VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ

ÚSTAV INFORMAČNÍCH SYSTÉMŮ

FACULTY OF INFORMATION TECHNOLOGY

DEPARTMENT OF INFORMATION SYSTEMS

WEBOVÝ SERVER PRO PREDIKCI 3D STRUKTURY PROTEINU

WEB SERVER FOR PROTEIN STRUCTURE PREDICTION

DIPLOMOVÁ PRÁCE

MASTER'S THESIS

AUTOR PRÁCE — **Bc. Lukáš Votroubek**

AUTHOR

VEDOUCÍ PRÁCE — **Ing. Ivana Burgetová, Ph.D.**

SUPERVISOR

BRNO 2013

Prohlášení

Prohlašuji, že jsem tento semestrální projekt vypracoval samostatně pod vedením Ing. Ivany Burgetové, Ph.D. Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

……

15.5.2013 Lukáš Votroubek

Poděkování

Moc rád bych poděkoval paní Burgetové za její rady a připomínky a za čas, který do této práce investovala.

Abstrakt

Tato práce se zabývá proteiny, především jejich strukturou a způsoby predikce terciární, neboli 3D, struktury. Predikce terciární struktury je důležitá pro zjištění funkce těchto životně důležitých látek. Využití metod bioinformatiky velice zefektivňuje a zrychluje predikci, protože klasické metody přímého zjišťování struktury z molekuly jsou drahé a pomalé. Na druhou stranu jsou zatím mnohem přesnější. Cílem práce je přiblížit metody používané pro predikci terciární struktury, popsat nástroje, které se používají a možnosti automatické komunikace s nimi. Dále je cílem popsat implementaci serveru, který bude sloužit proteinovým inženýrům pro efektivnější zjišťování informací o terciární struktuře z více nástrojů, aniž by museli zadávat požadavek na každý zvlášť. Také zde budou popsány výsledky testování.

Abstract

This work deals with proteins, especially with their structure and kinds of tertiary, or 3D, structure prediction. Tertiary structure prediction is very important for function prediction of this vitally important substance. Bioinformatics do this prediction much more effective and faster, because classical methods of structure prediction directly from molecule are very expensive and slow. On the other hand they are much more exact. Objective of this thesis is to describe tertiary structure prediction methods, describe used tools and possibility of automatic communication with them. Next objective is describe implementation of server, that will serve to protein engineers for more effective finding of information about tertiary structure from more servers without requesting each of them separately. Results of testing will be described in this work too.

Klíčová slova

3D struktura, terciární struktura, predikce, homologní modelování, threading, ab initio, de novo, Swiss model, CPH-models, ESyPred3D, Geno3D, AS2TS, 3D-Jigsaw, Phyre2, Fugue, HHPred, LOOPP, SAM-T08, PSI-PRED, I-TASSER, RaptorX, Robetta, iPBA, Superfammily, JSP, JAVA

Keywords

3D structure, tertiary structure, prediction, homology modeling, threading, ab initio, de novo, Swiss model, CPH-models, ESyPred3D, Geno3D, AS2TS, 3D-Jigsaw, Phyre2, Fugue, HHPred, LOOPP, SAM-T08, PSI-PRED, I-TASSER, RaptorX, Robetta, iPBA, Superfammily, JSP, JAVA

Citace

Bc. Lukáš Votroubek, Webový server pro predikci 3D struktury proteinu, diplomová práce, Brno, FIT VUT v Brně, 2013

Obsah

1 Úvod

Proteiny, známé také jako bílkoviny, jsou základním stavebním kamenem všech známých živých organizmů a vykonávají mnoho důležitých funkcí. Jsou nejen stavebními kameny buněk, ale také vykonávají životně důležité funkce jako transport látek, pohyb, řízení a regulace tvorby látek nebo obrana organizmu.

Veškeré funkce jsou určeny především strukturou proteinu. Proteiny jsou řetězce složeny z aminokyselin. Aminokyseliny svým pořadím strukturu proteinu, a tedy i jeho výslednou funkci, zásadně ovlivňují. Vliv na funkci a strukturu proteinu však mají i v okolí přítomné proteiny a různé další látky.

Většina z 21 aminokyselin, ze kterých se proteiny skládají, byla objevena v letech 1819 až 1904. Název protein navrhl pro složité organické sloučeniny bohaté na dusík, nalézané v buňkách živočichů a rostlin, pan Berzelius v roce 1838. Prvním důležitým objeveným proteinem byl hemoglobin (červený transportní protein červených krvinek), který krystaloval a pojmenoval Hoppe-Seyler v roce 1864. V roce 1897 Buchnerové položili základy enzymologii (vědě o enzymech). V roce 1933 pan Tiselius zavedl elektroforézu jako metodu pro dělení proteinů v roztoku. Jen o rok později provedli pánové Bernal a Crowfoot první podrobnou rentgenovou difrakci ve vzorcích proteinů. Chromatografie, široce používaná technika k dělení proteinů, byla vyvinuta pány Martin a Syngre v roce 1942. V roce 1951 pánové Pauling a Corey navrhli strukturu alfa-šroubovice a beta-listu, což jsou struktury, které byly následně nalezeny v mnoha proteinech. První protein, jehož přesné pořadí aminokyselin bylo stanoveno, byl insulin. Jeho analýzu dokončil roku 1955 pan Sanger. První podrobný popis struktury proteinu získali pánové Kendrew (vorvaní myoglobin) a Perutz (o něco detailnější struktura hemoglobinu). V roce 1963 pánové Monod, Jacob a Changeux poznali, že mnoho enzymů je regulováno změnami své konformace. [1]

V této práci se budu zabývat 3D, neboli terciární, strukturou proteinu. Touto strukturou je potřeba se zabývat zvláště z toho důvodu, že je nejdůležitější pro určení funkce proteinu a mechanizmu jeho fungování, což lze později mnoha způsoby využít (např. k urychlení jeho funkce nebo zvýšení odolnosti). Experimentální zjištění terciární struktury je finančně a časově velice náročné, u některých proteinů dokonce nelze experimentálním způsobem strukturu zjistit vůbec. Z těchto důvodů vzniklo několik metod, které se snaží predikovat strukturu proteinu pomocí počítačového zpracování sekvence aminokyselin.

Budu se zabývat samotnou funkcí proteinů, vznikem terciární struktury a jednotlivými metodami predikce této struktury. Dále se budu zabývat existujícími nástroji, které ze zadané sekvence dokážou terciární strukturu předpovědět. Poslední část práce se bude zabývat tvorbou webového serveru, který bude spolupracovat s existujícími nástroji a co nejefektivněji zpracovávat a zobrazovat jejich výsledky.

Text práce se skládá z několika částí. V kapitole 2 je popis jednotlivých úrovní organizace struktury proteinu. Bližší informace o jednotlivých úrovních jsou v podkapitolách 2.1 až 2.4. V kapitole 3 je

stručně popsáno, k čemu všemu jsou proteiny dobré. Kapitola 4 se zabývá metodami predikce terciární struktury. Podkapitola 4.1 se zabývá potenciální energií a silovými poli, pomocí kterých se vyhodnocuje, zda aktuální struktura proteinu je z energetického hlediska výhodná nebo nikoliv. Podkapitola 4.2 se zabývá predikcí terciární struktury metodou de novo, která využívá k predikci pouze znalosti sekvence aminokyselin. Metodou threading, která využívá ke své práci knihovny známých struktur, se zabývá podkapitola 4.3. Poslední používanou metodou, homologním modelováním, které využívá zarovnání a hledání nejpodobnějších proteinů k nějakému zkoumanému, se zabývá podkapitola 4.4. V kapitole 5 lze zjistit informace o nástrojích používaných pro predikci terciární struktury. Krátce se zmíním i o dalších nástrojích, používaných například k vizualizaci nebo verifikaci struktury. Podkapitola 5.1 se zabývá soutěžemi nástrojů pro predikci struktury proteinu, CASP a CAFASP. V dalších čtyřech podkapitolách lze najít bližší informace o jednotlivých nástrojích, každá metoda predikce má svou vlastní podkapitolu. V poslední podkapitole, 5.6, jsou popsány možnosti automatické komunikace s nástroji. V kapitole 6 je popsán návrh serveru pro predikci terciární struktury, především použitý programovací jazyk, uživatelské vstupy a výstupy serveru, metoda zvolení nástrojů k využití při implementaci apod. Způsobem testování nástrojů a jejich výběrem se zabývá kapitola 7 . V poslední podkapitole této kapitoly jsou výsledky testování. Kapitola 8 popisuje implementací serveru v jazyce JAVA. Testování výsledného serveru je popsáno a shrnuto v kapitole 9 . Jednotlivé její podkapitoly se zabývají jak výběrem testovacích dat, tak způsobem testování a zjištěnými výsledky. Kapitola 10 tvoří závěr, kde je stručné shrnutí dosažených výsledků a zjištěných informací. Použité zdroje jsou shrnuty v kapitole Zdroje.

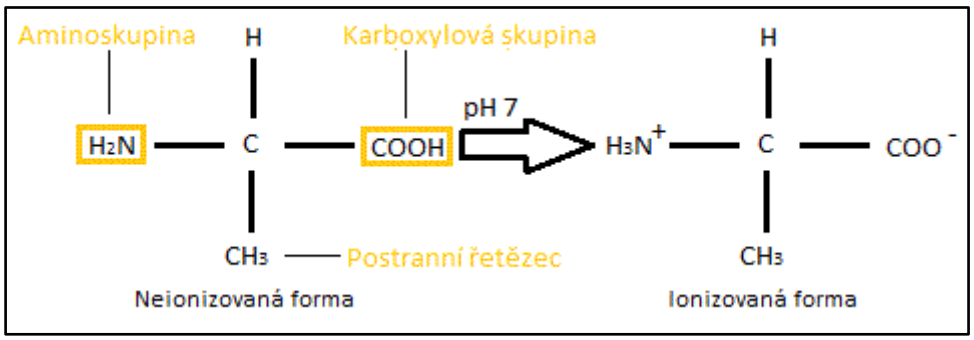
2 Úrovně organizace struktury

Text celé kapitoly bude čerpán z [2] a z menší části z [1]. Než se budeme zabývat tím, jak vzniká a predikuje se terciární struktura proteinu, ukážeme si, jaké existují další struktury a jaký je mezi nimi vztah.

Existují tyto následující úrovně struktury proteinů: primární struktura (primary structure), sekundární struktura (2D nebo secondary), terciární struktura (3D nebo tertiary) a kvartérní stuktura (quarternary).

2.1 Primární struktura

Primární struktura (primary structure) je sekvence aminokyselin tvořící protein. Aminokyseliny jsou molekuly, které vlastní karboxylovou a aminovou skupinu. Obě tyto skupiny jsou připojeny k jednomu uhlíkovému atomu zvanému uhlík alfa. Každá aminokyselina má jiné vlastnosti především kvůli postrannímu řetězci, který je k uhlíku alfa také připojen.



Obr. 1 Aminokyselina alanin [1]

Vazba mezi sousedními aminokyselinami v řetězci se nazývá polypeptidová vazba, řetězec aminokyselin je také známý jako polypeptid nebo polypeptidový řetězec. Existují ale specifické aminokyseliny, které jsou na koncích popypepdidu: NH2 se označuje jako N-konec, COOH se označuje jako C-konec. To dává řetězci směrovou vlastnost, strukturní polaritu. Pokud je protein syntetizován translací mRNA, pak je amino konec syntetizován jako první. Proto jsou sekvence zapisovány také s N-koncem na levé straně.

Každá z jednadvaceti druhů aminokyselin, které existují, se liší ve velikosti a složitosti postranního řetězce, který je připojen k uhlíku. Pět z jedenadvaceti druhů aminokyselin mají postranní řetězce schopné nést náboj.

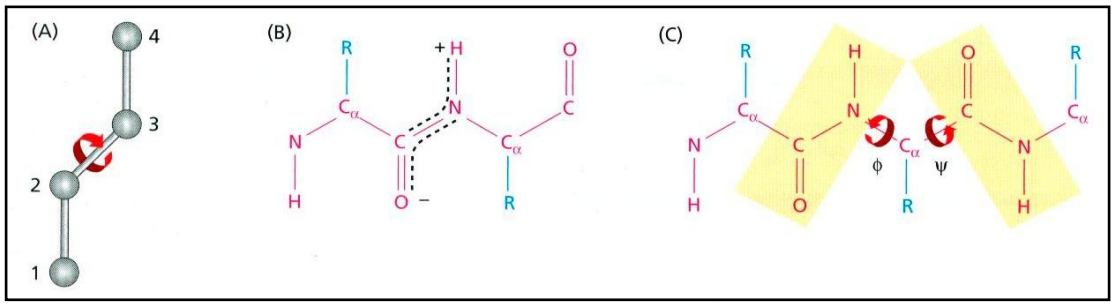
Ionizovaná forma na Obr. 1 je spíše pro zajímavost, vyskytuje se v buňkách, kde je pH blízké sedmi. Když se však zařadí do polypeptidového řetězce, náboje u aminoskupiny i karboxylové skupiny zmizí.

2.2 Sekundární (2D, Secondary) struktura

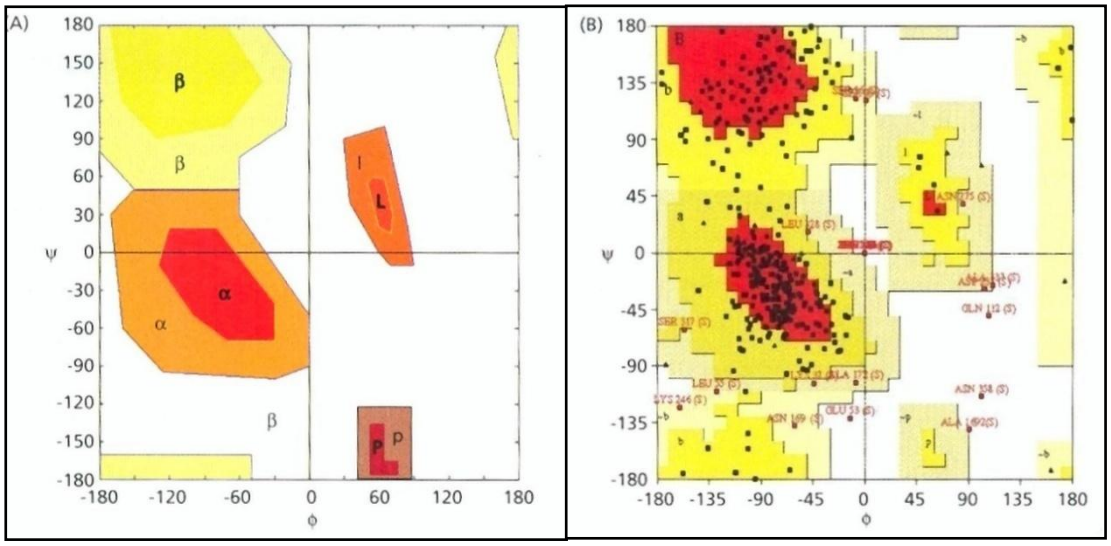
Sekundární struktura (Secondary structure), někdy také 2D vzniká tak, že aminokyseliny mají vliv na okolí a části řetězce se tak skládají do několika různých základních tvarů (např. alfa-šroubovice nebo beta-listy).

Sekundární struktura je ovlivněna torzními, jinak také dihedrálními, úhly, což jsou úhly mezi vazbami čtyř atomů A-B a C-D spojených v pořadí A-B-C-D (viz Obr. 2A,B) nebo mezi dvěma rovinami A-B-C a B-C-D (viz Obr. 2C). Nejčastěji jsou to úhly mezi peptidovými vazbami v kostře proteinu a jsou zapisovány jako ϕ a ψ. Tyto úhly jsou hlavním zdrojem flexibility polypeptidového řetězce a teoreticky mohou nabývat hodnot v rozsahu -180° až +180°, v praxi však nabývají pouze určitých mezí. Dále existuje úhel ω, což je úhel mezi atomy C a N.

Každý typ sekundární struktury se vyskytuje nejčastěji, pokud atomy svírají určité úhly. Například pravotočivá alfa-šroubovice preferuje úhly (ϕ , ψ), které jsou přibližně (-60°,-60°). Detailněji tyto úhly můžeme vidět na Obr. 3.



Obr. 2 Torzní úhly [2]



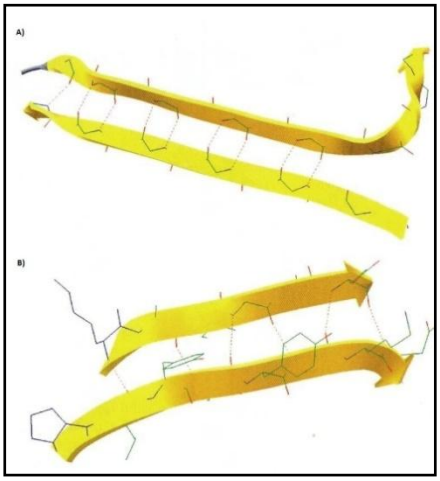
Obr. 3 Ramachadran plot – úhly ϕ a ψ jsou na vodorovné a svislé ose. Vykreslené oblasti reprezentují úhly, kde se nejpravděpodobněji struktura zformuje do některého tvaru: A) ideální graf (α je α-šroubovice, β reprezentuje β–listy, L je levotočivá šroubovice a P je struktura ε) B) skutečně naměřené hodnoty. V částech A) i B) platí, že čím tmavší oblast je, tím je pravděpodobnější výskyt) [2]

2.2.1 základní typy sekundární struktury

V zásadě existují tři nejčastěji se vyskytující typy sekundární struktury. Existují i další typy struktur, např. π-helix, 3_{10} helix, Beta-most apod. Tyto struktury však co do počtu nejsou tak významné. Tři základní typy sekundární struktury jsou:

a) α-šroubovice (α-helice)
Pravotočivá nebo levotočivá šroubovice, kde jeden závit šroubovice je složen z 3.6 reziduí

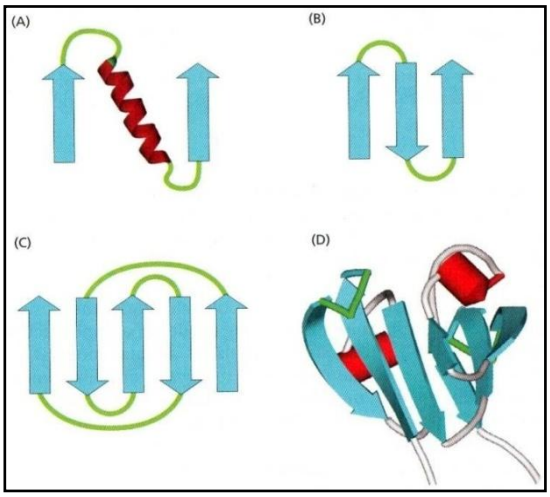
b) β-list (β-sheet)
Struktura odlišná od a), je tvořena z jednotlivých β-vláken (β-strand). Jednotlivá vlákna jsou mezi sebou stabilizována slabými vodíkovými vazbami.
Každé vlákno může mít nějaký směr – pak jsou v listu všechna vlákna buď stejným směrem (paralelní β-list), nebo se nějakým způsobem střídají (antiparalelní β-list). Speciálním typem β-listu je pak β-turn, což je struktura nacházející se obvykle mezi β-vláknem a α-šroubovicí



Obr. 4 Struktura Beta-listu (nahoře antiparalelní, dole paralelní uspořádání) [2]

c) Coil
Náhodně uspořádaná struktura.

