověřením metody jako takové. Tedy kapitola se zabývá i tím, zda neexistuje jednodušší možnost predikce se stejnou účinností. Rešerše toho, co bylo v rámci práce vykonáno, jakých výsledků bylo dosaženo, jaké jsou alternativní využití prediktoru a návaznosti na tuto práci spolu s celkovým zhodnocením této práce je v závěru.

Kapitola 2

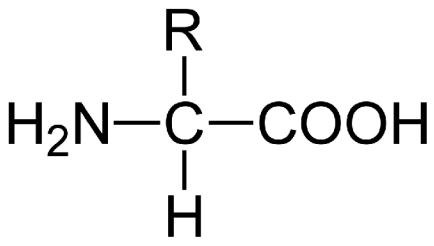
Protein a jeho rozpustnost

Tato kapitola slouží jako krátký úvod do *molekulární biologie*, jež se zabývá biologickými procesy a látkami na úrovni molekul. Obsahem jsou známá fakta o této problematice. V jednotlivých podkapitolách zjednodušeně vysvětluje: co je to *protein*, z čeho se skládá a jakou má strukturu; mechanismus vzniku proteinu – *syntéza*; a *rozpustnost proteinu*[1]. Kapitola využívá především informace z knihy [1].

2.1 Protein a jeho struktura

Protein (také *bílkovina*) je biologická makromolekula, která slouží jako základní stavební i funkční jednotka živých organismů. Jedná se o přírodní polymer, jehož podjednotkami jsou *aminokyseliny* (často zkracováno jako *AK*).

Existuje celá řada aminokyselin, nicméně obvykle se dělí na 22 druhů. Každá z nich má kromě svého celého jména také přiřazen *3písmenný* a *1písmenný kód*, zkráceně jako *3LC* a *1LC* (z angl. three- a one-letter code). Např. aminokyselina celým jménem „kyselina asparagová“ má kódy **Asp** a **D**. Aminokyseliny se skládají, až na výjimky, pouze z atomů vodíku, uhlíku, kyslíku a dusíku. Obecná struktura aminokyseliny je na obrázku 2.1. **R** (z angl. residue) označuje *postranní řetězec* aminokyseliny (také *zbytek*) – ten je odpovědný za to, že různé aminokyseliny mají různé vlastnosti. Ve vodě dochází k ionizaci skupin na obou koncích, což umožňuje spojení 2 a více aminokyselin v *peptid*. Řetězec opakující se sekvence N-konce, centrálního uhlíku (tzv. C-α) a C-konce v peptidu se nazývá *kostra* nebo páteř.



Obrázek 2.1: [2] *Chemický vzorec obecné aminokyseliny.* Horizontální vazby tvoří páteř proteinu vlevo s tzv. N-koncem (aminoskupina) a vpravo s C-koncem (karboxylová skupina).

[1] O tomto pojmu bude nadále referováno pouze jako o *rozpustnosti*.

Postranní řetězce, a v důsledku i jejich aminokyseliny, se liší především těmito několika vlastnostmi:

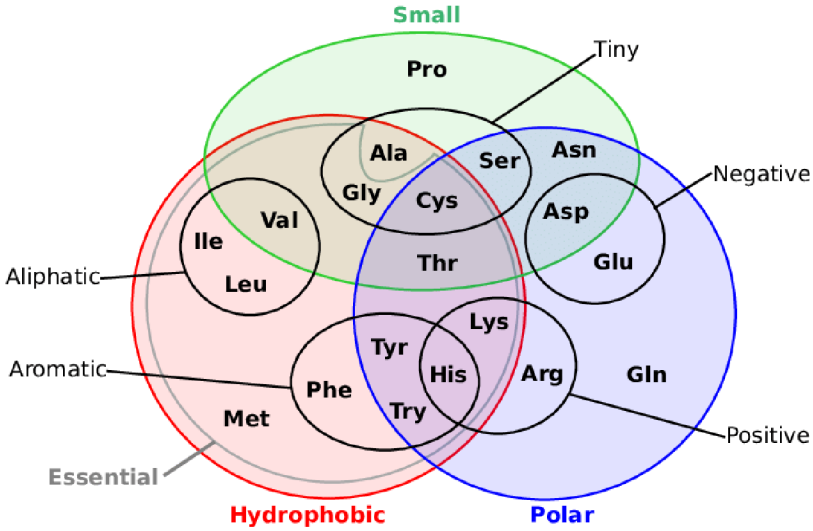
1. Polarita – nepolární, polární, nabité – viz dále
2. Interakce s vodou – hydrofobnost nebo hydrofilnost
3. Fyzická velikost - např. glycin (Gly/G), nejmenší aminokyselina, fakticky žádný postranní řetězec nemá
4. Přítomnost určité funkční skupiny – např. skupiny obsahující síru

Každá aminokyselina je v její nominální formě bez náboje (neobsahuje žádný ion). Prakticky u nich však, nacházejí-li se ve vodě, dochází k *ionizaci* (elektronizaci nebo protonizaci), kdy získávají náboj z roztoku. To je způsobeno tím, že voda je tzv. *polární rozpouštědlo*. Atom vodíku a kyslíku jsou v molekule vody vázány *kovalentní vazbou* – sdílejí elektron ve svých elektronových obalech. Tato vazba je ale *polární*, protože elektron je přitahován silněji atomem kyslíku nežli vodíku, proto se i povětšinou nachází v el. obalu kyslíku. Tím vzniká *elektrický dipól*, kdy *částečný záporný náboj* se nachází na kyslíku a *částečný kladný náboj* na vodíku. Molekula s dipólem se nazývá *polární* a znamená to, že je ovlivňována elektrostatickými silami. Polární molekula také ochotně reaguje za vzniku iontových vazeb. Voda díky tomu disociuje vodík, stejně tak jako ho navazuje iontovou vazbou. A to za vzniku hydroxidového anionu OH^- nebo hydroxoniového kationu H_3O^+. Pokud látka zvyšuje koncentraci H_3O^+, jedná se o *kyselinu*. Pokud naopak zvyšuje koncentraci OH^-, jedná se o *zásadu*. Za normálních okolností zvýšení koncentrace jednoho z ionů vede ke snížení koncentrace druhého a naopak. Kyselost roztoku se měří logaritmickou jednotkou *pH*, jež udává koncentraci H^+ (H_3O^+) v roztoku. Neutrální roztok má pH = 7, kyselý < 7 a zásaditý > 7. V buňce je udržováno pH kolem hodnoty 7,4.

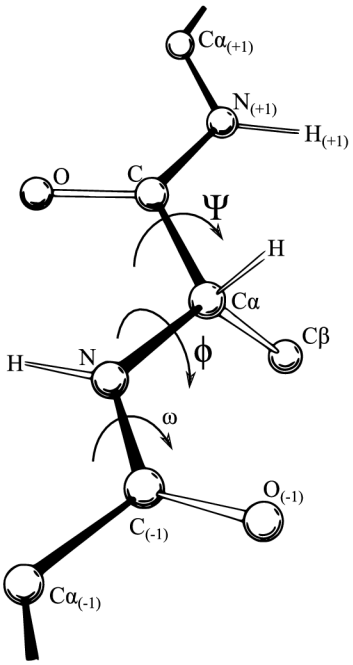
Aminokyseliny, které obsahují ve svém postranním řetězci funkční skupiny mohou být ve vodě nabité. Aminokyselina má kyselý zbytek, nachází-li se v něm karboxylová skupina (–COOH). Ve vodě dochází k elektronizaci takové aminokyseliny, neboť H^+ se disociuje od skupiny, zůstává $–COO^-$ a vzniká voda (reakcí s hydroxidovým anionem), nebo hydroxionový kation (reakcí s vodou). *Kyselé aminokyseliny* se tak stávají záporně nabitými – jsou to kyselina asparagová a glutamová. Naopak, nachází-li se ve zbytku aminoskupina ($–NH_2$), jedná se o *zásaditou aminokyselinu*. Dochází u ní k protonizaci přidružením vodíkového kationu ($–NH_3^+$) z roztoku (z vody nebo hydroxionového kationu). Zásaditá aminokyselina tak získává kladný náboj. Zásadité (záporné) aminokyseliny jsou lysin, arginin a histidin. Čím kyselejší je roztok, tím ochotněji zásady přijímají H^+ z roztoku (a tedy snižují kyselost). A naopak, čím zásaditější je roztok, tím ochotněji jej kyseliny uvolňují. Důležité je v tomto bodě poznamenat, že se nejedná o permanentní stav a v roztoku neustále probíhají asociace a disociace a atomy vodíku „cestují“ po molekulách obsažených v roztoku. Proto nabité aminokyselině nelze přiřadit náboj bez znalosti pH roztoku, neboť v nabitém stavu se nachází pouze po část doby, a to v závislosti na této hodnotě.

Pro získání obrazu o tom, kterými vlastnostmi disponují které základní aminokyseliny, nahlédněte na obrázek 2.2.

Protein (peptid) lze strukturně popsat na několika úrovních. Nejzákladnější je *primární struktura*, což je sekvence aminokyselin, které jej tvoří. Například sekvence `Val-Gly-Ala` (nebo zkráceně `VGA`) je tripeptid tvořený aminokyselinami Valin, Glycin a Alanin v tomto pořadí od N-konce. Velmi dlouhé peptidy čítající desítky až stovky aminokyselin se nazývají *proteiny*.



Obrázek 2.2: *Vennův diagram zachycující dělení 20 aminokyselin* do skupin podle 9 vlastností, podle toho zda jimi disponují. Vidíme kupř., že glycin (Gly) je drobná nepolární (hydrofobní) molekula, kdežto kys. asparagová (Asp) je nevelká záporně nabitá (tedy i polární) molekula. Převzato z [3].



Obrázek 2.3: [4] *Kostra proteinu.* Jedna z variant zobrazení primární struktury je strukturní vzorec. V něm ale není zachycené uspořádání molekuly v prostoru. Na obrázku je možno vidět tzv. *torzní úhly* ψ, ϕ (a také ω), které otáčí rovinami definovanými atomy (C-α, C, N_{+1}) pro ψ a rovinou definovanou (C_{-1}, N, C-α) pro ϕ kolem osy znázorněných vazeb. C-β je 1. uhlík v postranním řetězci.

Samotná primární struktura tvořená vazbami kostry nechává velkou volnost a části molekul můžou volně rotovat podle os tvořených vazbami mezi `N` a `C`-α (úhel ϕ), a `C`-α a `C` (úhel ψ). Schéma těchto úhlů na kostře peptidu je na obrázku 2.3. Primární struktura však neodráží skutečné prostorové uspořádání proteinu – protein ve skutečnosti zaujímá kompaktnější, složenou formu a ta je dána vyššími strukturami.

Sekundární struktura tvar molekuly omezuje vytvořenými vodíkovými můstky na kostře mezi aminovou skupinou jedné aminokyseliny a hydroxylovou skupinou jiné aminokyseliny. V proteinu vznikají struktury α-*helix* a β-*list*. První jmenovaná má tvar šroubovice a druhá skládaného listu.

Ještě na vyšší úrovni je *terciární struktura*. Ta tvar omezuje interakcemi způsobenými postranními řetězci aminokyselin. Jedná se opět zejména o vodíkové můstky anebo disulfidické vazby. Představuje kompletní prostorové uspořádání proteinu, neboli *konformaci*. Protein zaujímá takovou konformaci, která je pro něj energeticky nejvýhodnější.

Kvarterní struktura popisuje strukturu složenou z více než jednoho peptidového řetězce. Protein, jenž má kvarterní strukturu, se nazývá *proteinový komplex*. Podjednotky (monomery) drží pohromadě stejnými silami jako v terciální struktuře. Tyto proteiny můžou být *dimerní* (sestávajíce z 2 proteinů), trimerní (z 3), atd. Kromě toho se proteinové komplexy dělí podle počtu druhů podjednotek na *homomery* (pouze z 1 druhu proteinu) a *heteromery* (z vícero). Na obrázku 2.4 je ilustrace od primární struktury až po kvarterní strukturu trimerního komplexu. Každý z monomerů komplexu je zároveň ligandem. *Ligand* je jakákoliv látka, která je vázána k proteinu. Místo styku mezi ligandem a proteinem se nazývá *vazební místo*. Každý monomer na obrázku má dvě tato místa.

Poznatky o struktuře jsou velmi důležité, neboť strukturou je dána funkce proteinu. A ačkoliv primární struktura předurčuje i vyšší struktury, obvykle ji pouze ze znalosti primární struktury nedokážeme předpovědět. A pokud ano, je to velmi výpočetně náročné. Proto existuje databáze **PDB** (*Protein Data Bank*), jakožto svobodná databáze proteinů s jejich, již experimentálně zjištěnými, 3D strukturami. Databázi je možno prohlížet ve webovém prohlížeči např. přes stránky[2] organizace EBI[3]. Struktury proteinů je možné stahovat v strojově zpracovatelném textovém formátu PDB. Každý protein je označen svým *PDB ID*, které sestává ze 4 alfanumerických znaků. Např. PDB ID `3I40` označuje lidský inzulín.

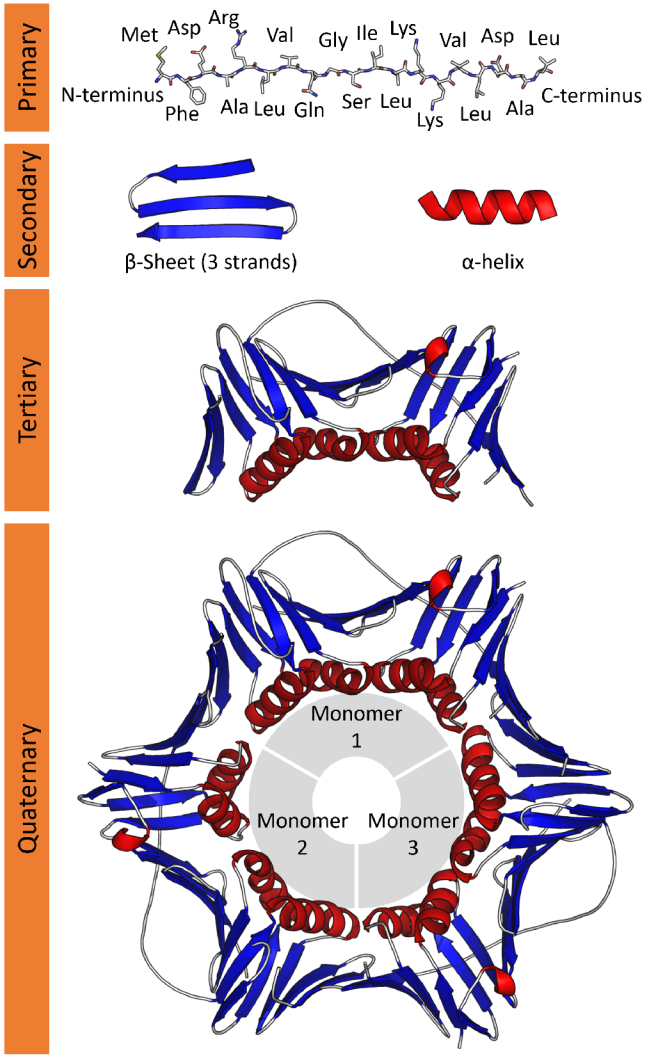
Struktury proteinů jsou obvykle získány experimentálně (*strukturní experiment*), občas ale také modelováním, neboli predikcí. Při experimentech se fyzicky analyzuje daný protein a výsledná data jsou konfrontována s předem známými daty a fakty – jeho primární struktura, obvyklé konformace aminokyselin, ap.

Nejčastějším způsobem strukturálního experimentu je *rentgenová krystalografie*, kdy je analyzovaný protein extrahován a vysušen. Poté je vyšetřován rentgenovými paprsky, pomocí kterých je zjištěna hustota elektronů v prostoru proteinu. Těmto datům je možno na základě známé primární struktury přiřadit jednotlivé atomy, tak aby byli dodrženy fyzikální zákony, čímž získáme 3D strukturu. Nevýhodou metody je, že flexibilní část proteinu je pro analýzu neviditelná a je důvodem chybějících souřadnic v datech PDB u některých proteinů. Dále, vysušený protein neprojevuje svůj náboj, či hydrofobnost a nemusí tedy vystihovat přesně konformaci v přirozeném prostředí buňky. Převážná část modelů v PDB je získaná touto metodou. Data získaná touto metodou jsou ilustrována na obrázku 2.5.

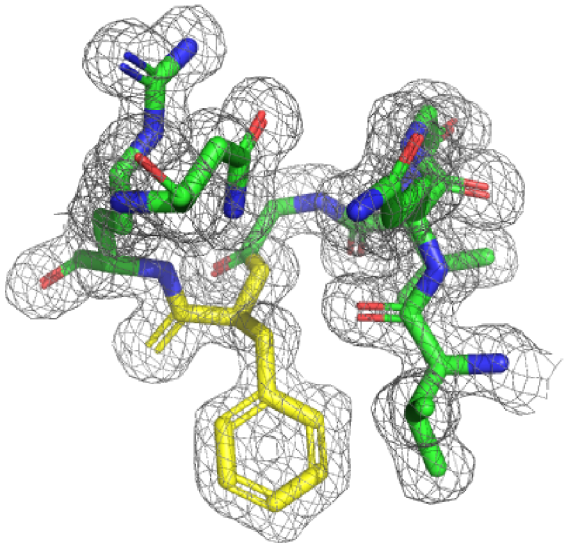
Dalším typem strukturálního experimentu je *NMR spektroskopie*, kdy se protein v roztoku vyšetřuje v magnetickém poli pomocí rádiových vln. Výsledkem je seznam atomů,

[2] https://www.ebi.ac.uk/pdbe/

[3] Evropský institut pro bioinformatiku



Obrázek 2.4: [5]
Znázornění primární až kvartérní struktury proteinového komplexu PCNA (PDB: 1AXC).
V primární struktuře vidíme peptidovou sekvenci zapsanou třípísmennými kódy aminokyselin.
V sekundární struktuře znázornění β-listu a α-šroubovice.
Terciární struktura zobrazuje prostorové uspořádání monomeru.
Kvartérní struktura zachycuje kompletní uspořádání komplexu. Všimněte si, že každý monomer má stejnou terciální strukturu a je pouze jinak natočený.
Všechny vyšší struktury než primární zobrazují na obrázku pouze kostry proteinů bez postranních řetězců. Ty jsou však ve skutečnosti odpovědné za výslednou terciární a kvarterní strukturu. Kostra je zobrazena pomocí barevných stuh – takovému zobrazení proteinu se říká *stužkový model*.



Obrázek 2.5: *Mapa hustoty elektronů* pro vybranou část proteinu (PDB: 1W2I), získaná rentgenovou krystalografií. Na obrázku je šedou sítí zobrazena izoplocha představující místa s konstantní úrovní hustoty v mapě. Strukturu získáme analýzou, při které se do sítě snažíme umístit atomy tvořícími daný protein. Nejnázorněji je to vidět na cyklické struktuře aminokyseliny fenylalaninu (zvýrazněn žlutě), jež je typická pro aromatickou skupinu v něm obsaženou.

které se vždy nacházejí poblíž sebe. U malých proteinů je to dostatečné na získání struktury. A v takovém případě se jedná o preferovanou metodu, protože takto získaná struktura nejlépe odpovídá skutečnosti.

Třetí nejčastější způsob strukturálního modelování je *3D elektronovou mikrosskopií* (3DEM). Dochází k vytváření tisíců snímků, které jsou poté výpočetně spojeny do strukturního modelu. Pochopitelně je také možné zkombinovat data z vícero experimentů pro získání přesnějšího modelu.

Avšak v některých případech nejsme schopni provést strukturální experiment, proto je snaha zjistit alespoň teoretickou strukturu modelací. Modelovat můžeme buď homologně, nebo tzv. „od nuly“ (lat. ab initio). Pokud známe strukturu proteinu, který se velmi podobá proteinu, jehož strukturu chceme znát, můžeme ho použít jako tzv. šablonu v *homologním modelování*. V cílovém proteinu se hledají strukturní motivy přítomné v šabloně. Naopak *ab initio modelování* nepoužívá žádné výchozí složení a hledá výslednou konformaci pomocí simulace tepelného pohybu, tak aby došlo k minimalizaci *volné energie* struktury – tedy situaci, kdy se protein dostane do konformace, z které se již náhodným pohybem nedostane. Jedním z nejuznávanějším nástrojů k tomuto účelu je program I-Tasser.

2.2 Syntéza proteinu

Hovoříme-li o vzniku proteinu, neboli jeho *syntéze*, může se jednat o *chemickou syntézu* (lat. *in vitro*) – jeho přípravu v laboratorním prostředí s využitím chemikálií a posloupnosti chemických reakcí podobně jako klasicky připravujeme jiné chemické látky; nebo o *biosyntézu* –

výrobu proteinu v buňce (lat. *in vivo*), tedy tak jak proteiny vznikají přirozeně. Biosyntéza probíhá za přítomnosti těchto faktorů:

- Předpis pro vytvoření proteinu – *genetický materiál* reprezentován *nukleovou kyselinou*
- Prostředí buňky s volnými aminokyselinami
- Molekulární komplex sestavující protein – nazývá se *ribozom*

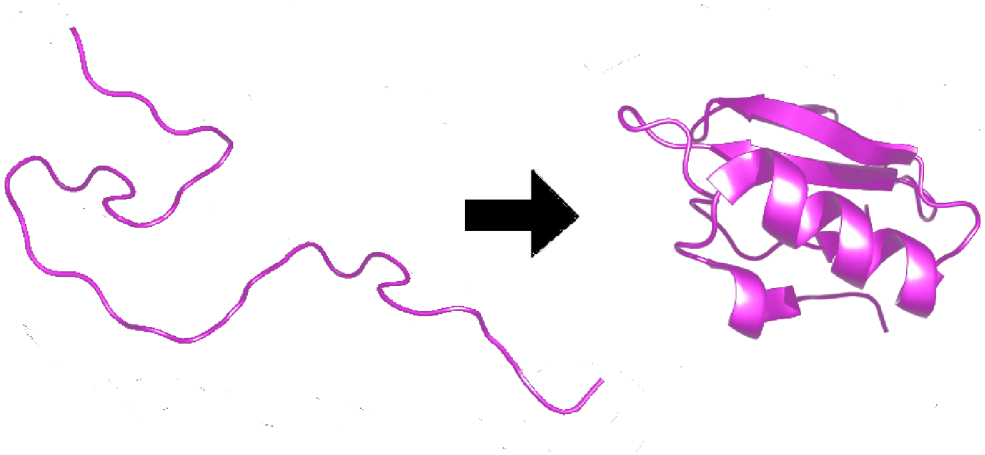
Nukleová kyselina obsahuje *genetický kód* představující určité *geny*. Gen, jenž kóduje nějaký protein se nazývá *strukturní gen*. Takový gen je čten v nukleové kyselině ribozomem, který podle něj vytváří peptidový řetězec. Dochází k tzv. *expresi genu*. *Hostitelskou buňku*, v které exprese (tedy syntéza proteinu v našem případě) probíhá, označujeme jako *expresní systém*. Nejoblíbenějším expresním systémem je bakterie *Escherichia coli*. Protein, který vznikne pozměněním (mutací) jeho struktury, nazýváme *mutant*. Původní struktura se označuje jako *wildtype* (WT).

