VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV INFORMAČNÍCH SYSTÉMŮ

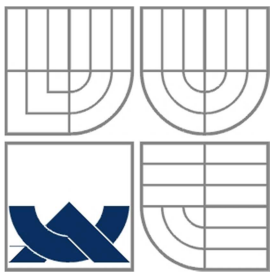
FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF INFORMATION SYSTEMS

WEBOVÝ SERVER PRO PREDIKCI SEKUNDÁRNÍ STRUKTURY PROTEÍNŮ

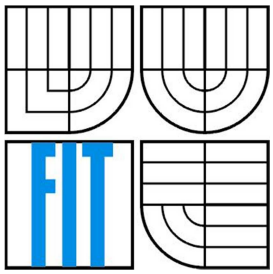
DIPLOMOVÁ PRÁCE
MASTER‘S THESIS

AUTOR PRÁCE BC. LUKÁŠ VILLEM

BRNO 2013



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV INFORMAČNÍCH SYSTÉMŮ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF INFORMATION SYSTEMS

WEBOVÝ SERVER PRO PREDIKCI SEKUNDÁRNÍ STRUKTURY PROTEINŮ

WEB SERVER FOR PROTEIN SECONDARY STRUCTURE PREDICTION

DIPLOMOVÁ PRÁCE
MASTER‘S THESIS

AUTOR PRÁCE
AUTHOR

BC. LUKÁŠ VILLEM

VEDOUCÍ PRÁCE
SUPERVISOR

ING. IVANA BURGETOVÁ, PH.D.

BRNO 2013

Vysoké učení technické v Brně - Fakulta informačních technologií

Ústav informačních systémů Akademický rok 2012/2013

Zadání diplomové práce

Řešitel: **Villem Lukáš, Bc.**
Obor: Bioinformatika a biocomputing
Téma: **Webový server pro predikci sekundární struktury proteinů**
Web Server for Protein Secondary Structure Prediction
Kategorie: Bioinformatika

Pokyny:

1. Seznamte se s problematikou predikce sekundární struktury proteinů a s nástroji pro predikci sekundární struktury proteinů.
2. Zvolené nástroje prostudujte detailně a seznamte se s možnostmi komunikace s těmito nástroji.
3. Navrhněte webový server, který pro zadanou sekvenci provede predikci její sekundární struktury pomocí zvolených existujících nástrojů a výsledky přehledně zobrazí uživateli. Dále pak na základě výsledků různých nástrojů vytvoří vlastní predikci sekundární struktury zadané sekvence.
4. Navržený webový server implementujte.
5. Otestujte funkčnost vytvořeného serveru a přesnost vlastního prediktoru na vhodném vzorku dat.
6. Zhodnoťte dosažené výsledky a diskutujte další rozvoj projektu.

Literatura:

- Zvelebil M., Baum J. O.: Understanding Bioinformatics, ISBN: 0-8153-4024-9, Garland Science, 2008
- Baxevanis A. D., Ouellette B. F. F.: Bioinformatics: A Practical Guide to the Analysis of Genes and Proteins, ISBN: 0-471-47878-4, Wiley-Interscience, 2005

Při obhajobě semestrální části diplomového projektu je požadováno:

- Body 1 až 3.

Podrobné závazné pokyny pro vypracování diplomové práce naleznete na adrese http://www.fit.vutbr.cz/info/szz/

Technická zpráva diplomové práce musí obsahovat formulaci cíle, charakteristiku současného stavu, teoretická a odborná východiska řešených problémů a specifikaci etap, které byly vyřešeny v rámci ročníkového a semestrálního projektu (30 až 40% celkového rozsahu technické zprávy).

Student odevzdá v jednom výtisku technickou zprávu a v elektronické podobě zdrojový text technické zprávy, úplnou programovou dokumentaci a zdrojové texty programů. Informace v elektronické podobě budou uloženy na standardním nepřepisovatelném paměťovém médiu (CD-R, DVD-R, apod.), které bude vloženo do písemné zprávy tak, aby nemohlo dojít k jeho ztrátě při běžné manipulaci.

Vedoucí: **Burgetová Ivana, Ing., Ph.D.**, UIFS FIT VUT
Datum zadání: 17. září 2012
Datum odevzdání: 22. května 2013

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
Fakulta informačních technologií
Ústav informačních systémů
612 66 Brno, Božetěchova 2

L.S.

doc. Dr. Ing. Dušan Kolář
vedoucí ústavu

Abstrakt

Tento diplomový projekt se zabývá problematikou predikce sekundární struktury proteinů. Po teoretickém úvodu následuje studie dostupných nástrojů, návrh a implementace webové aplikace, která kombinuje funkcionalitu několika webových nástrojů pro predikci sekundární struktury. Uživatel má možnost volit si počet využitých metod a vložit vybranou sekvenci jako prostý text nebo jako soubor ve standardním formátu. Získáním výsledků z vybraných nástrojů lze data konvertovat na unifikovaný formát, výsledek zobrazit uživateli a nad daty vytvořit vlastní predikci. Výsledná aplikace je testována a vliv jednotlivých nástrojů je upraven s cílem zvýšit úspešnost predikce. Výstupem aplikace je výsledek predikce, který je opět k dispozici jako text nebo soubor ke stažení.