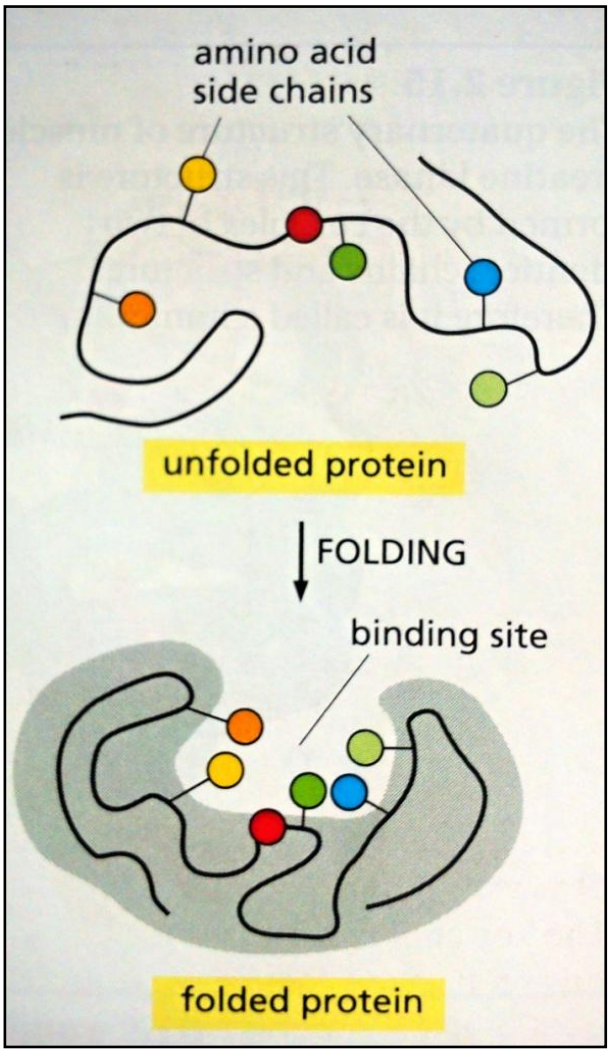
Dále existují supersekundární struktury, což je pojmenování struktur složených z nějaké kombinace α-šroubovic a β-listů (například βαβ repeat, β-mander, některé příklady viz Obr. 5).



Obr. 5 Supersekundární struktury: A) βαβ repeat B) β-mander C) Greek key D) jiná reprezentace struktury Greek key [2]

2.3 Terciární struktura

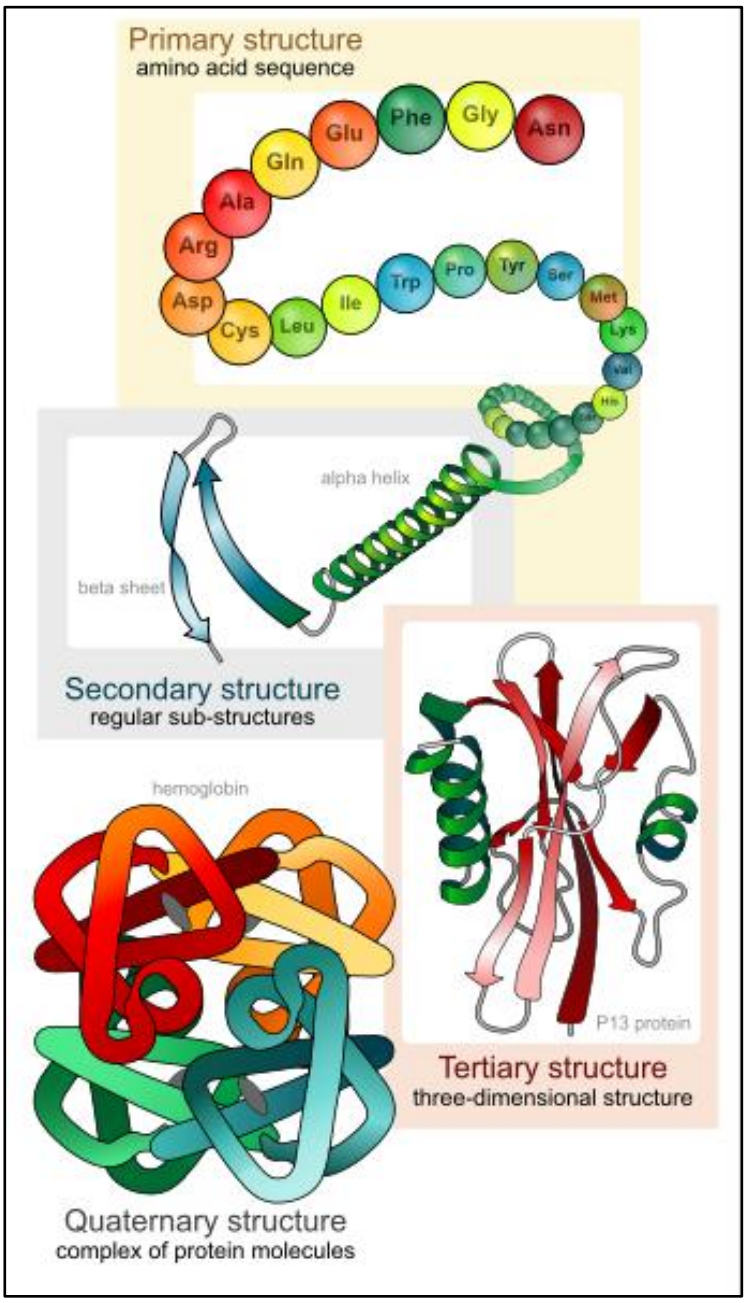
Terciární struktura (Tertiary structure), 3D struktura, nebo také proteinový fold, vzniká dalším působením aminokyselin na sebe poté, co utvořily sekundární strukturu. Až po zformování této struktury se protein stane aktivní a funkční biologickou jednotkou. Často má protein několik možností jak utvořit tuto strukturu, každá taková možnost se nazývá konformace. Každá konformace vzniká tak, aby protein měl co nejmenší energii – tedy aby všechny vazby v proteinu byly propojeny co nejstabilněji.



Obr. 6 Skládání proteinů (protein folding) – Postranní řetězce v sekvenci aminokyselin (amino acid side chains) se po složení proteinu do foldu přemění ve vazební místo (binding site) [2]

2.4 Kvartérní struktura

Kvartérní struktura (Quarternary structure) se vyskytuje, pokud je protein tvořen více než jedním řetězcem aminokyselin (je to více proteinů spojených v jeden). Každý jednotlivý řetězec je pak podjednotka, nebo monomer, nějakého oligomeru. Říkáme, že se protein skládá z více podjednotek. Protein se dvěma podjednotkami se nazývá dimer, se třemi trimer, se čtyřmi tetrametr, apod. Všechny podjednotky mohou být ze stejné sekvence proteinu.



Obr. 7 Úrovně organizace struktury proteinu [9]

3 Funkce proteinů

Kapitola čerpána z [1]. Především terciární a kvartérní struktura ovlivňuje výslednou funkci proteinu. Obecně dělíme funkce proteinů do těchto kategorií:

a) Enzymy
   Slouží ke katalýze (urychlení nebo zpomalení) různých reakcí, např. syntéza jiných proteinů, pepsin odbourávající proteiny z potravy v žaludku, ribulosabisfosfátkarboxyláza pomáhající přeměně oxidu uhličitého na cukry v rostlinách nebo DNA-polymeráza syntetizující DNA.

b) Strukturní proteiny
   Mechanická opora buňkám a tkáním, např. tubulin a aktin tvořící různá vlákna uvnitř buněk, alfa-keratin tvořící vlákna zpevňující epiteliální buňky a tvořící základní složku nehtů, vlasů a rohů.

c) Transportní proteiny
   Přenos malých molekul a iontů, např. přenos kyslíku a železa v krevním oběhu, různé proteiny v membránách buněk, které přes ně přenáší ionty a molekuly.

d) Pohybové proteiny
   Jsou původcem pohybu buněk a tkání, např. myozin v kosterním svalu, kinetin umožňující pohyb organelami uvnitř buňky, dynein umožňující pohyb eukaryotních bičíků a řasinek.

e) Zásobní proteiny
   Skladují malé molekuly a ionty, např. železo, které se ukládá v játrech navázáním na feritin, kasein v mléce jako zdroj aminokyselin pro novorozené živočichy.

f) Signální proteiny
   Přenášejí informační signály z buňky do buňky, např. insulin regulující hladinu glukosy v krvi, NGF stimulující některé typy nervových buněk k růstu axonů, EGF stimulující dělení epiteliálních buněk.

g) Receptorové proteiny
   V buňkách detekují chemické a fyzikální signály a předávají je ke zpracování buňce, např. rodopsin v oční sítnici zachycující světlo nebo acetylcholinový receptor v membráně svalové buňky přijímající signál z neuronu.

h) Regulační proteiny v genové expresi
   Váže se na DNA a spouští nebo vypíná transkripci některých genů.

i) Proteiny se zvláštním posláním
   Tato skupina je také velmi rozmanitá, např. protimrazové proteiny některých ryb, různé fosforeskující nebo adhezní proteiny, kterými se přichytávají mořští živočichové k mořskému dnu.

4 Predikce terciární struktury

Kapitola čerpána především z [2], podkapitola 4.2 čerpána z [3]. Predikce terciární struktury proteinu, je potřebná pro zjištění funkce proteinu, mechanizmu jak protein pracuje a ke zjištění jeho strukturně-funkčních vztahů. S pomocí těchto znalostí můžeme navíc přizpůsobit jeho funkčnost různým medicínským nebo průmyslovým účelům (zrychlení nebo zpomalení procesů, zvětšení vazebního místa umožňující proteinu pracovat s většími částicemi apod.)

Existuje několik experimentálních technik, které nám dovolují zjistit strukturu přímo z biologické molekuly. Jsou to rentgenová krystalografie (X-ray crystallography) a nukleární magnetická rezonance (NMR). Rentgenová krystalografie je sice velice přesná a zvládá rozsáhlé struktury, může však poskytnout pouze statický pohled na molekulu. Nukleární magnetická rezonance je méně přesná, na druhou stranu nám ale dává souřadnice v určitém čase, což nám umožňuje sledovat interní pohyb proteinu. Obě metody jsou však zatím finančně a časově velmi náročné a nelze tak zjistit strukturu všech molekul. U mnoha proteinů dokonce tyto techniky selhávají úplně (např. pokud nemůže krystalizovat, což je předpoklad pro rentgenovou krystalografii). Pro predikci struktury proto vzniklo několik metod vycházejících ze sekvence aminokyselin, které lze provádět s pomocí počítače.

Pro predikci terciární struktury existují tři základní metody, a to homologní modelování (jinak také komparativní modelování nebo modelování založené na znalostech, anglicky homology, comparative nebo knowledge-based modeling), threading (nebo také rozpoznávání foldů, anglicky fold recognition) a de novo. První dvě metody jsou běžně používané, dobře rozvinuté a vycházejí ze znalostí o přirozeném skládání proteinů do stabilní struktury. To znamená, že se protein poskládá do struktury, která má co nejmenší volnou energii (taková struktura se pak nazývá přirozená struktura nebo také přirozený stav). Entropickou část této volné energie je obtížné spočítat. Bylo však zjištěno, že ji lze bez důsledků ignorovat. Přirozená struktura je pak predikována na základě co nejmenší entalpie, která je definována jako minimální potenciální energie. Tato energie je například v kovalentních vazbách a nekovalentních interakcích. Zarovnání proteinu je tedy možné vypočítat pomocí funkce energetického potenciálu nebo silového pole odvozeného z interakcí mezi všemi atomy v proteinu.

Metoda de novo, často se jí říká také ab initio, predikuje strukturu pomocí principů termodynamiky a fyzikální chemie. Tyto principy jsou opět využity k identifikaci struktur s minimální energií. De novo však zatím s dostatečnou účinností dokáže predikovat pouze malé proteiny s jednou doménou.

Krom tří zmíněných základních metod existuje množství nástrojů, které různými způsoby kombinují více principů. Každá ze tří základních metod (ab initio, threading a homologní modelování) bude detailněji popsána v následujících podkapitolách. První podkapitola se však bude zabývat potenciální energií a silovými poli, jež nám pomáhají hodnotit energetickou výhodnost struktury proteinu.

4.1 Potenciální energie a silová pole

Při modelování struktury proteinu, ať už metodou ab initio nebo porovnáním s jiným foldem, je cílem zjistit strukturu, která má co možná nejmenší energii, a která zároveň splňuje stereochemická omezení na strukturu proteinu. Takovými omezeními jsou například torzní úhly ψ a ϕ pro kostru proteinu.

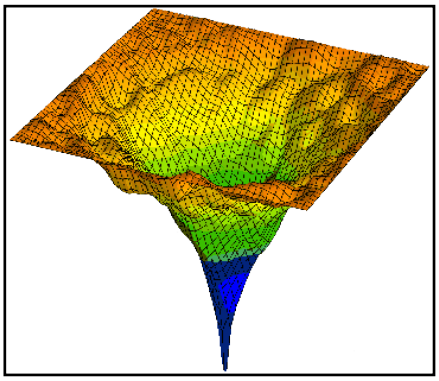
Abychom spočítali ohodnocení pro všechny konformace a zjistili, které konformace jsou energeticky preferované, musíme počítat s energetickým potenciálem proteinu jako s kolekcí rovnic. Tyto rovnice jsou známé jako funkce potenciální energie a reprezentují všechny komponenty, které přispívají k celkové potencionální energii. Kombinace všech takových energetických funkcí pro nějakou konkrétní konformaci se nazývá silové pole.

Existují dva významné typy silových polí:

a) Počítají potenciální energii konformace, která může obsahovat další molekuly (např. rozpouštědla)
b) Počítají energii, která obsahuje statisticky zprůměrované účinky na prostředí

Jakmile je vypočítáno silové pole, může být definována energie konformace a následně identifikována energeticky nevýhodná oblast. To může být vhodné k například k vyhodnocení navrhovaných struktur a komplexů. Pro predikci přirozených konformací existuje několik technik, ve kterých jsou tyto konformace predikovány na základě identifikace energetického minima. Těmito metodami se zabývá molekulární mechanika.

Potenciální energie molekuly závisí na polohách a částečně i na typech atomů a může být definována pro kteroukoli konformaci. Konformace malých molekul může být definována pomocí několika málo proměnných, často postačují tři souřadnice každého jednotlivého atomu. V případě větších molekul, jako i proteinů, jsou potřeba stovky nebo tisíce proměnných. Povrch, který reprezentuje odchylky potencionální energie, lze vykreslit jako „povrch potencionální energie“ (potential energy surface). Příkladem může být Ramachadran plot (viz Obr. 3), kde oblasti korespondují s nižší a vyšší potencionální energií.



Obr. 8 Graf volné energie - na horizontální ose jsou souřadnice konformace, na vertikální energie systému [10]

Z pohledu teorie termodynamiky můžeme říci, že se molekulární systémy většinou uspořádají do konformace, která má nejnižší volnou energii. Tato volná energie v sobě zahrnuje jak entropii, tak potenciální energii. Příklad zobrazení volné energie je na Obr. 8. Konformace, které bude struktura nejčastěji zaujímat, jsou ty, které leží nejníže v grafu.

Entropickou část volné energie je velice obtížné spočítat, protože vyžaduje zprůměrování velkého množství stavů systému. Ke zjednodušení problému nám nepomůže ani to, že mnoho stavů může být vyjádřeno konstantou, nebo dokonce může být zcela ignorováno.

4.2 De novo modelování

Cílem metody de novo je predikce struktury proteinu, a to pouze na základě sekvence aminokyselin. Obecně se očekává, že se protein zabalí do nějaké výhodné konformace a že tato konformace je stavem energetického minima nebo mu se blíží. Problém hledání stavu energetického minima lze rozložit na dva menší problémy. Zaprvé zjištění přesného potenciálu, zadruhé vývoj účinné metody pro hledání energetického obrazu, který zjišťujeme z potenciálu.

Mnoho metod, které jsou dnes vedeny jako de novo, byly dříve vedeny jako ab initio. Obvykle jsou jako de novo nazývány metody, které se nespoléhají na homologii mezi dotazovanou sekvencí a sekvencí v PDB. Metody de novo jsou teoreticky určeny pro mnohem rozsáhlejší obrazy než při homologním modelování a metodě threading, které limituje vzor, ze kterého se vychází při predikci. Predikce de novo je však výpočetně náročná a proto zatím není vhodná pro běžné aplikace. Ani zatím není dostatečně přesná pro více než 150 reziduí. S tímto problémem se dá vypořádat například tak, že rozdělíme sekvenci na menší domény. Pro mnoho proteinů je takovéto rozdělení snadné, zatímco u některých je problém takovéto domény detekovat. Určení doménové rodiny (domain family) a hranic domény pro více-doménové proteiny je tak vhodným začátkem predikce. Většina metod pro rozpoznávání domén se spoléhá na hierarchické prohledávání dotazované sekvence. Při tomto prohledávání si vypomáhá nahlížením do knihovny domén a do databáze PDB.

Teoreticky existují dva principy, jak by mohla být struktura proteinu predikována. První je založený na statistice, druhý na fyzice. Většina metod dnes využívá spíše statistiku.

Abychom zrychlili predikci a předešli problémům s dělením na domény, většina metod využívá různá zjednodušení modelu (zanedbává nějaké skutečnosti). Metody pro redukci složitější struktury proteinu na jednotlivé jednodušší modely mohou být rozděleny do dvou tříd: mřížkové (lattice), a bez mřížky (off-lattice).

Mřížkové modely mají dlouhou historii v modelování chování polymerů, a to především kvůli jejich analytické a výpočetní jednoduchosti. Vyhodnocení energií v mřížce může být dosáhnuto poměrně účinně a metody umožňující úplné prohledávání prostoru možných konformací jsou také poměrně dostupné a efektivní. Tyto metody však mají omezenou schopnost reprezentovat drobné aspekty a mohou reprodukovat kostru s přesností, která není o moc větší než přibližně poloviční rozteč mřížky. Nejběžnější systematická chyba pozorovaná pro velké množství mřížkových modelů je v jejich neschopnosti reprodukovat šroubovice, což je z důvodu důležitosti sekundární struktury problém. Výpočetní výhody však převažují nad těmito problémy.

Dále existují diskrétní stavové bezmřížkové (Discrete state off-lattice) modely. Většina takto redukovaných modelů zjednodušuje postranní řetězce na jediný rotamer nebo několik centroidů s kosterními atomy. Také jsou často redukovány úhly ψ a ϕ, obvykle na čtyři až třicet dva stavů, reprezentujících všechny typy sekundární struktury.

Když máme model, který co nejefektivněji redukuje složitost prohledávání, je třeba mít nějakou skórovací nebo energetickou funkci. Energetická funkce musí adekvátně reprezentovat síly zodpovědné za strukturu proteinu. Protože máme zjednodušený model, který nereprezentuje

všechny atomy, je přesnost výpočtu nižší a funkce proto musí být robustní a s chybami se co nejúčinněji vypořádat. Také musí být výpočetně efektivní, protože v počáteční fázi vyhledávání konformace je prováděno velké množství vyhodnocení.

Skóre může být vypočteno například na základě solvatace (solvation-based scores). Běžný způsob je klasifikovat síť v proteinu na základě působení rozpouštědel. Pak je buď vystavený působení celý povrch, nebo nějaký počet blízkých reziduí. Pak jsou určeny frekvence výskytu aminokyselin každého typu a z tohoto vycházejí další výpočty, ze kterých vychází výsledné skóre. Další běžný způsob klasifikace vychází z globální míry hydrofobního uspořádání. Jednoduchý způsob takovéto klasifikace je vzdálenost rezidua od těžiště konformace, které může být použito k výpočtu množství, které je analogické hydrofobickému poloměru otáčení. Takové metody však mají problém s malými vzdálenými podoblastmi.

4.3 Threading

Další možný název pro tento způsob modelování je rozpoznávání foldů (fold recognition). Tato metoda na rozdíl od homologního modelování (popsaného dále) nepotřebuje homologní strukturu proteinu. Místo toho strukturu porovnává se všemi známými foldy a zjišťuje, který z nich se zdá být energeticky a stereochemicky nejvýhodnější. Jakmile je vhodný fold nalezen, může být pomocí technik homologního modelování zjištěn realističtější strukturální model. Od metody ab initio se tento způsob liší především tím, že není tolik výpočetně náročný.

Z analýz a velkého množství experimentů bylo zjištěno, že existuje pouze omezené množství způsobů, jak se protein může zarovnat, a to možná i méně než 2000. U sekundární struktury platí, že stejná struktura může být formována různými sekvencemi, a toto tvrzení platí i u terciární struktury.

Protože techniky rozpoznávání foldů nezávisí primárně na porovnávání sekvencí, vztahy mezi strukturami různých proteinů mohou být zjištěny i když podobnost sekvencí je velmi malá nebo dokonce žádná. Zachování struktury může být způsobeno jednak společnými předky a jednak faktem, že fyzická omezení limitují počet foldů, které protein může zaujmout. Proto mohou být stejné foldy nalezeny ve velkém množství různých proteinů.