Během procesu syntézy vzniká nejen primární struktura proteinu, ale také dochází k jeho *skládání* do výsledné podoby. Postup tohoto procesu rovněž ilustruje obrázek 2.4. V důsledku vzniků vodíkových můstků mezi kostrou, mezi zbytky, příp. i zbytky a kostrou, disulfidických vazeb mezi sirnými zbytky a Van der Waalsovými silami se do sebe protein začne proplétat a začne vznikat jeho sekundární/terciální struktura. Další vliv má také hydrofobnost reziduí – hydrofobní aminokyseliny se snaží stočit dovnitř proteinu, tak aby je od vody ostatní aminokyseliny oddělovaly. Jeho konformace je tedy daná jeho sekvencí, nicméně může být ovlivněna prostředím nebo přítomností jiných proteinů (vznik komplexu). Protože je však prostředí v buňkách stabilní, obvykle se každý složí stejně. Protein se při skládání snaží dosáhnout energeticky výhodného tvaru, tedy stavu s co nejnižší možnou *volnou energií* **G** [J/mol]. Dokončením skládání se dostává protein do *aktivního stavu*, tedy do stavu, kdy je schopen plnit svou biologickou funkci. Rozdíl mezi proteinem v čerstvě syntetizovaném stavu a v jeho složeném stavu je na obrázku 2.6. Jeho nesložený stav i v 3D připomíná jeho primární strukturu, poté ale dochází k jeho zkompaktnění. Po ustanovení terciální struktury dochází k tomu, že po určitém čase najde protein svůj ligand a vzniká kvartérní struktura.

Oblast výroby proteinů je zajímavá – chceme vytvářet proteiny požadovaných vlastností, což může znamenat jak vytvoření zcela nového proteinu (lat. *de novo*), tak pouze genetickou úpravu známého strukturního genu. Ale i pokud chceme vyrábět nějaký běžně vyskytující se protein, spíše nežli extrahovat daný protein z organismu, může být výhodnější použít k tomu *in vivo* systém, příp. *heterologní expresi* v *in vitro* systému (vyrábění proteinu v hostiteli, v němž se pro něj přirozeně nevyskytuje gen). Nicméně při této umělé výrobě nás často omezuje *proteinová agregace* [6]. Při skládání totiž může docházet k soutěži mezi terciárními vazbami a kvartérními vazbami s jinými peptidy, především při vysoké koncentraci vyráběného proteinu. Takto získaný protein není užitečný, protože se nenachází ve svém aktivním stavu [7]. Kromě koncentrace proteinu má na skládání vliv také přítomnost solí, pH a teplota prostředí.

2.3 Rozpustnost proteinů

Rozpustnost je vlastnost rozpouštěné látky rozptýlit se v rozpouštědle (solventu), čili tvořit roztok. Látka je tím rozpustnější, čím více látky je nutné do rozpouštědla přidat, než se



Obrázek 2.6: *Stužkový model proteinu* v denaturovaném (vlevo) a aktivním stavu (vpravo). Protein se od počátku syntézy dostává ze svého denaturovaného do svého složeného stavu, tak jak postupně dochází k vazebným interakcím mezi (prostorově) blízkými aminokyseliny v řetězci.

začnou tvořit sraženiny (tj. než dojde k nasycení roztoku). Pokud mluvíme o rozpustnosti látky, je nutné určit pro jaké rozpouštědlo. Biosyntéza však probíhá v *cytoplazmě*, proto je tato informace často zřejmá z kontextu. Rozpouštědla se dělí na *polární* a *nepolární*. Polární rozpouštědlo je tvořeno z molekul, ve kterých je elektrické pole. V těchto se rozpouštějí *hydrofilní* látky, naopak *hydrofobní* se rozpouštějí v nepolárních rozpouštědlech. Voda, a tedy i *cytoplazma*, jsou polárními rozpouštědly. Hydrofilní molekula se rozpouští ve vodě, protože interaguje s molekulami vody a je tak vtahována do roztoku. Naopak hydrofobní je z roztoku vypuzována a tyto látky začnou vytvářet sraženiny. K měření rozpustnosti může být použita separace sraženiny z roztoku, nebo jiné měření její velikosti v roztoku.

Rozpustnost proteinů je problematika, kterou zatím neumíme uspokojivě řešit. Závisí na množství faktorů – vnitřních (vlastnosti proteinu samotného) a vnějších (vlastnosti prostředí). Jak již bylo řečeno, vnější závisí na pH, množství solí a teplotě. Protože ty jsou většinou již dány předem, jsou z hlediska této práce zajímavější vlivy vnitřní. Těmi jsou myšleny povrchové vlastnosti proteinu, tedy vlastnosti aminokyselin, které se nachází na povrchu proteinu – jejich hydrofobnost, funkční skupiny, náboj. Samozřejmě, jde-li se do detailu, zajímají nás vlastnosti všech reziduí, neboť z těch také plyne, která rezidua se v konečné podobě proteinu nachází na povrchu, a která uvnitř. Typicky se hydrofilní aminokyseliny rozmístí, tak aby mohli reagovat s vodou, kdežto hydrofobní se naopak ukryjí vevnitř proteinu, aby s vodou do styku nepřicházeli.

Míra, jak moc se nachází aminokyselina na povrchu se nazývá *přístupnost* (angl. accessibility). Opačná vlastnost pak *zanořenost* (angl. buriedness[4]), tedy jak moc je aminokyselina přikryta jinými v roztoku. Oba pojmy jsou však spíše koncepční, proto se přesnější údaje vyjadřují pomocí proměnné *SASA* (Solvent Accessible Surface Area, také jen ASA), což je plocha aminokyseliny přístupná rozpouštědlu. Udává se obvykle v $Å^2$ (Ångström čtvereční). Jednotka má velikost 1 Å = 0,1 nm a často se používá pro měření vzdálenosti na molekulární úrovni (např. vazba H–O má délku 0,96 Å a H–N 1,04 Å). Užitečnější však mnohdy může být přístupná plocha normalizovaná velikostí aminokyseliny – takovou hodnotou je *RSA* (také rASA) udávající, jak velká část povrchu rezidua je exponována do roztoku. Počítá

[4]Doslova „zahrabanost“.

se jako RSA = SASA/MaxASA, kde *MaxASA* určuje nejvyšší možnou SASA pro danou aminokyselinu.

Přestože rozumíme, čím je způsobena (ne)rozpustnost určitého proteinu, nedokážeme už dost dobře určit, jak pozměnit protein, abychom vylepšili jeho rozpustnost bez negativního dopadu na ostatní vlastnosti. Obvykle se to musí řešit tak, že se mutantní protein vyrobí v laboratoři a jeho rozpustnost změří. Možných mutací je ale příliš mnoho – jen pro jednobodovou mutaci je to délka sekvence $\times$ 19 (zbylých) aminokyselin. *Jednobodová mutace* mění v proteinu pouze jedno reziduum. Odtud plyne finanční náročnost takového postupu. Dobrá predikce rozpustnosti je proto v odvětví žádaná.

Kapitola 3

Predikce rozpustnosti

Ačkoliv již existuje celá řada predikčních nástrojů, obvykle se snaží o predikci z primární struktury. Vzhledem k tomu, že rozpustnost vychází spíše z vlastností proteinu v prostoru a tvaru samotného, nejeví se tyto řešení robustními. Nicméně existují dostatečně veliké datové sady známých sekvencí a jejich rozpustností. Naopak rozsáhlé datové sady obsahující struktury proteinů a jejich rozpustnost v současnosti nejsou. Tzn. tento přístup umožňuje lépe aplikovat strojové učení, což pravděpodobně také přináší větší zájem výzkumníků.

Tři zajímavé predikční nástroje jsou popsány v následující podkapitole; následuje popis predikční metodiky, kterou využívá i tato práce; a poslední podkapitola se věnuje existujícím datovým sadám pro predikci změny rozpustnosti.

3.1 Existující nástroje

Některé predikční nástroje jsou popsány v této podkapitole. Dají se rozdělit dle jejich cíle na tyto skupiny:

- Predikce (absolutní) rozpustnosti
- Predikce změny rozpustnosti
- Návrh potenciálně vhodné mutace pro zlepšení rozpustnosti

Jak u absolutní, tak relativní rozpustnosti může být výsledek predikce binární (rozpustný/nerozpustný – každý nástroj si stanoví svou mez; zlepšení/zhoršení) nebo nebinární (míra rozpustnosti; míra zlepšení). Většina existujících nástrojů však predikuje rozpustnost jako takovou – obvykle binárně. Pokud je predikce nebinární, teoreticky ji lze použít i k predikci změny rozdílem, nicméně na toto dané nástroje nejsou obvykle natrénovány. I experimentální výsledky potvrzují [8], že v tomto případě nedosahují vysoké úspěšnosti, ačkoliv samotnou rozpustnost dokáží predikovat poměrně dobře. Pro tuto práci jsou však podstatnější ty, které predikují změnu, a tak je tato kapitola zaměřena především na ně. Na rozdíl od zbylých 2 skupin, kterým fundamentálně postačuje jako vstup protein, tato vyžaduje (kromě originálního proteinu) navíc také zvolenou mutaci/mutanta.

Prvním nástrojem schopným predikovat rozpustnost je pravděpodobně **SOLpro** [9] pocházející z roku 2009 – predikuje binárně a využívá pouze primární strukturu. Následně se objevil v 2010 nástroj **OptSolMut** [10], který naopak predikuje změnu a sice ze struktury. Pro absolutní predikci po vydání SOLpro vznikla celá řada nástrojů – PROSO, ccSOL, A3D, několik dalších, a nejnověji SOLart (2019). Naopak pro predikci změny jich vzniklo

pouze pár – CamSol (2014), PON-Sol (2016) a **SODA** ([8], 2017). Zvýrazněné budou probrány v následujících podkapitolách. Vybrány jsou tak, že každý představují výrazně jiný přístup k problému.

3.1.1 SOLpro

SOLpro (2009, [9]) je nástroj binárně predikující rozpustnost proteinu z jeho primární struktury natrénovaný na ~17 000 proteinech exprimovaných v *E. coli*. Vytvořený dataset, nazvaný SOLP, je vybalancovaný (stejné množství rozpustných a nerozpustných proteinů) a sloučen z databází PDB, TargetDB a SwissProt (databáze proteinových sekvencí). Architektura prediktoru je založela na dvouvrstvém SVM. V první vrstvě je 21 SVM klasifikujících podle $3 \cdot 7$ skupin rysů, a sice frekvence monomerů, dimerů a trimerů v sekvenci podle 7 různých dělení s různou aritou:

- všechny aminokyseliny – 20
- hydrofobnost – 5
- podobnost – 7
- chemické vlastnosti – 7 – kladná/záporná, sirná/alkoholová/amidová/aromatická skupina, alifatická
- a další...

V důsledku tedy mají rysy 5 až 8000 hodnot (nejnižší hodnota u monomerů dle hydrofobnosti, nejvyšší pak u trimerů všech aminokyselin). V druhé vrstvě se klasifikuje podle výstupů první vrstvy, k tomu navíc několik dalších hodnot určených přímo ze sekvence (délka, hmotnost molekuly, průměrný náboj, ...) a hodnoty predikované ze sekvence (poměr zbytků v α-helixu, v β-listu, ...). Během optimalizace z klasifikace byly odebrány 3 rysové skupiny a u ostatních skupin značně zmenšen celkový počet rysů a finální prediktor dosáhl složením všech těchto vlastností přesnosti ~75 %.

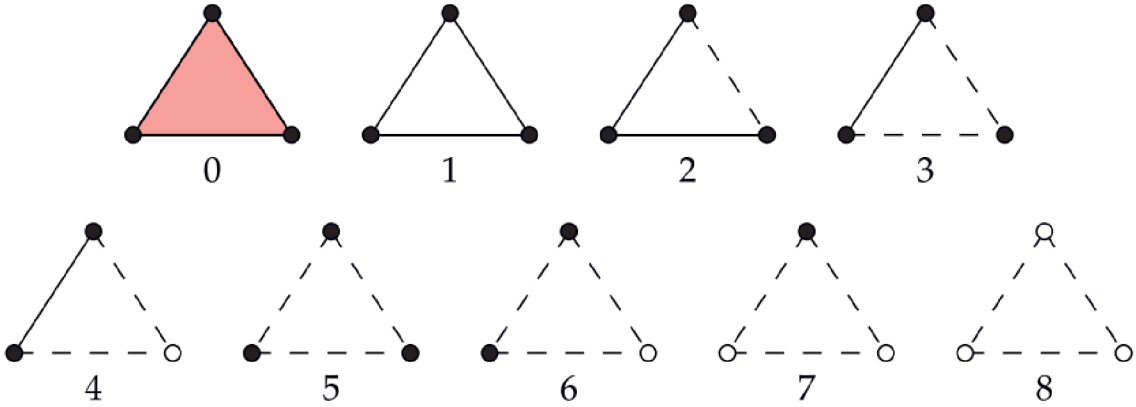
3.1.2 OptSolMut

Tvůrci nástroje **OptSolMut**[1] (2010, [10]) vymysleli neotřelý přístup, jak předpovědět vliv mutace na rozpustnost. Používají totiž strukturální predikci, do které započítávají především ty aminokyseliny, které se nachází na nebo poblíž vrcholu. Využívají k tomu koncept *zanořenosti* (angl. buriedness), což je vlastnost aminokyseliny v řetězci, kterou získává po složení proteinu. Význam je, jak hluboko se aminokyselina nachází od povrchu proteinu, neboli jak moc je vevnitř proteinu. Dále vytvořili ohodnocující funkci „rozpustnosti mutace“ (volně přeloženo). Do výpočtu zahrnují frekvenci výskytů pouze takových trojic monomerů (prostorově, nikoliv sekvenčně sousedících), které nejsou zanořené.

Problém určení zanořenosti vyřešili tak, že vypočítají nejdříve *3D Delaunayovu teselaci*. To je proces, který prostor vyplní nepřekrývajícími se čtyřstěny, které sdružují nejbližší 4 body v prostoru. Body představují aminokyseliny v proteinu. Samotné čtyřstěny jsou ale vlastně vedlejším produktem. Hlavním produktem jsou jejich trojúhelníky a těleso s povrchem tvořeným některými z nich. Tento povrch je zároveň přibližně (přibližně protože aminokyselina je určena pouze jedním bodem, což je zjednodušení) i povrchem proteinu. Tyto povrchové trojúhelníky a aminokyseliny, které tvoří jejich vrcholy, a také hrany těchto

[1] Tvůrci jej sice takto explicitně nenazvali v citovaném článku, nicméně webserver s běžícím nástrojem jej takto označuje.

trojúhelníku, jsou *nezanořené*. Mají stupeň zanoření = 0. Všechno ostatní je *zanořené* (stupeň zanoření $\geqslant 1$). *Stupeň zanoření* trojúhelníku je hodnota 0–8, kde hodnoty vyšší jak 0 jsou dány počtem nezanořených hran a vrcholů daného trojúhelníku, což je lépe vidět na obrázku 3.1.



Obrázek 3.1: (Převzato z [10]) **Stupeň zanoření** trojúhelníku. Nultý stupeň představuje nezanořený (jsoucí na povrchu) trojúhelník. Plná čára a vybarvený vrchol představují nezanořenou hranu, vrchol. Za povšimnutí stojí, že je-li vrchol zanořený, je zanořená i hrana, na které se nachází. Nebo duálně, je-li některá hrana nezanořená, platí, že i na ní ležící body nejsou. To plyne z toho, že trojúhelníky jsou částí čtyřstěnů získaných teselací.

Další problém, s kterým se potýkali autoři, byla neexistující datová sada pro strojové učení na tento problém v té době. Podařilo se jim však shromáždit vlastní datovou sadu 137 proteinů – proteiny se strukturou, mutací a binární změnou rozpustnosti. Autoři tuto sadu označili jako „největší jim známý dataset pro tento problém“. Tedy shromáždili datovou sadu o velikosti více než 100× menší než datová sada o rok staršího SOLpro. Tato absence dat jen dokládá, proč je predikci rozpustnosti ze 3D struktury věnována značně menší pozornost než predikci ze sekvence.

Pro predikci rozpustnosti použili rozdíl ohodnocující funkce pro WT a mutanta – kladný = zlepšení, záporný = zhoršení. Ohodnocující funkce se skládá z váženého součtu vlivu Q_t každé 3-jice ({i, j, k}, b, c), kde i, j, k jsou aminokyseliny, b je stupeň zanoření a c je stupeň konektivity (určuje zda a nakolik trojúhelník leží na kostře). Vliv 3-jice $Q_t = \log[\frac{f_t}{p_t}]$ je dán poměrem mezi frekvencí dané 3-jice v proteinu (f_t) a běžné frekvence (p_t), přičemž ty jsou získány statistickou analýzou z odlišného datasetu proteinů, které jsou rozpustné. Pouze 3-jice s zanořeností $\leqslant 4$ byly zahrnuty do výpočtů.

Pro strojové učení pro váhy zmíněné v předešlém odstavci autoři nejprve zkusili použít SVM a Lasso, nicméně podané výsledky nenaplnili očekávání, zejména protože v datové sadě je jenom 3895 různých trojic (z 8000 všech teoreticky možných), z toho mnohé z nich se vyskytují pouze jednou, zatímco počet proteinů je 137. Proto vyvinuli vlastní metodu inspirovanou oběma zmíněnými a dosáhli tak lepších výsledků. Během učení 3-jice, které nejvíce (anti)korelují se změnou rozpustnosti dostanou větší váhu (resp. váhu blížící se 0). Ve finální variantě dosáhli přesnosti 76,6 %.

3.1.3 SODA

Proti předchozím nástrojům je tento nástroj (2017, [8]) koncepčně jednoduchým, protože pouze počítá rozpustnost jako vážený součet skóre jiných predikčních nástrojů predikujících

jiné (související) vlastnosti. Pracuje se sekvenční informací, ačkoliv dokáže využít i strukturní informace (o tom dále) a kromě substitučních mutací zvládne i deleci a inserci. Toto jsou využité nástroje/vlastnosti SODou:

- **PASTA** – sklon k agregaci
- **ESpritz** – predikce vnitřní neuspořádanosti (IDP)
- Profil hydrofobnosti podle **Kyte-Doolittle**
- **FESS** – sklon k vzniku sekundárních motivů

Jiné vědecké práce dokázali korelaci těchto vlastností s nerozpustností, ostatně některé z nich již byly vysvětleny i v této práci. *Vnitřně neuspořádaný protein* (angl. intrinsically disordered protein – *IDP*) je pak vlastnost proteinu, kdy obsahuje části (proto vnitřně), které nemají žádnou vyšší strukturu. Tím pádem nemají ani stálý tvar a také experimentálně zjištěná struktura se vždy značně liší. Je to však důležitá vlastnost některých proteinů, které hrají roli v životně důležitých procesech v organismu, které by uspořádané proteiny (tj. s neflexibilní konformitou) plnit ani nemohly. [11] Příklad proteinu s IDP je na obrázku 3.2 – pouze část proteinu má pevně danou strukturu. Vrchní obrázek představuje příklad, jak může zobrazený protein vypadat v určitém okamžiku. Spodní je naopak složení z více jeho konformací zachycených v různých momentech.

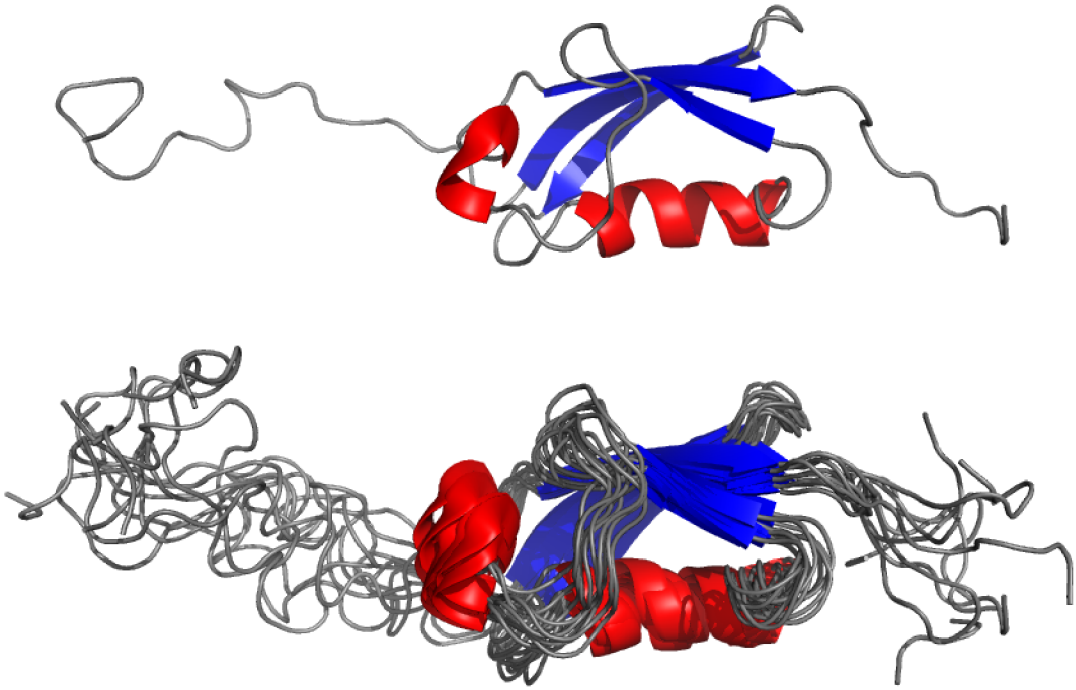
Skóre pro každý nástroj je vypočtena jako suma ohodnocení daným nástrojem pro každé reziduum. Ta je normalizována délkou sekvence (tj. nástroj tak umožňuje pracovat i s delecí/insercí). Toto sumární skóre je odečteno WT od mutanta. Takto získané rozdíly pro každý faktor, resp. jejich váha, byly poté předmětem strojového učení. Získány byly váhy jsou 1, 2, −50, 2, 2 pro agregaci, neuspořádanost, hydrofobicitu (nepřímá úměra), α-helix a β-list. Nemá smysl tyto hodnoty porovnávat, protože jsou ovlivněny taky měřítkami zmíněných nástrojů, nicméně znaménka výsledných vah korelují s teorií.