Abstract

This master's thesis deals with protein secondary structure prediction. There is a theoretical introduction followed by study of available tools, proposal and implementation of web application, which combines functionality of several web tools used to predict secondary structure. User is asked to choose prediction methods and insert input sequence as plain text or upload a file. Results collected from selected tools serve to convert data into common format, show the result and create new type of prediction. Finally, the testing is applied and influences of tools are adjusted in order to increase percentage of prediction. The output of application is a result of prediction also available as plain text or as a file.

Klíčová slova

Proteiny, predikce, sekundární struktura, strukturální stav, metody predikce, webová aplikace, nástroj, webové služby.

Keywords

Proteins, prediction, secondary structure, structural state, prediction methods, web application, tool, web services.

Citace

Villem Lukáš: Webový server pro predikci sekundární struktury proteinů, diplomová práce, Brno, FIT VUT v Brně, 2013

Webový server pro predikci sekundární struktury proteinů

Prohlášení

Prohlašuji, že jsem tuto diplomovou práci vypracoval samostatně pod vedením Ing. Ivany Burgetové, Ph.D.
Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

……
Lukáš Villem
18. 5. 2013

Poděkování

Velmi rád bych poděkoval Ing. Ivane Burgetové, Ph.D. za odbornou pomoc a rady při tvorbě diplomové práce. Dále bych chtěl poděkovat Ing. Mariánovi Javorkovi, Ing. Jakubovi Janovičovi a Ing. Filipovi Janovičovi za pomoc a podporu při implementaci.

Obsah

1 Úvod

Molekuly proteínov majú tendenciu vytvárať mnohé komplikované tvary. Motívy, ktoré sa v štruktúre vyskytujú opakovane nazývame štrukturálne stavy sekundárnej štruktúry. Na základe znalosti sekundárnej štruktúry je možné odvodiť celkové priestorové usporiadanie molekuly a následne odhadnúť približnú funkciu proteínu. Práve funkcia proteínov je kľúčovou znalosťou a umožňuje nám lepšie pochopiť ako proteíny fungujú. V prvej časti práce sa nachádza teoretický úvod do problematiky proteínov, ktorý v krátkosti popisuje ich vznik, zloženie a štruktúru. Predpokladá sa, že čitateľ tejto práce má základné znalosti molekulárnej biológie, bioinformatiky a programovania, umelej inteligencie a soft computingu. Nezainteresovanému čitateľovi poskytuje druhá kapitola úvod do problematiky proteínov, ktorý sa však môže hodiť aj skúseným bioinformatikom.

Sekundárnu štruktúru možno určovať experimentálne alebo výpočtom. Vývoj sa pre viaceré problémy spojené s laboratórnym spracovaním zameral na metódy výpočtové. Tretia kapitola sa zaoberá práve problematikou predpovedania štrukturálnych stavov s využitím rôznych výpočtových techník. Medzi najznámejšie spôsoby predikcie patria pravdepodobnostný odhad, využitie podobnosti s homologickými molekulami, výpočet pomocou neurónovej siete alebo konsenzus nad výsledkami viacerých metód. Faktom je, že žiadna z metód nie je bezchybná, preto vznikli rôzne metódy hodnotenia úspešnosti, ktoré sú rovnako popísané v tejto kapitole.

Výpočtové metódy sú implementované a voľne dostupné najčastejšie v podobe webových aplikácií. Jedným z cieľov tejto práce je štúdia existujúcich nástrojov a následná voľba niektorých z nich. Spôsob akým tieto nástroje pracujú je rôzny, preto je potrebné vyextrahovať spoločné črty a zapracovať nástroje univerzálne. Pre tieto účely je potrebné s nástrojmi komunikovať bez interakcie užívateľa a výsledky získavať automatizovane. Štvrtá kapitola obsahuje detailnú analýzu zvolených nástrojov s ohľadom na vstupy, výstupy a možnosti komunikácie.