Metoda threading se tedy snaží najít foldy, které jsou kompatibilní se sekvencí zkoumaného proteinu. Zkoumaná sekvence je zarovnána ke každé proteinové struktuře v knihovně foldů a je vypočítáno skóre každého takového zarovnání. Pokud je v knihovně nalezena struktura, která má toto skóre dostatečně vysoké, pak je velice pravděpodobné, že zarovnání zkoumané sekvence bude z větší části stejné. Výsledek může být navíc porovnán s další často používanou technikou, běžně známou jako inverzní skládání proteinů (inverse protein folding), která spočívá v prohledání databáze proteinových sekvencí a abychom predikovali sekvence, které mají podobný fold, využíváme znalosti struktury.

V databázi PDB (Protein Data Bank) existuje mnoho proteinů, kterým odpovídá mnoho homologů. Abychom co nejpřesněji predikovali výslednou strukturu, je nejlepší využít tyto homology všechny. K efektivnější práci bylo proto vytvořeno několik knihoven neopakujících se foldů. Vyhodnocení kompatibility zkoumané sekvence a proteinového foldu vyžaduje skórovací schéma. Cílová sekvence může být zarovnána s foldem také mnoha způsoby a metoda threading musí určit pro každou

knihovnu foldů tu nejlepší. Metody zarovnání, které se zde využívají, jsou až na některé rozdíly stejné jako v zarovnání sekvencí.

4.3.1 Knihovny a databáze foldů používané v metodě threading

Dnes je v databázi PDB okolo 80 000 proteinů [4]. Pokud bychom je chtěli všechny prohledat, abychom našli jednu cílovou položku, potřebovali bychom mnoho času. Proto byly vytvořeny knihovny, které obsahují pouze ty unikátní, a je tak drasticky snížen potřebný čas pro jejich zpracování. Abychom reprezentovali všechny experimentálně určené struktury, potřebujeme pouze několik málo tisíc struktur [5].

Asi nejvýznamnějšími knihovnami unikátních foldů jsou CATH a SCOP. Tyto dvě knihovny se liší v tom, jak klasifikují proteinové foldy a také v tom jak jsou jednotlivé foldy identifikovány. Knihovna CATH foldy klasifikuje do čtyř úrovní: třída (Class, zn. C), arcihtektura (Architecture, zn. A), topologie (Topology, zn. T) a homologní nadtřída (Homologous superfamily, zn. H). Pro nás je v tomto okamžiku významná třída T, ve které jsou popsány a klasifikovány foldy. Knihovna SCOP,z anglického Structural Clasification Of Protein, klasifikuje do tří úrovní: foldy, superrodiny a rodiny. Úroveň fold je ekvivalentní úrovni T v knihovně CATH.

4.3.2 Používaná skórovací schémata

V metodě threading se nejčastěji používají dva typy skórovacích schémat. Prvním typem je schéma využívající modifikace klasických skórovacích matic pro aminokyseliny. Druhý typ schémat explicitně obsahuje detaily o struktuře v okolí každého rezidua, a to včetně mezi-atomických vzdáleností nebo počtů reziduí uvnitř nějaké vzdálenosti.

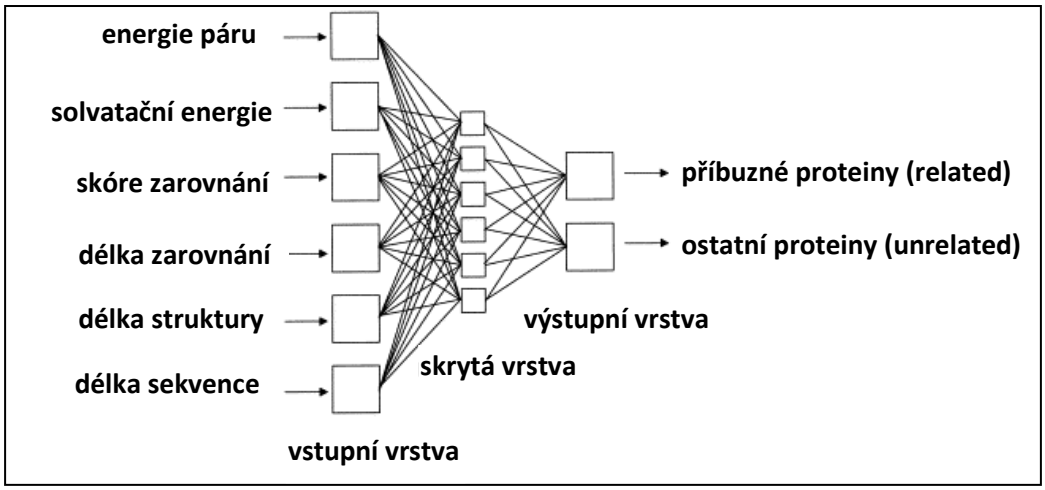
Substituční matice používané pro threading se od těch používaných v zarovnání sekvencí liší v jedné základní věci a to v tom, že si dvě zarovnaná rezidua nejsou ekvivalentní. U jednoho rezidua totiž známe jeho strukturu, zatímco u druhého, toho zkoumaného proteinu, strukturu neznáme. Další věc, kterou musíme brát v úvahu je, že substituční matice jsou nesymetrické (pokud alanin ve vzorovém reziduu nahradíme lysinem v tom zkoumaném, není to stejné jako bychom lysin ve vzorovém reziduu nahradili alaninem v tom zkoumaném).

Například metoda FUGUE, popsaná panem Tomem Blundellem a jeho kolegy, používá šedesát čtyři různých skórovacích matic, každá z nich definuje specifické podmínky pro každou část foldu. Podmínky jsou definovány ve třech kategoriích. Jsou definovány čtyři třídy pro konformaci hlavního řetězce, dvě třídy pro dostupnost rozpouštědel, „solvent accessibility“, a osm tříd pro vodíkové vazby, přičemž lze rozlišit vazby navazující postranní řetězec, skupinu NH na hlavním řetězci, a skupinu CO na hlavním řetězci. Tyto tři kategorie podmínek jsou na sobě nezávislé a dávají dohromady oněch šedesát čtyři možností. Používané matice byly odvozeny podobným způsobem jako BLOSUM. Další existující metodou je LOOPP od pana Rona Elbera a jeho kolegů, kde matice podobného charakteru jako v metodě FUGUE byly odvozeny jiným způsobem.

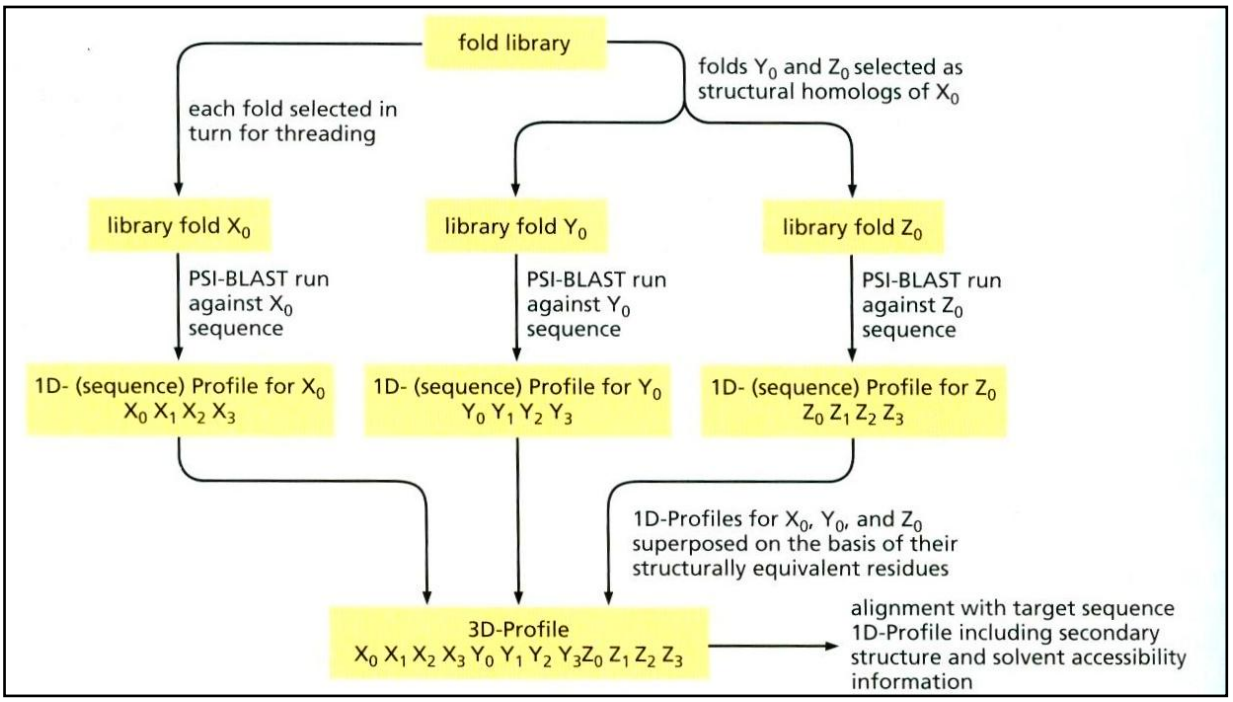
Druhá třída skórovacích schémat v sobě zahrnuje funkce, které v nějaké míře závisí na vzdálenostech mezi specifickými atomy foldu. Páteřní N a O atomy a postranní řetězec C_β každého rezidua jsou

využívány k rozlišení interakcí na krátké (méně než 11 reziduí), střední a dlouhé (více než 22 reziduí). Počet pozorovaných specifických párů atomů (např. $C_\beta \rightarrow C_\beta$) na dané vzdálenosti je konvertován na energii.

Metoda GenThreader používá pro každé reziduum solvatační potenciál, který závisí na počtu atomů C_β umístěných ve vzdálenosti deset Ångstromů od daného atomu C_β. Dále existuje např. metoda 3D-PSSM, která využívá kombinaci obou typů skórovacích schémat. Místo využití substitučních matic využívá PSSM (Pozičně Specifické Skórovací Matice) a ke generování profilů pro každý fold v knihovně využívá PSI-BLAST. Knihovna obsahuje identické foldy pro nehomologní sekvence a pro každý z takových foldů je generován odlišný profil. Výsledný profil je generován zkombinováním pozic na odpovídajících místech různých foldů a k určení ekvivalence využívá principu superpozice.



Obr. 9 Architektura neuronové sítě používané v algoritmu GenThreader (Místo abychom využili k predikci foldů pouze energetické skóre, zkombinujeme jej s dalšími hodnotami. Poté využijeme klasickou dopřednou neuronovou síť.) **[2]**



Obr. 10 Flow diagram algoritmu 3D-PSSM (Každá sekvence x_0 v knihovně foldů je zarovnána se svými homology x_1,x_2, … s využitím PSI-BLAST. Poté je k identifikaci dalších homologů y_0 a z_0 využita databáze SCOP, tyto homology jsou zarovnány rovněž pomocí PSI-BLAST. Pro různé rodiny jsou vytvořeny PSSM, a ty jsou asociovány se stejným proteinem jako x_0. Ty jsou pak zkombinovány s daty o sekundární struktuře a přístupností rozpouštědel a je vyprodukován 3D profil foldu. Obdobný profil je zkonstruován pro dotazovanou sekvenci a je rovněž založen na homologní sekvenci, i když v tomto případě predikované, sekundární struktury. Nakonec jsou k zarovnání dotazovaného profilu s 3D profilem každého foldu v knihovně foldů využity metody dynamického programování.)**[2]**

4.3.3 Zarovnání sekvencí a foldů

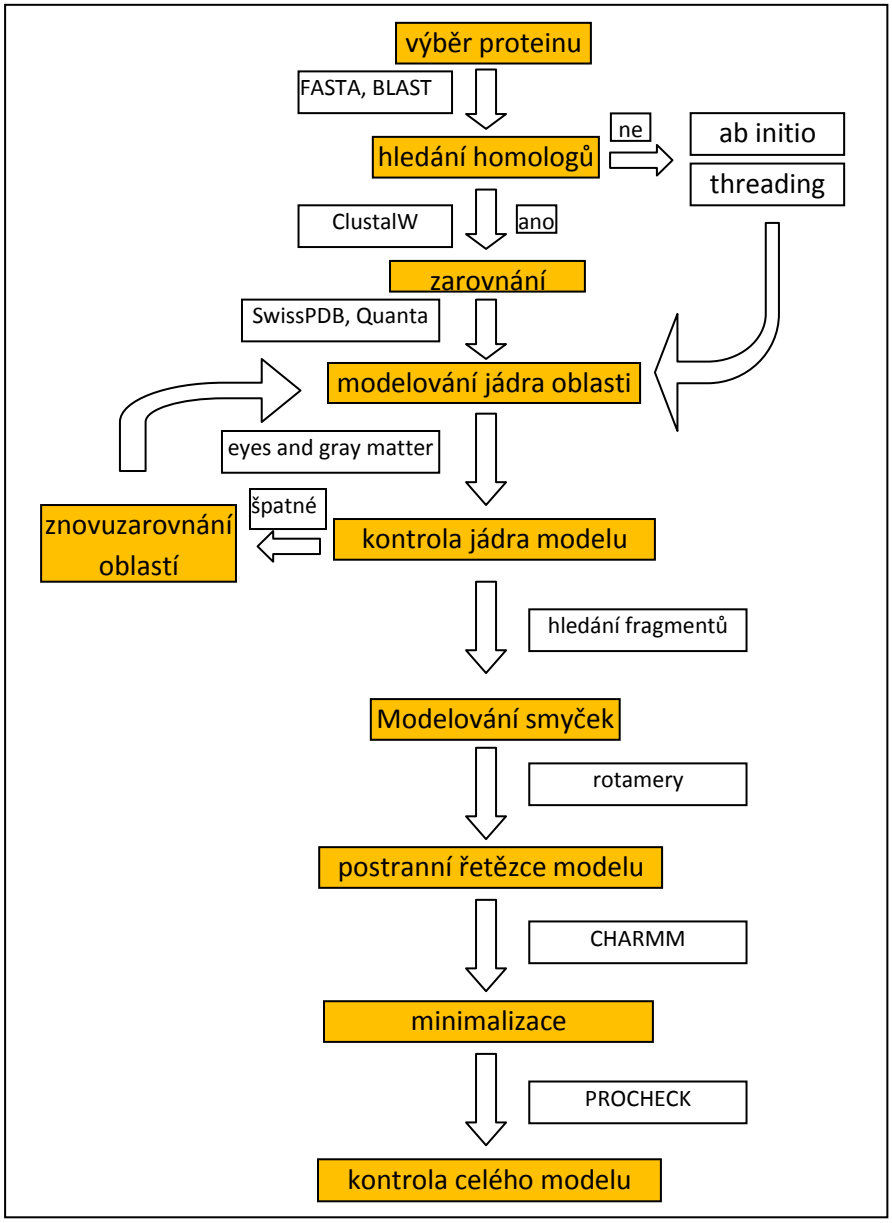
Pro optimální zarovnání cílových sekvencí a foldů s výhodou využíváme metod dynamického programování. Protože však ve skórovacích schématech využíváme odlišné skutečnosti, budeme muset provést oproti základním metodám zarovnání sekvencí některé modifikace, jako jsou například dvojité dynamické programování (double dynamic programming, které bylo navrženo panem Willie Taylorem a kolegy) nebo schémata dynamického programování založená na iteraci.

Nejvyšší skóre zarovnání pak může být použito k okamžitému návrhu foldu cílové sekvence. Protože však bylo skóre odhadováno a není přesné, nemusí být tento způsob nejefektivnější a využívá se komplexnějších metod, například neuronových sítí (jako je vidět na Obr. 9).

4.4 Homologní modelování

Jinak také nazýváno komparativní modelování nebo modelování založené na znalostech (knowledge-based modeling). Tato metoda využívá již známých struktur, a na základně homologie hledané neznámé struktury s některou již známou provádí predikci struktury. Vše je založeno na faktu, že struktury homologních proteinů jsou během evoluce stálejší než sekvence aminokyselin. Nejčastěji se homology vyhledávají v databázi Protein Data Bank (PDB). Pokud homolog zkoumaného proteinu v této databázi neexistuje, je lepší využít např. threading.

První homologní modely byly sestaveny v šedesátých letech pomocí drátů reprezentujících vazby a plastikových kuliček místo atomů. První takový publikovaný model se však objevil až v roce 1969 (byl to kulovitý protein α-lactalbumin).



Obr. 11 Flow diagram homologního modelování [2]

Sestavování modelu proteinu pomocí homologního modelování je, jak je vidět na Obr. 11, vícekrokový proces. Ve většině kroků, ať už využívají automatického nebo manuálního modelování, jsou rozhodnutí o výběru zakládány převážně na experimentálních znalostech. Počet možných výběrů a přesnost modelu silně závisí na podobnosti mezi modelovaným proteinem – cílovou strukturou – a proteinem, nebo proteiny, které jsou použity jako model.

Pokud identita přesáhne 90%, kostra modelu bude stejně dobrá jako při krystalografii. Pokud ale identita spadne pod 25%, odhad není příliš spolehlivý a je třeba použít další experimentální data, která pomohou odhad zpřesnit. Identita však závisí na délce, proto je třeba říci, že 25% platí pro délku proteinu nad sto aminokyselin (pro méně než sto aminokyselin je třeba pro spolehlivý odhad zvyšovat minimální identitu).

Homologní modelování si klade dva předpoklady. Očekává se, že polypeptidová kostra konzervovaných regionů má u vzoru i cíle identické prostorové souřadnice. Také se očekává, že vkládání a mazání při zarovnávání sekvencí bude spadat hlavně do regionů, které se skládají z neuspořádané struktury, tedy struktury typu coil.

4.4.1 Kroky homologního modelování

Proces homologního modelování terciární struktury se skládá z několika oddělených kroků, které jsou znázorněny na Obr. 11. Existuji např. programy Swiss-PdbViewer nebo MolIDE, které jsou volně dostupné a pracují na tomto principu, část rozhodnutí je však třeba provést manuálně.

Před začátkem vlastního modelovacího procesu se musíme rozhodnout co dělat v případě, že jsme našli více homologů s řešeným problémem. Můžeme buď jednoduše vzít ten nejpodobnější vzor, udělat průměr založený na všech vzorech nebo použít fragmenty z různých vzorů. Každá z možností má své výhody a nevýhody. Model založený na jednom nejpodobnějším vzoru může být přesný, pokud jsou si vzor a zkoumaná sekvence velice blízké. Někdy však například nevíme nic o zkoumaném proteinu a průměr nám může dát mnohem bližší reprezentaci. Při spojování fragmentů nám může vzniknout mnoho chyb, může však dát nejlepší výsledky.

Prvním krokem je nalezení strukturních homologů ke zkoumanému proteinu. Ty můžeme nalézt například v databázi PDB (Protein Data Bank), kde je uloženo nejvíce experimentálně vyřešených struktur. Jako vzor je pak vybrán protein s největším skóre podobnosti. Když známe funkční rodinu cílového proteinu, můžeme v této rodině nalézt ideální vzor. Pokud nenajdeme žádný vhodný vzor, můžeme buď provést predikci ab initio, predikci typu threading, nebo můžeme hledat homology k jednotlivým krátkým regionům. Pokud vybereme poslední možnost, můžeme nalézt alespoň částečné řešení.

Když jsme našli vhodný vzor, cílová a vzorová sekvence musí být zarovnána. Tato fáze je klíčová. Model proteinu považujeme za třídimenzionální reprezentaci zarovnání sekvencí. Pokud je zarovnání špatné, přičemž chyba o jedno reziduum je již pokládána za vážnou chybu, dostaneme i špatnou terciární strukturu. Oblasti, které je obtížné zarovnat, by proto měly být překontrolovány. Taková

kontrola je založena na několika jednoduchých základních pravidlech, jako například že nejvíce nabitá rezidua by měla být na povrchu proteinu, dokud nejsou jejich náboje překonány ostatními náboji ve struktuře nebo není jiný důvod, proč by měl být atom zatlačen dovnitř struktury. Pokud jádro algoritmu nedokáže upravit zarovnání, bude nutné provést znovu celé zarovnání a znovu vše vymodelovat.

Nejdříve jsou modelovány konzervované oblasti struktury, tzv. jádra. Modelování jader je provedeno transformací souřadnic (x, y, z) každého odpovídajícího atomu zarovnaného rezidua ze vzorové do cílové molekuly. Tyto atomy jsou pak spojeny dohromady, aby zformovaly peptidové vazby se správnými úhly. Obvykle je možné zkopírovat souřadnice pouze některých postranních řetězců, protože mnoho z postranních řetězců nebude identických těm ve vzorové struktuře. Oblasti s inzercemi a delecemi jsou ponechány na později.