Pokud je poskytna struktura, je použita pouze k tomu, aby rezidua s méně než 20 % přístupu k rozpouštědlu, byla ignorovány ve výpočtu. Výpočet je proveden programem DSSP. Kromě vlivu mutace umí spočítat i profil proteinu – vliv všech možných jednobodových mutací na rozpustnost.

SODA byla otestována na datové sadě konkurenčního CamSOLu, kde dosáhla 100% úspěšnosti. Nicméně tento dataset obsahuje pouze 56 variant mutací z 19 proteinů. Trénovacím datasetem byl PON-Sol, kde SODA získala pouhých 59 %. V článku dochází také k porovnání pouze s nástroji, které jsou zamýšleny jako prediktory rozpustnosti, OptSolMut ve srovnání chybí.

3.2 Predikce analýzou povrchového potenciálu

Jedná se o jednu z aktuálně rozvíjených metodik (**Surface Patches**, 2019) pro návrh příznivé mutace vzhledem k rozpustnosti, jež byla popsána a experimentálně ověřena v článku [12]. Její autoři ji označují za slibnou a také podle ní navrhují vytvořit nástroj k predikci celkové rozpustnosti. Metoda využívá strukturální informace a výsledkem je binární predikce. Principiálně je založena na analýze stejných autorů [13] z roku 2013, kde došli k závěru, že nerozpustnost proteinu koreluje s výskytem velkých kladně nabitých ploch na jeho povrchu, neboli kladných *(povrchových) patchí*. Tento pojem bude dále nazýván jednoduše



Obrázek 3.2: *Vnitřní neuspořádanost zobrazená na proteinu SUMO* (PDB: 1A5R). Na vrchním obrázku je struktura proteinu zachycená jedním během strukturního experimentu. Poměrně značná část jak od N-konce (vlevo), tak od C-konce (vpravo) absentují jakoukoli sekundární strukturu, co už naznačuje, že konce budou flexibilní. Na dalším obrázku je pak složenina získaná ze struktur z 10 experimentů. Zde již vidíme, že zatímco středová část proteinu obsahující dostatek strukturních motivů si svůj tvar docela zachovává, konce různě mění svůj tvar. Za zmínku snad stojí ještě to, že malinká část N-konce si zachovává své zatočení, což ale není na obrázku dobře patrné.

jako kladná *oblast*[2]. Korelace je pravděpodobně způsobena tím, že takové proteiny se agregují s (ribo)nukleovými kyselinami, které jsou mírně záporně nabité a vzájemně se tedy přitahují. Navrhovaný postup je pak:

1. Vytvoření modelu vnějšího el. pole molekuly (na povrchu):
   Do úvahy se, ideálně, počítají všechny polární zbytky, nejen nabité. Snahou je zjistit, které oblasti na povrchu proteinu jsou nabité.

2. Nalezení největší kladné oblasti:
   Může jít také o jinou význačnou oblast, hledáme však tu s kladným potenciálem. A tu je cílem rozbít.

[2] *Patch* je část povrchu, která se nějak odlišuje od jeho zbytku. Variantami pro překlad jsou tedy flek, skvrna, stopa, záplata či náplast. Jakkoliv jde ale pouze o hledání části povrchu mající určitou skrytou vlastnost, která se ale nemusí projevit vizuálně, ani se nejedná o nějakou k povrchu přilnuvší strukturu a i zbylé varianty mají své nevhodné konotace, byl překlad jednoduše zvolen jako *oblast*, což je dostatečně unikátní v rámci práce.

3. Výběr vhodného rezidua pro mutaci se změnou znaménka:
   Vybíráme kladně nabité, či jiné polární[3] reziduum nenesoucí záporný náboj, které se nachází v této oblasti, či dostatečně blízko. Provedeme substituci na záporně nabité reziduum, tak abychom do dané oblasti zanesli opačný – záporný – náboj (mutace původně neutrálního rezidua), případně zároveň odebrali původní kladný náboj (mutace kladného). Optimální variantou je tedy druhá zmíněná.

4. Ověření (podle genové databáze), zda je dané reziduum konzervované:
   Abychom nepoškodili funkci daného proteinu, provedeme analýzu *konzervovanosti* daného rezidua. Konzervované reziduum znamená, že konkrétní aminokyselina na této pozici má klíčový význam, proto během evoluce nedocházelo k náhodným změnám na této pozici. V genové databázi hledáme různé varianty téhož genu – tzv. *homology*. Pokud je najdeme a dané reziduum je v různých homolozích tvořeno jinými aminokyselinami, jedná se o reziduum vhodné k mutaci. V opačném případě je vhodné najít a analyzovat jiné kandidátní reziduum.

5. Aplikace mutace:
   Nalezené nekonzervované reziduum můžeme mutovat, případně můžeme použít i více mutací a předpokládat synergický efekt. Zde je nutno poznamenat, a autoři se k tomuto kroku nevyjadřují, že mutované reziduum by stále mělo splňovat fyzikální zákony – nacházet se v povolených torzních úhlech a neokupovat svým zbytkem prostor již zaujímaný jiným reziduem.

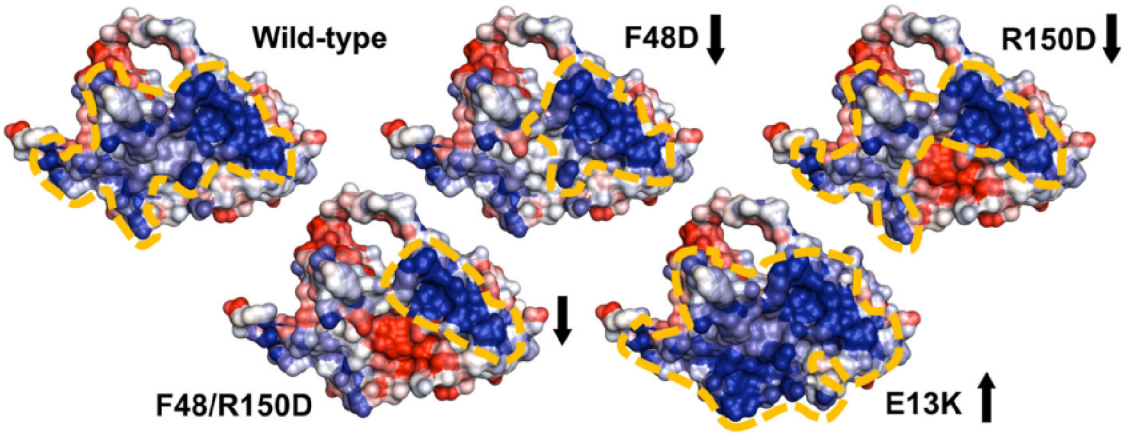
Navržená metoda byla otestována na rekombinantním lidském hormonálním proteinu erythropoetinu (zkráceně **rHuEPO**). V těle reguluje výrobu červených krvinek a podává se jako medikament při léčbě mnoha souvisejících onemocnění. Pro toto použití se průmyslově vyrábí v *E. coli*. Při jeho výrobě vzniká v buňce nepřirozeně velká koncentrace[4] této látky a dochází k agregaci namísto správného skládání. To je, jak již bylo řečeno, běžná příčina nerozpustnosti.

Mutace byly patrně provedeny pouze nahrazením aminokyselinového zbytku bez jakéhokoliv přizpůsobení okolní struktury. Jako substituent v mutaci využili kys. asparagovou (asp/D), která nese záporný náboj. Druhou možností by byla kys. glutamová (glu/E), nicméně kys. asparagová je defenzivní varianta, která snižuje šanci na změnu vyšších struktur proteinu v důsledku kratšího postranního řetězce (méně příležitostí pro terciární vazby). Celkově ověřili metodu na 4 mutacích, z nichž dvě jsou příznivé jednobodové mutace (fenylalaninu F48D a argininu R150D), třetí je kombinace předchozích a čtvrtá je nepříznivá (E13K na lysin). Z daných reziduí bylo jen reziduum 150 zjištěno jako konzervované. Nepříznivá mutace byla zařazena jako oboustranný test, tedy pro ověření, zda opačný postup bude mít na rozpustnost i opačný vliv. Vizualizace potenciálu na povrchu přirozeného rHuEPO i jeho mutantech je na obrázku 3.3. K povšimnutí je to, nakolik jediná mutace v 166-meru může pozměnit jeho povrchový náboj.

Všechny 4 varianty byly syntetizovány ve dvou *E. coli* systémech (SHuffle, BL21 pLysS) při neutrálním pH. V prvním ze systémů se potvrdili predikované změny rozpustnosti, v druhém však pouze pro mutace E13K a R150D, tedy pro ty neobsahující mutaci F48D. Autoři předpokládají, že je to způsobeno tím, že dochází k obnažení hydrofobních reziduí nacházejících se pod tímto fenylalaninem, neboť kys. asparagová je podstatně menší AK

[3] Vyhýbáme se mutaci nepolárního rezidua, což by způsobilo změnu hydrofobicity.

[4] To je běžný výrok používaný v těchto situacích, ve skutečnosti je však jakékoliv množství této lidské látky v cizorodém organismu nepřirozeně velké.



Obrázek 3.3: (Převzato z [12]) Zobrazení el. potenciálu na povrchu erythropoetinu (PDB: 1EER) a jeho mutantů. **Modrá** představuje **kladný** potenciál (+25 mV a výše) a červená naopak (−25 mV a méně). Bezbarvé části povrchu nenesou náboj. Žlutou přerušovanou čárou je ohraničena největší kladná oblast (souvislá plocha kladného potenciálu na povrchu). Šipka označuje zvětšení/zmenšení této plochy vůči WT. Mutace F48D a R150D jsou zasazeny přibližně do středu plochy WT, tak aby ji rozrušili na 2 části. Mutovanou aminokyselinu na povrchu lze na obrázku zaměřit podle místa s nejkontrastnější změnou náboje vůči WT. Všechny modely mají stejný tvar – to poukazuje na neprovedení optimalizace po mutaci.

(mutací přichází o aromatickou skupinu). To je dohad autorů, nicméně může to být už tím, že zde dochází k náhradě nepolární látky (fenylalanin) za polární (kys. asparagová), a tudíž i změně hydrofilnosti daného rezidua, což v důsledku může mít vliv na skládání proteinu. S mutanty EPO navíc údajně nelze provést strukturální experiment, pro ověření spočteného modelu – tj. nelze vyloučit, že některá z provedených mutací i výrazně pozměnila strukturu proteinu.

Předchozí 4 (3 + kombinace) zmíněné mutace byli ověřovány v expresním systému. Ty však byli autory vybrány jako 3 zajímavé mutace z celkově 34 vytvořených modelů – každý vytvořený jedinou mutací určité aminokyseliny nacházející se v největší kladné oblasti. Jsou rozděleny do 3 skupin, z nichž z každé byla vzata jedna:

1. Kladná AK $\Longrightarrow$ asp: příznivá mutace se změnou polarity
   Z 9 mutací vybrána R150D. 5 mutací predikováno jako rozpustných.

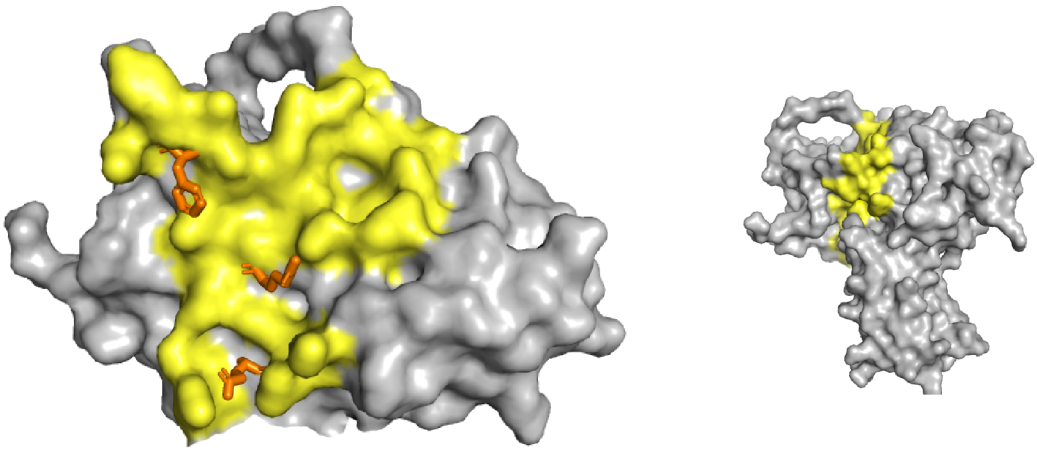
2. Nenabitá AK $\Longrightarrow$ asp: příznivá mutace se zavedením náboje
   Z 19 mutací vybrána F48D. Rovněž 5 mutací predikováno jako rozpustných

3. Záporná AK $\Longrightarrow$ kladná AK: nepříznivá mutace se změnou polarity
   Z 5 mutací vybrána E13K. Všechny predikovány jako nerozpustné.

Nepřekvapivě, u 1. a 2. skupiny došlo vždy k zmenšení oblasti vůči WT a stejně tak u 3. skupiny k jejímu zvětšení. K drastickému zmenšení o více než polovinu došlo v případech, kdy byla oblast mutací rozdělena ve dví, jako u mutace F48D – viz zvýrazněná oblast mutace F48D na obrázku 3.3.

Strukturu proteinu pro modelování mutací autoři získali z erytropoetinového komplexu (PDB: 1EER), což trimer sestávající, jak ze samotného rHuEPO, tak i jeho dvou *receptorů* – tedy látek, které se na něj specificky vážou. *Rozhraní* (angl. interface) je povrch

proteinu v místě, kde má kontakt s jiným proteinem komplexu. Ačkoliv to nemá vliv na jádro problému (návrh mutace pro zlepšení rozpustnosti), autory zvolená rezidua k mutaci jsou právě součástí jednoho z rozhraní – viz obrázek 3.4. To je pravděpodobně také důvod konzervovanosti u nabitého rezidua R150. Přínosnost takového mutanta, nehledě na jeho lepší rozpustnost, je tak značně diskutabilní, uvážíme-li, že jeho funkce je pravděpodobně narušena.



Obrázek 3.4: Zobrazení jednoho z rozhraní proteinu rHuEPO (PDB: 1EER) na jeho povrchu: samostatně (obr. vlevo), s navázaným receptorem (obr. vpravo). Rozhraní mezi oběma proteiny je zvýrazněno žlutě a je definováno jako skupina reziduí, jejichž atomy se dotýkají atomů toho druhého proteinu. Trojice oranžově zvýrazněných reziduí na obr. vlevo není vykreslena jako součást povrchu a jedná se právě o 3 mutované rezidua z článku (seshora F48, R150 a E13).

Autoři nezveřejnili SW, pomocí kterého výpočty provedli a v článcích [12, 13] jsou pouze kusé informace. Avšak některé podstatné chybějící informace o jejich implementaci se podařilo získat pomocí e-mailové komunikace s jedním z autorů [14]. Uvedl, že si jej napsali svépomocí – výpočetní jádru ve FORTRANu s rozhraním v Perlu.

Výpočet el. pole byl proveden pro mřížku s roztečí 0,6 Å podle Poisson-Boltzmannovi rovnice při neutrálním pH, iontové síle I = 0,15 M. Aproximace povrchu proteinu je spočtena jako množina bodů proteinu v mřížce ne zcela nejbližších solventu („not quite nearest neighbour points (nn+1)“)[14]. Pravděpodobně za cílem povrch částečně vyhladit. Zvolená rozteč má totiž velikost poloviny Van der Waalsova poloměru nejmenšího atomu (H – 1,2 Å), což znamená, že členitostí by mohlo vznikat značné množství. Následně byl povrch proteinu rozdělen na jednotlivé kladné ($\geqslant$ +25 mV), záporné ($\leqslant$ −25 mV) a neutrální oblasti procházením mřížky v 6 směrech [14]. Autor proces nazývá *2D countouring*, protože myšleně probíhá na ploše povrchu, ačkoliv prakticky v prostoru. Pro predikci je použita velikost největší kladné oblasti, vyjádřena počtem bodů mřížky, jimiž je tvořena. Přesněji je použita její relativní velikost vůči předem stanovenému *prahu* (viz dále) a to tak, že hodnota větší než 1 predikuje nerozpustnost a naopak menší než 1 rozpustnost.

Výpočet je založen na zjištěních z článku [13]. V něm analyzovali, jak spolu statisticky souvisí el. pole proteinu a jeho rozpustnost. Pro sadu proteinů se známými jak strukturami, tak rozpustností vypočítali jejich povrchové náboje a ty analyzovali. Použili několik kvantitativních vlastností proteinu a zjišťovali, která z nich nejlépe dovede dataset rozdělit na

rozpustné a nerozpustné proteiny pomocí Mann-Whitneyho statistického testu. Nejspolehlivější rozdělení bylo získáno právě podle velikosti největší kladné oblasti s nejlepší hodnotou prahu 3000 bodů mřížky. Suma náboje v největší kladné oblasti také dataset rozděluje spolehlivě, zdaleka však ne tolik jako předchozí míra. Naopak podle celkového náboje proteinu dataset nelze rozdělit. Dále je nutno zmínit, že rozpustnosti proteinů z tohoto datasetu byly zjištěny při chemické syntéze (ve zkumavce), nikoliv jako v navazujícím článku v *E. coli*. To znamená, že tato zjištění nemusí nutně platit i v buněčném systému.

3.3 Datové sady

Pro tvorbu schopného predikčního nástroje jsou nezbytná data. I když o rozpustnosti je již známo mnoho, stále to není dost a unikají nám souvislosti. Takže i když do predikčního nástroje můžeme vložit velkou informaci, data jsou stále nezbytná alespoň pro evaluaci. Pro predikci změny z rozpustnosti potřebujeme datovou sadu obsahující:

A protein se strukturou + jeho rozpustnost, mutanta se strukturou + jeho rozpustnost v témže expresním systému, nebo

B protein se strukturou, mutanta se strukturou + informaci o zlepšení (zhoršení) rozpustnosti

Bohužel, taková data **prakticky neexistují.** Přístup stejný jako v Surface Patches, tedy získání struktury mutanta svépomocí, je tak téměř nutný, a proto využitelných dat není mnoho. Pokud je informace o rozpustnosti jistého proteinu, pro který byla zjištěna struktura, většinou se nikdo nepokoušel provést totéž pro jeho mutanta. Pokud naopak taková informace je, často není zjištěna struktura, protože se experiment prováděl za cílem zajistit lepší rozpustnost, přičemž nízká rozpustnost také komplikuje strukturní experimenty.

Existují datové sady, které měly za účel sjednotit data z těchto experimentů právě pro výzkum nástrojů pro predikci rozpustnosti. Jejich problém je však *vysoký šum* v datech, přítomný právě z toho důvodu, že data pocházejí z různých zdrojů. Zdrojové experimenty byly prováděny v jiných systémech, za různého pH, za různé teploty, s různou přesností, měřilo se různými metodami a celé to prováděli jiní lidé. Kromě toho se výsledky experimentů na molekulární úrovni můžou lišit i za stejných podmínek vlivem náhody. Proto se biologické experimenty většinou opakují vícekrát, aby byla zaručena přesnost dat. S jakou přesností však vědci pracovali také nevíme. Tím pádem zjištěná vlastnost týkající se rozpustnosti na části takového datasetu nemusí být přenositelná na zbytek.

Byly nalezeny 2 použitelné datové sady, jsou popsány v následujících odstavcích.

Dataset OptSolMut

Je z roku 2010 a vytvořili jej autoři nástroje OptSolMutu [10] jako první známý dataset pro predikci změny rozpustnosti. Autoři tento dataset získali ručním prohledáváním vědeckých prací, které se zabývaly rozpustností mutantů. Autoři umístili dataset volně ke stažení ve formátu .xls. Celkově obsahuje mutace 15 proteinů a 137 mutací. Zdrojových prací je také 15, nicméně jedná se pouze o shodu náhod – ve skutečnosti v některých pracích byli měřeni mutanti vícera proteinů, a naopak jistými proteiny se zabývalo vícero prací. Dataset obsahuje pro každou mutaci tyto údaje:

- ID zdroje unikátní v rámci datasetu
- Pojmenování zdrojové práce
- Název proteinu
- PDB ID + identifikátor řetězce
- Seznam mutací
- Rozpustnost mutanta (binární) – není u všech
- Změna rozpustnosti
- Predikce změny
- Změna stability

Většina mutací (105) je jednobodových, ostatní jsou 2bodové až 9bodové. Pro práci je zajímavých pouze těchto **105** variant. Dále víme, že všechny mutované proteiny jsou v původní variantě rozpustné. Až na 16 případů známe i rozpustnost po mutaci – mutanti kromě 2 výjimek zůstali rozpustní. Nevíme však, jak v jednotlivých pracích stanovili mez rozpustnosti, proto se nedá na tuto hodnotu spoléhat. Dataset je nevyvážený z hlediska predikovaných změn. Obsahuje 59:78 zlepšení:zhoršení, resp. 44:61 pro 105 vybraných mutací. Dataset není ani vyvážený, ani se nejeví jako přirozený, protože složený z náhodných mutací by měl menší poměr zlepšujících mutací. Nevyvážení je ještě patrnější na jednotlivých proteinech a jeho mutantech, kdy mutace většiny z proteinů jsou buď všechny zlepšující, nebo všechny zhoršující. Může to být ale také jednoduše tím, že protein, jehož drtivá většina mutantů měla horší rozpustnost, je relativně lépe rozpustný, než protein, který to má naopak. Vzhledem k vyváženosti počtu hydrofobních a hydrofilních aminokyselin bychom totiž náhodnou mutací „průměrně rozpustného proteinu“ neměli jeho rozpustnost v průměru ovlivnit.

Dataset Whitehead

Dataset Whitehead ([15], 2017 a [16], 2019) vznikl zřejmě v první vědecké práci, která masivně zkoumala vliv mutací na rozpustnost včetně syntézy a změření rozpustnosti všech mutantů. Mutovány byly proteiny TEM1 beta-laktamáza (**TEM1 BLA**, PDB ID: 1M40), levoglukosan kináza (**LGK**, PDB ID: 4ZLU) a pyrrolidinketidová syntáza (**PKS**). Rozpustnost byla změřena v 3 různých systémech: Yeast Surface Display (YSD), Tat export a *E. coli* GFP.

Dataset vznikl ve dvou částech. První část v roce 2017 obsahovala proteiny TEM1 a LGK exprimované v systémech Tat export a YSD. Velmi podezřelé bylo, že LGK dosáhlo relativní v prvním zmíněném rozpustnosti 33,1 %, což se silně odlišovalo od hodnoty naměřené v 2. z nich. Autoři v roce 2019 vyzkoušeli další expresní systém, opět na proteinu LGK a navíc přídali další protein PKS. Přehled počtu jednotlivých mutací obsažených v datasetu spolu s poměrem zlepšujících mutací je v tabulce 3.1.