V této fázi je důležité zkontrolovat, zda jádro neobsahuje chyby, protože následují časově náročné procesy modelování smyček a minimalizace energie. Jakmile je sestavena struktura jádra, musí být zkontrolovány oblasti s inzercemi a delecemi a oblasti, které bylo obtížné zarovnat. Oblasti, kde zarovnání vyžadovalo vložení, by neměly být součástí nebo narušovat elementy sekundární struktury. V regionech, které bylo obtížné zarovnat, bychom měli prověřit, zda posun inzerce o jedno nebo dvě rezidua nevytvoří lepší konformaci.

Pokud inzerce narušila jádro nebo sekundární strukturu, pak je nutné se vrátit o krok zpět a znovu zarovnat vzor, vymodelovat jádro a provést kontrolu.

Jakmile jsme získali dobré zarovnání, provedeme další důležitou proceduru, modelování smyček. Smyčky jsou oblasti, které obvykle obsahují inzerce a delece a jsou to nejvariabilnější oblasti v sekvenci. Protože jejich variabilita je jak v samotné sekvenci, tak v délce, vymodelování těchto oblastí je velice obtížné. Smyčky jsou často využívány v rozpoznávání ligandů, ve vazbách ligandů a při hledání vazebních míst proteinu. Je proto důležité je modelovat co nejpřesněji.

Když zkoumaný protein obsahuje inzerci tak nemáme souřadnice, se kterými lze pracovat. Nejjednodušší je najít homolog, který má stejnou inzerci a následně k modelování chybějících částí využít souřadnice z homologu. Často však takový homolog nelze najít, pak se pomocí manuálních nebo automatických procedur hledají v databázi struktur s vysokým rozlišením (high-resolution structures) fragmenty stejné délky. Fragmenty jsou pak připojovány k jádru struktury. Místa, na která se fragment váže, se nazývají kotevní body (anchor points). Při zkoumání vhodnosti připojení fragmentu se kromě aminokyselin smyčky připojují ještě dvě aminokyseliny z každého konce. Při vyhodnocení vhodnosti fragmentů je počítán vliv každého z nich na okolí a ten, který interferuje s okolím nejméně a má s cílem největší podobnost, je do struktury připojen.

S krátkými delecemi se lze vypořádat pomocí lokální minimalizace energie, která nám spojí hranice smyčky. S většími delecemi je extrémně obtížné se vypořádat.

Některé programy, jako je např. COMPOSER, zkouší modelovat smyčky pouze pomocí homologních struktur. Tento způsob, který je založen na analýze smyček nalezených v homologních strukturách, je přesnější než jiné způsoby.

Postranní řetězce ovlivňují charakteristiky proteinu jak z pohledu struktury, tak funkce. Abychom predikovali konformaci postranních řetězců, když zarovnaná rezidua nejsou identická, využijeme knihovnu rotamerů. Rozsáhlé analýzy konformací postranních řetězců známých struktur ukázaly, že každý postranní řetězec je limitován na relativně malé množství energetických minim (resp. několik málo možných kombinací dihedrálních úhlů). Také bylo dokázáno, že konformace postranního řetězce je ovlivněna konformací hlavního řetězce (především typem jeho sekundární struktury). Proto byly vytvořeny knihovny rotamerů závislých na kostře (backbone dependent rotamer libraries). K modelování postranních řetězců bylo vyvinuto několik algoritmů využívajících různé knihovny rotamerů a lišících se v energetických funkcích. Která metoda je použita v praxi často závisí na tom, jakou program využívá pro modelování.

Když máme model, který se skládá z jader a smyček, je na smyčky aplikován algoritmus minimalizace energie. Ten se provádí především z toho důvodu, že oblasti smyček mají vlivem spojení z více různých proteinů špatnou geometrii. Existuje lokální minimalizace energie, při které je dovolen posun pouze atomů nebo jejich blízkého okolí. Dále existuje globální minimalizace energie, která se používá, když potřebujeme upravit úhly v kostře proteinu. U obou způsobů se minimalizace energie provádí běžně v padesáti až sto krocích.

V dalším kroku je důležité ověřit přesnost modelu a odhadnout pravděpodobnost a závažnost potenciálních chyb. Abychom odhadli chybu, porovnáme parametry struktury se strukturou zjištěnou krystalografií. Bylo vyvinuto několik metod, které by dokázaly chybovost odhalit jiným způsobem. Například prohledávání různých energetických kritérií nebylo nakonec dostatečně citlivé. Také může být počítáno s různými statistickými kritérii, jako jsou torzní úhly, úhly a délky vazeb, rozložení polárních a hydrofobických reziduí a vazby mezi rezidui. Na tomto principu funguje několik programů.

Program PROCHECK pracuje na principu sledování stereochemické kvality struktury. Zkoumá jaká je geometrie zkoumaného proteinu v porovnání s přesněji vypočítanými strukturami s vysokým rozlišením. Oblasti označené tímto programem nemusí nutně být chyby, mohou to být pouze méně obvyklé struktury. Vstupem takového programu je soubor se souřadnicemi modelu, výstupem pak soubory PostScript s grafem Ramachadran plot. Webový nástroj MolProbity pracuje na podobném principu. Dále existuje program WHAT_CHECK, jež je součástí balíčku WHAT IF. Ten porovnává vzory lokálních vazeb s průměrnými vzory vazeb podobných kontaktů dvou reziduí v databázi. Výstupem jsou aminokyseliny, které mají nějaké neobvyklé vlastnosti. Výstupem tohoto programu je nepřehledný a dlouhý seznam aminokyselin. Nástroj ANOLEA využívá k ověření životaschopnosti modelu různých znalostí. Životaschopnost ověřuje porovnáváním vztahů mezi různými typy reziduí pozorovanými ve skutečných strukturách proteinu. Z informací o interakcích je pak vypočítána energie. Pokud je tato energie vysoká značí to, že protein je v daném segmentu chybně zarovnán.

Výstupem programu ANOLEA je seznam všech reziduí s vysokou energií a sekvence, která je barevně obarvená.

Základními programy využívanými při automatizovaném homologním modelování jsou programy MODELLER a COMPOSER. MODELLER vytváří teoretickou cílovou strukturu na základě známé vzorové struktury a využívá při tom myšlenku splňování omezení, což jsou například vzdálenosti dvou atomů a dihedrální úhly. Nejdříve jsou tato omezení extrahována ze vzorové struktury a použity ve struktuře cílové a jsou pak kombinovány se základními pravidly využívanými ve struktuře proteinu, což jsou například délky vazeb a preferované úhly. K určení ekvivalentních reziduí mezi vzorem a cílem je využito zarovnání. Nakonec je provedena optimalizace, aby byla splněna prostorová omezení.

COMPOSER získává strukturu spojováním fragmentů z proteinů, které mají podobnou sekvenci. V prvním kroku jsou vyhledány v databázi homology k cílovému proteinu a je provedeno zarovnání. To je provedeno tak, že jsou nalezena ve všech homolozích nejméně tři topologicky ekvivalentní rezidua. Ta jsou pak využita k nalezení dalších topologicky ekvivalentních reziduí a ke znovu zarovnání struktury. Tyto kroky jsou opakovány, dokud lze najít další topologicky ekvivalentní rezidua. Všechna tato rezidua pak definují strukturně konzervované regiony (Structurally conserved regions, SCR), které následně definují průměrnou kostru. Dále je vypočítán přínos každého vzoru do této kostry a vzor s nejvyšším přínosem je pak považován za strukturu dané části proteinu. Postranní řetězce jsou predikovány pomocí výpočtů energie, podobně jako bylo popsáno výše, bylo však přidáno dalších 1200 pravidel. Výsledný model může být vylepšen například pomocí minimalizace energie nebo pomocí modelování založeného na omezeních.

5 Nástroje pro predikci

Seznam nástrojů pro tuto kapitolu čerpán především z [6] a [7]. Odkazy na detailnější informace o nástrojích budou uvedeny u popisu každého jednotlivého nástroje. Nástrojů využívaných pro predikci terciární struktury existuje velké množství, proto uvedu pouze příklad těch zajímavějších. Všechny nástroje pro modelování mají vesměs podobný charakter: jedná se o webovou službu, kde na vstupu je třeba vyplnit formulář (nejčastěji očekávající emailovou adresu a FASTA soubor se zkoumanou sekvencí) a výstup je odeslán buď na email (takovým výstupem je buď soubor PDB nebo odkaz na webovou stránku s výsledky), je vrácen PDB soubor přímo na stránce, nebo jsou výsledky graficky zobrazeny pomocí nějakého nástroje. Na stránce s graficky zobrazenými výsledky je obvykle i výsledný PDB soubor.

Obecně lze je rozdělit nástroje do čtyř kategorií podle metody, kterou využívají. Toto rozdělení je třeba vzhledem ke složitému principu fungování některých nástrojů brát spíše obrazně. Každý nástroj bude detailněji rozepsán v podkapitolách, které budou následovat. Jména podkapitol odpovídají kategoriím, úvodní podkapitolu však bude tvořit testování nástrojů. Dále popsané nástroje a kategorie nástrojů:

Homologní modelování

a) Swiss model
b) CPHmodels
c) ESyPred3D
d) Geno3d
e) AS2TS
f) 3D-JIGSAW

Threading

g) HHpred
h) SAM-T08
i) PSIpred
j) RaptorX

Ab initio

k) Rosetta

Kombinace více metod:

l) LOOPP
m) Fugue
n) Phyre^2 (následník 3D-PSSM)
o) Robetta (novější verze nástroje Rosetta)
p) I-TASSER (metaserver – kombinuje v sobě mnoho nástrojů)

Dále existuje několik nástrojů pro vyhodnocení a analýzu terciární struktury (pouze stručně):

a) Anolea
   Nástroj pro testování nelokálního prostředí atomů (Atomic Non-Local Environment).
b) LiveBench
   Slouží pro testování serverů pro predikci struktury
c) NQ-Flipper
   Poskytuje nástroje pro validaci a korekci asparaginových a glutaminových postranních řetězců rotamerů v proteinových strukturách, řešených pomocí rentgenové krystalografie
d) PROCHECK
   Nástroj pro verifikaci stereochemické kvality struktury proteinu.
   další informace: http://www.ebi.ac.uk/thornton-srv/software/PROCHECK/
e) ProSA-web
   Slouží k identifikaci chyb v terciární struktuře proteinu.
   další informace: https://prosa.services.came.sbg.ac.at/prosa.php (jen online)
f) QMEAN
   Server pro odhad kvality modelu
   další informace: http://swissmodel.expasy.org/qmean/cgi/index.cgi
g) What If
   Analytický program pro predikci mutací a verifikaci struktury.
   další informace: http://swift.cmbi.ru.nl/gv/whatcheck/

Příklady programů pro vizualizaci terciární struktury (existuje jich mnohem více):

a) Swiss-PdbViewer
   Program schopný analyzovat a vizualizovat terciární strukturu.
   další informace: http://spdbv.vital-it.ch/
b) MarvinSpace
   Vizualizační nástroj dostupný v různých jazycích
   další informace: http://www.chemaxon.com/download/marvin/
c) Jmol
   Volně dostupný nástroj v Javě sloužící pro zobrazení chemických struktur.
   další informace: http://jmol.sourceforge.net/
d) PyMOL
   Vizualizace molekul
   další informace: http://www.pymol.org/
e) VMD
   Program pro vizualizaci, animaci a analýzu velkých biomolekulárních systémů.
   další informace: http://www.ks.uiuc.edu/Research/vmd/

5.1 Testování modelovacích nástrojů (CASP, CAFASP)

Nástroje, jejichž detailnější informace budou popsány dále, jsou porovnávány ve dvou otevřených soutěžích. První soutěží je CASP, Critical Assessment of Strucure Prediction, druhou je CAFASP, Critical Assesment of Fully Automated Structure Prediction. Porovnávání se provádí predikcí struktury na předdefinovaných vzorcích, jejichž struktura byla již dříve zjištěna nukleární magnetickou rezonancí nebo rentgenovou krystalografií. Predikce je prováděna zadáním sekvence aminokyselin. Soutěže se konají každoročně, a každoročně se koná v Asilomar v USA setkání, kde jsou porovnávány výsledky.

Žádný ze způsobů predikce není absolutně přesný, homologní modelování s detailními vzory však dávají nejlepší výsledky. Pokud není žádný homolog nalezen, využívá se metody ab initio. V menší míře se pak využívá metody threading. Čerpáno z [2].

5.2 Nástroje homologního modelování

V této podkapitole popíšu nástroje využívající pouze metodu homologního modelování. Jedná se o nástroje Swiss model, CPHModels, ESyPred3D, Geno3D, AS2TS a 3D-Jigsaw.

5.2.1 Swiss-Model

Byl vyvinut na institutu bioinformatiky univerzity v Baselu (University of Basel). Jedná se o nástroj, který automaticky generuje modely pomocí techniky homologního modelování. Je založený na znalostech.

Nástroj je určen k tomu, aby poskytoval informace vědecké komunitě, a vyhrazuje si proto právo ukládat informace o využití svých stránek, a to pouze k internímu využití. Pro nekomerční využití nejsou kladena žádná zvláštní omezení. Je však zakázáno stahování celé databáze nebo jakékoli automatické stahování dat z ní, zrcadlení serveru a operace, které by mohly ohrozit jeho funkčnost. Komerční využití je také zdarma, ale vyžaduje podepsanou licenci.

Server je dostupný přes webový server ExPASy nebo pomoci programu DeepView (Swiss Pdb-Viewer).

Nástroj pracuje ve dvou režimech. Režim automatického modelování přijímá vstupní soubor typu FASTA se sekvencí aminokyselin, a volitelně také vzor který se má použít (buď jako PDB-ID nebo jako soubor). Alignment mód přijímá soubor se zarovnáním, a to buď ve formátu FASTA, CRUSTALW, DeepView, MSF, PFAM nebo SELEX. Tento mód se využívá především pokud známe strukturu alespoň jedné ze zarovnaných sekvencí.

Výstupem jsou, krom dalších informací o průběhu zpracování, soubory pro program DeepView a soubor ve formátu PDB. Odkaz na výsledky je odeslán emailem, zobrazí se i po odeslání formuláře se vstupními údaji (vstupní sekvence ve FASTA formátu a email kam poslat upozornění) a jejich zpracování.

Odkaz pro další informace: http://swissmodel.expasy.org/

5.2.2 CPHmodels

Tento server vznikl v centru pro analýzu biologických sekvencí (CBS) na oddělení systémové biologie technické univerzity v Dánsku (DTU, Technical Univerzity of Denmark). Současná verze nástroje má číslo 3.2. Pracuje na principu homologního modelování a k rozpoznávání vzorů využívá zarovnání profilů. Pro lepší efektivitu pracuje se sekundární strukturou a s predikcemi vlivu na okolí.

Ke své činnosti využívá dva nástroje. Nástroj Sowhat, což je neuronová síť schopná predikovat kontakty mezi atomy C-alfa, a nástroj RedHom, který umí najít podmnožinu dat s málo podobnými sekvencemi.

Přístup na predikční servery CBS je pro akademické uživatele volný a bez limitů. Pro všechny ostatní uživatele je také volný a limity jsou obecně pro všechny servery CBS nastaveny na padesát dotazů denně s maximálně dvěma tisíci sekvencemi na dotaz. Na samotném serveru CPHmodels je však omezení trochu jiné, a to na počet aminokyselin, který je v rozmezí od 15 do 4 000.

Vstup tvoří soubor FASTA s jednou nebo více sekvencemi, výstup tvoří soubor PDB (lze jej nalít na linku s výsledky spolu s dalšími informacemi o zpracování, který se zobrazil buď po odeslání formuláře a zpracování, nebo dorazil emailem).

Pro některé servery je umožněn programátorský přístup pomocí technologie SOAP. Tyto služby jsou zatím volné a neomezené pro všechny uživatele. Služba CPHmodels bohužel mezi podporované služby nepatří. Je však možné pomocí e-mailu získat přenosnou verzi.

Odkaz pro další informace: http://www.cbs.dtu.dk/services/CPHmodels/

5.2.3 ESyPred3D

Automatizovaný nástroj ESyPred3D byl vyvinut na výzkumném institutu molekulární biologie na univerzitě v Namuru, Belgie (The Univerzity of Namur). Současná verze 1.0.

Nástroj má vylepšenou efektivitu při zarovnávání, protože využívá neuronové sítě a výsledky zarovnávání kombinuje z několika nástrojů pro vícenásobná zarovnání. Výsledná terciární struktura je vytvořena pomocí modelovacího balíčku MODELLER.

Přesné informace o omezeních provozu, licencích apod. jsem nenašel. Na stránkách není ani odkaz na nějakou dokumentaci, fórum nebo něco podobného.

Odkaz pro další informace: http://www.fundp.ac.be/sciences/biologie/urbm/bioinfo/esypred/

5.2.4 Geno3D

Nástroj byl vyvinut na institutu biologie a chemie proteinů v Lyon-Gerland (PBIL-IBCP Lyon-Gerland). Současná verze má číslo 2. Na využívání tohoto nástroje pro neziskové organizace nejsou kladena žádná licenční omezení. Server verze 2 však generuje modely pouze do pětiset aminokyselin.

Ve druhé verzi je možno vybrat vzory s nízkou identitou (pod 20%, ale je zdůrazněno, že se má využívat opatrně), je možno pomocí predikce sekundární struktury validovat výběr a zarovnání vzoru a je možno vybrat více vzorů. Také byla přidána dostupnost nástroje pomocí webového rozhranní.

Nástroj na vstupu přijímá sekvenci aminokyselin. Následně uživatel vybere vzory, jakožto některé z vypočítaných homologů z PDB. Odkaz na výsledky je zobrazen po odeslání vstupního formuláře.

Odkaz pro další informace: http://geno3d-pbil.ibcp.fr/cgi-bin/geno3d_automat.pl?page=/GENO3D/geno3d_home.html

5.2.5 AS2TS

Tento systém vyvinula Lawrence Livermore National Laboratory nacházející se v Livermore, USA. Informace o licenci nebo omezení použití jsem nenašel. Je zde obsaženo několik nástrojů na analýzu sekvencí, analýzu a modelování struktury, některé z nich je možno si stáhnout. Jedná se např. o nástroje AL2TS (viz dále) a nástroje pro vizualizace RasTop a RasMol2.

Nejzajímavějšími nástroji jsou AL2TS (ALignment TO Tertiary Structure), jehož vstupem je soubor se zarovnáním sekvence a struktury ve formátu AL, SAL nebo BLAST. Pokud nemáme potřebná data, je pro nás vhodný nástroj AS2TS (Aminoacid Sequence TO Tertiary Structure).

Pro nástroj AS2TS existují dvě verze, které se liší pouze v interpretaci výsledků. Klasická verze zobrazí výsledky ve formě textových informací. Verze AS2TS – Model Builder (v době psaní textu mimo provoz z důvodu údržby) zobrazuje výsledky v přehlednější grafické formě. Obě verze přijímají jako vstup soubor ve formátu FASTA.

Na dokončení zpracování můžeme být upozorněni emailem, pokud jsme jej zadali. Jinak je zobrazen odkaz na výsledky ihned po odeslání formuláře.

Odkaz pro další informace: http://proteinmodel.org/AS2TS/system.html

5.2.6 3D-JIGSAW

Využití nástroje je zdarma pro výzkumné účely, nesmí být použita pro přímý komerční zisk. Při nadměrném využívání služby si Cancer Research UK vyhrazuje právo zablokovat službu nebo vyžadovat platbu.

Nástroj je nyní ve verzi 2. Tato verze slibuje lepší výsledky za cenu delšího zpracování. Výhodou je možnost výběru automatického nebo interaktivního přístupu (kdy uživatel může rozdělit sekvenci na domény, vybrat vzory a editovat zarovnání). Na vstupu jsou vyžadovány klasické informace, e-mail, na který je odeslán výsledek zpracování formou PDB souboru, a sekvence aminokyselin. Navíc je vyžadováno zadání identifikátoru proteinu.

Odkaz pro další informace: http://bmm.cancerresearchuk.org/~3djigsaw/

5.3 Nástroje threading

V této podkapitole se budu zabývat nástroji, které využívají pouze metodu threading. Jedná se o nástroje HHpred, SAM-T08, PSIpred a RaptorX.

5.3.1 HHpred

Pochází z „planckova institutu pro vývojovou biologii“ (Max Planck Institute for Developmental Biology), jenž se nachází ve městě Tübingen v Německu. Krom nástroje HHpred lze na stránkách naleznout mnoho nástrojů pro další práci s 3D strukturou, 2D stukturou, klasifikací, analýzu sekvencí, zarovnání apod.