Z důvodu velkého rozdílu u naměřených rozpustností v systému Tat export a ve zbylých dvou systémech (zbylé dva se navíc téměř shodnou u proteinu LGK) a vzhledem k tomu, že běžný poměr lépe rozpustných mutantů pro průměrný protein je spíše kolem jednoho mutanta z desíti, jsou mutace z tohoto systému vyřazeny, neboť s velkou pravděpodobností tyto hodnoty nejsou relevantní. Kromě toho jsou u ostatních proteinů ještě vyřazeny pozice mutací, které autoři datasetu označili za nespolehlivé. Výsledné počty mutací jsou reflektovány ve sloupci změřeno stejné tabulky. Ignorování vzorků ze systému Tat není tak významné, jak by se mohlo zdát, protože se tím nezmění ani počet proteinů datasetu a ani počet mutací.

TEM1 BLA je nejmenší z proteinů o 263 reziduích, PKS má 388 reziduí a LGK je největší s 439 rezidui. Pro každý protein byli vytvořeny v podstatě všechny možné jednobodové

mutace, tedy mutace každého rezidua na 1 ze zbývajících 19 standardních aminokyselin. Pro každou mutaci je změřena míra změny rozpustnosti. Dohromady je pro experimenty použito 14 254 mutací pro všechny proteiny (bez započtení těch, které byly z důvodu dříve uvedených odebrány). Naměřené mutace pro LGK v *E. coli* a YSD mají naměřeny i vzájemně exkluzivní mutace, takže po sjednocení mutací proteinu z obou systémů je jejich počet 6947. Drtivá většina mutací je nepříznivých, data tedy nejsou symetrická. Pro trénování by tak mohly být vyváženy převzorkováním. Jelikož se ale jedná o přirozené nevyvážení, převzorkování nebude provedeno. Velkou výhodou datasetu je, že obsahuje všechny možné mutace. Nehrozí tedy špatné statistické vyhodnocení z důvodu vychýleného datasetu.

	syntetizováno	systém	změřeno	zlepšení	v %	zhoršení
TEM1	4997	YST	2467	253	10,3	2214
		Tat	4467	1101	24,6	3102
LGK	7945	YST	6264	309	4,9	5955
		Tat	7111	2353	33,1	4330
	8341	*E. coli*	6427	422	6,6	6005
PKS	6060	*E. coli*	4840	1420	29,3	3420

Tabulka 3.1: *Tabulka obsažených mutací v datové sadě Whitehead dle proteinu a expresního systému.* Ne pro všechny z naměřených hodnot se podařilo autorům naměřit rozpustnost, avšak v počtu změřených nejsou zahrnuty ani mutace, které sice byly změřeny, ale autoři je označili za nespolehlivé. K povšimnutí je, že rozpustnost proteinů v systému Tat se značně rozlišuje od naměřené rozpustnosti ve zbylých dvou.

Kapitola 4

Návrh predikčního nástroje

Pro dosavadní návrh predikčního nástroje využívám myšlenku popsanou v podkapitole 3.2 – predikce podle povrchového potenciálu proteinu (Surface Patches). Odkazovaná metoda se snaží zlepšit rozpustnost pomocí rozbíjení největší kladně nabité plochy na povrchu molekul, neboť již byla experimentálně dokázána souvislost s nízkou rozpustností v předešlých vědeckých pracích. Tato metoda tedy pracuje implicitně s intuitivní premisou (tu využívají obvykle i ostatní nástroje), podle které je třeba se zabývat povrchem proteinů pro analýzu rozpustnosti. Nástroj samotný bude vytvořen v jazyce Python, pro jeho širokou interoperabilitu a s ovládáním CLI. Formálně by nástroj měl pracovat s těmito vstupy a provádět tuto činnost:

Vstupy

WT Mutovaný protein se strukturou (= wildtype, formát PDB)

i, x Mutace daná pozicí a substituentem

Výstupy

Δs Binární predikce změny rozpustnosti varianty proteinu danou mutací spočtená metodou povrchového potenciálu

Predikci je tedy principiálně možné provádět *měřením změn elektrického potenciálu na povrchu molekuly*. K výpočtu vnějšího el. pole by ale bylo potřeba strukturu zkoumaného proteinu a nejen ji, ale i jeho mutanta. Jak bylo uvedeno v podkapitole 3.3, takové datasety nejsou. Proto je potřeba obstarat strukturu mutanta jiným způsobem. Nabízí se tentýž způsob, jakým to dělá metoda Surface Patches, tedy pouhým nahrazením zbytku mutovaného rezidua., tj. se zachováním pozicí atomů C, C_α, N a zřejmě i C_β (1. C atom postranního řetězce). Je možné domnívat se, že při malém množství mutací, které zkoumali, vytvářeli mutanty manuálně bez použití sofistikovanějšího nástroje. U automatického nástroje pochopitelně tato cesta nepřipadá v úvahu. Kromě toho je vyloučeno, že by mutace do jisté, byť malé, míry nezměnila i konformaci proteinu. Obzvláště u těch mutací, které mění fundamentální vlastnosti rezidua (např. polaritu nebo hydrofobnost). Může se pochopitelně změnit i sekundární a především terciární struktura. Z toho důvodu se jeví lépe využít specializovaného nástroje na výpočet mutantní struktury.

Jedním z takových nástrojů je **FoldX** – nástroj (binární CLI aplikace) pro *proteinové inženýrství* [17]. Má víceúčelové použití týkající se struktury proteinů – výpočty provádí podle kalkulace silového pole v proteinu a volné energie $\triangle G$ – započteny jsou síly vodíkových můstků, hydrofobní interakce, interakce vevnitř proteinu a další faktory. Pro výpočet potřebuje co nejpřesnější 3D strukturu ve formátu PDB. Nástroj je komerční, umožňuje však

použití pro výukové/výzkumné účely zdarma. Pro tuto práci jsou nejzajímavější 2 funkce FoldX – RepairPDB a BuildModel. *RepairPDB* slouží k *optimalizaci struktury*, kdy nástroj manipuluje s postranními řetězci proteinu, tak aby minimalizovala jeho ΔG a tedy dosáhl konformace, jakou by pravděpodobně měl přirozeně v buňce. Funkce *BuildModel* pak umožňuje vytvořit mutantní strukturu (umí i více mutací najednou), kdy dojde nejen k nahrazení aminokyseliny v řetězci, ale i optimalizaci výsledné struktury. Vzniknuvší mutant by tedy měl mít přirozenou strukturu. $\Delta\Delta G$ mutanta určuje, zda se jedná o stabilizující ($-$, snížení volné energie), či destabilizující ($+$, zvýšení volné energie) mutaci. Alternativou pro nástroj FoldX, která stojí za bližší prozkoumání, je **Modeller**. To je také nástroj pro proteinové inženýrství, nicméně ve formě Python knihovny.

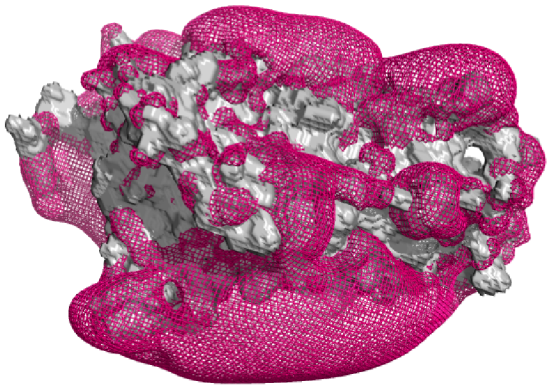
Je-li už k dispozici originální i mutovanou strukturu, je možné přistoupit k výpočtům povrchového potenciálu. Při využití libovolného nástroje je nutné podotknout, že před měřením rozdílu, by i WT měl projít optimalizací (u FoldX pomocí funkce RepairPDB), aby byly porovnávány stavy s nejnižší energií. Struktury totiž můžou odpovídat např. přímo datům z krystalografického experimentu[1], kdy je protein v odlišné, zhuštěné konformaci způsobené absencí interakcí se solventem. U takové struktury by se povrchový potenciál změnil ve velké míře u libovolného mutanta a hledané změny by byly zastíněny tímto jevem. Výsledné stavy WT i mutanta by dále měli být získané pomocí téhož nástroje. Opět aby majoritní podíl na změně neměl rozdíl v implementaci energetické optimalizace, ale samotná mutace. Tyto jevy by v opačném případě působily jako *šum*.

Abychom spočítali vlastní povrchový potenciál, musíme spočítat el. potenciál v proteinu a jeho okolí a zároveň analyzovat povrch proteinu. S oběma úkoly je schopen pomoci nástroj **APBS** (binární CLI program). Ten je schopen pro zadanou PDB strukturu a parametry okolí (pH, teplota, atd.) vypočíst *hodnoty el. potenciálu* v celém diskrétním objemu kvádru daným velikostí proteinu se zadanou roztečí bodů. Vedlejším výstupem APBS je binární určení, zda se *místo v proteinu nachází vevnitř, nebo vně*. Oba výstupy jsou ve formě volumetrického souboru ve formátu OpenDX (.dx). Ilustrace využití obou výstupů pro zobrazení povrchu proteinu a jeho kladně nabitých izoploch současně je na obrázku 4.1. Hodnota pH bude zvolena 7 pro porovnatelnost výsledků s metodou Surface Patches.

Provedeme-li analýzu pomocí APBS pro WT i mutanta, můžeme poté ze získaných dat pro oba spočítat jejich kladné izoplochy na určité úrovni potenciálu **V** – např. na +25 mV jako zvolili tvůrci Surface Patches – není však zaručeno, že je to nejvhodnější hodnota. Izoplochu z mřížky reprezentované .dx souborem je možné získat např. pomocí algoritmu *pochodujících kostek* (angl. marching cubes), který takovou plochu nalezne ve formě trojúhelníkové sítě. Nevýhodou tohoto řešení je, že je zřejmě výpočetně náročnější, než pouze označovat body mřížky jako v originální metodě. Dále produkuje nestejně velké trojúhelníky a nelze tedy velikost určit pouhým součtem vrcholů v izoploše, jako se děje v originální metodě a je nutné vypočíst její obsah. Na druhou stranu je takové řešení přesnější a je možné ho přímo vizualizovat.

Zde chci podotknout, že se tímto navržený prediktor odlišuje od originální metody, protože měří velikost izoplochy, nikoliv kladné oblasti na povrchu. Je tak možné předpokládat odlišné výsledky, přestože protein s větším kladnou oblastí bude mít větší i příslušnou izoplochu a naopak. Velikost izoplochy je totiž více spjata s nábojem dané oblasti nežli s její plochou. Při větší hustotě náboje bude izoplocha vypouklá, kdežto při nízké hustotě bude přiléhat k povrchu a více korelovat s velikostí. To by však nemělo mít negativní důsledky,

[1] Typické třeba pro data z PDB.



Obrázek 4.1: *Surface Patches.* Zobrazení povrchu (šedou barvou) proteinu (PDB: 1EER) a jeho kladně nabitých izoploch (fialovou mřížkou) po analýze PDB souboru programem APBS s výchozími předvolbami. Z tohoto úhlu to není příliš patrné, ale spodní izoplocha pokračuje dozadu a pokrývá téměř celou odvrácenou stranu povrchu proteinu. Ve skutečnosti tak pokrývá odhadem třetinu povrchu – tento protein je do jisté míry extrémem. Takto velké kladně nabité pole zásadně zvyšuje pravděpodobnost „lepení se“ na záporně nabité makromolekuly. Toto je ta izoplocha, kterou by se metoda Surface Patches snažila rozbít. Ilustrace je vyrobena programem PyMol, vč. výpočtu povrchu a mřížky z volumetrických dat získaných pomocí APBS.

protože jak bylo zmíněno v kapitole 3.2, byla prokázána i korelace mezi součtem náboje v největší oblasti a jeho rozpustností.

Jelikož zajímavý je pouze potenciál nad povrchem atomu, je třeba izoplochu oříznout povrchem molekuly. U metody Surface Patches je použitá největší kladně nabitá izoplocha, nicméně vyvíjený nástroj má jiný cíl než originální metoda. A sice predikci, nikoliv návrh mutace, proto pro zjednodušení bude počítáno se všemi izoplochami. Porovnání WT a mutanta bude provedeno podle hodnot jedné z následujících dvou proměnných:

- Součtem obsahu všech izoploch (`area`)
- Obsahem největší izoplochy (`largest`)

Zmíněné zjednodušení spočívá především v tom, že v mutantu nemusí být hledána plocha, která by odpovídala největší nalezené ploše ve WT. Taková ani v mutantu nemusí existovat – původní může být roztříštěna do více izoploch, zhroucena do větší izoplochy či vychýlena. Proto je vhodnější porovnávat pouze celkový obsah izoploch, čímž se vyhneme problému jak stanovit relaci mezi izoplochami WT a mutanta. Lepších výsledků by nicméně mohlo být dosaženo, pokud by byla nastavena mez velikosti obsahu ploch určující to, jestli budou započteny do celkové plochy. Mez by mohla být stanovena např. jako podíl z celkového součtu. Tímto postupem by se myšlenka i znovu přiblížila k myšlence Surface Patches, která se nezabývá nevýznamnými plochami.

Pro vyhledání největší plochy bude trojúhelníková síť analyzována na nalezení izolovaných celků (izoploch, které se nedotýkají). Největší je pak pochopitelně ta s největším obsahem. Pro případnou pozdější analýzu mutací, bude prediktor produkovat výstup jednotlivých nalezených izoploch ve formátu *.r3d* programu Raster3D [18], který se používá

pro vykreslování makromolekul v oblasti molekulární biologie a je podporován i programem PyMOL. Formát se je určen pro přímou interpetaci renderovacím programem, takže obsahuje také specifikaci o světelném zdroji, ap – ty však budou ignorovány a použito bude generické nastavení. Formát podporuje celou řadu primitiv jako je text, strukturní motivy, trojúhelníky a další, včetně specifikace barvy. Pro uložení izoploch bude využita trojúhelníková síť. Největší izoplocha bude vybarvena bíle pro usnadnění analýzy predikce podle největší izoplochy. Ostatní potom budou mít barvy přiřazené náhodně, aby je bylo možno odlišit při vizuálním vyhodnocení.

Pro rekapitulaci, navržený prediktor vykonává následující činnost:

1. Vytvoření mutanta `M` ze vstupního proteinu `WT` zadanou mutací rezidua `i` na aminokyselinu `x` specializovaným nástrojem
2. Provedení optimalizace `WT` ekvivalentní s tou provedenou pro `M`
3. Analýza optimalizovaného `WT`, `M` programem APBS při $\mathtt{pH} = 7$, $\mathtt{I} = 0{,}15\,\mathrm{M}$ za vzniku volumetrických souborů el. potenciálu a povrchu ve formátu .dx

   (následující kroky provádí s oběma proteiny `WT`, `M` zvlášť)
4. Vyhledání izoplochy `P` v el. poli na potenciálu $\mathtt{V} = +0{,}25\,\mathrm{mV}$ ve formě trojúhelníkové sítě
5. Odečtení povrchu proteinů od nalezených izoploch – zbývá tak pouze část nad povrchem
6. Rozdělení celkové izoplochy na její souvislé části[2]
7. Nalezení největší z izoploch $\mathtt{P}_{max}$[2]
8. Výpočet obsahu $\mathtt{S}_{max}$[2] největší izoplochy $\mathtt{P}_{max}$ a celkové sumy obsahu `S` izoploch `P`
9. Binární predikce změny rozpustnosti $\Delta\mathtt{s}$ ze získaných proměnných

[2]Tento krok není nutný, nemíníme-li použít největší izoplochu pro rozhodování.

Kapitola 5

Implementace a použití

Tato kapitola slouží jako implementační dokumentace a návod k zdrojovým souborům práce. Začíná použitím a instalací prediktoru a pokračuje 3 částmi dokumentující implementaci: prediktoru, experimentů a použitých vizualizací. Čtenář, kterého tak technické detaily nezajímají, může přejít k následující kapitole s experimenty. Programové podklady pro práci jsou vytvořeny v následujících jazycích: Python (prediktor), Shell, PowerShell (experimenty) a R (analýza). Zdrojové soubory samotného textu pro LaTeX jsou v adresáři `/text/`.

Pro použití prediktoru stačí využít připravené CLI, které je přístupné přes soubor `predictor.py`. Toto CLI zapouzdřuje všechny kroky predikce. Nápovědu je možné zobrazit standardně spuštěním `python predictor.py --help` (výstup příkazu viz příloha B). Rozhraní umožňuje:

- Volbu mutačního nástroje: FoldX nebo Modeller (výchozí, preferovaná volba)
- Nastavit míru informací tisknutých na standardní výstup o průběhu výpočtu
- Pouze vypočíst .r3d soubor s izoplochami a neprovádět predikci
- Nastavit cílovou mutaci a cílovou AK specifikovat i v 1LC i 3LC

Mandatorním argumentem je vstupní PDB soubor a v případě predikce také specifikace mutace. Oproti návrhovým požadavkům je podporována i vícenásobná mutace, nicméně nebyla použita v experimentech. Vícenásobná mutace je specifikována více pozicemi oddělenými čárkou. Cílová AK může být uvedena jedna, mají-li všechny mutace stejnou substituční AK, nebo musí obsahovat stejný počet cílových AK. Příkladem použití budiž následující příkaz: `./predictor.py 1eer.pdb predict 48,150 D ++verbose -e FoldX`, který provede 2 mutace na pozicích 48 a 150 na kys. asparagovou v proteinu 1EER pomocí FoldX s tisknutím veškerých informací. Zároveň provede globální optimalizaci vstupního PDB, vytvoří soubor s izoplochami a nakonec stanoví predikci.

Instalace: K použití prediktoru je potřeba Python 3. Instalaci vyžadovaných knihoven je možné provést pomocí skriptu `setup.py`, který v lokálním adresáři doinstaluje chybějící knihovny. Pokyny pro manuální instalaci jsou v souboru `readme.txt`. Co je však dále nutné, je nainstalovat FoldX 4 stažením binárního souboru a jeho umístěním do adresáře `/foldx/`. FoldX je volně k použití v akademické sféře a je možné stáhnout jej na stránkách FoldX Suite[1]. Alternativou je nainstalovat Modeller, který je rovněž k volnému akademickému použití.

[1] http://foldxsuite.crg.eu/academic-license-info

5.1 Prediktor

Prediktor samotný je naprogramovaný kompletně v jazyce Python 3 pro snadnou znovupoužitelnost. Je rozdělen do 3 souborů: `predictor.py` jež slouží jako CLI prediktoru a řídí celý výpočet, `isosurface.py` se samotným hledáním izoploch v elektrickém poli proteinu a `export.py` pro grafický export spočtených izoploch. Využívá tyto knihovny:

- scikit-image pro hledání izoplochy ve volumetrických datech
- gridDataFormats pro načítání volumetrických dat ve formátu OpenDX .dx
- Colour pro generování náhodných, avšak unikátních barev pro objekty
- Biopython pro převod textových reprezentací AK

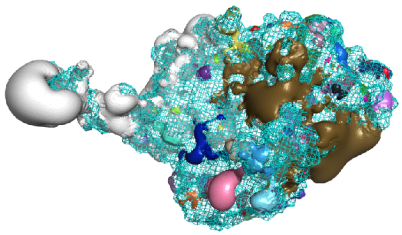
Všechny vyjmenované knihovny jsou stažitelné z repozitáře PyPI. Pro funkci prediktoru musí být dále nainstalovaný FoldX (ve formě binární aplikace) nebo Modeller (ve formě Python knihovny). Prediktor vyžaduje alespoň jeden z nich na vytváření mutantních struktur. Oba nástroje jsou sice zdarma pro akademické použití, nicméně licence neumožňují jejich šíření, což je důvod proč nejsou zahrnuty v souborech projektu. Dalšími externími nástroji, které prediktor používá jsou APBS a PDB2PQR sloužící pro výpočet elektrického pole proteinu. Jedná se o otevřené SW a je přiložen – licence umožňuje šíření.

Pro stanovení predikce jsou porovnány hodnoty mutanta a WT. Mutant vytvořený Modellerem podstupuje lokální optimalizaci v místě mutace, tak aby výsledná mutace byla fyzikálně reálná. Kdežto mutant vytvořený FoldX je vytvořen tak, že nejprve podstupuje globální optimalizací WT („opravu“, funkce RepairPDB) a v rámci provedení mutace (funkce BuildModel) je provedena optimalizace znova – nicméně připomíná spíše už lokální optimalizaci. Ve výsledku nicméně mutant z FoldX podstoupil globální optimalizaci. Při *lokální optimalizaci* nástroj pohybuje pouze s určeným reziduem a jeho bezprostředním okolím. Do úvahy se neberou všechna fyzikální omezení. Při *globální optimalizaci* se naopak pohybuje s celou strukturou a ve výsledku by měla odpovídat struktuře v solventu, nikoliv krystalové struktuře vzešlé z krystalografického experimentu.

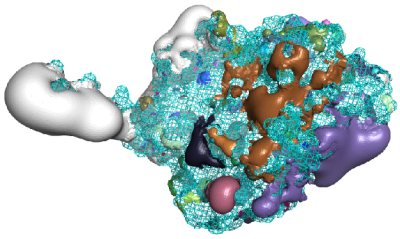
Proto se dá očekávat, že struktura mutanta se může i výrazně lišit od WT. A to nikoliv vlivem samotné mutace, nýbrž vlivem optimalizace. Proto se zdá nevhodné porovnávat přímo hodnoty získané na WT s hodnotami mutanta. Je tedy nutné rozhodnout, jakou strukturu použít jako referenční. Dále vyjmenované referenční možnosti jsou ilustrovány na proteinu PKS na obrázku 5.1. Nasnadě je porovnat mutanta z FoldX s optimalizovaným WT také pomocí FoldX (obrázek 5.1B). Toto pravděpodobně není dobrá reference pro porovnání s mutantem vytvořeným Modellerem, protože globální optimalizace FoldX je agresivnější. Vhodné však není ani porovnání s originálním WT (obrázek 5.1A), neboť v něm zase není provedena optimalizace žádná. Řešením je vytvořit lokálně optimalizovaný WT v místě mutace, ale bez provedení mutace samotné (tzv. *selfmutant*[2], obrázek 5.1C). Nepraktičností tohoto řešení spočívá v tom, že přináší režii ve formě nutnosti vypočítat selfmutanta pro každou mutovanou pozici, což v případě datasetu o velikosti Whiteheadu může být výrazná režie. Provedené experimenty (viz kapitola 6.2) potvrdily domněnku, že jediné dvě smysluplné varianty predikce jsou porovnání FoldX mutanta s glob. optimalizovaným WT a Modeller mutanta s Modellerem selfmutantem. CLI prediktoru proto nabízí na výběr jen nástroj pro tvorbu mutace mezi FoldX a Modellerem a neumožňuje vytvářet jmenované

[2]Tedy mutant získaný mutací WT sám na sebe.