HHpred je založen na dvou nástrojích zpracovávajících markovovy modely (HMM), a to HHblits pro detekci homologií a HHsearch, který umí vyhledat v databázi vzory podobné profilu HMM nebo vícenásobnému zarovnání sekvencí. Oba nástroje jsou dostupné v open-source balíčku HH-suite 2.0 a fungují na platformách Linux a MAC OS X. Pro webovou službu samotného nástroje HHpred jsem nenašel žádná licenční omezení.

Na webovém serveru je vyžadováno zadání sekvence ve formátu FASTA nebo čistě jako sekvence nebo je možno zadat vícenásobné zarovnání v těchto formátech: CLUSTAL, Stockholm, A2M, A3M, EMBL, MEGA, GSG/MSF, PIR/NBRF, TREECON. Dále je možno nastavit některá nastavení týkající se vyhledávání, např. vybrat databáze pro HMM, zda se má použít Psiblast nebo HHblits apod.

Po odeslání vstupního formuláře je stránka obnovována, dokud nejsou zobrazeny částečné výsledky. Např. při zadání FASTA na vstupu je však zobrazena volba, zda chceme výsledný model vytvořit manuálně (ručně vybrat vzory) nebo automaticky. Dále je pak potřeba vybrat kolik vzorů se má zarovnávat – jeden nebo více. Tyto výsledky zarovnání jsou pak odeslány do nástroje MODELLER, kde zadáme jméno a odešleme na další zpracování. Výstupem nástroje MODELLER je již kýžený PDB soubor. Krom 3D struktury jsou zobrazeny i hodnocení kvality nástroji SOLVX, VERIFY3D a ANOLEA.

Odkaz pro další informace: http://toolkit.tuebingen.mpg.de/hhpred

5.3.2 SAM-T08

Tento server vznikl a vyvíjel se pod vedením Kevina Karpluse. Byl testován na CASP8 a jeho výsledky byly dobré na primární server, metaservery kombinující výsledky z více serverů měly však celkové výsledky lepší.

Informace o licenci nebyla na stránkách nalezena. Sekvence by však z důvodu menšího přetěžování serveru neměly přesahovat 700 aminokyselin. Zpracování je omezeno na jednu sekvenci (nelze zpracovávat více sekvencí paralelně).

Server hledá podobné proteiny v NR a zarovnává je. Výstupem této fáze jsou sekvenční loga, která ukazují relativní konzervovanost jednotlivých pozic v sekvenci. Predikce lokální struktury je prováděna pomocí neuronové sítě a pomocí HMM, které jsou pro zvýšení efektivity také vytvořeny. Následně jsou prohledávány foldy a jsou prováděna zarovnávání v PDB. Podle výsledného

zarovnávání je pak konstruován 3D model. U výsledku je třeba se dívat na hodnotu E (Error, chyba), protože se zde 3D modely konstruují i když je tato hodnota vysoká.

Na vstupním webovém formuláři je vyžadována emailová adresa, na kterou mohou být odeslány výsledky. Je možno zvolit, zda chceme emailem poslat predikovanou sekundární strukturu (formát CASP SS), predikci kontaktu dvou reziduí (formát CASP RR) a nejlepších pět modelů (formát CASP TS). Dále je vyžadována vstupní sekvence ve formátu FASTA. Odkaz na výsledky se zobrazí po zadání formuláře přímo na webu.

Odkaz pro další informace: http://compbio.soe.ucsc.edu/SAM_T08/

5.3.3 PSIpred

Nástroj byl vytvořený v Bloomsburyho centru pro bioinformatiku, které náleží k Univerzity of London. Obsahuje několik různých predikčních metod. Metoda PSIPRED slouží pro predikci sekundární struktury. Pro nás jsou však významnější metody GenTHREADER a pGenTHREADER. Metoda pGenTHREADER je vylepšením GenTHREADER, je úspěšnější, ale o něco pomalejší, protože navíc využívá zarovnání dvou profilů (profile-profile alignments) a jako vstup přijímá sekundární strukturu (predikovanou pomocí PSIPRED).

Nástroj je volně dostupný i pro komerční použití avšak s tím, že veškerá data jsou nějakou dobu (v současnosti asi 5 dní) veřejně dostupná. Pokud je opravdu nutné utajení dat, je možnost spolupráce. Uživatelé používající automatické skripty bez předchozího svolení budou potrestáni. Počet úloh není omezen, zároveň jich však může běžet pouze deset.

Pokud je třeba vyšší priorita úloh, vyšší kapacita apod. je možno si zaplatit např. samostatný server.

Na webovém formuláři je třeba nejdříve vybrat metodu predikce (např. GenTHREADER, pGenTHREADER, predikce sekundární struktury apod.). Dále je třeba zadat vstupní sekvenci, a to buď jako vícenásobné zarovnání (ve formátu FASTA, pozor aby byly všechny sekvence stejně dlouhé), jednu sekvenci ve formátu FASTA nebo čistě jako posloupnost znaků. Dále lze nastavit několik možností týkajících se filtrování a zarovnání a lze nastavit emailovou adresu, kam přijde upozornění o dokončení práce. Je vyžadováno ještě zadání pojmenování zadaného úkolu. Odkaz na stránku s výsledky je odeslán emailem. Stránka s výsledky je zobrazena ihned po odeslání formuláře.

Odkaz pro další informace: http://bioinf.cs.ucl.ac.uk/psipred/

5.3.4 RaptorX

Nástroj vyvinula skupina Xo, a slouží pro predikci sekundární a terciární struktury. Také umí provádět zarovnání struktury k záznamu v PDB. Na webu nebyla nalezena žádná licenční omezení, pouze že každý může zadat do fronty pouze dvacet sekvencí, jeden job může obsahovat maximálně deset sekvencí. Více zdrojů lze domluvit prostřednictvím emailu. Nástroj měl výborné výsledky v soutěži CASP9 a je velice vhodný na struktury, které mají s vyřešenými strukturami v PDB identitu méně než 30%.

V současnosti se skládá ze čtyř hlavních modulů: Threading s jedním vzorem (single-template threading), testování kvality zarovnání, threading s více vzory (multi-template threading), modelování nevyužívající fragmentaci.

RaptorX lze stáhnout jako balíček, je však třeba do formuláře na webu zadat jméno, jméno organizace a email s koncovkou .edu, .edu.* nebo .gov. Pro ostatní emailové adresy, nejlépe poskytované přímo organizací namísto např. gmail, je třeba kontaktovat přímo vývojovou skupinu.

Webový formulář vyžaduje na vstupu obvyklé údaje, email a vstupní sekvenci ve formátu FASTA. Dále můžeme vybrat, zda chceme predikovat sekundární strukturu, terciární strukturu nebo obojí. Po odeslání vstupního formuláře lze zobrazit stránku s výsledky, výsledky jsou rovněž odeslány na email.

Odkaz pro další informace: http://raptorx.uchicago.edu/predict/

5.4 Nástroje ab initio

Kategorie nástrojů využívající pouze metodu ab initio má jednoho jediného zástupce, Rosettu.

5.4.1 Rosetta

Tento software byl vytvořen na University of Washington. Univerzita negarantuje soukromí (data nahraná na stránku mohou být veřejně přístupná).

Zpracování na serveru Rosetta může trvat několik hodin, proto je doporučeno, pokud máme k dispozici vhodný hardware, stáhnout balíček programů Rosetta ™, který je po zřízení licence dostupný pro nekomerční i komerční užití (komerční užití je však zpoplatněno částkou 40 000$).

Webový nástroj je přístupný pomocí rozhranní PyRosetta, které je napsané v pythonu. Vstup tvoří soubor FASTA, kde počet aminokyselin, je omezený od 40 do 650. Doporučeno jich je přibližně 200. Lze zadat i sekvenci DNA, protože ji lze přímo ve formuláři převést na sekvenci aminokyselin. Výstupem je pak soubor ve formátu PDB, který se zobrazuje nástrojem RasMol.

Na vstupním formuláři je vyžadováno zadání registrované emailové adresy nebo registrovaného jména. Dále je vyžadováno zadání sekvence ve formátu FASTA a jméno úlohy.

Odkaz pro další informace: http://boinc.bakerlab.org/rosetta/

5.5 Nástroje kombinující více metod

Tato podkapitola se bude zabývat nástroji, které kombinují různým způsobem metody threading, homologní modelování a ab initio. Jedná se o nástroje LOOPP, Fugue, Phyre2, I-Tasser a Robetta (novější verze již zmíněného nástroje Rosetta).

5.5.1 LOOPP

Vznikl v rámci CBSU, Computational Biology Service Unit, který je podporován Microsoftem. Nástroj je open-source, zdrojové kódy jsou napsané v Perlu, C++ a FORTRANu a je dostupný přes SVN. Je platformě agnostický, testován byl primárně na clusterech založených na Linuxu.

Rozpoznávání foldů je založeno na sběru několika signálů a jejich spojování do jednoho výsledného skóre. Následně jsou na základě zarovnání k homologní struktuře generovány souřadnice atomů. Signály, které se používají, mohou vycházet např. ze zarovnání sekvencí, profilu sekvence, ze samotného procesu metody threading, sekundární struktury nebo z údajů predikce exponovaných ploch.

Webový formulář vyžaduje na vstupu obvyklé základní údaje, a to sekvenci a emailovou adresu, kam budou odeslány výsledky zpracování. Navíc vyžaduje jméno zadané úlohy. Odkaz na výsledky je odeslán emailem.

Odkaz pro další informace: http://loopp.org/

5.5.2 Fugue

Tento nástroj vznikl na ústavu biochemie na univerzitě v Cambridge a spolupracuje s knihovnou HOMSTRAD (Homologous Structure Alignment Database). Současná verze 2.0.

Nástroj je psaný v ANSI C a je volně dostupný ke stažení. Lze jej využít na platformách typu UNIX a po obstarání licence je povolen pro akademické uživatele. Pro webové služby hledání homologů a zarovnání jsem nenašel žádná speciální omezení.

FUGUE je program pro rozpoznávání vzdálených homologů pomocí porovnávání sekvence a struktury. Využívá substituční tabulky závislé na okolí a na penalizacích za mezery. Skóre pak závisí na okolí každé aminokyseliny v nějaké již známé struktuře. Po zadání sekvence (nebo zarovnání sekvencí) je skenována databáze struktur a je vypočítáno skóre kompatibility. Následně je vyprodukován list potenciálních homologů a zarovnání.

Webový server požaduje na vstupu e-mailovou adresu, kam pošle výsledky. Dále buď sekvenci aminokyselin ve formátu FASTA, nebo zarovnání sekvencí ve formátu FASTA, NBRF, CLUSTAL nebo MSF, přičemž je třeba vybrat, kterou variantu jsme zadali (jedna sekvence a chceme hledat homolohy pomocí PSI-BLAST, jedna sekvence ale chceme predikovat přímo bez homolohů, nebo zarovnání více sekvencí). Dále existuje několik doplňujících volitelných parametrů.

Odkaz na výstup je zobrazen ihned po odeslání formuláře. Je zobrazen také odkaz na stránku, kde můžeme sledovat status námi zadané predikce. Stejné odkazy také dorazí na zadanou emailovou adresu.

Odkaz pro další informace: http://tardis.nibio.go.jp/fugue/

5.5.3 Phyre^2

Název pochází z anglického Protein Homolohy/analogY Recognition Engine verze 2, v překladu nástroj pro rozpoznávání homologie/analogie proteinů. Nástroj byl vytvořen na Imperial College v Londýně.

Nástroj je pouze pro akademické použití. Materiály přístupné přes stránky projektu jsou k dispozici pouze pro osobní, akademické, nekomerční a informační účely. Komerční užití může být potrestáno zakázáním nástroje nebo vyžadováním poplatku.

Jedná se o automatizovaný nástroj využívající porovnání pomocí dvou skrytých markovových modelů. Jeden model byl získán pomocí hledání homologů z prohledávané sekvence, druhý transformací foldu z databáze (výběr vzorů proveden pomocí nástroje HHpred). Tento princip podstatně zvyšuje účinnost zarovnání a schopnost detekce (je schopen detekovat velmi vzdálenou homologii a vytvářet poměrně přesné modely i při identitě menší než 15%). Také je schopen provést simulaci typu ab initio, nazývanou Poing, kterou využívá u proteinů, kde nebyla detekována žádná homologie. Nástroj má dva módy modelování, normální je rychlý, ale méně přesný a intenzivní provádí porovnávání podle většího množství vzorů a využívá již zmíněnou metodu ab initio.

Na vstupu je vyžadována sekvence aminokyselin a emailová adresa, výsledky jsou zobrazovány průběžně po odeslání formuláře (a lze je pak stáhnout jako zazipovaný soubor, nebo lze stáhnout PDB soubor), případně po zpracování dorazí e-mailem.

Odkaz pro další informace: http://www.sbg.bio.ic.ac.uk/phyre2/html/page.cgi?id=index

5.5.4 Robetta

Tento software byl vytvořen na University of Washington a je vylepšením nástroje HMMSTR/Rosetta. Aktuálně je pouze ve verzi Beta a je pouze pro nekomerční použití. Univerzita negarantuje soukromí (data nahraná na stránku mohou být veřejně přístupná).

Základní princip a podmínky použití jsou stejné jako u původní verze nástroje.

Nejdříve je protein pomocí protokolu Ginzu rozdělen na domnělé domény, a na tyto domény je použito buď homologní nebo ab initio modelování. Další služby, které tento server poskytuje, je rozlišování domén, 3D modelování, generování knihovny fragmentů a „Interface Alanine Scaning“ (pokouší se odhadnout, jak hodně přispívá každé reziduum v rozhranní mezi dvěma proteiny k volné vazební energii).

Odkaz pro další informace: http://robetta.bakerlab.org/index.html

5.5.5 I-TASSER

Nástroj patří univerzitě v Michiganu a je volně dostupný pro nekomerční použití. Modelování je založeno na vícenásobném zarovnání. I-TASSER se stal vítězem v oblasti predikce struktury na soutěžích CASP7 až CASP10 a v soutěži CASP9 zvítězil i v predikci funkce.

Nástroj je dostupný v balíčku, který lze stáhnout na domovské stránce. Akademická licence je pak zdarma (po registraci), komerční licence také zdarma, ale po podpisu smlouvy.

Predikce na samotném serveru I-TASSER trvá obvykle jeden až dva dny, při větším vytížení i déle. Záleží také na velikosti sekvence proteinu.

Krom hlavního serveru je možno využít server LOMETS (LOcal MEta-Threading Server), kde je zakomponováno deset lokálně nainstalovaných programů využívajících threading (FUGUE, HHsearch, MUSTER, PPA, PROSPECT2, SAM-T02, SPARKS, SP3, FFAS a PRC). Z těchto nástrojů je vygenerováno 200 modelů, každý nástroj jich generuje deset. Z těchto všech modelů je pak podle skórovací funkce vybráno 10 nejlepších.

Dále je možno využít server MUSTER (MUlti-Source ThreadER). Tento server, který pracuje metodou threading, využívá šest různých zdrojů: profily odvozené ze sekvence, sekundární struktury, profily odvozené ze struktury, dostupnost rozpouštědel, torzní úhly Phi a Psi a skórovací matici. Výstup serverů MUSTER a LOMETS je vytvořen pomocí balíčku MODELLER.

Samotný server I-TASSER pracuje tak, že zkusí pomocí LOMETS vybrat z PDB vzory, které mají podobný fold nebo supersekundární strukturu. Pokud takové vzory nejsou nalezeny, je využita metoda ab initio. Potom jsou pomocí nástroje TMalign, který vyvinula stejná instituce, vzory z nástroje LOMETS a struktury z PDB zarovnány a s tímto zarovnáním je dále pracováno. Pro lepší výsledky je provedena ještě jedna další iterace. Nakonec jsou z výsledků vybrány struktury s nejmenší energií. Výsledné modely atomů jsou získány pomocí nástroje REMO, což je opět jejich vlastní nástroj, který výsledné modely konstruuje optimalizací páteřních vodíkových vazeb.

Na vstupu je na všech třech serverech vyžadována emailová adresa, kam budou odeslány výsledky, a sekvence ve formátu FASTA. Na samotném serveru I-TASSER je navíc vyžadována registrace – při zadání sekvence je pak krom emailové adresy vyžadováno heslo a stránka s výsledky je dostupná ihned po odeslání formuláře.

Odkaz pro další informace: http://zhanglab.ccmb.med.umich.edu/I-TASSER/
http://zhanglab.ccmb.med.umich.edu/LOMETS/
http://zhanglab.ccmb.med.umich.edu/MUSTER/

5.6 Možnosti automatické komunikace s nástroji

V této kapitole se budu zabývat možností automatické komunikace s jednotlivými nástroji. Nabízí se několik možností jak takovou komunikaci provést, a to buď přímo využít webový formulář, lokálně nainstalovaný program nebo využít rozhranní (API). Každá z možností má své nesporné výhody a nevýhody.

5.6.1 Komunikace pomocí formuláře

Většina nástrojů přijímá vstupy pomocí webového formuláře, nabízí se tedy možnost komunikace pomocí nich. Nejčastěji proces zadávání dat probíhá tak, že do formuláře zadáme vstupní údaje a odešleme je na server ke zpracování. Ve většině případů je pak zobrazen odkaz na stránku s výsledky, případně jsou přímo výsledky zobrazeny. Takovýto formulář může být převzat přímo z webu a po částečné úpravě by byl využitelný externím programem. Problém tohoto způsobu je především ten, že čas od času jsou stránky aktualizovány, a drobná změna v HTML kódu může způsobit nefunkčnost komunikace. Dalším problémem je to, že musíme po určité době zkoušet získat stránku s výsledky, dokud výsledky nejsou zobrazeny (neexistuje možnost, jak zjistit efektivněji zda bylo dokončeno nebo jak dlouho bude proces ještě trvat).

5.6.2 Lokální instalace nástroje

Druhá možnost, kterou však nabízejí pouze některé nástroje, je stáhnout si daný nástroj a nainstalovat lokálně na server. Odpadala by tak nutnost komunikace s ostatními servery, což je poměrně velká výhoda, a při dostatečném výkonu by mohl být protein i rychleji zpracovaný. Problém však nastává v několika záležitostech. Zaprvé by byla poměrně složitá údržba, hlavně z důvodu časté aktualizace některých nástrojů. Zadruhé je problém se zmíněným dostatečným výkonem, těžko můžeme docílit rychlejšího zpracování, než když každý nástroj poběží na svém vlastním serveru.

Možnost stažení a instalace nástroje a vhodné licenční podmínky má pouze část z výše popsaných nástrojů. Je možnost lokálně nainstalovat nástroj CPH-Models, kde je akademická licence bez omezení a stažení je možné přes email (bližší informace o této přenosné verzi nejsou k nalezení). Bez omezení lze také stáhnout a využívat AL2TS, což je open-source nástroj, napsaný v jazyce C patřící k serveru AS2TS.

Dále je možnost stáhnout nástroj Fugue, který je open-source, open-source balíček HH-suite, ve kterém jsou obsaženy např. nástroje HHblits a HHsearch (ze kterých vychází predikční nástroj HHpred), open-source nástroj LOOPP, po obstarání licence emailem nástroj RaptorX, a I-TASSER SUITE, sadu nástrojů meta-serveru I-TASSER.

5.6.3 API

Nejideálnější možností vzdálené komunikace je API, nebo obecně jakékoli rozhranní pro vzdálenou komunikaci se serverem. Vzhledem k tomu, že se většina serverů a nástrojů spoléhá nejvíce sama na sebe, případně na lokálně nainstalované nástroje, není tato možnost příliš rozšířená. Přináší však výhodu jak využitelnosti výkonu ostatních serverů, tak efektivity práce s nimi.

Rozhranní API, mají z popsaných nástrojů pouze dva. Je to ROBETTA/ROSETTA, která má interface psaný v jazyce Python a je volně dostupný pro nekomerční užití a nástroj PSI-PRED, který má interface napsaný v jazyce RUBY.

6 Návrh serveru

Jedním z hlavních cílů této práce je navrhnout server pro predikcí terciární struktury proteinu. Jedná se o server, který přijme vstupy od uživatele, automaticky zvolí vhodné nástroje, se kterými bude sám komunikovat, získá od nich výsledky predikce a tyto výsledky vhodně uživateli zobrazí.

6.1 Programovací jazyk

Možností volby jazyka, ve kterém bude server naprogramován, je mnoho, a každý má své nesporné výhody a nevýhody. Tento server bude vytvořen v jazyce JSP, Java Server Pages. Nejdříve jsem přemýšlel o vytvoření serveru formou klient-server aplikace, která by však měla nevýhodu potřeby instalace, resp. stažení, klienta, který by uměl komunikovat se serverem.

Naproti tomu JSP je jazyk, který využívá velice rozšířený, propracovaný a moderní jazyk JAVA, a kombinuje jej s klasickým HTML. Základní principy fungování jazyka JSP jsou podobné i dalším v této oblasti často používaným programovacím jazykům, PHP a ASP.NET Web Forms. ASP.NET je designovaný spíše pro Windows, což z důvodu potřebné nezávislosti na operačním systému není výhodné. Jazyky JSP a PHP jsou v tomto ohledu agnostické. S ohledem na mé zkušenosti s programováním v jazyce JAVA jsem zvolil z této dvojice JSP. [8]

6.2 Vstupy uživatele

Vstupy od uživatele budou formou formuláře vyžadujícího několik základních údajů. Především vstupní sekvenci proteinu, a to buď ve formátu FASTA (samostatný soubor nebo jako text, vložený do textového pole) nebo jako čistý text (sekvence nukleotidů). Dále bude vyžadována emailová adresa, protože některé nástroje ji vyžadují.

Dále bude vstupní formulář obsahovat několik málo polí, kde budou možnosti uživatelské volby některých proměnných, které ovlivní výběr použitých nástrojů, případně jejich nastavení.

6.3 Princip činnosti serveru

Cílem serveru je propojit výsledky predikce struktury proteinu z několika nástrojů a poskytnout tak přesnější výstup. Základním výstupem všech nástrojů je soubor ve formátu pdb, ve kterém je uložena výsledná predikovaná struktura proteinu. V tomto souboru jsou obsaženy pozice a typ každého atomu, jejich propojení, informace o postranních řetězcích a další důležité informace.

Propojení výsledků by se dalo provést několika způsoby, jako např. rozdělením sekvence proteinu na části a predikce každé části jedním nástrojem. Dále by se dalo procházet jednotlivé atomy, a souřadnice každého z nich predikovat dle výsledků z více nástrojů (např. pomocí aritmetického průměru nebo výběrem nejčastější hodnoty). Tyto metody by však pravděpodobně byly poměrně složité a přesnost by příliš nezvýšily, spíše naopak.

Hlavním důvodem pravděpodobného snížení přesnosti metodou průchodu jednotlivými atomy je, že terciární struktura příliš nezávisí na jednotlivých aminokyselinách, ale na komponentách sekundární struktury. Kdybychom přiřazovali jednotlivé atomy místo celků, mohli bychom nejen porušit

celistvost aminokyselin (protože procházíme jednotlivé atomy, ze kterých se aminokyseliny skládají), ale i pozměnit sekundární strukturu a celkovou strukturu proteinu tak ještě více zdeformovat.

Rozdělením na menší části bychom došli k obdobnému problému, s velkou pravděpodobností bychom porušili sekundární strukturu. Jediná situace, kdy by takovýto princip byl použitelný, je rozdělení vícedoménových proteinů na jednotlivé domény a na každou doménu pak použít jiný nástroj. Jednotlivé domény by však ze stejných důvodů nebylo vhodné rozdělovat.

Pro implementaci serveru jsem zvolil jiný způsob. V první fázi otestuji přesnost predikce jednotlivých nástrojů pro různé typy proteinů. V ideálním případě zjistíme, že každý nástroj je vhodný na určitý typ proteinu, a ten bychom potom ve vhodné kombinaci s dalšími nástroji využili ke zjištění výsledné struktury.

Server přijme vstupy od uživatele a kontaktuje některé z nástrojů za účelem zjištění typu proteinu. Pro rozdělení proteinů na jednotlivé typy se nabízí jejich sekundární struktura, lze využít např. tříd proteinů v databázi SCOP. Za účelem zjištění typu proteinu tedy kontaktuje některý z nástrojů, které umí v databázi SCOP vyhledávat. Poté podle zjištěného typu proteinu vybere nástroj, který je pro daný typ proteinu nejvhodnější. Předá data zvolenému nástroji a zjistí výsledky (stáhne výsledné pdb soubory a případně další informace jako odhadovaná přesnost predikce, identita s použitým homologem, apod.). Výsledky z různých použitých nástrojů poté vhodně vizualizuje.

6.4 Princip výběru nástrojů

Ideálním způsobem pro automatickou komunikaci s nástroji by bylo, kdyby poskytovaly API. To však poskytuje zanedbatelné množství nástrojů, je třeba je tedy vybírat podle jiných kritérií. Velice důležité jsou údaje týkající se samotné predikce, například její rychlost a přesnost, forma předání výsledků, složitost implementace automatické komunikace s nimi, nebo princip funkčnosti (zda pracuje metodou homologního modelování, metodou threading nebo je nějak kombinuje), apod.

Pro správný výběr nástrojů je bude tedy potřeba vhodně otestovat a rozdělit podle toho, na které typy proteinů je jejich predikce nejpřesnější. Ideální bude zkombinovat několik nástrojů pracujících metodou homologního modelování s několika dalšími, které pracují metodou threading (případně kombinují více metod), pro případ, že homologní modelování selže (nebude existovat vhodný homolog v databázi proteinů).

7 Testování nástrojů

7.1 Výběr vzorových proteinů a způsob provedení testů

Testování nástrojů bude prováděno pomocí experimentálně zjištěných proteinů, které lze stáhnout na stránkách PDB. Použiji proteiny zjištěné pomocí rentgenové krystalografie s rozlišením 1.5 až 2Å, abych zajistil určitou spolehlivost a přesnost testování. Takto vybrané proteiny lze rozdělit do tříd dle databáze SCOP (Structural Classification Of Proteins) do šesti kategorií:

a) A/B - hlavně paralelní beta-listy
b) A+B - hlavně anti-paralelní beta-listy
c) all alpha - domény skládající se pouze z alfa-šroubovic
d) all beta - domény skládající se pouze z beta-listů
e) coiled coil - obsahuje více alfa-šroubovic stočených dohromady
f) alpha and beta - obsahuje 2 a více domén, které patří do více tříd

Seznam pdb id jednotlivých vybraných testovacích proteinů v jednotlivých SCOP třídách lze nalézt v příloze2.

Protože se jedná o vybírání nástrojů, které budou začleněny do serveru až později, bude testování prováděno z větší části ručně a na menším množství vzorků, přibližně jich bude pět v každé kategorii. Ruční způsob testování je výhodný mimo jiné pro zjištění dalších dat o nástrojích, jako je přibližná doba zpracování, přívětivost uživatelského rozhranní, detailnost informací, které jsou zobrazeny stránce s výsledky a v e-mailu apod. Zjištění všech těchto dat by bylo z takového množství nástrojů poměrně složité.

Výsledky predikce, které budou především ve formátu pdb, bude potřeba porovnat s pdb soubory experimentálně zjištěnými. K tomu využiji nástroj iPBA[1], který umí porovnat dva soubory ve formátu pdb a poskytnout o něm různé statistické informace.

Pro efektivnější testování, resp. porovnávání, výsledků z nástrojů implementuji v jazyce JAVA program, který projde zadané složky s referenčními pdb soubory a s pdb soubory, které byly výstupem z jednotlivých nástrojů, a provede jejich porovnání. Výsledné hodnoty z nástroje iPBA poté uloží do textového souboru tak, aby šly zpracovat v jiných programech, např. v programu Microsoft Excel.

[1] iPBA -- http://www.dsimb.inserm.fr/dsimb_tools/ipba/index.php

7.2 Nástroj pro automatické porovnávání pdb souborů

Vstupem implementovaného programu budou cesty ke dvěma složkám. K jedné, která obsahuje referenční pdb soubory, a druhé, která obsahuje pdb soubory z nástrojů získané. Výstup porovnání bude určen třetí zadanou cestou, tentokrát k jednomu konkrétnímu souboru, do kterého se výsledek bude ukládat.

Program přečte soubory ze vstupních cest a uloží si jejich seznam. Pak bude procházet jednotlivé referenční pdb soubory, a podle jména souboru, které je tvořeno identifikátorem proteinu z databáze pdb (tedy jméno souboru je ve formátu [PDB_ID].pdb), bude vybírat odpovídající výsledné pdb soubory nástrojů, které jsou ve formě [PDB_ID]_[jméno nástroje].pdb. Cesty k vybraným pdb souborům spolu s dalšími informacemi, jako např. identifikátor konkrétního řetězce v pdb souboru, je následně vložen do vstupního formuláře nástroje iPBA. Po nějaké době jsou staženy a dekódovány výsledky zpracování.

Výsledné hodnoty jsou ukládány po řádcích do výsledného souboru. Každá z následujících hodnot je oddělena středníkem, po poslední hodnotě je místo středníku znak konce řádku:

a) jméno nástroje (zjištěno ze jména vstupního souboru)
b) ID proteinu (zjištěno ze jména vstupního souboru, následující údaje jsou již z nástroje iPBA)
c) nscore – celkové skóre zarovnání
d) RMSD (Root Mean Square Deviation) – míra, která je dána průměrnou vzdáleností mezi atomy, nejčastěji těmi páteřními (v našem případě se tedy jedná o atomy Cα)
e) Alignment length – délka zarovnání
f) Number of aligned residues – počet zarovnaných reziduí
g) GDT_TS (Global Distance Test – Total Score) – jedná se o skóre využívané na soutěžích CASP, běžně se využívá při porovnávání výsledků predikce struktury proteinu vzhledem k experimentálně zjištěným strukturám. Toto skóre využiji při porovnávání nástrojů a to především z toho důvodu, že je pro tyto účely odzkoušené a používané.

7.3 Výsledky testování a výběr nástrojů

Hodnoty porovnaných souborů budu porovnávat podle soutěžemi CASP ověřeného skóre GDT_TS. Výsledné hodnoty z nástroje pro automatické testování výstupních pdb souborů (využívajícího nástroj iPBA) byly zpracovány programem Microsoft Excel tak, že byly z výsledného souboru překopírovány do Excelu. Následně byla data rozdělena podle oddělovače (středníku) do sloupců. Poté byly řádky seřazeny podle prvního sloupce, názvu nástroje. Nakonec byl vypočítán průměr hodnot GDT_TS každého z nástrojů, a hodnoty byly porovnány. Celý proces byl prováděn na každou kategorii proteinů zvlášť, abychom zjistili, jak moc je každý z nástrojů vhodný na kterou z kategorií.

Výsledné ohodnocení nástrojů v různých SCOP třídách můžeme vidět v Tab. 1. Celkově nejlépe vyšel z hodnocení nástroj Swiss Model, který se v téměř všech třídách umístil na prvním místě (ve třídě all alpha se umístil na druhém místě). Krom dobrého umístění je výhodou tohoto nástroje také rychlost zpracování. Jedná se však o nástroj využívající principu homologního modelování a testovací proteiny nalezl v databázi pdb, jeho úspěšnost pro nové proteiny je tedy pochybná. Nástroj tedy využiji pro rychlé prohledání databáze pdb a pokus o zpracování proteinu metodou homologního modelování.

Jako další nástroje se budu snažit vybírat ty, které pracují metodou threading (resp. kombinují více metod) a zároveň mají dobré skóre a přiměřenou dobu zpracování. Tyto nástroje by neměly být tolik ovlivněny tím, zda protein existuje v databázi pdb nebo nikoliv. Následuje seznam nástrojů a tříd, ve kterých budou využity:

a) Psi Pred - a/b, alpha and beta
b) Phyre 2 - a+b, coiled coil
c) Fugue - all alpha
d) Loopp - all beta

Zajímavostí je nástroj Raptor X v kategorii multidoménových proteinů (alpha and beta), kde na výstupu protein rozdělil na jednotlivé domény a pro každou doménu poskytl jeden pdb soubor. Nic takového žádný jiný nástroj neposkytl. Jeho využití však bude případným vylepšením do budoucna, protože by bylo složité zhodnotit správnost predikce. Navíc jeho webové rozhranní není pro účely automatické komunikace příliš přívětivé. Z důvodu pro komunikaci nevhodného rozhranní nebyl využit nástroj HH pred, i když např. v kategorii a+b má velice dobré výsledky.

a/b	
swissm	**98,5**
esypred3D	98,3
psipred	**97,5**
loopp	97,4
raptorx	97,0
hhpred	96,3
phyre2	95,7
3DJigsaw	94,9
cphmodels	94,1
fugue	82,1

a+b	
swissm	**97,1**
hhpred	96,8
phyre2	**96,0**
esypred3d	94,9
psipred	93,0
loopp	89,3
cphmodels	88,7
3Djigsaw	87,6
raptorx	87,1
fugue	78,7

all beta	
swissm	**96,7**
loopp	**93,7**
psipred	92,5
3djigsaw	92,5
hhpred	92,4
phyre2	92,4
raptorx	92,4
esypred3d	91,6
cphmodels	90,6
fugue	69,0

all alpha	
fugue	**97,4**
swissm	**96,2**
psipred	91,8
hhpred	90,6
3djigsaw	89,9
phyre2	88,1
raptorx	84,5
esypred3d	82,6
loopp	80,3
cphmodels	78,2

coiled coil	
swissm	**96,2**
3djigsaw	91,6
cphmodels	78,2
esypred3d	74,8
phyre2	**74,4**
loopp	66,4
hhpred	65,7
psipred	63,0
fugue	60,4
raptorx	???

Multi domain	
swissm	**87.4**
hhpred	81.8
phyre2	80.5
psipred	**80.5**
fugue	75.8
loopp	75.4
esypred3d	74.0
cphmodels	69.5
3djigsaw	51.8
raptorx	???

Tab. 1 Ohodnocení nástrojů v různých SCOP třídách

V Tab. 1 je zobrazeno průměrné skóre GDT_TS z přibližně deseti testovaných proteinů každým nástrojem v každé kategorii. Skóre GDT_TS je využívané při ohodnocování kvality predikce terciární struktury v různých soutěžích (např. CASP). Vidíme, že nástroj založený na principu homologního modelování Swiss model (v tabulce označen swissm) se umístil na horních místech ve všech kategoriích, bude jej tedy vhodné využít při predikci struktury každého typu proteinu. Vzhledem k použité metodě, při které hledá co nejpodobnější proteiny v databázi PDB, je však poměrně pravděpodobné, že nebude příliš přesný na sekvence, ke kterým žádný podobný protein nenalezne (resp. pokud najde, ale bude mít nízkou identitu). Swiss model, spolu s dalšími nástroji, které budou využity při implementaci serveru, je zvýrazněn tučným písmem. Ostatní vybrané (a tedy zvýrazněné) nástroje byly voleny nejen na základě hodnot skóre GDT_TS ale také tak, aby vyplňovaly nedostatek nástroje swiss model. To lze zajistit tak, že vybereme nástroje pracující na jiném principu než homologní modelování (např. pomocí metody threading, nebo kombinovali homologní modelování s jinými metodami).

Kolizi stejného skóre GDT_TS u multi-doménových proteinů rozhodla doba zpracování, psipred je dle skóre stejně kvalitní, avšak jeho zpracování je rychlejší. Nástroj phyre2 však z důvodu několika

výpadků nástroje Psi Pred byl vybrán jako náhradní. Stejně tak bude nástrojem Phyre 2 predikována struktura pokud některý z nástrojů selhal.

8 Implementace serveru

Server byl implementován v programu Eclipse, s využitím serveru Apache Tomcat 7.0.33 a byl vytvořen jako Dynamic Web Project. Server pracuje na principu třívrstvé architektury. Nejnižší vrstvu pro uchování dat a přímou komunikaci s nástroji tvoří balíček *processing*. Prostřední vrstvu pro zpracování a parsování dat a kontrolu vstupního formuláře tvoří balíček *preparing*. Nejvyšší vrstvu určenou pro prezentaci dat tvoří především hlavní stránka *mainForm.jsp* a stránka pro zobrazení výsledků.

8.1 Nejnižší vrstva

Je tvořena dvěma třídami. V první jsou uloženy formuláře všech využitých nástrojů (ve formátu MultipartEntity) a adresy jednotlivých nástrojů (jedná se o adresy, kam se odesílají formuláře). Druhá třída slouží především pro přímou komunikaci s nástroji, je v ní implementován http klient. K jeho implementaci jsem využil knihovny HttpClient verze 4.2.3[2] (org.apache).

8.2 Střední vrstva

Tato vrstva je tvořena třemi částmi. První část je tvořena třídou *FormProcessing*, která je po zadání správných hodnot volána z hlavního formuláře a řídí celé následné zpracování proteinu. Další součást této vrstvy tvoří dva balíčky. Jeden obsahuje třídy řídící zpracování jednotlivých nástrojů (stahování různých podstránek a zpracování údajů z nich), a druhý obsahuje třídy, které umí zjistit různé požadované údaje z jednotlivých webových stránek nástrojů. Krom dalších informací jsou výstupem této vrstvy pdb soubory z jednotlivých nástrojů. Tyto pdb soubory budou ukládány přímo ve složce s webovým obsahem projektu (složka *WebContent*), konkrétně v podsložce *Jmol/outputs/[jedinečné číslo predikce]* z důvodu správné funkčnosti vizualizačního appletu Jmol.

8.3 Prezentační vrstva

Jedná se o vrstvu pro interakci s uživatelem. Obsahuje stránku s hlavním formulářem, přes který je možno zahájit predikci struktury, a stránky pro zobrazení výsledků. Pro lepší prezentaci struktury je využit nástroj Jmol. Tomuto nástroji je možno předat pdb soubor, a on následně zobrazí vizualizaci terciární struktury. Jmol je vložen na stránku s výsledky formou appletu. Pro zobrazení vizualizace z pdb souboru je potřeba, aby byl pdb soubor ve stejné složce jako třída tento nástroj implementující, je s tím třeba počítat při ukládání pdb souborů z nástrojů (podsložky jsou také dovoleny).

[2] Nejnovější verzi knihovny lze stáhnout na http://hc.apache.org/downloads.cgi

8.4 Postup zpracování

Hlavní stránka, která umožní spuštění celého procesu, je v prezentační vrstvě, v souboru *mainForm.jsp*. Tato stránka obsahuje formulář, kam uživatel vloží základní údaje o predikci, především svůj email, krátký popis predikce a vloží sekvenci aminokyselin buď jako soubor, nebo přímo jako sekvenci aminokyselin do textového pole. Vyplněný formulář je pak odeslán opět na stránku *mainForm.jsp*, která zajistí kontrolu správnosti údajů.

Kontrola správnosti údajů spočívá v ověření emailové adresy, kontrole délky popisu a jeho případném zkrácení a kontrola vložené sekvence proteinu. Pokud je sekvence ve FASTA formátu, je hlavička odříznuta. Stejně tak jsou odříznuty případné další řetězce nacházející se v témže souboru. Následně je zkontrolováno, že sekvence neobsahuje žádné jiné znaky než ty, které označují aminokyseliny. Když jsou všechny údaje v pořádku, může se pokračovat ve zpracování. Nejdříve se vygeneruje jednoznačné identifikační číslo procesu predikce a je vytvořena složka se jménem stejným jako číslo tohoto procesu. Složka *outputs*, která všechny výstupní podsložky obsahuje, se kvůli správnému fungování nástroje Jmol nachází uvnitř projektu ve složce *WebContext* a podsložce *Jmol*. Protože se cesta k projektu téměř vůbec nemění (pouze při přenesení projektu), zadává se úplná cesta k projektu do proměnné *projectHome* nacházející se na hlavní stránce (*mainForm.jsp*). Při zadávání je třeba si dávat pozor na oddělovače, které se liší v různých systémech.

Když je vygenerováno jednoznačné číslo zpracování a je vytvořena složka pro výstupy, pokračuje zpracování ve třídě *FormProcessing*, která rozšiřuje třídu *Thread*. Vytvoří se tak nové vlákno, které se stará o požadavek, a klient může s aktuálním oknem dále pracovat. Třídě *FormProcessing* jsou předány všechny základní údaje jako vložený email, popis a sekvence, fyzická cesta k domovské složce projektu (ta je potřebná pro pozdější ukládání výsledných souborů), vygenerované číslo úlohy (job id) a URI adresa serveru. Tyto údaje jsou uloženy a dále předávány pomocí třídy *InputData*.