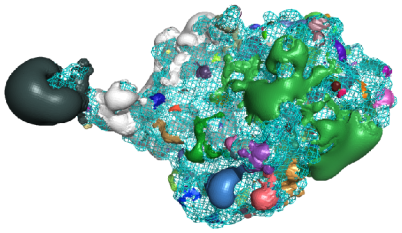
nevalidní kombinace. Přičemž preferovaná varianta je Modeller, neboť originální struktura je upravena v menší míře, což zmenšuje chybu zanesení chyby mutačním nástrojem.



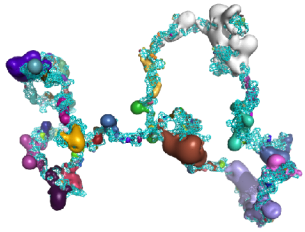
(A) Originální struktura



(B) Glob. optimalizovaná struktura FoldX



(C) Lok. optimalizovaná struktura Modellerem



(D) Lok. optimalizace struktury Modellerem aplikována globálně

Obrázek 5.1: *Ilustrace izoploch různě optimalizovaných variant WT proteinu PKS.* Každý z obrázku zobrazuje nalezené kladné izoplochy proteinu. Bíle je přitom znázorněna největší z nich. Ostatní jsou obarveny náhodně, aby je bylo snadné odlišit. Azurová mřížka představuje povrch atomu. Vlevo nahoře jsou izoplochy nemodifikované originální struktury. Vpravo nahoře je opravená struktura FoldX – globálně optimalizovaná. Je možné povšimnout si změn nejen tvaru povrchu, ale i tvaru, velikosti a souvislosti samotných izoploch. Vlevo dole je struktura lokálně optimalizovaná Modellerem. Ta se naopak liší prakticky jenom v místech provedené optimalizace (výstupek na levé straně proteinu), což zde však dokonce vede k rozpadu největší izoplochy ve dví. Vpravo dole je pouze ilustrační varianta výsledku stejné lokální optimalizace, avšak aplikované v rozsahu celého proteinu. V důsledku toho, že některé vazby jsou při ní ignorovány, dojde k rozpadu vyšších struktur proteinu.

Vzniklý mutant i optimalizovaná struktura jsou analyzovány programem APBS, jehož činnost je určena instrukcemi a parametry ve speciálním souboru s příponou *.in*. Ten je sekundárním výstupem programu PDB2PQR (vynucený přepínačem `--apbs-input`). Hlavním výstupem je PDB soubor upravený (s příponou *.pqr*, argument `--ff=parse`) pro běh v APBS, přičemž hlavní změnou je doplnění vodíků do struktury. Ty se běžně v PDB souborech vynechávají, neboť jejich pozice je možné odvodit. Výstupní .in soubor je poté upraven tak, aby APBS zapsal také volumetrický soubor s povrchem proteinu (.smol.dx) a nastavil žádané ionické síly pro výpočet. Ostatní parametry jsou ponechány – výchozí teplota je 298,15 K (přibližně 25 °C).

Výstupní soubor s elektrickým polem (.pot.dx) je prohledán na izoplochy na žádané úrovni el. potenciálu metodou Marching Cubes implementované v knihovně scikit-image. Výsledkem je trojúhelníková síť definovaná množinou vrcholů a množinou trojúhelníků danými odkazy na své vrcholy. Jsou to však izoplochy přes veškerý objem specifikovaný .in souborem. Jsou tedy následně oříznuty tak, že je ponechána pouze část nad povrchem proteinu. Použit je k tomu výstup z APBS s informací o povrchu. Trojúhelníky tvořící izo-

plochu jsou procházeny a pokud některý z nich má alespoň 2 vrcholy vevnitř proteinu, je z izoplochy odebrán. Trojúhelníková síť je dostatečně jemná na to, aby pouhé odstraňování nezpůsobovalo nepřesnosti. Výpočet celkové plochy je proveden jako suma obsahů jednotlivých trojúhelníků, jejichž obsah je vypočten známým vzorcem pro vektorový součin, resp. pomocí délky takto vzniklého vektoru.

Pro nalezení největší plochy je však nutné provést analýzu souvislosti izoploch vzniklých v předchozím kroku. Ta je provedena přes analýzu souvislostí vrcholů ve funkci `distincty _areas`. Pro každou potenciálně samostatnou plochu je vytvořen objekt typu `surface` obsahující seznam trojúhelníků ji tvořící. Zároveň jsou všechny vrcholy buď k nějaké z nich přiřazeny, nebo ještě nebyly zpracovány. Trojúhelníky jsou jednotlivě procházeny a je-li některý z vrcholů trojúhelníků součástí některé plochy, je k ní přiřazen i on. Ostatní vrcholy jsou také přiřazeny této ploše a pouze tehdy, pokud trojúhelník obsahuje konfliktní vrcholy přiřazené rozdílným plochám, dojde k sjednocení ploch a přiřazení dotčených vrcholů se aktualizuje. V případě, že žádný vrchol není přiřazen, je pro něj a trojúhelník vytvořena nová plocha. Po průchodu všemi trojúhelníky zůstává množina nespojitých izoploch, které lze obarvit.

Prediktor výsledek finální výpočetní fáze exportuje ve formátu *.r3d* (Raster3D formát) [18, 19], což je textový formát specifikující grafickou scénu ve formě zdroje světla a seznamu grafických primitiv, vyvinutý pro program Raster3D, jejž autoři popisují jako program na tvorbu „fotorealistické molekulární grafiky“. Tento výstup je zamýšlen především pro pozdější vizuálně-manuální validaci a analýzu konkrétního případu predikce v programu PyMOL (viz podkapitola 5.3), přičemž tento formát je jediný geometrický formát, který PyMOL umí načíst[3], čímž je volba jasně dána [20]. Výstupní soubor má následující gramatiku:

```
SOUBOR    := HLAVIČKA GEOMETRIE*
HLAVIČKA  := POPIS nl NASTAVENÍ
POPIS     := text                ; maximálně 80 znaků
NASTAVENÍ := ...
GEOMETRIE := TYP nl HODNOTY nl | ; geometrické primitivum
             # text nl           ; komentář
TYP       := 0 | 1 | ... | 19    ; typ geometrického primitiva
HODNOTY   := ...                 ; dle typu i na více řádcích
```

kde `nl` a `text` jsou terminály pro nový řádek a libovolný jednořádkový text. Hlavička má 20 řádků specifikující vlastnosti scény, osvětlení a kvality výstupu. První řádek `POPIS` obsahuje název souboru (např. `Generic R3D file header (Molscript V2.02)`[4]). Ze zkušenosti plyne, že hlavička je PyMOLem ignorována, výstupní soubory tedy mají generickou hlavičku pro soulad se syntaxí. Soubor obsahuje geometrické objekty reprezentované několika druhy geometrických primitiv (např. trojúhelník, koule, text, ap.) parametrizované číselnými hodnotami oddělenými bílým znakem.

Pro tuto práci je zajímavý typ 1 – trojúhelník – s formátem hodnot: `X1 Y1 Z1 X2 Y2 Z2 X3 Y3 X3 R G B`, kde první 3 trojice specifikují vrcholy trojúhelníku (pomocí reálných čísel) a poslední trojice barvu (složky barvy jsou čísla od 0 do 1, takže např. bílá je: `1.0 1.0 1.0`). Je možné specifikovat i průhlednost pomocí materiálu – typ 8 – ve formátu: `_ _ SR SG SB CLRITY _ _ _ MORE` (znak `_` značí vynechání popisu terminálu), kde trojice

[3] Zajímavostí je, že PyMOL kromě toho umí exportovat do řady dalších geometrických formátů, ale ani z jednoho z nich neumí číst. Čili nepodporuje ani jeden geometrický formát zároveň pro zápis i čtení.

[4] Molscript je jeden z programů umožňující výstup do .r3d.

`SR SG SB` slouží k nastavení barvy odraženého světla (záporná hodnota znamená, že daná složka světla není definována materiálem, ale příslušným primitivem), `CLRITY` pak průhlednost (číslo od 0 do 1 – od zcela neprůhledného po zcela průhledný). Protože materiál může být specifikován na více řádcích, hodnota `MORE` určuje počet těchto řádků navíc. Ostatní primitiva v souboru přebírají vlastnosti posledního před nimi specifikovaného (je-li takový) materiálu. Toto přebírání vlastnosti se dá zrušit primitivem typu 9. Nicméně, zdá se, že PyMOL ignoruje všechny hodnoty kromě `CLRITY` a `MORE` a také typ 9. Nastavení průhlednosti je tak možné zrušit nastavením nového materiálu s nastavením původní průhlednosti. PyMOL nepodporuje změnu průhlednosti objektu načteného z formátu .r3d, proto je potřeba průhlednost zanést do souboru.

5.2 Experimenty

Implementace experimentů je umístěna v adresáři `/experiments/`, kde každý z experimentů má svůj vlastní podadresář, ve společném adresáři jsou pouze sdílené soubory. Experimenty je zde míněna cílená produkce nových dat k analýze ze zdrojových datasetů. Naprogramovány jsou především skripty v jazyku PowerShell (ve verzi 7), ale také Shell nebo Python dle experimentu. PowerShell byl zvolen proto, že se jedná o multiplatformní systémový skriptovací jazyk. PowerShell je objektový (a tedy typovaný) jazyk s podporou paralelizace a zřetězení příkazů operátorem |[5]. Umožnil rychlé a přehledné prototypování experimentů. V experimentech slouží na řízení – spouštění jiných skriptů a binárních aplikací, předzpracování jejich vstupních dat a argumentů, postprocessing a agregaci dat do výstupních souborů.

PowerShell je také použit k paralelizaci výpočtů a parametrické generování spouštěcích Shell skriptů (`create-job.ps1`) pro cloudové výpočty. Výpočty nad datasetem Whitehead běželi totiž výlučně ve výpočetním gridu MetaCentrum[6]. Výpočet predikce pro jednoho mutanta totiž **trvá kolem 2 minut strojového času**. Výpočet na všech téměř 20 tisících mutantech tak zhruba **zabere 28 CPU dní**.

MetaCentrum je výpočetní grid, tedy spojení mnoha výpočetních clusterů. Skládá se ze sítě uzlů, které lze rozdělit na: čelní, datové a výpočetní. K čelním uzlům je možné se připojit z vnější sítě a zařadit do fronty tzv. „job“ (dále úloha) reprezentovaný cestou k shellovému skriptu. Datové uzly obsahují disková pole o velikostech stovek TB až jednotek PB volně k využití pro uživatele MetaCentra. Datové uzly jsou namapovány do souborových systémů čelních a výpočetních uzlů protokolem NFS. Na výpočetních uzlech následně běží samotná úloha. Při tvorbě úlohy je nutné specifikovat plánovanou dobu běhu, počet jader (příp. i více CPU v clusteru), vyžadovanou operační paměť a volitelně také vyžadovanou lokální diskovou paměť. Plánovač následně naplánuje úlohu k běhu v určitém časovém slotu určitého uzlu za zohlednění těchto požadavků. Je možné specifikovat požadavky i detailněji, např. na konkrétní výpočetní cluster, na konkrétní město, na CPU s podporou určitých instrukcí, ap.

Plánovač následně kontroluje, zda úloha nevyužívá více prostředků, než o kolik bylo zažádáno, a v takovém případě úlohu „zabije“. V případě překročení rezervovaného času zabíjí okamžitě. Nicméně ze zkušenosti plyne, že v ostatních případech zabíjí pouze pokud dojde k výraznému překročení – tedy asi k situaci, kdy procesy dané úlohy omezují i ostatní úlohy běžící na stejném uzlu, příp. využijí veškeré zdroje uzlu. Plánovač nesleduje množství

[5] Podobně jako roury v Shellu, nicméně zde roura přenáší „proud objektů“, nikoliv proud bajtů.
[6] https://metavo.metacentrum.cz/

současně využitých jader, nicméně průběžný strojový čas (dále *CPU čas*) nesmí překročit poměrný využitý *rezervovaný strojový čas* (= rezervovaný čas · počet jader). Proto řídící skript (`run-parallel.ps1`) spouští stejný počet podprocesů, jako je počet „rezervovaných jader“ – najednou tak může běžet i o 1 proces více, než je počet rezervovaných jader. Každý podproces zpracuje dávku mutantů přidělených řídícím procesem. Neprobouzí-li se řídící proces příliš často, tento využitý CPU čas „navíc“ je vyrovnán časem čekání na souborové operace a celkový poměrný čas není překročen. Ve většině běhů byl tento čas čekání zodpovědný za 10 až 20 % nevyužitého rezervovaného CPU času. Avšak výjimečně tato technika neuspěla, když byly souborové operace příliš rychlé a poměrný využitý čas překročil 100 % – v takovém případě bylo nutné úlohu spustit znovu s nevyužitým posledním rezervovaným jádrem.[7]

Větší požadavky při vytváření úlohy většinou přirozeně vedou na její pozdější zahájení. Avšak požadovat např. konkrétní město může být výhodné kvůli rychlosti v případě velkého datového toku mezi výpočetním a určitým datovým uzlem, neboť linky z výpočetních uzlů jsou často přetíženy. V takovém případě jednoduše dojde k zabití úlohy z důvodu překročení rezervovaného času způsobené čekáním na souborové operace. Ne všechny výpočetní uzly navíc mají přímý přístup (přes NFS) ke všem datovým uzlům. Doporučený je požadavek na přidělení lokálního místa (tzv. `SCRATCHDIR`) pro soubory. Je to už z vyřčeného důvodu, kdy souborové operace přes síť jsou velmi pomalé a určení přiměřeného rezervačního času se stává velmi problematické. Proto experimenty běžely lokálně. Vstupní skript vždy zkopíruje program a vstupní data na lokální disk a teprve potom se zahájí samotný experiment. Na konci jsou data nahrána zpět do datového uzlu. Je důležité před startem experimentu nastavit proměnou prostředí `CLEAN_SCRATCH=false`, aby nedošlo k urychlenému smazání lokálních dat po ukončení úlohy. V případech, kdy dojde k předčasnému ukončení nebo chybě je tak možné data manuálně vyzvednout pomocí SSH – výpočetní uzly jsou přes něj přístupné z čelních uzlů.

Velkou pozornost je třeba věnovat plánování času. Na úlohy, které žádají o přidělení velkého množství času, přichází řada později než na kratší úlohy. Nestačí pouze nastavit rezervační čas podle doby výpočtu 1 mutanta, neboť i délka samotného výpočtu kolísá jak rychlostí komunikace s datovými uzly, tak rychlostí lokálních diskových operací. To je ovlivněno i ostatními úlohami běžícími na clusteru. Je tak třeba dbát na to, aby v rezervovaném času byla dostatečná rezerva – např. o polovinu času více.

Uzly MetaCentra mají nainstalovanou pouze základní množinu programů a knihoven. Některé programy lze v prostředí MetaCentra donačíst příkazem `module add`. Verze těchto programů navíc nejsou aktuální a ani konzistentní mezi výpočetními uzly. Proto jsem z rejstříku předpřipravených programů využil pouze Python verze 3.6 (`module add python36-modules-gcc`). Zbytek programů byl doinstalován do uživatelského adresáře (umístěného na některém z datovým uzlů) a vyžadované Python knihovny byly nainstalovány nástrojem `pip` do lokálního adresáře. Takto je nutné při každém novém sezení aktualizovat proměnou `PATH` pro nalezení binárních programů, aktualizovat `LD_LIBRARY_PATH` o `lib` adresáře instalace APBS a Modelleru, a také `PYTHON_PATH` o adresář s lokálně nainstalovanými rozšířeními a knihovnami Modelleru. Toto předpřipravené prostředí bylo vždy na začátku experimentu přeneseno na lokální úložiště výpočetního uzlu spolu s ostatními daty.

Protože výpočet je nezanedbatelně náročný, byly zachovány výstupní soubory z jednotlivých mezikroků výpočtu pro dataset Whitehead na serverech MetaCentra. To umožňuje případnou novou analýzu bez nutnosti znovu všechny data vypočíst, nebo jejich pozdější

[7] Možná „způsobeno“ nižším využitím MetaCentra o Velikonocích.

vizualizaci. Všechny soubory jsou komprimované a to nejen z důvodů šetření místa MetaCentra, ale především pro zkrácení doby nahrávání těchto souborů z výpočetních uzlů ke konci úlohy. Jedná se o výstupy z APBS – soubory `*.pot.dx` a `*.smol.dx` s el. potenciálem a povrchem proteinu. Dále o výstup prediktoru `*.r3d` s izoplochami. Velikosti těchto souborů po kompresi jsou od 12,6 MB, 144 kB a 0,5 MB pro každou mutaci nejmenšího z proteinů TEM1 až po 39,5 MB, 460 kB a 1,3 MB největšího z proteinů LGK. Celkově přibližně 32 GB pro všechny mutanty TEM1, 138 GB pro PKS, 280 GB pro LGK a dalších 30 GB pro selfmutanty.

Analýzy výstupů z experimentů jsou v adresáři `/analysis/` rozděleny do složek dle zaměření. Tyto analýzy jsou podklady pro psaní následující kapitoly. Vytvořeny jsou v jazyce R a uloženy v tzv. R Markdown souborech (.Rmd), jež je možné pohodlně prohlížet, spouštět a upravovat např. programem RStudio. Využívají sdíleného souboru `./data.r`, který definuje některé společné funkce. Především to je funkce `load_data`, která unifikuje načítání dat z jednotlivých experimentů do R dataframů. Dále funkce `compute_changes` pro výpočet relativních a absolutních změn proměnných `area` a `largest` pro mutanty.

5.2.1 Vstupní a výstupní data

Zdrojová data pro experimenty jsou umístěny v adresáři `/data/`. Obsahují jednotlivé datové sady, PDB soubory (`./pdbs/`) a výstupy programu DSSP (`./DSSP/`) pro zkoumané proteiny. Datové sady mutací jsou normalizovány do formátu .csv a mají společné sloupce dle tabulky 5.1:

sloupec	pdb	chain	location	mutation	solubility
hodnota	`PDB ID`	`písmeno`	`přiroz. číslo`*	`1LC`	`číslo`
význam	`protein`	`místo mutace`		`substituent`	`zlepšení`

Tabulka 5.1: Formát tabulek představujících datové sady.

*V jednom případě (dataset Surface Patches) sloupec `location` obsahuje 2 pozice oddělené čárkou specifikující dvojmutaci.

Proteiny jsou specifikovány jejich PDB ID, výjimkou jsou proteiny PKS a TEM1, jež nejsou původem přímo z PDB, ale z homologního modelování a převzaty z datasetu Whitehead. Změna rozpustnosti může být binární (1 znamená zlepšení, jinak 0) – OptSolMut a Surface Patches, anebo celočíselná – Whitehead (zde je autory zlepšení stanoveno jako $> 0{,}15$). Mutace s neznámou rozpustností (většina Surface Patches) mají toto pole prázdné. Whitehead dataset je rozdělen do dvou souborů podle expresního systému – *E. coli* a kvasinka (YSD). U Surface Patches je navíc přítomen sloupec `patch_ratio`, který odpovídá naměřené relativní hodnotě největší oblasti zjištěné autory – hodnoty jsou ručně doplněny z obrázkové přílohy jejich článku.

Ne všechny experimenty byly prováděny s originálním PDB souborem, proto jsou všechny použité PDB soubory přiloženy. Je u nich prověřeno, že se nejedná o tzv. „asymetrické jednotky“ (angl. asymmetric unit), ale přirozené uspořádání (biological assembly), tak aby v experimentech nebyl šum v důsledku měření na shluku tvořeného daným proteinem, místo na něm samotném. PDB soubor 1EER je dále upraven, aby obsahoval jen samotné rHuEPO a ne celý proteinový komplex obsažený v originálním PDB, jehož je rHuEPO součástí.

Výsledná data z experimentů s mutanty jsou ve složce `/results/` (v případě výpočtu v MetaCentru) nebo ve složkách jednotlivých experimentů `/experiments/*/`. Výsledná data mají stejný formát, jako byl uveden výše, nicméně přítomné sloupce se lisší. Obsahují

navíc některé z následujících sloupců dle experimentu s následujícím významem pro každého z mutantů:

- `area`: Celkový obsah všech kladných izoploch v $Å^2$
- `largest/largest_area`: Obsah největší z nalezených izoploch v $Å^2$
- `asa`: Hodnota SASA vypočtená nástrojem DSSP v $Å^2$
- `rsa`: Hodnota RSA vypočtená nástrojem DSSP (bezrozměrná)
- `netQ`: Celkový náboj mutanta při neutrálním pH
- `posQ`: Suma částečných kladných nábojů při neutrálním pH

Obecně dále platí, že neobsahuje-li soubor sloupec `mutation` ani `location`, znamená to, že se jedná o výsledky pro originální PDB (WT). Pokud přitom ale název proteinu obsahuje sufix `_Repair`, jedná se sice o WT, ale globálně optimalizovaný nástrojem FoldX. Pokud obsahuje ze dvou sloupců pouze sloupec `location`, jedná se o WT s lokální optimalizací Modellerem.

5.3 Vizualizace

Ilustrace proteinů jsou vytvořeny pomocí nástroje **PyMOL** [21] sloužícího pro vizualizaci různých biologických dat prostorového charakteru. Podporuje jak práci se strukturálními daty (PDB), tak volumetrickými daty či geometrickými primitivy, ale i různé další typy dat. V práci byl použit PyMOL verze 2.3.

Zdrojové kódy vizualizací proteinů použité v práci se nacházejí buď v samostatných adresářích v adresáři `/figures/` (ilustrace k textu), nebo jsou umístěny přímo ve zdrojových adresářích některých analýz, pakliže jsou součástí analýzy. Vizualizace jsou zachovány v jedné ze dvou forem:

- PyMOL sezení
  Uloženo ve formátu *.pse* (PyMOL Session). Sezení ukládá data všech použitých entit v jejich momentálním stavu, jejich zobrazení (typ vizualizace, obarvení, viditelnost, aj.) spolu s definicí pohledu. Obrázek je možné exportovat použitím GUI `File/Export Image As/PNG`.

- PyMOL skript
  Skript (*.pml*) sestává z PyMOL příkazů, které tvoří jakýsi skriptovací jazyk podobný jazyku Tcl. Ne všechna funkcionalita má své API.