Poté vlákno předá údaje do další vrstvy modelu, konkrétně třídě *processSwissModel*, která zajistí zpracování sekvence proteinu pomocí nástroje Swiss model a uloží některé potřebné údaje (o postupu zjišťování údajů z nástrojů se detailněji zmíním později). Hlavním výstupem z tohoto nástroje je identifikátor použitého templátu v databázi pdb (pdb id), procento podobnosti zkoumané sekvence s templátem, chybová hodnota, a odkaz na soubor pdb. Pro budoucí využití jsou také uloženy odkazy na detaily o templátu v některých dalších nástrojích nebo databázích.

Když jsou zpracovány potřebné údaje z nástroje Swiss model, zjistím třídu sekundární struktury podle členění, které používá databáze SCOP. K tomu použiji nástroj SuperFamily, který umí predikovat SCOP rodinu, jejíž částí je hledaná třída sekundární struktury. To SuperFamily provádí buď přímo vyhledáním v databázi SCOP pomocí PDB id, nebo použitím skrytých markovových modelů (HMM) na zadanou sekvenci aminokyselin. Pokud je identita s templátem po zpracování nástrojem Swiss model dostatečně velká a chybová hodnota dostatečně malá, mohu použít zjištěné PDB id. Pokud podobnost s templátem není dostatečná nebo nic v databázi SCOP nebylo nalezeno, predikuji rodinu pomocí HMM ze sekvence.

Nástroj SuperFamily mi však nevrátí pouze jednu třídu, ale třídu pro každou doménu, která byla nalezena. Výsledná třída, podle které bude použit jeden z nástrojů, bude zjištěna z množiny tříd prioritním způsobem, jak vyplývá z definice konkrétních tříd. Pokud je v množině zjištěných tříd třída coiled coil, bude také výsledná, protože je poměrně hodně specifická. Třída multi-doménových proteinů bude výsledkem, pokud je nalezeno více domén a alespoň dvě domény patří do různých tříd. Ostatní třídy all alpha, all beta, a+b a a/b nastanou v situacích, kdy buď byla nalezena jedna doména daného typu, nebo více domén, avšak stejného typu.

Když víme SCOP třídu proteinu, můžeme dle výsledků předchozího ručního testování nástrojů vybrat odpovídající nástroj a v něm zpracovat vstupní údaje podobně jako v nástroji Swiss model. Výjimkou, kdy vybereme nástroj podle jiných kritérií, jsou situace jako příliš dlouhá sekvence, nebo jiná SCOP třída proteinu než se kterými počítáme. Až máme zpracovány i údaje z druhého nástroje, vytvoříme výslednou stránku s některými výstupními hodnotami nástrojů a především s jejich výslednými modely. Modely jsou na výsledné stránce vykresleny pomocí nástroje Jmol.

Stránka s výsledky je tvořena pomocí třídy *OutputSummary* a jsou do ní průběžně ukládány výsledky zpracování. Po získání některých důležitějších údajů třída odešle data metodou GET stránce *outputSummary.jsp*, která vygeneruje HTML stránku s výstupními údaji. Třída *OutputSummary* tuto HTML stránku stáhne, a uloží do odpovídající výstupní složky.

Zpracování každého z nástrojů je zajištěno odpovídající třídou v balíčku *preparing.process*, a to velice podobným způsobem (vyjma nástroje LOOPP, který v určité situaci pracuje jiným způsobem, o kterém se zmíním později). Nejdříve je pomocí metod třídy *ToolForms* vygenerován formulář, který je odeslán na adresu uloženou v téže třídě. Tím je zadáno zpracování proteinu odpovídajícímu nástroji. Následně je navázána spolupráce s balíčkem *preparing.parse*, který obsahuje třídy schopné zjistit některé údaje z nástrojů. Jsou tak postupně zjišťovány adresy přesměrování na stránky s výsledky, adresy na pdb soubory a další zajímavé údaje. Zjišťování dat ze stránky je provedeno hledáním významných sousloví na stránce, podle kterých je možno se zorientovat, a následným uložením dané části kódu html. Snažil jsem se vše provést tak, aby drobné změny v html kódu měly co nejmenší efekt na zpracování.

Trochu složitější situace je se zpracováním nástroje LOOPP, a to v případě, že sekvence nebyla v blízké době zpracovávána. V této situaci nelze zjistit odkaz na stránku s výsledky, adresa na výsledky je odeslána na zadanou emailovou adresu. V této situaci je výstupní stránku vložen formulář, do kterého uživatel vloží celý obsah emailu, ze kterého si z důvodu přívětivosti pro uživatele server sám zjistí adresu na výsledky. O zpracování tohoto formuláře a jeho přepsání na stránce s výsledky se stará stránka *loopContinue.jsp* a s ní spolupracující třída *processLoopp*.

9 Testování serveru

V této kapitole bude popsán způsob a průběh testování implementovaného serveru a zhodnocení výsledků z tohoto testování.

9.1 Výběr testovacích dat

Testovací sada proteinů byla vybrána obdobným způsobem jako při předchozím ručním testování. Byly vybrány proteiny získané experimentální metodou, rentgenovou krystalografií, s přesností 1.5 až 3Å (o něco větší rozptyl než při ručním, aby byl větší výběr vzorků). Opět jsem vybral tyto SCOP třídy:

a) A/B - hlavně paralelní beta-listy
b) A+B - hlavně anti-paralelní beta-listy
c) all alpha - domény skládající se pouze z alfa-šroubovic
d) all beta - domény skládající se pouze z beta-listů
e) coiled coil - obsahuje více alfa-šroubovic stočených dohromady
f) alpha and beta - obsahuje 2 a více domén, které patří do více tříd

Každý protein jsem zařadil do složky, která odpovídala kategorii. To mi umožní krom použitých nástrojů pro predikci struktury také kontrolu nástroje predikujícího SCOP třídu.

9.2 Způsob testování

Hlavní část testování bude opět automatická komunikace s nástrojem iPBA, který dokáže porovnat dva pdb soubory. Po dokončení zpracování nástroje Swiss model porovná výsledný pdb soubor s referenčním souborem. Následující fáze zjištění typu sekundární struktury, tedy SCOP třídy, je ověřena porovnáním zjištěné struktury se strukturou referenční, kterou zjistíme ze jména složky, ve které je referenční protein umístěn. Po zpracování dalším nástrojem je jeho výsledný pdb soubor opět porovnán pomocí nástroje iPBA s tím referenčním.

Pro komunikaci s nástrojem iPBA jsem použil předchozí implementaci, kterou jsem zabudoval do serveru. Pro zjednodušení jsem však udělal několik úprav, aby bylo porovnávání výsledků nástrojů plně automatické, ale úprav na serveru bylo potřeba co nejméně. Každý porovnávaný protein jsem zadával ze souboru, což byl FASTA soubor stažený z rcsb pdb, pojmenovaný dle pdb id, a koncovkou fasta. Ze jména souboru si server zjistil id zkoumaného proteinu, a to použil jako popis predikce (label). Vstupem třídy pro zpracování nástroje iPBA a výsledků z něj (*PDBCompare*) byly pak pdb soubory zjištěné z nástrojů, jejich jména se nemění (mění se složka, ve které jsou umístěny, a ta má stejné jméno jako job id), a referenční pdb soubor, který byl vyhledán dle nastaveného labelu (jména složek, které prohledávat, byly pro zjednodušení předem nastaveny v poli řetězců).

Výsledky každého z porovnání jsou ukládány do jednoho souboru. Výstupem jsou tedy tři soubory, v jednom jsou výsledky porovnávání nástroje Swiss model, v jednom výsledky porovnání SCOP třídy, a ve třetím jsou výsledky porovnání dalšího použitého nástroje. Cesty a jména souborů, do kterých se mají informace ukládat, jsou opět nastaveny v proměnných typu řetězec.

Pro spojení tabulek a jejich základní zpracování jsem použil nejdříve program RapidMiner 5, jehož vstupem byly csv soubory, a výstupem tabulka xls, která byla dále zpracovávána programem Microsoft Excel.

9.3 Výsledky testování

Výstupními hodnotami testování jsou především informace o podobnosti modelů ve formě skóre GDT_TS a informace o predikci SCOP třídy proteinu nástrojem Superfammily. Tabulky se všemi výstupními hodnotami budou obsaženy v příloze 3, v této podkapitole bude souhrn a zhodnocení těchto výstupních hodnot.

Nástroj využívající homologního modelování Swiss model, ačkoli ve většině případů udával identitu 100 a chybovou hodnotu nulovou, nebo blízkou nuly, měl celkové průměrné skóre GDT_TS přibližně 90, což se dá hodnotit jako vysoká přesnost. Nejnižší vyskytující se skóre bylo 40.3, jinak se ty nižší skóre pohybovaly kolem 70 – 80. Rychlost zpracování byla vysoká a jeho použití ve všech SCOP třídách pro prvotní zpracování se tak zdá být výhodné.

Nejméně využitým nástrojem je nástroj fugue, který při předchozím ručním testování podal dobré výsledky u proteinů třídy all alpha. Zpracování sice bylo rychlé, průměrné skóre však má hodnotu přibližně 65, což není příliš uspokojivé. Nejnižší skóre bylo 7.5, několik se jich pohybovalo kolem 50, a několik i kolem hodnoty 95.

Dalším použitým nástrojem, který byl při ručním testování velice dobrý pro třídu proteinu all beta, je nástroj LOOPP. Skóre sice má uspokojivé, pohybuje se kolem hodnoty 90, je však velice pomalý na zpracování. Čistě zpracování ne zřídka trvá i deset hodin, nějakou dobu se také čeká ve frontě. Výsledky, které přicházejí pouze e-mailem, by při menším množství proteinů zpracovávaným tímto nástrojem a troše pozornosti uživatele, neměly být problém.

Dvěma nejlepšími nástroji byly PsiPred a Phyre2, jejichž skóre se pohybovalo kolem hodnoty 90. Doba zpracování nástrojem PsiPred se pohybovala od půl hodiny do několika hodin (i 12ti), a to spíše v závislosti na vytížení serveru. Doba zpracování nástrojem phyre2 závisí ve větší míře na délce sekvence. Z části samozřejmě doba zpracování závisí na počtu běžících predikcí, kterých zde běží obvykle mnoho zároveň. Pro sekvence délky větší než tisíc aminokyselin se ukázal nepoužitelný nástroj PsiPred, protože takto dlouhé sekvence odmítá zpracovat. Dříve jsem o tomto chování nenalezl nikde zmínku.

Výsledkem hodnocení serveru je tedy to, že nástroje Phyre2, PsiPred a Swiss model jsou jeho vhodnou součástí. Nástroje fugue z důvodu nižší přesnosti a LOOPP z důvodu pomalejšího zpracování by bylo vhodné využívat obezřetněji.

Přesnost nástroje Superfammily pro zjišťování SCOP třídy proteinu byla vysoká. Třída asi tří čtvrtin zkoumaných proteinů byla získána přímo pomocí identifikátoru templátu použitému nástrojem Swiss model, a téměř všechny odpovídaly té referenční. Při odhadu třídy zbylých proteinů ze sekvence byla úspěšnost přes 75%, což je také vyhovující.

Nástroj iPBA, který byl využit pro porovnávání dvou pdb souborů a je využit pro zobrazení modelů dvou nástrojů v jednom výstupu, se ukázal být čas od času problémovým. I když totiž nástroj Jmol oba pdb soubory bez problémů zpracoval, stávalo se, že vrátil chybu týkající se správnosti vloženého pdb souboru. Chyba však byla příliš obecná na to, aby se dala zjistit přesná příčina tohoto chování.

Kuriozitou se stala třída zahrnující v sobě nejmenší množství proteinů, coiled coil. I nástroj phyre2, který byl jinak poměrně přesný, v této třídě nepodával příliš dobré výsledky. Potvrdily se tedy celkově nižší výsledky všech nástrojů v této třídě, které byly zjištěny při ručním testování. Průměrné skóre nástroje phyre2 se v této kategorii pohybovalo kolem hodnoty 60.

10 Závěr

Prostudoval jsem několik publikací týkajících se jak proteinů obecně, např. Základy buněčné biologie: úvod do molekulární biologie buňky od pana Bruce Albertse, tak druhů zarovnání proteinů a způsobů jejich predikce, např. Structural bioinformatics od autorů Jenny Gu a Philip Eric Bourne a Understanding bioinformatics od autorů Marketa Zvelebil a Jeremy Baum. Z těchto publikací jsem pochopil k čemu jsou nám dobré proteiny, proč potřebujeme predikovat terciární strukturu a princip základních metod používaných pro její predikci. Dále se mi podařilo vyhledat a detailněji prozkoumat různé nástroje pro predikci terciární struktury a několik nástrojů pro její analýzu, verifikaci a vizualizaci.

Na základě získaných znalostí jsem navrhl server pro predikci terciární struktury, který využívá některé z prozkoumaných nástrojů. Pro jejich vhodnější výběr jsem provedl testování, jehož výsledkem bylo zjištění přesnosti predikce většiny zkoumaných nástrojů pro různé třídy proteinu. Tato třída byla zjišťována na základě rozdělení proteinů do rodin využitých v databázi SCOP. Nejlépe z testování vyšel nástroj využívající homologní modelování Swiss Model, který však dle očekávání většinu testovacích proteinů nalezl v databázi PDB. Velice dobře dopadlo také testování některých nástrojů využívající metody threading nebo kombinace různých metod. Protože byl takový nástroj často vhodný pouze pro některou třídu proteinů, ale na jinou třídu byla přesnost nižší, daly se zkoumané nástroje zkombinovat tak, že je nástroj využit pouze na typ proteinu, pro který je jeho predikce nejpřesnější. Server tedy nejdříve s pomocí nástroje Swiss model a nástroje Superfammily, schopného pomocí skrytých markovových modelů predikovat rodinu proteinu, předpoví třídu proteinu. Následně podle předpovězené třídy použije některý z nástrojů.

Následně jsem navržený server implementoval v jazyce JAVA, s využitím technologie Java Server Pages. Implementace však nebyla vůbec jednoduchá, protože většina nástrojů s automatickou komunikací vůbec nepočítá, a je tak složité zjistit jaké informace přesně odesílat ve formulářích a jak z přijatých dat vhodně zjistit potřebné údaje. Navzdory všem problémům je server funkční a použitelný pro predikci struktury. Vzhledem k prvotnímu ručnímu testování, díky němuž byly vybrány pouze spolehlivější nástroje, a jejich vhodnému zkombinování pevně věřím, že server poskytuje dostatečně kvalitní modely. Nelze však říci, že by byla predikce stoprocentně spolehlivá pro všechny proteiny, protože dnešní techniky predikce struktury nejsou zatím dostatečně dokonalé.

To se potvrdilo i při následném testování serveru. Celková přesnost se pohybovala kolem osmdesáti bodů ze sta. Dobře vyšel z hodnocení co se týče přesnosti nástroj LOOPP, který měl skóre přes devadesát, jeho rychlost zpracování však byla velice nízká. Nástroj fugue měl pro změnu nižší přesnost predikce. Ostatní z použitých nástrojů, tedy Swiss model, PsiPred a Phyre2, měly skóre kolem devadesáti, jejich využití v implementovaném serveru bylo tedy velice vhodné.

Výsledkem práce je server, který může sloužit proteinovým inženýrům pro zjišťování informací zkombinovaných z různých serverů tak, aby nemuseli zadávat dotazy postupně na více míst. Dále byly doporučeny některé nástroje, které se osvědčily pro různé třídy proteinů. V budoucnu by bylo vhodné testování rozšířit na více tříd a více nástrojů. Kvalitní nástroje pro různé třídy proteinů by se

pak mohly stát dalšími součástmi serveru. Server by bylo také možné rozšířit tak, aby poskytoval co nejkomplexnější informace o struktuře, účelu a principu fungování proteinu. Některými možnými rozšířeními, jako je predikce sekundární struktury, predikce vlivu aminokyselinových mutací na sekundární strukturu, nebo predikce proteinových domén a kontaktů, se zabývaly letošní diplomové práce a bylo by vhodné je zkombinovat s tímto serverem.

Zdroje

[1]. **Albets, Bruce.** *Základy buněčné biologie : úvod do molekulární biologie buňky.* Ústí nad Labem : Espero Publishing, 2001.

[2]. **Zvelebil, Marketa a Baum, Jeremy.** *Understanding Bioinformatics.* místo neznámé : Garland Science, 2008.

[3]. **Gu, Jenny a Bourne, Philip Eric.** *Structural Bioinformatics.* New Jersey : Jon Wiley & sons, Inc., 2009.

[4]. **Berman, H M, a další.** Yearly Growth of Protein Structures. *RCSB - Protein Data Bank.* [Online] http://www.rcsb.org/pdb/statistics/contentGrowthChart.do?content=molType-protein&seqid=100, 1. Leden 2013. [Citace: 4. Leden 2013.]

[5]. **Murzin, A G, a další.** Scop Classification Statistics. *Structural Classification of Proteins.* [Online] 1.75, Únor 2009. http://scop.mrc-lmb.cam.ac.uk/scop/count.html#scop-1.75.

[6]. **Artimo, P, a další.** *ExPASy - Bioinformatics Resource Portal.* [Online] 27. Září 2012. http://us.expasy.org/tools/#primary.

[7]. **Kropinski, Andrew.** Protein_tertiary_structure. *ONLINE ANALYSIS TOOLS.* [Online] Říjen 2012. http://www.molbiol-tools.ca/Protein_tertiary_structure.htm.

[8]. **wikipedia, en.** JavaServer Pages. [Online] 14. 12 2012. [Citace: 9. 1 2013.] http://en.wikipedia.org/wiki/Java_server_pages.

[9]. **LadyofHats.** Wikipedia, the free encyclopedia. *Main protein structures levels.* [Online] 7. Říjen 2008. http://en.wikipedia.org/w/index.php?title=File:Main_protein_structure_levels_en.svg&page=1.

[10]. **Chaplin, Martin.** Protein Folding and Denaturation. *Water Structure and Science.* [Online] Prosinec 2012. http://www.lsbu.ac.uk/water/protein2.html.

[11]. **Swiss model.** fully automated protein structure homology-modeling server. [Online] Swiss Institute of bioinformatics. [Citace: 14. květen 2013.] http://swissmodel.expasy.org/workspace/index.php?func=modelling_simple1.

[12]. **Fugue.** Sequence-structure homology recognition. [Online] 2010. [Citace: 14. Květen 2013.] http://tardis.nibio.go.jp/fugue/prfsearch.html.

[13]. **PsiPred.** Protein Sequence Analysis Workbench. [Online] The Bloomsbury Centre for Bioinformatics. [Citace: 14. Květen 2013.] http://bioinf.cs.ucl.ac.uk/psipred/.

[14]. **LOOPP.** Learning, Observing and Outputting Protein Patterns. [Online] Center for Computational Life Science and Biology, The University of Texas at Austin. http://clsb.ices.utexas.edu/loopp/web/.

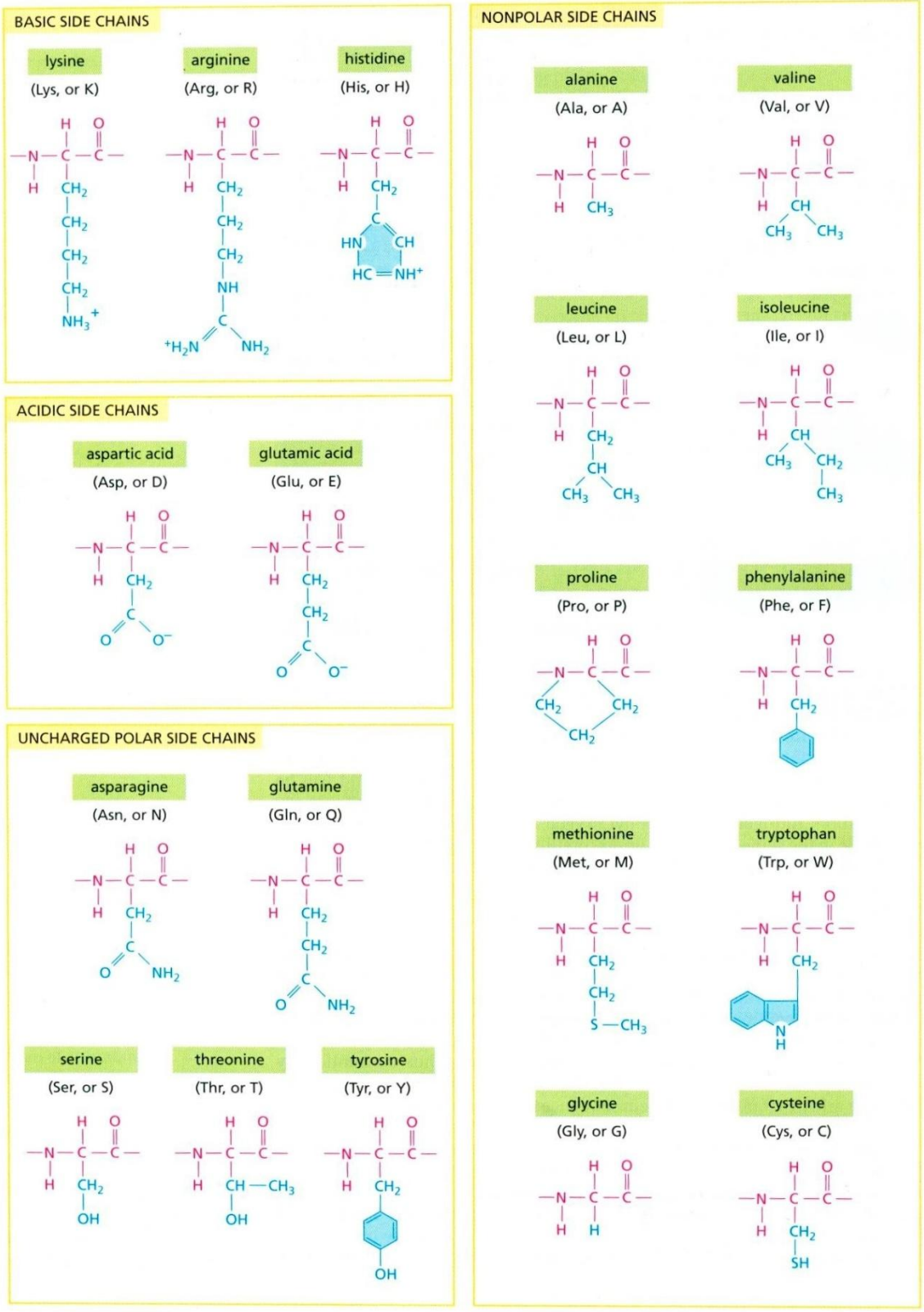
[15]. **Phyre2.** Protein Homology/analogY Recognition Engine V 2.0. [Online] Structural Bioinformatics Group, Imperial College, London. [Citace: 14. květen 2013.] http://www.sbg.bio.ic.ac.uk/phyre2/html/page.cgi?id=index.

[16]. **iPBA.** A powerful method for structural alignment based on a structural alphabet. [Online] INSERM, Universite Paris Diderot. [Citace: 14. Květen 2013.] http://www.dsimb.inserm.fr/dsimb_tools/ipba/index.php#.

[17]. **Superfammily.** Hmm library and genome assignments server. [Online] 2013. [Citace: 14. Květen 2013.] http://supfam.cs.bris.ac.uk/SUPERFAMILY/.

Pozn.: Informace o nástrojích získány přímo na webových stránkách jednotlivých nástrojů. Odkazy na jejich domovské stránky jsou uvedeny přímo v textu.

Příloha 1 – Aminokyseliny včetně postranních řetězců a reprezentujících symbolů[2]



Příloha 2 – PDB ID proteinů použitých pro první, ruční, testování

V této příloze je seznam proteinů použitých v první fázi testování nástrojů, rozčleněný do jednotlivých SCOP tříd:

a) **A+B**
   2EZQ, 3BH8, 3E9V, 3ECF, 3ETH
b) **A/B**
   2W6K, 3DYB, 2QX8, 3BSG, 2V7X
c) **all alpha**
   3CR2, 2ZG7, 3CSI, 2VVN, 2GZJ
d) **all beta**
   3DD8, 3DJ9, 2V7V, 3B6S, 2ZQN
e) **coiled coil**
   2HV8, 2GGX, 2E42, 2Q5U
f) **alpha and beta**
   2JEJ, 2PU2, 2PY5, 2ZD1, 3DLK

Příloha 3 – Výsledky testování serveru

První tabulka: Přesnost predikce druhého použitého nástroje (především skóre GDT_TS) spojená s výsledky predikce SCOP třídy proteinu (v tabulce predikovaná a referenční třída, a typ jejího zjištění – dle PDB ID nebo ze sekvence). Vše je doplněno jménem použitého nástroje a počtem animokyselin (sloupec amino). Sloupec průměr udává průměrnou hodnotu GDT_TS z každé třídy, dole pak celkové skóre implementovaného serveru.

pdb id	jmeno	predikovaná třída	referenční třída	zjištění	amino	GDT_TS	průměr
1N8F	PsiPred	a_lomeno_b	a_lomeno_b	id	350	95.9	
1FJ2	PsiPred	a_lomeno_b	a_lomeno_b	id	232	98.7	
1OAB	PsiPred	a_lomeno_b	a_lomeno_b	id	370	78.9	
2BI5	PsiPred	a_lomeno_b	a_lomeno_b	id	360	99.5	
1KIM	PsiPred	a_lomeno_b	a_lomeno_b	sekv	366	80.1	
1ZAH	PsiPred	a_lomeno_b	a_lomeno_b	id	363	86.0	
3D4N	PsiPred	a_lomeno_b	a_lomeno_b	id	286	91.7	
2ZB2	PsiPred	a_lomeno_b	a_lomeno_b	Id	849	85.34	89.5
1YRX	Phyre2	a+b	a+b	id	121	79.4	
3C8W	Phyre2	a+b	a+b	id	255	87.5	
1J3W	Phyre2	a+b	a+b	id	163	82.1	
3D3H	Phyre2	a+b	a+b	sekv	200	72.0	
3D03	Phyre2	a+b	a+b	id	576	89.8	
2V4J	Phyre2	a+b	a+b	id	437	95.0	
2HZ5	Phyre2	a+b	a+b	id	106	66.1	
3E88	Phyre2	a+b	a+b	id	335	85.9	
2RA5	PsiPred	alpha_and_beta	a+b	sekv	247	37.4	
3DS6	Phyre2	a+b	a+b	id	366	76.0	77.1
1HCI	Fugue	all_alpha	all_alpha	id	476	100.0	
2HR2	Fugue	all_alpha	all_alpha	id	159	53.2	
2VUE	Fugue	all_alpha	all_alpha	id	585	7.5	
1QSJ	Fugue	all_alpha	all_alpha	id	278	66.2	
1ZO9	Fugue	all_alpha	all_alpha	id	473	94.7	
1H4I	PsiPred	alpha_and_beta	all_alpha	id	554	94.7	
1KKQ	Fugue	All alpha proteins	all_alpha	id	269	45.5	
3DY6	Fugue	All alpha proteins	all_alpha	sekv	271	49.8	
3EJN	Fugue	All alpha proteins	all_alpha	id	474	64.5	
3DPY	Fugue	All alpha proteins	all_alpha	sekv	377	90.4	66.6
1FUJ	LOOPP	All beta proteins	all_beta	id	221	99.6	
1JCC	Phyre2	coiled-coil proteins	all_beta	sekv	56	61.9	
2EDM	LOOPP	All beta proteins	all_beta	id	161	95.8	
2VDL	LOOPP	All beta proteins	all_beta	sekv	452	98.0	
3EHW	-	All beta proteins	all_beta	sekv	164	-	

2IH8	-	All beta proteins	all_beta	sekv	559	-	
2IX0	-	All beta proteins	all_beta	id	663	-	
2PHF	LOOPP	All beta proteins	all_beta	id	252	99.3	
2QYI	-	All beta proteins	all_beta	sekv	223	-	
3BU7	-	All beta proteins	all_beta	id	394	-	90.9
1EI5	PsiPred	alpha_and_beta	alpha_and_beta	id	520	99.3	
1F06	PsiPred	alpha_and_beta	alpha_and_beta	id	320	98.6	
1JXL	PsiPred	alpha_and_beta	alpha_and_beta	id	352	85.5	
1ECF	PsiPred	alpha_and_beta	alpha_and_beta	id	504	96.8	
1K1Y	PsiPred	alpha_and_beta	alpha_and_beta	id	659	93.8	
1CVU	Phyre2	alpha_and_beta	alpha_and_beta	sekv	552	99.3	
2P0M	Phyre2	alpha_and_beta	alpha_and_beta	id	662	88.0	
1KRC	Phyre2	a+b	alpha_and_beta	sekv	100	97.2	
1JVA	PsiPred	alpha_and_beta	alpha_and_beta	id	475	84.6	
1CKN	PsiPred	alpha_and_beta	alpha_and_beta	id	330	95.3	93.8
1RF1	Phyre2	coiled_coil	coiled_coil	sekv	66	73.9	
1J1D	Fugue	all_alpha	coiled_coil	sekv	161	5.2	
1QU1	Phyre2	coiled_coil	coiled_coil	id	155	9.6	
2OYH	Phyre2	coiled_coil	coiled_coil	sekv	66	76.3	
1EPW	Phyre2	a+b	coiled_coil	id	1290	91.8	
2FYZ	Phyre2	coiled-coil proteins	coiled_coil	sekv	63	43.1	
1EBO	Phyre2	coiled-coil proteins	coiled_coil	id	131	41.8	
2VSG	Phyre2	coiled-coil proteins	coiled_coil	id	358	88.5	
-	-	-	-	-	-	-	53.8
				Celkový průměr:			78.6

Tabulka druhá: Predikované hodnoty nástroje Swiss Model“ a srovnání s referenčním souborem

PDB ID	GDT_TS	Identita	Evalue
1B0G	93.26	100	5.2E-150
1CKN	99.63	100	0
1CVU	98.87	100	0
1EBO	85.44	100	8.85E-57
1ECF	99.95	100	0
1EI5	100	100	0
1EPW	25.63	100	0
1F06	63.03	100	0
1FJ2	98.36	100	6.3E-131
1FUJ	100	100	3.7E-129
1GJQ	95.32	100	0
1GWF	99.95	100	0
1H4I	93.7	99.83	0
1HCI	100	100	0
1J1D	59.7	100	1.33E-61
1J3W	93.4	100	1.9E-59
1JCC	93.57	100	7.62E-05
1JVA	76.19	98.23	0
1JXL	96.69	100	4.7E-173
1K1Y	94.81	96.51	0
1K90	68.11	100	0
1KIM	88.08	97.26	4.5E-177
1KKQ	76.47	100	1.4E-152
1KRC	100	100	6.39E-46
1N7K	100	100	6.9E-114
1N8F	99.76	100	0
1NO7	96.42	96.44	0
1OAB	97.93	100	0
1QSJ	97.67	100	2.5E-153
1QU1	70.02	100	4.19E-74
1RF1	89.67	100	4.97E-17
1XG5	96.86	100	1.1E-148
1YRX	100	100	1.03E-60
1ZAH	92.07	100	0
1ZO9	99.11	100	0
2EDM	100	100	1.74E-68
2EFY	97.3	100	2.48E-156
2EQB	100	100	1.44E-89
2FYZ	86.15	100	1.78E-16

2HR2	98.63	100	4.23E-77
2HZ5	64.54	100	6.72E-46
2IH8	100	100	0
2IX0	77.12	99.84	0
2OYH	90.5	100	4.97E-17
2P0M	100	100	0
2PHF	99.38	100	3.5E-120
2QYI	98.22	100	6.8E-110
2RA5	68.76	43.24	2.4E-24
2V4J	100	100	0
2VDL	99.11	100	0
2VSG	90.51	100	0
2VUE	40.29	100	0
2YSU	78.09	100	0
2ZB2	78.68	100	0.0
2ZLB	81.13	100	2.32E-121
3BU7	100	100	0
3C8W	99.86	99.6	3.4E-139
3CMM	77.81	99.21	0
3D03	100	100	2.7E-165
3D3H	93.22	99.45	3.49E-90
3D4N	94.19	100	3.8E-147
3DPY	99.31	100	1.3E-170
3DS6	73.35	100	0
3DY6	83.95	100	3E-153
3E88	91.77	100	8.06E-179
3EHW	85.47	100	1.55E-66
3EJN	98.01	98.04	0
Průměr:	89.32866	98.92015	1.25E-06

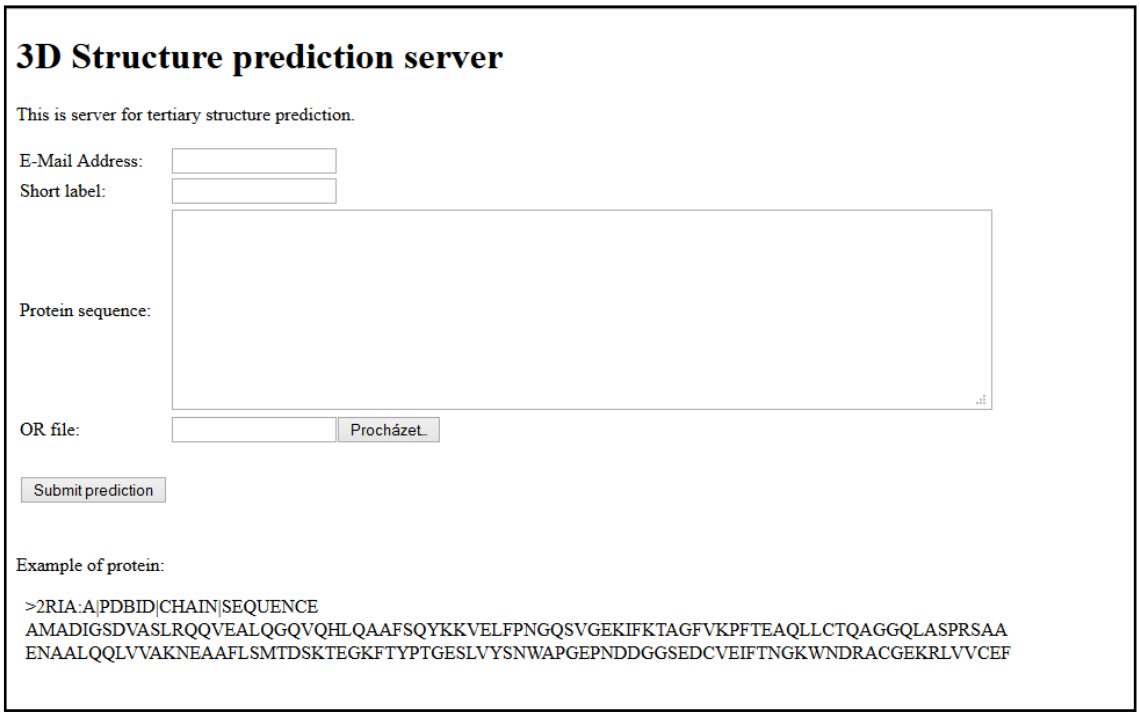
Příloha 4 – Instalace a spuštění serveru

Projekt byl implementován ve Windows v prostředí Eclipse, nejjednodušší jej tedy bude zprovoznit zde. Výhodou tohoto prostředí je, že umí sámo nastavit spuštění projektu na aplikačním serveru, např. na mnou použitém serveru Apache Tomcat v7.0. Verze javy, která byla použita při implementaci, je JavaSE-1.7.

Pro spuštění je tedy vhodné mít nainstalovaný Eclipse, Apache Tomcat v7.0 a správnou verzi Javy. Dále je potřeba vložit na aplikační server dodatečné knihovny využité v projektu. Tyto knihovny lze nalézt v projektu v adresáři lib, a je potřeba je nakopírovat do adresáře lib, který se nachází na místě, kam byl aplikační server nainstalován. Po spuštění aplikačního serveru už stačí jen otevřít v prohlížeči hlavní stránku projektu, *predictionServer/mainForm.jsp*, který se implicitně nachází na adrese *localhost*, na portu 8080. Celá adresa, pokud není na serveru nastavena jinak, je tedy takováto:

http://localhost:8080/predictionServer/mainForm.jsp

Nyní bychom měli vidět hlavní stránku serveru, která obsahuje vstupní formulář a příklad zadání sekvence proteinu, můžeme ji vidět na Obr. 12.



Obr. 12 Hlavní stránka implementovaného serveru

Pro další správnou funkčnost je třeba nastavit ručně cestu k domovskému adresáři projektu (cesta kam jsme projekt umístili), především z důvodu správného ukládání výsledných souborů. Ta se nastavuje v hlavním vstupním souboru, *mainForm.jsp*. Jedná se o proměnnou *projectHome* typu *String*. Buď je možno vložit přímo cestu do uvozovek, nebo pokud nevíme, jaké oddělovače přesně

systém používá, použijeme místo něj proměnnou *separator*, ve které je správný separátor již uložen. Vložená cesta pak může vypadat např. takto:

projectHome = "G:"+separator+"projekty"+separator+"implementaceServeru"

Ve složce implementaceServeru by se tedy měly nacházet základní složky projektu jako src, lib nebo WebContext. Doporučením, které není potřebné pro instalaci a spuštění, ale zjednoduší další práci se serverem, je nastavení automatická aktualizace workspace. Když v eclipse zobrazíme Window – Preferences – Workspace, tak by mělo být zaškrtnuto (krom implicitních hodnot jako „Build automaticcaly“) také hodnoty „Refresh using native hooks or polling“ a „Refresh on access“. Jinak bychom museli jednou za čas dát obnovu projektu (pravým kliknutím na projekt a refresh, nebo levým kliknutím a stisk klávesy F5), aby se na server načetly soubory, které v projektu přibyly.

Použití serveru a jeho výstupy

Při správném spuštění serveru se nám tedy objeví vstupní formulář, který bude očekávat data. Po odeslání formuláře budou vložená data překontrolována a zobrazí se údaje o tom, co chybí, nebo co je špatně. Při kontrole údajů je provedeno ověření správnosti emailu, email musí být v obvyklém formátu. Dále je provedena kontrola popisu. V popisu jsou dovoleny malé a velké znaky anglické abecedy, číslice a podtržítko. Mezery jsou nahrazeny podtržítkem automaticky. Délka popisu by neměla být větší než dvacet znaků. Poslední kontrola, která je provedena, je správnost vložené sekvence a případné oříznutí sekvence ve fasta formátu tak, aby neobsahovala hlavičku ani další sekvence, pokud se jednalo o multi-fasta vstup, ať už se jednalo o zadání formou souboru nebo formou textu v textovém poli.

Po úspěšném ověření všech údajů jsme přesměrováni na stránku, kde jsou zobrazeny vložené údaje, a především odkaz na stránku s výsledky. Stránka s výsledky není vygenerována ihned po odeslání formuláře a zobrazení stránky přesměrování, nesmíme se tedy divit, když se nám po kliknutí na odkaz objeví stránka nenalezena.

Zpočátku na výstupní stránce, když už je vygenerována, jsou pouze informace o tom, že predikce není hotova. Postupně pak přibudou výsledky z nástroje Swiss model, výstup nástroje Superfammily a výstup z dalšího nástroje, resp. textové pole pro email z nástroje LOOPP, pokud je potřeba. Na konci výstupní stránky se, pokud vše dobře proběhlo, objeví informace o porovnání výstupů obou nástrojů pomocí iPBA, především 3D model vykreslující řetězce z obou nástrojů. Z důvodu občasného odmítnutí zpracování pdb souborů nástrojem iPBA, se občas stane, že tento model kombinující výsledky z obou nástrojů nebude přítomen.

Výstupem nástroje Swiss model je pdb id použitého templátu (po kliknutí na něj se zobrazí stránka v rcsb pdb obsahující informace o něm), hodnotu udávající identitu s templátem a chybovou hodnotu. Na výstupu jak nástroje Swiss model, tak dalšího zpracovaného nástroje, je zobrazen 3D model proteinu v nástroji Jmol. Při kliknutí na jména obou nástrojů se zobrazí odkaz na výslednou stránku nástroje. Ta je však obvykle ukládána pouze po nějakou dobu, a po jejím uplynutí je smazána. Delší dobu uchování a přehled výsledků za poslední nabízí nástroj swiss model, když se registrujeme.

Pokud je druhým zpracovávaným nástrojem loopp a výsledky nebyly nedávno zpracovávány (nejsou v paměti nástroje), zobrazí se textové pole pro obsah emailu s výsledky, které dorazí na zadanou emailovou adresu, když je zpracování dokončeno (je třeba sledovat, aby email byl opravdu od nástroje LOOPP, a jednu část předmětu emailu tvořil zadaný popis). Po zadání obsahu je po krátké době výstupní stránka znovu vygenerována. Formulář je nahrazen výstupem nástroje, krom 3D modelu je zobrazeno také id stromu, který byl použit (pořadí důvěryhodností jednotlivých id jsou rovněž zobrazeny).

Seznam příloh