Nevýhodou PyMOL sezení je jeho poměrně velká velikost (v řádech desítek MB) a také to, že neuchovává posloupnost vykonaných příkazů – nelze je tedy zpětně odvodit. Naopak ale zapouzdřuje všechno ostatní do jediného souboru. Také je jedinou možností, je-li využita nějaká funkce přístupná pouze skrze GUI.

Kapitola 6

Experimenty

Tato kapitola obsahuje naměřené výsledky získané navrženým nástrojem – součet obsahu izoploch (`area`) či velikost největší z nich (`largest`) – na datasetech uvedených v podkapitole 3.3 a také na proteinu rHuEPO z článku Surface Patches pro srovnání výsledků originální a aktuální metody (kapitoly 3.2 a 4). Implementační detaily experimentů jsou v předchozí podkapitole 5.2. Kapitola začíná rozsáhlým úvodem o přístupu k experimentům – vlastní experimenty jsou v podkapitolách.

Hlavním cílem experimentů je ověřit, zda jedna z vlastností `area`, `largest` je schopna predikovat vliv mutace na rozpustnost i u jiných proteinů než u těch, na které se zaměřili autoři Surface Patches, případně při jakých parametrech predikce dosahuje nejlepších výsledků. Základní vyhodnocení bude provedeno pomocí těchto 3 indikátorů:

- Pearsonův korelační koeficient ρ
- Matice záměn a z ní vypočtené tradiční hodnoty *úspěšnost* (accuracy, ACC), *přesnost* (precision, PPV[1]) a *citlivost* (sensitivity/recall, TPR[2]), a dále také MCC (Matthewsův korelační koeficient)
- XY závislostní graf změny rozpustnosti na naměřených hodnotách

Každý z těchto indikátorů má však svoje nedostatky.

U Pearsonovy korelace je to fakt, že upozorňuje na lineární závislost mezi dvěma veličinami. Nehodí se tak pro srovnání spojitého souboru dat (naměřená data) a diskrétního souboru (binární změna rozpustnosti – dataset OptSolMut). Ačkoliv v případě zřejmé souvislosti se korelace také projeví. Druhou a závažnější nevýhodou Pearsonova koeficientu je, že ho lze zmást smíšeným souborem, kdy je soubor sestaven z několika podsouborů, jež mají rozdílné statistické rozložení (zejména průměr) obou sledovaných proměnných. To je možné si představit na libovolném datasetu o více než jednom proteinu. Proteiny mají různé elektrické pole (např. v důsledku prostého rozdílu ve velikosti molekuly) a ve výsledku i různé naměřené hodnoty `area`, `largest` – v naměřených hodnotách lze proteiny od sebe *segmentovat* (odlišit). Dále, každý protein má svoji *relativní rozpustnost* vůči svým homologům. Lze si to představit jako pravděpodobnost s jakou zlepšíme jeho rozpustnost náhodnou mutací. Jednoduše tedy dojdeme k závěru, že je-li v souboru nějaký podsoubor proteinu s výrazně odlišnou velikostí naměřených hodnot od průměru celého souboru

[1] Positive Prediction Value
[2] True Positive Rate

a zároveň odlišnou relativní rozpustností, soubor vykáže (anti)korelaci. Slabá korelace tedy může být matoucí.

Zaveďme nové relativní proměnné `*_relative` a `*_diff` – relativní a absolutní změna příslušné proměnné u mutanta vůči WT. Takto zmíněný nedostatek způsobený segmentací proteinů v naměřených hodnotách minimalizujeme. Avšak stále se lze domnívat, že větší protein bude mít i podobně menší rozsah relativní změny oproti menšímu proteinu, uvážíme-li, že stále provádíme jednobodovou mutaci. Pomocí těchto odvozených proměnných je navíc možné pohodlně porovnávat úspěšnost mutací mezi sebou. Vyrušit nedostatek zcela by bylo možné spočtením korelace zvláště pro jednotlivé podsoubory (proteiny) a agregovat výslednou hodnotu.

Matice záměn je běžný nástroj k vyhodnocování binárních predikcí. Vyhodnocovaný jev je zde změna rozpustnosti a za pozitivní případy jsou případy jejího zlepšení. Ze statistických proměnných s ní svázaných budou použity úspěšnost a především přesnost a citlivost spočtené v tomto pořadí jako:

$$ACC = \frac{TP + TN}{N} \qquad PPV = \frac{TP}{TP + FP} \qquad TPR = \frac{TP}{TP + FN}$$

kde TP, TN jsou správně klasifikované pozitivní a negativní případy a FP, FN špatně klasifikované. N je celkový počet případů.

Úspěšnost představuje poměr správně klasifikovaných případů. Jelikož ale datasety nejsou vybalancované z hlediska sledovaného jevu a konstantní predikcí zhoršení by tak bylo dosaženo úspěšností lepší než náhodného prediktoru, je úspěšnost signifikantní pouze tehdy když $ACC > \max(pos, neg)\ /\ N$ (pos/neg – pozitivní/negativní případy celkem). Přesnost je snad nejzajímavější ze tří proměnných, protože představuje poměr správně klasifikovaných pozitivních případů. Vysoká přesnost by znamenala, že prediktor dokáže poukázat na skupinu mutací, mezi nimiž se nachází ty perspektivní a především tyto má smysl prověřit experimentálně – to by vedlo k úspoře prostředků. Citlivost pak určuje poměr, kolik pozitivních případů bylo nalezeno z jejich celkového počtu.

Tyto proměnné ale nelze vyhodnocovat izolovaně, protože je jednoduché je vychýlit. Kupř. konstantní predikce zlepšení vede na 100% citlivost. Naopak v situaci (extrémní pro ilustraci) kdy $TP = 1$, $FP = 0$ a $pos = 100$ je přesnost 100 %, ale přitom je opomenuto 99 % všech pozitivních případů. Řešení je tak vyhodnocovat tyto proměnné v kontextu té druhé – hledat vysokou přesnost a zároveň nenechat klesnout citlivost příliš nízko. Pro pohodlnost ale bude uvedena také hodnota MCC, která se často používá právě v oblasti bioinformatiky [22]:

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

Tato hodnota je ve skutečnosti korelační koeficient pro binární data a samostatně umožňuje objektivně zhodnotit prediktor. Je to díky tomu, že je *symetrická*. Tzn. že obou třídám je implicitně přiřazena stejná váha a tedy není ovlivněna nevyvážeností datasetu. Stejně jako u Pearsonovy korelace nabývá hodnot od -1 (prediktor se nikdy netrefí: $TP = TN = 0$) přes 0 (odpovídá náhodě) do 1 (dokonalý prediktor: $FP = FN = 0$).

Matici záměn nelze použít samotnou, protože může zakrýt špatnou predikcí i případnou existující korelaci. Avšak také Pearsonova korelace může nezachytit závilost, pokud je jiná než lineární. P roto bude použit i závislostní graf pro vizuální evaluaci.

6.1 Komparace originální a aktuální metody

Tvůrci originální metody svoji predikci vyzkoušeli pouze na mutantech rHuEPO. Ačkoliv existují odlišnosti aktuální metody od originální (viz kapitola 4), neměli by metody dosahovat protichůdných výsledků. Avšak, zvolený protein má tu vlastnost, že velmi velká část jeho povrchu je kladně nabitá. To je také příčinou toho, že jeho el. pole ve formě izoplochy je tvořeno především jedinou souvislou plochou nadkrývající téměř všechny kladné oblasti – tvoří 90 % celkové izoplochy. Jelikož jsou tak izoplochy spočtené aktuální metodou spojeny, kdežto oblasti originální metodou jsou méně koherentně spojeny, či i oddělené, dá se očekávat, že změny hodnot u mutantů budou u originální metody markantnější. Stále by však měli zachovávat stejnou tendenci. Ze stejného důvodu je očekávatelné, že hodnoty `area_change` a `largest_change` budou navzájem velmi podobné. Surová data z experimentu jsou v souboru `/experiments/1EER/areas.modeller.csv`.

Srovnání naměřených hodnot v práci (`area`, `largest`) a v Surface Patches (`patch_ratio`) jsou na grafu obrázku 6.1 nahoře. Všechny hodnoty v grafu jsou relativní vůči hodnotám získaným na WT. Předešlé předpoklady na výsledky se tak naplňují. Přestože na první pohled se na grafu hodnota `patch_change` výrazně odlišuje od zbylých dvou, při bližším zkoumání se potvrzuje, že metody jsou výrazně podobné a odpovídají předešlým předpokladům a poukazuje na to ostatně i hodnota korelace (viz tabulka 6.1). Hovoří pro to jednak to, že u 4 hlavních sledovaných (exprimovaných) mutací je jejich vzájemné pořadí stejné u všech 3 proměnných. Jednak také to, že pro 3 sledované skupiny, rozdělené dle očekávaného efektu mutace:

1. Skupina s poklesem – bez obrácení polarity náboje (v grafu prostřední, resp. šedá)
2. Skupina s *vyrázným* poklesem – s obrácením polarity (levá, resp. zelená)
3. Skupina s nárůstem (pravá, resp. červená)

je tendence konzistentní v rámci skupiny pro každou z nich, a to opět u všech proměnných. V 1. a 2. skupině tak k poklesu vždy dojde, a opačně u 3. skupiny vždy dojde k nárůstu. Přičemž ve skupině s obrácením polarity náboje dochází častěji k výraznějšímu ponížení, než u druhé skupiny bez obrácení polarity. A to je také konzistentní bez ohledu na proměnnou.

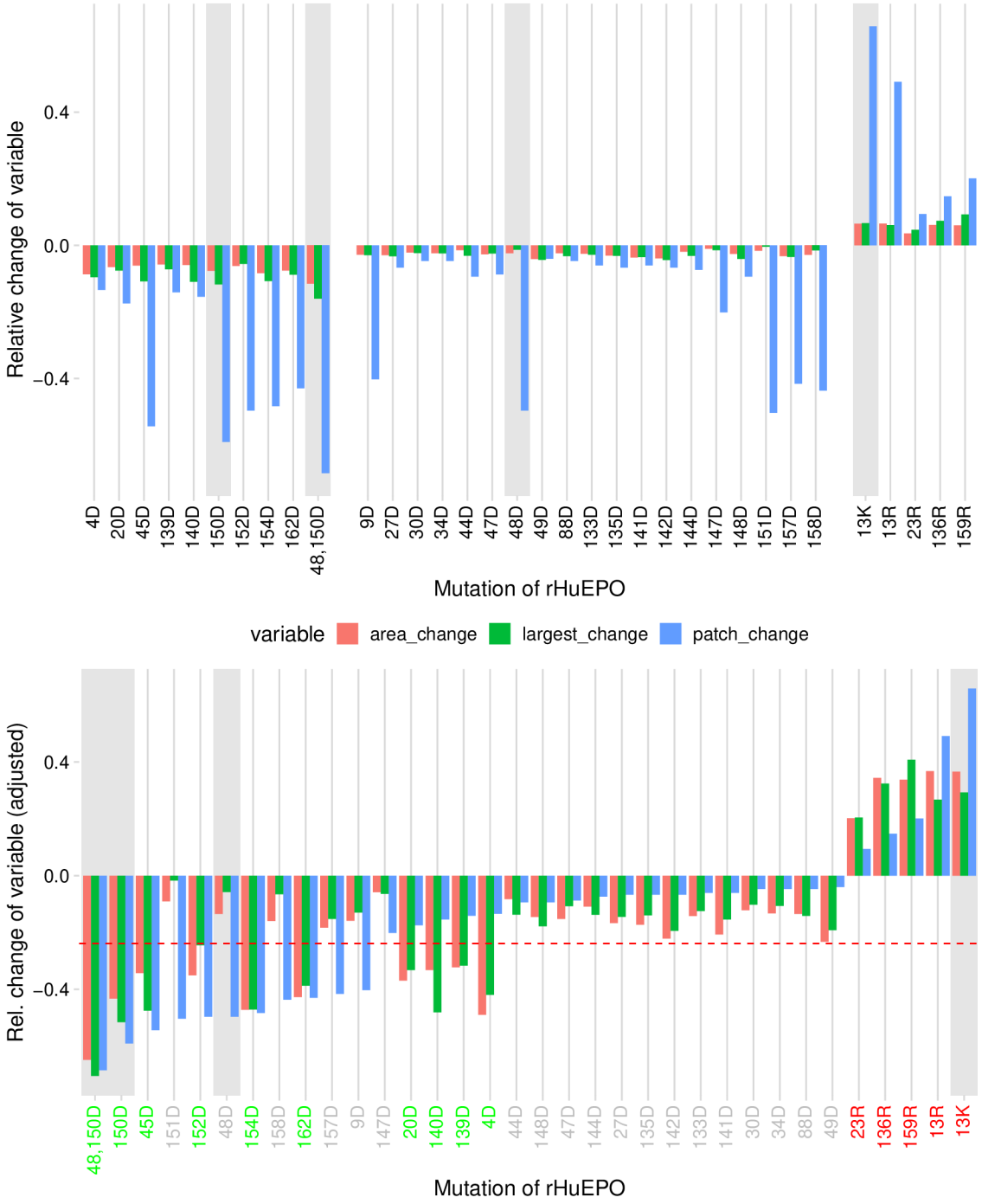
ρ	patch_change	area_change	largest_change
patch_change	1		
area_change	0.776	1	
largest_change	0.721	0.969	1

Tabulka 6.1: Pearsonova korelace mezi zkoumanými proměnnými s hodnotami získanými na datasetu Surface Patches.

Odlišnosti jsou lépe patrné, seřadí-li se data dle hodnoty `patch_ratio` – viz spodní graf obrázku 6.1. Aby byly rozdíly viditelnější a srovnatelnější, směrodatná odchylka proměnných byla normalizována (`patch_change` beze změny):

$$\mathrm{SD}(area\_change) = \mathrm{SD}(largest\_change) = \mathrm{SD}(patch\_change)$$

Ukazuje se, že aktuální metoda v případě tohoto proteinu podává výsledky podobnější teoretickému očekávání o změně celkového náboje. Neboli, je možné dle změn oddělit mutace spadající do 1. skupiny a do 2. skupiny – to je prezentováno dělící čárou na spodním grafu



Obrázek 6.1: *Graf získaných hodnot na datasetu Surface Patches* spolu s hodnotami `patch_ratio` převzatých z článku demonstrují, že jak aktuální, tak originální metoda při bližším zkoumání podávají velmi (ne však zcela) podobné výsledky. Všechny proměnné zobrazují relativní pokles/nárůst (osa y) těchto hodnot u daného mutanta (osa x) vůči hodnotám WT. Zvýrazněny jsou 4 mutace se známou rozpustností. Na vrchním grafu jsou mutace roztříděny horizontálně na 3 kategorie podle očekávání – zleva: příznivé obracející polaritu náboje (na ose zeleně), ostatní příznivé (šedě), nepříznivé (červeně). Spodní graf je seřazen, a proměnné jsou normalizovány, dle hodnot `patch_change`. Použitá metoda: lokální optimalizace Modellerem + Modeller mutant.

– hodnoty 1. skupiny jsou vždy nižší, než hodnoty 2. skupiny. U originální metody tento jev nefunguje zcela – viz levá část spodního grafu, kde jsou promíchány mutace z obou skupin.

Zatímco spousta mutací si u obou metod spíše vzájemně odpovídá, u jiných jsou výrazné rozdíly. Mutace, kdy originální metoda zaznamenala značně menší změnu než aktuální metoda mají několik možných vysvětlení. V případech, kdy originální metoda detekuje podstatně menší změnu než aktuální (např. 140D, 4D a množství mutací 2. a 3. skupiny), se pravděpodobně jedná o to, že mutace jsou zamířeny do míst, kde je i po mutaci náboj nad prahem a velikost kladné oblasti se tedy nezmění natolik, jako samotná izoplocha, která změnu reflektuje v plné míře. Opačné výkyvy jsou pak způsobeny prakticky výhradně 2. skupinou a můžou být vysvětleny tím, že úbytek stejného náboje, způsobí větší zmenšení největší kladné oblasti, než největší izoplochy. U největších rozdílů (G151D nebo F48D) ale pravděpodobně dochází k rozbití největší oblasti ve dví, čímž je vysvětlen drastický propad. Může zde ale roli hrát i pravděpodobná změna lokální konformace v důsledku změny hydrofobnosti rezidua (glycin i fenylalanin jsou hydrofobní látky), která je započtena pouze v aktuální metodě díky přítomnosti lokální optimalizace.

Dále je vhodné připomenout, že ve výsledcích článku neodpovídala zjištěná rozpustnost mutace 48D předpovědi, ale naopak by odpovídala předpovědi aktuální metodou (více viz kapitola 3.2). Kromě toho není cílem, aby aktuální metoda podávala naprosto stejné výsledky jako originální, neboť pouze 4 vybrané mutace byly prověřeny na rozpustnost. Upustíme-li od porovnávání hodnot používaných vnitřně pro predikci, pro samotnou predikci, platí, že se **originální i aktuální metoda shodují ve všech 34 případech**.

6.2 OptSolMut dataset

V této podkapitole je výkon prediktoru otestován v různých nastaveních na datasetu OptSolMut, který je variabilní ve více hlediscích, zejména obsahuje mutace různých proteinů. Mutanti byli pro každou z celkových 105 mutací datasetu vypočteni jak Modellerem, tak FoldX. U druhého zmíněného nástroje však s výjimkou mutací proteinu 1NZX. FoldX 4, ve verzi ke stažení k 23. 1. 2020[3], při výpočtu mutací pro tento protein neočekávaně ukončí činnost bez jakékoliv chybové či nezvyklé hlášky. Autoři byli uvědoměni o chybě, spolu s postupem reprodukce chyby, neprojevili však žádnou reakci.

Analýza el. potenciálu byla kromě mutantů provedena i pro originální struktury, globálně opravené (optimalizace FoldX) a lokálně opravené (selfmutanti z Modelleru). Celkově je tedy kombinací, jakým způsobem lze přistoupit k výpočtu predikce:

$$\Big|\{\text{FoldX, Modeller}\}\times \{\texttt{area\_change}, \texttt{largest\_change}\}\times \{\text{originální, glob. opravená, lok. opravená}\}\Big| = 12$$

Pro obraznost byly vypočteny hodnoty MCC pro všechny z nich, ačkoliv v kapitole 5.1 je objasněno, že v případě některých z nich se nejedná o validní přístup – např. porovnání FoldX mutanta s originální strukturou, tedy porovnání krystalické struktury se silně op-

[3]Tvůrci bohužel neuvádějí verzi ani datum sestavení programu a parametr `--version`, přítomný v nápovědě, nefunguje.

timalizovanou. Všechny hodnoty MCC jsou v tabulce 6.2. V ní je možné spatřit několik jevů:

1. Ve většině nastavení se projevuje antikorelace, a sice v některých případech (červeně zvýrazněny) na nezanedbatelné úrovni.
2. Existuje i nastavení s nezanedbatelnou korelací (zvýrazněno zeleně) – tedy fungující podle předpokladů.
3. Glob. optimalizovaná struktura v kombinaci s `area_change` nemá naprosto žádný význam – podává výsledky na úrovni náhody.
4. Pokud je použit FoldX s referencí jím neoptimalizovanou, dochází k splývání výkonosti `area_change` a `largest_change`. Při použití Modelleru k jevu nedochází.
5. Nejsilnějších (anti)korelací je dosaženo při použití originálního PDB jako reference.

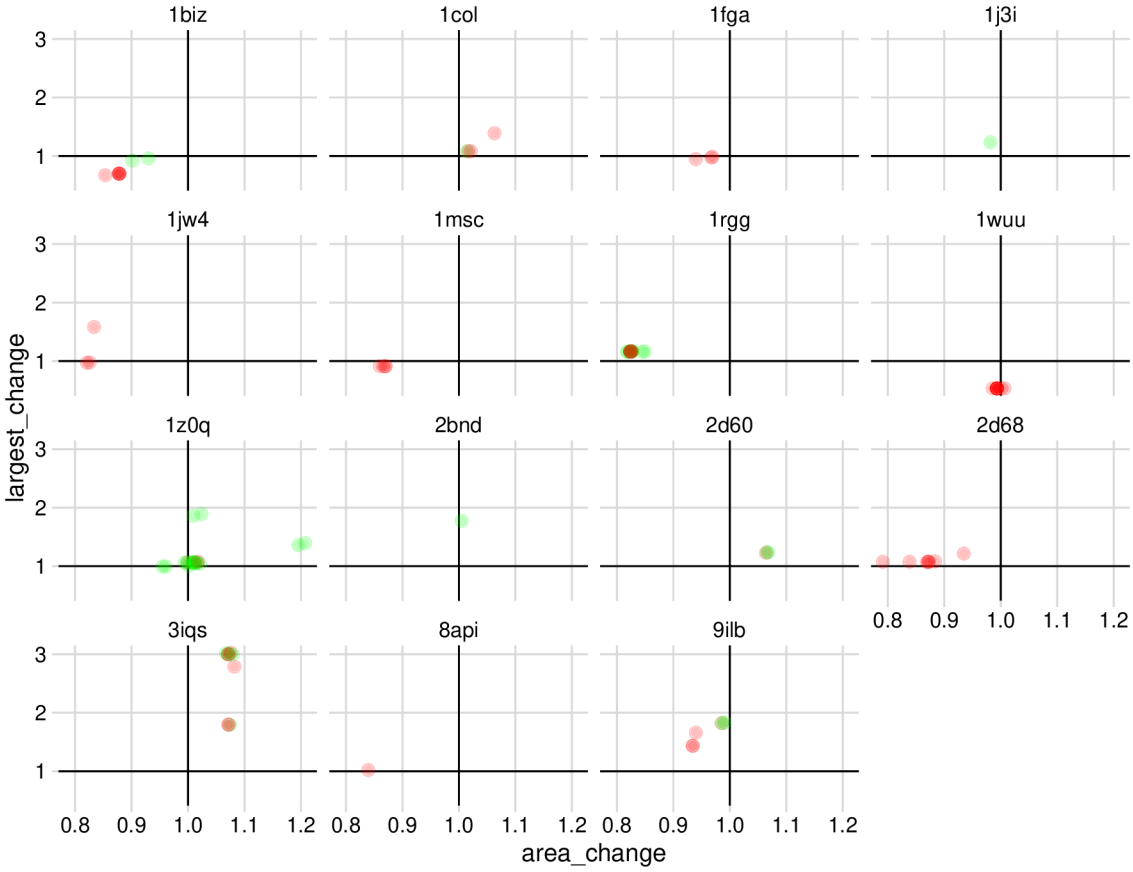
První jev by znamenal, že predikce funguje obráceně, než bylo navrženo v článku. To by ale bylo v rozporu s jevem druhým. Vodítko k tomu, čím je to způsobeno, je nasnadě při zohlednění jevu číslo 5. Glob. optimalizace nebo mutace pomocí FoldX je poměrně agresivní změna struktury – viz hodnoty FoldX mutanta vs. originální struktury. Kdežto lok. optimalizace nebo mutace Modellerem je méně význačná – viz Modeller mutant vs. originální struktura, příp. FoldX mutant vs. lok. optimalizovaná struktura. Ze stejného důvodu také mají lok. optimalizované struktury v prvních 3 sloupcích slabější korelaci než u originální struktury.

MCC: nástroj / proměnná / reference	FoldX*		Modeller	
	`area_change`	`largest_change`	`area_change`	`largest_change`
originální	−0,363	−0,365	−0,267	−0,123
glob. opravená	−0,012	−0,179	−0,045	+0,274
lok. opravená	−0,215	−0,214	−0,166	+0,105

Tabulka 6.2: *Matthewsův korelační koeficient mezi predikcí změny rozpustnosti a reálnou změnou získaných na datasetu OptSolMut.* V každé buňce je MCC spočteno podle parametrů určených v řádcích a sloupcích. Sloupce ukazují, podle které proměnné se predikuje – podle změny velikosti sumy izoploch (`area_change`), nebo největší izoplochy (`largest_change`); a pomocí kterého z obou nástrojů je vytvořen mutant. Řádky určují referenční strukturu pro výpočet změny. Červeně zvýrazněny jsou výrazné antikorelace, kdežto zeleně jsou korelace. Výsledky je možné číst tak, že funguje-li na tomto datasetu metoda Surface Patches, funguje spíše opačně, než bylo zamýšleno.
*Pro FoldX nebyla spočtena ani 1 ze 3 mutací struktury PDB 1NZX, neboť s ní měl nástroj potíže.

Příčinu výskytu jevu 5 je možné odhalit na obrázku 6.2. Graf zobrazuje změnu obou proměnných pro jednotlivé mutace rozdělených do mřížky dle proteinu. Je vidět, že téměř všechny mutace v rámci jednoho proteinu jsou vychýleny stejným nebo podobným směrem

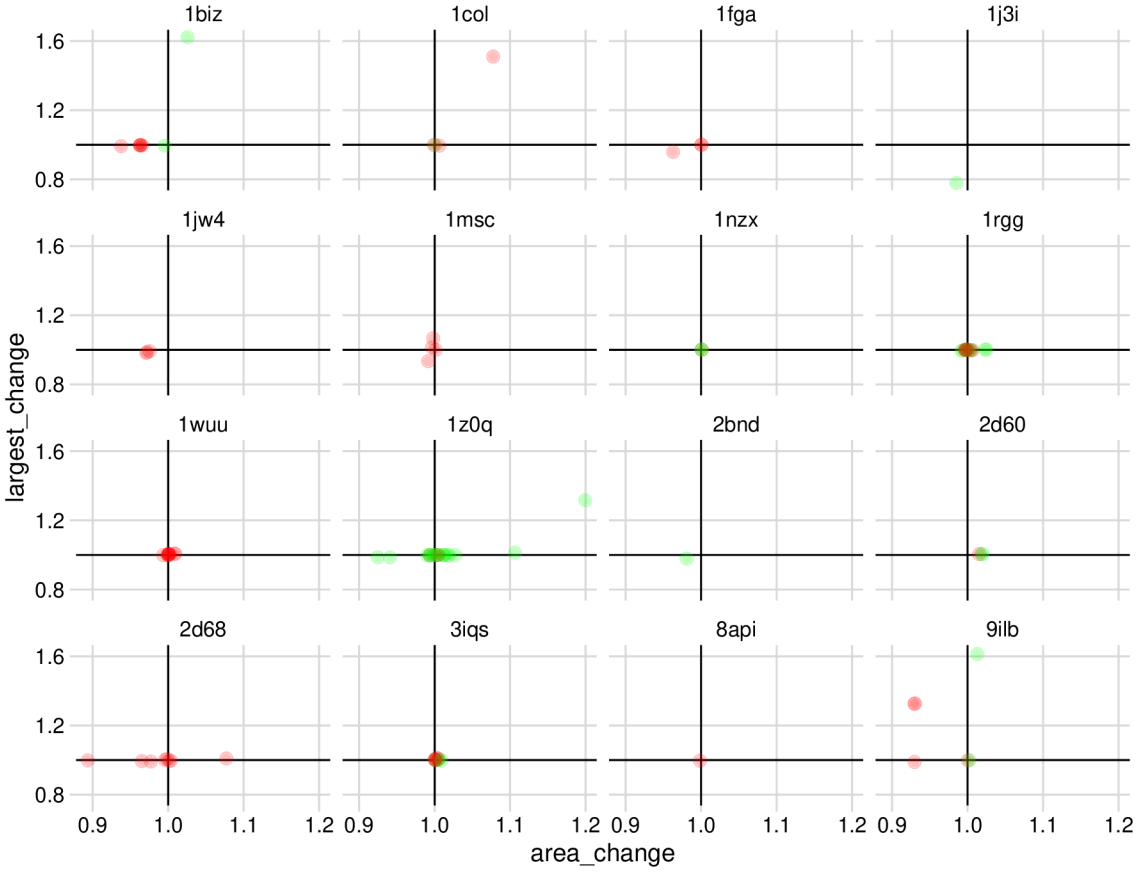
od středu, a to i ve stejné míře. Kromě proteinů 1COL a 1Z0Q je to možné pozorovat na všech ostatních. Nejlépe však na proteinu 1RGG, kde se mutace shlukují v tomtéž místě vzdáleného od středu nehledě na to, zda se jedná o příznivou, či nepříznivou mutaci. Tento efekt spolu s tím, že dataset je nevyvážený vzhledem na příznivost mutace (61 nepříznivých vs. 44 příznivých) vede na zdánlivou korelaci. Tím je tedy vysvětlen původ antikorelace u všech variant, které mají jako referenci originální strukturu. Ale také o něco slabší korelaci mezi FoldX mutanty a lok. optimalizovanými strukturami, neboť optimalizace FoldX je výraznější. A tedy tímto je vysvětlen i jev první. A ve skutečnosti je toto i příčinou jevu druhého, kde jsou pouze nástroje FoldX a Modellerem použity naopak (FoldX s výraznější glob. optimalizací pro referenci a méně výrazný Modeller pro mutaci), což vede na obrácení znaménka korelačního koeficientu.



Obrázek 6.2: *Změny sledovaných proměnných u FoldX mutantů seřazených v mřížce dle proteinu.* Změna je určena relativně vůči originální PDB struktuře. Body v grafu jsou jednotlivé mutace – zelené jsou příznivé, naopak červené nepříznivé. Body ve středu zvýrazněných os představují mutace bez velkých změn proměnných. Nadpisy jednotlivých grafů specifikují protein podle jeho PDB ID. Graf zobrazuje nastavení, jež je zajímavé tím, že v predikci vykazuje významnou korelaci se skutečnou změnou rozpustnosti. Avšak tato mřížka odhaluje, že se jedná o korelaci falešnou. Všechny mutace téměř každého z proteinů jsou totiž značně vychýleny od středu stejným směrem (nejzřejmější to je na proteinu 1RGG).

Ačkoliv obrázek 6.2 zobrazuje mutace z FoldX. Odpovídající graf pro Modeller by byl prakticky identický. Jak situace vypadá, je-li použita validní nastavení predikce je na ob-

rázku 6.3, kde jsou mutanti vytvořeni Modellerem a jako reference jsou použity ním lok. optimalizované struktury. Lze vidět, že hodnoty v grafu jsou nyní vždy vycentrovány kolem jeho středu.



Obrázek 6.3: *Změny sledovaných proměnných u Modeller mutantů seřazených v mřížce dle proteinu.* Zde je změna, na rozdíl od předchozího grafu, určena vůči lokálně optimalizované struktuře WT stejným nástrojem. Body v grafu jsou jednotlivé mutace – zelené jsou příznivé, naopak červené nepříznivé. Nadpisy jednotlivých grafů specifikují protein podle jeho PDB ID. Oproti předchozímu grafu jsou zde mutace vycentrovány kolem středu (hodnoty [1,1]) a jedná se o příklad validního nastavení prediktoru.

Pro ty z validních nastavení je čistý Pearsonův korelační koeficient mezi proměnnou a změnou rozpustnosti následující $\rho =$

- FoldX `area_change`: 0,204
- FoldX `largest_change`: 0,205
- Modeller `area_change`: 0,250
- Modeller `largest_change`: 0,047

Tyto vyšší hodnoty při porovnání s nižšími výsledky MCC v předchozí tabulce (viz 6.2) vedou k úvaze, zda by nebylo vhodné použít jiný než návrhový práh (1,0) pro predikci. Tedy

použít dataset OptSolMut jako trénovací a výslednou hodnotu otestovat na jiném datasetu. Pokud bychom toto trénování provedli pro kombinaci Modelleru, lokální optimalizace a `area_change` (nejvyšší ρ), jako nejvhodnější práh se bude jevit hodnota 1,01, tedy zvětšení izoplochy o 1 %. Pro tento optimální práh je MCC = −0,360, kdežto pro návrhový práh je to MCC = −0,166. To je podstatné zlepšení. Avšak kromě toho, že pro jiný práh změny této fyzické hodnoty by se těžko hledalo fyzikální vysvětlení, je zde i jiný problém. Pokud z datasetu odstraníme mutace proteinů 1WUU, 1FGA a 3IQS, dosáhneme při návrhovém prahu velmi podobné hodnoty: MCC = −0,300. Toto je výsledek po odstranění proteinů, jež „kazí“ predikci. Je to způsobeno tím, že dané proteiny mají dvě společné vlastnosti. Každý z proteinů má (téměř) všechny své mutace s hodnotami `area_change` v rozmezí 1,0 a 1,01. A zároveň jsou všechny jeho mutace, nebo téměř všechny, nepříznivé. Z toho vychází, že **tato optimalizace ve skutečnosti silně optimalizuje prediktor na tento konkrétní dataset**, resp. na jeho nedostatky. Tím hlavním nedostatkem je v tomto případě absence mutací relativně dobře rozpustného proteinu větší velikosti (tj. mající vícero příznivých mutací, a přitom s malými výkyvy `area_change`). Jinými slovy, optimalizovaný práh rozděluje mutace spíše podle jejich příslušnosti k určitým proteinům, než podle samotné podstaty proměnné `area_change`.

Závěry z otestování metody na datasetu OptSolMut jsou následující: Je důležité použít prediktor ve validním nastavení, jinak šum značně převýší skutečné změny měřených proměnných. Prediktor dosahuje korelace pouze MCC = −0,179 pro FoldX/`largest_change` a MCC = −0,166 pro Modeller/`area_change`. Dále, `area_change` nefunguje jako kladné oblasti v Surface Patches, ale spíše naopak (zvětšení, a nikoliv zmenšení, predikuje zlepšení). `largest_change` směrově funguje jen s Modellerem a funguje opačně s FoldX. Z tohoto důvodu se jeví `area_change` jako spolehlivější. Nepřesvědčivé výsledky prediktoru by bylo možno viditelně zlepšit trénováním prahu za předpokladu, že by k tomu posloužil dataset, jehož už tak nevelká velikost by se vyvážením nedostala pod únosnou mez.

6.3 Dataset Whitehead

Tento dataset se svými tisíci mutanty vyžadoval odlišný přístup. Predikce byly vypočteny s velkou pomocí gridu MetaCentrum[4] dostupného volně pro českou akademickou sféru. Bylo tam **spotřebováno 117 dnů procesorového času a vyprodukovány byly stovky GB dat**. Z toho více než 500 GB komprimovaných dat z mezivýpočtů bylo ponecháno pro případnou reanalýzu, nebo pro ušetření výpočetního času při příp. změně výpočtu predikce, jež by neovlivnila všechny fáze výpočtu. Vypočteno bylo 6937 mutací proteinu LGK, 4840 mutací proteinu PKS a 2470 mutací proteinu TEM1. Všichni mutanti byli spočteni pomocí Modelleru a zároveň většina také pomocí FoldX. Dohromady také Modellerem bylo vypočteno 891 lokálních optimalizací (selfmutantů) pro všechny z mutovaných pozicí. Velkou výhodou oproti datasetu OptSolMut je zde, že obsahuje i míru změny rozpustnosti mutace. Data z tohoto experimentu jsou v adresáři `/results/`. Při uvážení množství daných mutací v tisících a toho, že výběr mutací je statisticky validní (dataset obsahuje všechny varianty – není vzorkován) jsou všechna dále prezentované data statisticky významná na hladině $p = 0{,}05$.

Všechny zkoumané varianty mutací z Modelleru jsou v tabulce 6.3. Zjištěné korelace jsou minimální a vymyká se jim pouze korelace u proteinu TEM1. Navíc i tyto mizivé korelace nejsou konzistentní napříč různými proteiny a různými expresními systémy. Je

[4] https://metavo.metacentrum.cz/

možné vyvodit závěr, že skutečně velikost kladné izoplochy souvisí s rozpustností proteinů. Zároveň je zřejmé, že ne natolik, aby to bylo použitelné všeobecně. Je pravděpodobné, že prediktor může fungovat pro některé skupiny proteinů mající specifickou vlastnost. Bez bohatšího datasetu je však těžké o této vlastnosti spekulovat. Pro doplnění, mutanti pro expresní systém *E. coli* byli vypočteni také pomocí FoldX se zjištěnou korelací $\rho = -0{,}024$.

Metoda nejlépe funguje pro TEM1 (samostatně), kde je hodnota MCC pouhých 0,086. Graf pro tento protein je na obrázku 6.4. Potvrzuje také intuitivní předpoklad, že `largest_area` dosahuje mnohem větších změn u mutantů vůči WT než `area` a to od zmenšení na polovinu až na zvětšení více než na dvojnásobek.

ρ	*E. coli*			YSD		
proteiny	oba	LGK	PKS	oba	LGK	TEM1
`area_change`	−0.020	0.013	−0.034	−0.068	−0.034	−0.137
`largest_change`	−0.002	0.051	−0.048	−0.025	−0.015	−0.085

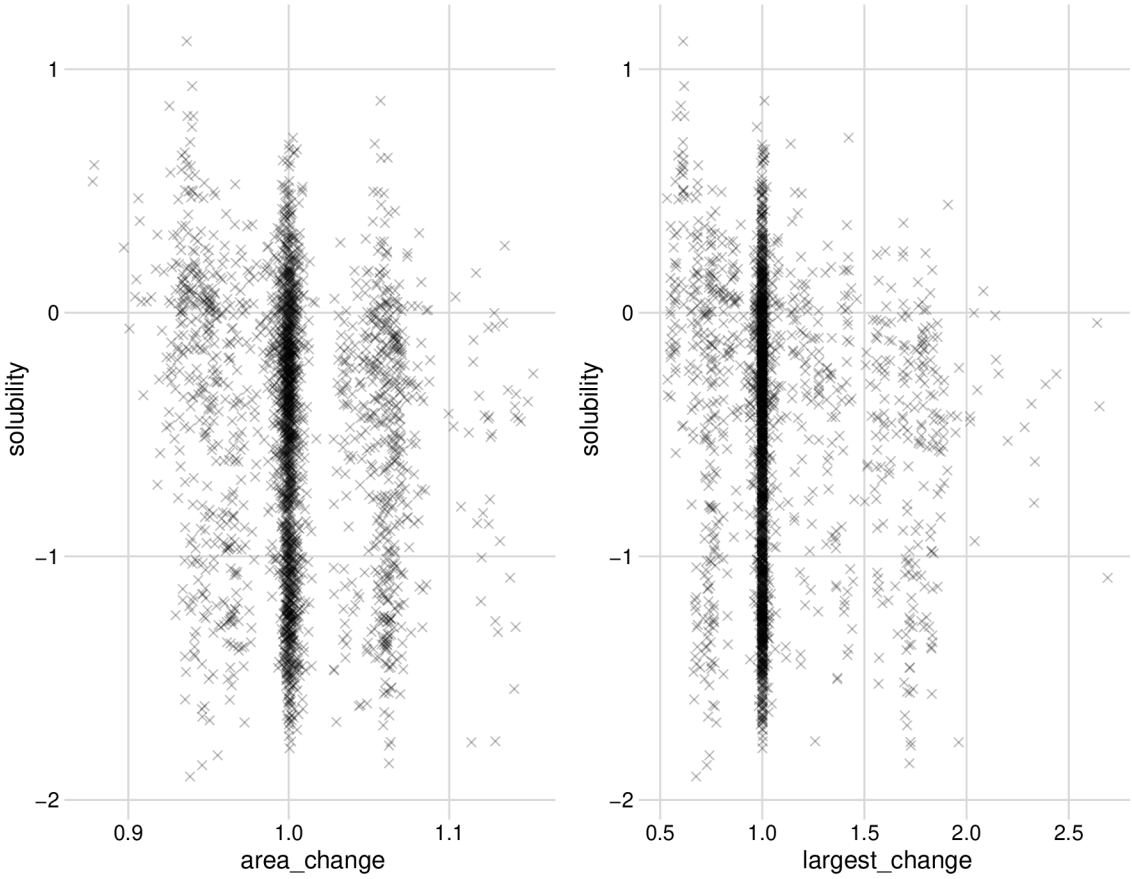
Tabulka 6.3: *Pearsonův korelační koeficient mezi změnou rozpustnosti a sledovanými proměnnými na datasetu Whitehead.* Mutace jsou rozděleny podle expresních systémů. Zeleně zvýrazněna je proměnná, která podává lepší výsledky pro danou množinu mutací. Avšak jediná množina, na které predikce funguje výrazněji, je osamocený protein TEM1 v expresním systému YSD pro proměnou `area_change`. Statisticky významné jsou však všechny zvýrazněné hodnoty.

6.4 Alternativní experimenty

Všechny 3 predikční proměnné zkoumané v práci, tedy jak `area_change` s `largest_change` v aktuální metodě, tak `patch_ratio` v originální metodě (souhrnně *plošné proměnné*), jsou zřejmě a silně provázány s nábojem obsaženým v proteinu. Nasnadě je tak otázka, neexistuje-li elementárnější vlastnost proteinu, která tyto proměnné spojuje a je snadnější ji vypočíst. Surface Patches totiž vyžaduje spočítat model elektrické pole v objemu proteinu a jeho okolí pro predikci, což je výpočetně i paměťově náročná operace. Navíc úměrná (fyzické) velikosti proteinu.

V kapitole 3.2 je vysvětleno, že principem metody je hledáním mutací na povrchu proteinu v neprospěch lokálních kladných nábojů. Měření efektu mutace je počítáno v originální metodě jako změna velikosti největší kladné oblasti (v té je provedena mutace) na povrchu proteinu. V aktuální jako změna kladně nabité izoplochy v prostoru nad povrchem proteinu. Rozdílem je tak de facto způsob měření efektu mutace, avšak cílem je v obou případech totéž.

Tyto proměnné budou pochopitelně svázány s celkovým nábojem proteinu, nebo podobnou veličinou. V kapitole 2.1 je vysvětleno, jak vzniká náboj v aminokyselině. Každá aminokyselina tak při daném pH může (ale nemusí – v závislosti na pH) nést tzv. *částečný náboj*. Sečtením těchto částečných nábojů získáme *celkový náboj*. Označme jej jako `netQ`. Výpočet je možné provést jako: `netQ` = `posQ` − `negQ`. Tedy celkový náboj je rozdíl *sumy*



Obrázek 6.4: *Bodový graf změn sledovaných proměnných a rozpustnosti pro mutace proteinu TEM1* z expresního systému YSD datasetu Whitehead. Každý bod v grafu představuje jednu mutaci. Ačkoliv korelace je velmi nízká, na grafu je poměrně zřetelně vidět – rozpustnost mutantů (osa y) se zvyšuje s poklesem velikosti izoploch (osa x). Může se zdát neintuitivní, že `largest_change` zůstává častěji naprosto neměnná, ale je potřeba připomenout, že ne každá mutace ovlivňuje největší izoplochu.

částečných kladných nábojů (`posQ`[5]) a sumy částečných záporných nábojů (`negQ`). Souhrnně je nazvěme *nábojové proměnné*.

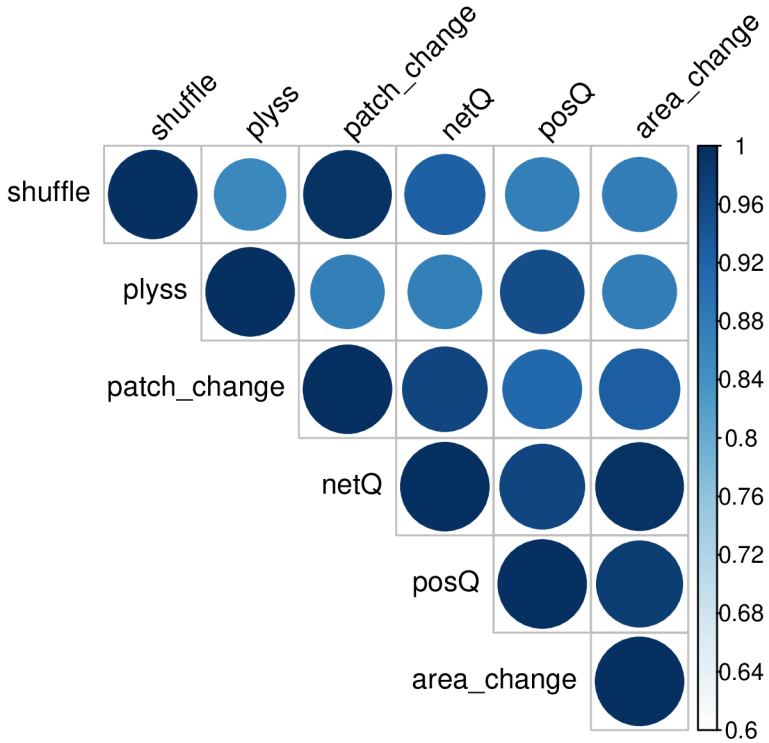
Metoda ale ve skutečnosti explicitně počítá pouze s částečným kladným nábojem, neboť počítá pouze s kladnou oblastí, nebo kladnou izoplochou. Záporný náboj je zahrnut do výpočtu pouze implicitně tím, že může oblast/izoplochu zmenšit, pokud se v ní, či její blízkosti, octne. Z tohoto důvodu je možné se domnívat, že zkoumané proměnné si budou odpovídat spíše s hodnotami `posQ` nežli `netQ`. Na druhou stranu `posQ` nebere v úvahu vůbec žádné záporně nabité AK, což může být vnímáno jako druhý extrém. V situacích, kdy dojde ke zmenšení oblasti vlivem mutace **neměnící polaritu náboje** v mutované pozici (tj. odstranění záporného náboje bez zavedení opačného náboje), bude zlepšení indikovat pouze `netQ` a nikoliv `posQ`.

Změna nábojových proměnných se od plošných bude vymykat ještě v případech, kdy dojde ke spojení nebo segregaci nabitých ploch. Spojení/rozdělení mutací např. dvou ploch

[5] V článku Surface Patches se objevuje stejně pojmenovaná proměnná. Tam má nicméně význam hodnoty `patch_ratio` pro největší kladnou oblast.

teoreticky může vést na velkou změnu plošné proměnné, ale nábojovou příliš nezmění. Z hlediska cíle této práce, a sice binární predikce, je tento rozdíl nepodstatný – tendence zůstává. K situaci ale také může dojít v důsledku změny povrchu/tvaru proteinu, a to i beze změny samotného náboje. V takovém případě již nebude zachována ani tendence. Není však pravděpodobné, že by takových speciálních situací bylo mnoho.

Cílem článku Surface Patches bylo především představit a najít lépe rozpustné mutace pro protein rHuEPO. Autoři však své výsledky svým narativem také pokládají za důkaz funkčnosti jejich metody. Avšak zde nastolenými otázkami se nezabývali. Proto pro každou mutaci z článku byly vypočteny nábojové proměnné – data viz `/data/experiments/net-charge/`. Korelace mezi všemi zmíněnými proměnnými na mutantech rHuEPO jsou na obrázku 6.5. Jsou také zahrnuty hodnoty[6] naměřené rozpustnosti z článku v expresních systémech SHuffle a pLysS, které autoři provedli pro 4 z mutací.



Obrázek 6.5: *Korelační matice mezi proměnnými na 4 mutantech proteinu 1EER z článku Surface Patches.* Proměnné `shuffle` a `plyss` označují naměřenou rozpustnost ve stejnojmenných expresních systémech. Další dvě proměnné – `netQ` a `posQ` – jsou celkový náboj proteinu a suma jeho kladné části v tomto pořadí při neutrálním pH. Sytost barvy (viz legenda) a velikost kruhu představují míru korelace. Všechny proměnné velmi korelují, proto je, pro zvýraznění odlišností, osa ohraničena na hodnoty korelace od $\geqslant 0{,}6$. Zajímavá je korelace mezi `shuffle` a `netQ` a mezi `plyss` a `posQ`.

Graf ukazuje, že na těchto 4 mutacích je možné vzít jako predikční proměnou snad libovolnou proměnnou svázanou s nábojem a predikce stále bude fungovat. Jistě nelze autorům vyčítat, že podrobně neanalyzovali více mutací, což by bylo nákladné. Co je ale možné vyčíst, je volba těchto mutací. Problémem je především ostrá korelace mezi `patch_change`

[6]Hodnoty jsou graficky odečteny z publikovaného grafu – autoři prosté hodnoty neuvedli. Pro toto použití je to postačující.

a `netQ`. K tomuto zjištění měli autoři přijít také a připravit alespoň jednu mutaci, která je mimo tento vzor. Tedy např. se snížením `netQ` při současném zachování `patch_ratio` – např. příznivá mutace v záporné oblasti.

Přesto nelze autorům upřít úspěšnost u systému SHuffle vyjádřenou následující „korelační netranzitivitou“: $\rho(\texttt{patch\_change}, \texttt{shuffle}) \approx \rho(\texttt{patch\_change}, \texttt{netQ}) > \rho(\texttt{netQ}, \texttt{shuffle})$. Tedy přestože `patch_ratio` velmi dobře koreluje jak s `netQ`, tak s rozpustností v tomto systému, korelace mezi rozpustností tam a `netQ` zdaleka není tak silná. Nicméně na tento výsledek je nutné pohlížet s vědomím toho, že u systému pLysS naopak rozpusnost koreluje nejlépe s `posQ`.

Na proteinu TEM1, z datasetu Whitehead, na němž jediném vytvořený prediktor funguje, bylo ověřeno, že i zde nábojové proměnné silně korelují s plošnými proměnnými. Avšak ne tak silně jako v předchozích případech. Nejlepší korelaci si zde zachovává `netQ` a `area_change` s $\rho = 0{,}96$. Ostatní varianty zdaleka tak nekorelují. Avšak `netQ` koreluje lépe než `posQ` i s `largest_change`, a sice $\rho = 0{,}61$.

Nicméně, navržený prediktor se nesnaží predikovat míru změny, ale pouze její směr. Proto je porovnání pouze korelací zavádějící. Všechny mutace TEM1 proto byly analyzovány na přítomnost 7 vlastností. Z nichž 4 jsou definovány jako pokles/zmenšení hodnoty `posQ`, `netQ`, `area` a `largest_area` oproti WT. Další dvě jsou platnost alespoň jedné z a platnost obou předchozích dvou zmíněných. Poslední vlastnost je zlepšení rozpustnosti. Koincidence mezi vlastnostmi je v koincidenční matici na obrázku 6.6. Vyjmenované vlastnosti odpovídají pořadím vlastnostem v hlavičce řádků. Koincidenční matice je v %, nikoliv v absolutním počtu. Buňky tak lze také číst jako podmíněnou pravděpodobnost P(řádek | sloupec) – tedy pravděpodobnost, že platí vlastnost řádku za předpokladu, že platí vlastnost sloupce.

Nejzajímavějšími místy tabulky je první a poslední sloupec a rovněž poslední řádek. Poslední řádek odhaluje, že všechny vlastnosti můžou být použitý pro predikci a budou úspěšnější než náhodný prediktor, který by se trefil v 1 případě z 10. Nejlepší přesnost má `posQ`, která se trefí v 1 z 5 případů – následuje matice záměn:

		Skutečnost		
		Zlepšení	Zhoršení	Celkově
Predikce	Zlepšení	62	241	303
	Zhoršení	191	1973	2164
	Celkově	253	2214	2467

Favorizuje pouze 303 z celkových 2467, z nich však najde pouze 62 rozpustnějších mutací. Citlivost je tedy 24,5 %. Pokud je cílem najít efektivně co nejvíce rozpustných mutací, je lépe využít souhlasný pokles `largest_area` a `area`, tedy vlastnost `both`:

		Skutečnost		
		Zlepšení	Zhoršení	Celkově
Predikce	Zlepšení	120	688	808
	Zhoršení	133	1526	1659
	Celkově	253	2214	2467

Tehdy najde téměř polovinu (citlivost: 47,4 %) všech zlepšujících mutací za procházení jen třetiny celkového počtu. Ve většině případů ale bude preference pro přesnost, protože možných mutací proteinu je zkrátka příliš mnoho.

Analýza koincidence ukazuje, že funguje-li metoda Surface Patches pro daný protein, fungují i značně jednodušší proměnné `posQ` a `netQ`. Navíc v podstatě platí, že vlastnost

Obrázek 6.6: *Koincidenční matice vlastností mutantů proteinu TEM1.* Každá vlastnost představuje změnu proměnné vůči WT danou znamínkem. Vlastnost `all` platí pro všechny mutanty. Vlastnosti `aone-`, `both-` představují platnost alespoň jedné z vlastností `area-` nebo `larg(est)-`, nebo obou vlastností zároveň. Koincidence je uvedena v %. Hodnoty v matici jsou podíl Ř řádkové vlastnosti v mutantech s vlastností S sloupce – P(Ř | S). Například P(solu+ | posQ-) = 0,205 znamená, že pouze 1/5 mutantů, jimž se `posQ` sníží, se také zlepší rozpustnost (`solu`) – to je *přesnost*. Zároveň platí P(posQ- | solu+) = 0,245 – u mutantů s lepší rozpustností se pouze v 1/4 případů také snížila `posQ` – to je *senzitivita*. Celkový počet mutantů (100 % `all`) je 2467.

`area-` jednostranně zahrnuje `netQ-`, která zase pochopitelně zahrnuje `posQ-`. Opačným směrem nedochází k zisku téměř žádné informace. Pro `largest-` platí podobný závěr, nicméně už méně odpovídá nábojovým proměnným. K situacím, kdy sice dojde k snížení celkového náboje, ale izoplocha naroste dochází jen výjimečně. To tedy znamená, že **aktuální metoda je snadno překonatelná** a pro původní záměr (hledání mutantů s lepší rozpustností) pravděpodobně nenajde uplatnění v praxi.

Kapitola 7

Závěr

V této diplomové práci jsem čtenáře seznámil s problematikou rozpustnosti proteinů, byly vysvětleny a zavedeny související pojmy z molekulární biologie. Dále jsem popsal stávající (odlišně fungující) nástroje na predikci rozpustnosti nebo její změny, rozvedl problematiku predikce změny rozpustnosti ze struktury a shromáždil 2 datové sady použitelné pro vývoj nástroje predikujícího změnu rozpustnosti z tohoto typu dat. Druhá z těchto sad vznikla teprve v roce 2019 a jedná se o první velký dataset pro tento druh problému.

Surface Patches, článek s poměrně novou metodikou na hledání lépe rozpustných mutací proteinů, jsem rozebral podrobněji. S inspirací z tohoto článku jsem provedl návrh nástroje pro predikci změny rozpustnosti (prediktor) a implementoval jej v Pythonu. Některé části jsem se rozhodl udělat jinak, než v citovaném článku. U jiných, ke kterým autoři nenapsali podrobnosti, jsem musel analyzovat možnosti a rozhodnout samostatně. V obou případech jsou však rozhodnutí zdůvodněna.

Nástroj dokáže spočítat změnu obsahu kladných izoploch elektrického potenciálu proteinu a jeho mutanta, a podle toho predikovat, zda se jedná o příznivou mutaci, či nikoliv. Vstupem je pouze PDB struktura proteinu a zamýšlená mutace. V současné době je to jeden z mála nástrojů, který k predikci využívá především strukturu proteinu. K nástroji jsem vytvořil intuitivní CLI a nápověda k němu je přiložena v příloze B.

S prediktorem jsem provedl řadu experimentů za účelem prověření funkčnosti. Nejprve byl ověřen vůči výsledkům z citovaného článku na 34 mutantech proteinu rHuEPO, kde dochází prakticky k naprosté shodě. K zajímavému obratu dochází na datové sadě OptSolMut čítající 105 případů mutací. Prediktor tam funguje velmi nepřesvědčivě, a navíc v opačném směru. To znamená, že by byl úspěšnější kdyby se směr predikce obrátil. Je to však způsobeno spíše nevyvážeností datasetu na příznivost mutací dle proteinů a také jejich velikostí. Na datasetu Whitehead funguje pouze na jediném ze 3 proteinů – TEM1.

Zjištění, že prediktor v experimentech neobstál, není jediný výstup. Z provedených experimentů je zřejmé, že metoda přece jen na některých proteinech funguje. Zřejmě tak může fungovat na skupině proteinů splňující určitou vlastnost, kterou by mohla být např. neutrálnost celkového náboje proteinu, či jeho velikost. Tímto směrem by mohlo vést další úsilí, z použitých datasetů to totiž zjistit nelze. Neobsahují na to dostatek dat.

Nejpodstatnější zjištění je však, že metoda dosahuje téměř maximální korelace se změnou celkového náboje proteinu, což je proměnná, jejíž výpočet je mnohem jednodušší a není k němu třeba ani znalost struktury proteinu. Analýzu na tento jev autoři Surface Patches opomněli provést. Kromě tohoto opomenutí jsem zjistil, že autoři tuto metodu testovali na proteinu rHuEPO, jehož povrchové elektrostatické pole (utvářející tzv. rozhraní) je podstatné pro jeho funkci. Mutace byly dokonce provedeny přímo v místě rozhraní, jak jsem

v práci demonstroval, tedy s vysokou šancí na poškození funkce proteinu. Výběr mutací také nebyl proveden optimálně, nicméně to souvisí s opomenutím analýzy korelací s triviálními proměnnými.

Mnou navržená (aktuální) metoda má několik rozdílů oproti originální metodě z článku Surface Patches. Je to už to, že originální metoda si dává za cíl predikovat rozpustnost jako takovou za stanovení jisté umělé hranice rozpustnosti. Kdežto aktuální metoda má za cíl predikovat změnu rozpustnosti pro mutanty. Hlavními rozdíly ve výpočtu jsou to, že povrch proteinu není aproximován do spojnic bodů v mřížce, ale je reprezentován přesnější trojúhelníkovou sítí. Dále to, že se pro predikci používá velikost kladné izoplochy namísto velikosti největší kladně nabité oblasti na povrchu. A potencionálně také to, že je k tvorbě modelu mutantů použit běžný SW k tomu určený s optimalizací mutantní struktury, kdežto v článku autoři o tom, jak vytvořili modely mutací, hovoří jen velmi vágně, což kontrastuje s poměrně rigorózním popisem toho, jak vytvářeli fyzické mutanty. Přes tyto rozdíly je aktuální metoda výsledky velmi podobná ($\rho = 0{,}776$) s originální metodou z článku.

Vytvořený prediktor je možné snadno upravit na výpočet největší kladné oblasti jako v článku Surface Patches za ponechání rozdílů v reprezentaci povrchu a domnělých rozdílů tvorby mutantů. Jako autor bych považoval za zajímavé provést experimenty i s takto upraveným prediktorem. A to z toho důvodu, že přestože jsou si má a originální metoda, dle měření, velmi podobné, je to ověřeno pouze na mutacích, které autoři článku provedli a zveřejnili. A ty jsou všechny vytvořeny na jediném proteinu. Nelze tedy vyvrátit, že v jiných případech můžou podávat metody výrazně odlišné výsledky. Domnívám se však, že tak tomu nebude. Zajímavé to považuji spíše z toho důvodu, že tak by bylo (při stejných výsledcích) možné definitivně označit Surface Patches za neperspektivní metodu.

Velkým problémem pro návrh prediktoru rozpustnosti ze struktury zůstává absence dobrého datasetu mutantů proteinů se známou strukturou i rozpustností. Ten by obsahoval řadu proteinů jak menších, tak větších, více nabitých i méně nabitých resp. neutrálních. Problematické proteiny z hlediska rozpustnosti i ty, u kterých problémy s rozpustností nejsou. Pro každý protein reprezentativní vzorek mutací měnící polaritu v místě mutace a zvláště mutací ostatních. A to jak pro příznivé mutace z hlediska změny rozpustnosti, tak i pro nepříznivé. Se všemi mutacemi vytvořenými v identickém expresním systému. Takový dataset by byl podstatně větší než dataset OptSolMut, na druhou stranu stále by při uvážlivém výběru mohl být menší než dataset Whitehead, který právě postrádá variabilitu z hlediska počtu proteinů.

Prediktor může sloužit také pro jiné než zamýšlené účely. Třeba k analýze, jakou má protein tendenci k elektrostatické interakci se záporně nabitými biogenními molekulami – to jsou třeba molekuly RNA. Pro analýzu míst jejich potencionálního styku mohou sloužit právě kladné izoplochy získané nástrojem.

Literatura

[1] Alberts, B. *Základy buněčné biologie: úvod do molekulární biologie buňky*. 2. vyd. Ústí nad Labem: Espero, 2006. ISBN 80-902906-2-0.

[2] Wikimedia Commons. *Alpha-amino-acid-condensed-2D-flat*. 2007. Dostupné z: https://commons.wikimedia.org/wiki/File:Alpha-amino-acid-condensed-2D-flat.png.

[3] Lamy, J.-B., Berthelot, H. a Favre, M. Rainbow Boxes: A Technique for Visualizing Overlapping Sets and an Application to the Comparison of Drugs Properties. In:. červenec 2016, s. 253–260.

[4] Richardson, J. S. *Protein backbone dihedral angles phi, psi, and omega*. 2011. Dostupné z: https://foundation.wikimedia.org/wiki/File:Protein_backbone_PhiPsiOmega_drawing.svg.

[5] Shafee, T. *Protein structure (full)*. 2019. Dostupné z: https://foundation.wikimedia.org/wiki/File:Protein_structure_%28full%29.png.

[6] Nilsson, B. L., Soellner, M. B. a Raines, R. T. Chemical synthesis of proteins. *Annual review of biophysics and biomolecular structure*. 2005, roč. 34, s. 91–118. 15869385[pmid]. Dostupné z: https://www.ncbi.nlm.nih.gov/pubmed/15869385. ISSN 1056-8700.

[7] Kiefhaber, T., Rudolph, R., Kohler, H.-H. a Buchner, J. Protein Aggregation in vitro and in vivo: A Quantitative Model of the Kinetic Competition between Folding and Aggregation. *Bio/Technology*. 1991, roč. 9, č. 9, s. 825–829. Dostupné z: https://doi.org/10.1038/nbt0991-825. ISSN 1546-1696.

[8] Paladin, L., Piovesan, D. a Tosatto, S. C. E. SODA: prediction of protein solubility from disorder and aggregation propensity. *Nucleic Acids Research*. Oxford University Press (OUP). květen 2017, roč. 45, W1, s. W236–W240. Dostupné z: https://doi.org/10.1093/nar/gkx412.

[9] Magnan, C. N., Randall, A. a Baldi, P. SOLpro: accurate sequence-based prediction of protein solubility. *Bioinformatics*. Červen 2009, roč. 25, č. 17, s. 2200–2207. Dostupné z: https://doi.org/10.1093/bioinformatics/btp386. ISSN 1367-4803.

[10] Tian, Y., Deutsch, C. a Krishnamoorthy, B. Scoring function to predict solubility mutagenesis. *Algorithms for Molecular Biology*. Springer Science and Business Media LLC. 2010, roč. 5, č. 1, s. 33. Dostupné z: https://doi.org/10.1186/1748-7188-5-33.

[11] Uversky, V. N. Intrinsically Disordered Proteins and Their "Mysterious" (Meta)Physics. *Frontiers in Physics*. 2019, roč. 7, s. 10. Dostupné z: https://www.frontiersin.org/article/10.3389/fphy.2019.00010. ISSN 2296-424X.

[12] Carballo Amador, M. A., McKenzie, E. A., Dickson, A. J. a Warwicker, J. Surface patches on recombinant erythropoietin predict protein solubility: engineering proteins to minimise aggregation. *BMC Biotechnology*. 2019, roč. 19, č. 1, s. 26. Dostupné z: https://doi.org/10.1186/s12896-019-0520-z. ISSN 1472-6750.

[13] Chan, P., Curtis, R. A. a Warwicker, J. Soluble expression of proteins correlates with a lack of positively-charged surface. *Scientific Reports*. Springer Science and Business Media LLC. listopad 2013, roč. 3, č. 1. Dostupné z: https://doi.org/10.1038/srep03333.

[14] Warwicker, J. *The size of the largest positive patch* [osobní e-mailová komunikace]. Červenec 2019.

[15] Klesmith, J. R., Bacik, J.-P., Wrenbeck, E. E., Michalczyk, R. a Whitehead, T. A. Trade-offs between enzyme fitness and solubility illuminated by deep mutational scanning. *Proceedings of the National Academy of Sciences*. 2017, roč. 114, č. 9, s. 2265–2270. Dostupné z: https://www.pnas.org/content/114/9/2265. ISSN 0027-8424.

[16] Wrenbeck, E., Bedewitz, M., Klesmith, J., Noshin, S., Barry, C. et al. An Automated Data-Driven Pipeline for Improving Heterologous Enzyme Expression. *ACS Synthetic Biology*. Únor 2019, roč. 8.

[17] Buss, O., Rudat, J. a Ochsenreither, K. FoldX as Protein Engineering Tool: Better Than Random Based Approaches? *Computational and Structural Biotechnology Journal*. 2018, roč. 16, s. 25 – 33. Dostupné z: http://www.sciencedirect.com/science/article/pii/S2001037017301186. ISSN 2001-0370.

[18] Merritt, E. A. *Raster3D Manual* [online]. Srpen 2003 [cit. 2020-05-22]. Dostupné z: http://www.physics.ohio-state.edu/doco/Raster3D/R3D_manual.pdf.

[19] Merritt, E. A. *Render - Raster3D package* [online]. [cit. 2020-05-22]. Dostupné z: http://skuld.bmsc.washington.edu/raster3d/html/render.html.

[20] Schrödinger, LLC. *File Formats [PyMOL Documentation]* [online]. [cit. 2020-05-20]. Dostupné z: https://pymol.org/dokuwiki/doku.php?id=format.

[21] Schrödinger, LLC. *The PyMOL Molecular Graphics System, Version 2.3*. listopad 2019. Dostupné z: https://pymol.org/2/.

[22] Shmueli, B. Matthews Correlation Coefficient Is The Best Classification Metric You've Never Heard Of. *Towards Data Science*. 2019, [cit. 2020-05-15]. Dostupné z: https://towardsdatascience.com/the-best-classification-metric-youve-never-heard-of-the-matthews-correlation-coefficient-3bf50a2f3e9a.

Příloha A

Obsah přiloženého média

```
/
├── analysis/ ..... RMd soubory s analýzami experimentů
├── data/ ..... vstupní data experimentů
│   ├── DSSP/ ..... analýzy PDB souborů pomocí DSSP
│   ├── pdbs/ ..... zkoumané PDB soubory
│   └── whitehead/ ..... Whitehead dataset
├── experiments/ ..... zdrojové soubory experimentů + výsledky
├── external/ ..... převzaté skripty a binárky
├── figures/ ..... zdrojové soubory ilustrací v programu PyMOL
├── foldx/ ..... složka určená pro instalaci FoldX
├── results/ ..... výsledky některých experimentů
├── text/ ..... zdrojové soubory tohoto dokumentu pro systém LaTeX
├── predictor.py ..... CLI prediktoru
├── readme.txt ..... návod k použití prediktoru
└── xvelec07-solubility-predictor.pdf ..... tento dokument
```

Příloha B

Nápověda CLI prediktoru

```
usage: predictor.py [-h] [-e {FoldX,Modeller}] [+v]
                    input {isosurface,predict} [chain] [location] [mutation]

Solubility change preditor:
  Predicts the change in the solubility of a given protein by measuring its el.
  potential isosurface.

Can operate in these modes:
  predict - create a specified mutant and make a prediction
    (includes isosurface; requires mutation)
  isosurface - only compute an isosurface of the given protein or its mutant
    (depends on whether a mutation is specified)

positional arguments:
  input                 input file in PDB format
  {isosurface,predict}  action to be performed
  chain                 target chain (letter; default=A)
                        (requires location)
  location              target residuum(s) (int)[]
  mutation              substituents(s) (1-letter or 3-letter code)[]
                        (requires location)

optional arguments:
  -h, --help            show this help message and exit
  -e {FoldX,Modeller}, --engine {FoldX,Modeller}
                        engine for the mutation (default=FoldX if available)
  +v, ++verbose         inform about the progress and show logs from the
                        underlaying tools

example of use: ./predictor.py 1eer.pdb predict 48,150 D ++verbose
```