Na základe teoretickej štúdie a analýzy dostupných nástrojov možno získať predstavu o tom, ako má navrhovaný nástroj pracovať. Užívateľ vyplní vstupný formulár a vložené hodnoty odošle. Spustí sa extrakcia dát z jednotlivých nástrojov na základe vložených vstupov. Nad týmito dátami je ďalej vytvorená vlastná predikcia využívajúca percentuálne zastúpenie jednotlivých štrukturálnych stavov. Návrh aplikácie v piatej kapitole obsahuje definíciu hlavných cieľov projektu a výber použitých technológií. Implementácia aplikácie je popísaná v šiestej kapitole a obsahuje detailný popis komunikačného protokolu a spôsob, akým je vytvorený klient simulovanej webovej služby. Významnou časťou kapitoly je popis realizácie vlastnej predikcie.

Po implementácii nástroja nasleduje jeho dôkladné testovanie s cieľom zistiť kvalitu výstupu zapracovaných nástrojov a zvýšiť úspešnosť vlastnej predikcie. Šiesta kapitola obsahuje výber množiny testovacích dát, popis priebehu testovania, zhodnotenie testov a úpravu vplyvu nástrojov na celkový výsledok.

Mnoho súčasných biológov a bioinformatikov používa spomínané nástroje ako jednoduchý a rýchly spôsob získania predikcie sekundárnej štruktúry. Webový server je teda navrhnutý tak, aby bol konkurencie schopný v otázkach bezpečnosti, integrity dát, spoľahlivosti a osvojiteľnosti užívateľského rozhrania. Cieľová skupina užívateľov teda vďaka tomuto nástroju získa prístup k rôznym nástrojom z jedného miesta a pre zadávanie vstupov a získavanie výstupov im stačí použiť iba navrhnutý nástroj.

2 Proteíny

Z pohľadu molekulárnej biológie sú proteíny najzložitejšie a funkčne najdômyselnejšie známe molekuly. Pri pozorovaní bunky pod mikroskopom, skúmaní biochemickej či elektrickej aktivity bunky pozorujeme hlavne proteíny. Patria medzi hlavné stavebné elementy bunky, ale zároveň vykonávajú väčšinu bunečných funkcií. Funkcia proteínov je veľmi rozmanitá. Z mnohých možno spomenúť nasledovné:

- Stavebná,
- Zásobná,
- Katalýza chemických reakcií,
- Transport látok v organizme,
- Pohyb buniek a tkanív,
- Prenos informačných signálov medzi bunkami,
- Regulácia génovej expresie.

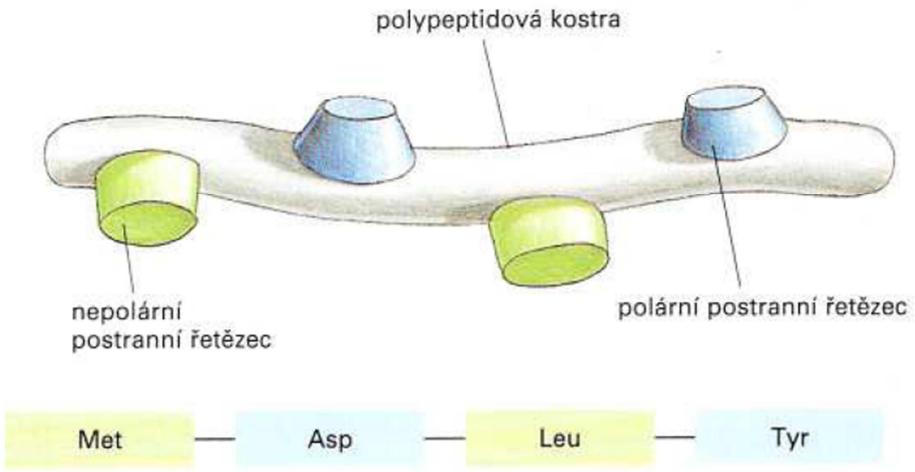
Pre pochopenie toho, ktorý proteín vykonáva akú funkciu je potrebné najprv detailne pochopiť ich štruktúru na atomárnej úrovni. V tejto kapitole možno nájsť stručný úvod do problematiky zloženia a štruktúry proteínov. Dôležitou časťou je definícia sekundárnych štruktúr, ktoré sú analyzované v ďalšom texte.

2.1 Vznik a zloženie

Proteíny vznikajú v procese translácie. Matrica mRNA sa prekladá do výslednej molekuly na ribozómoch za prítomnosti tRNA. Molekula proteínu je tvorená dlhým reťazcom aminokyselín vzájomne prepojených kovalentnou peptidovou väzbou. Opakujúce sa poradie atómov pozdĺž reťazca sa nazýva polypeptidová kostra, na ktorú sú pripojené postranné reťazce rôznych aminokyselín. Každý typ proteínu má jedinečné poradie aminokyselín.

Na stabilizácii celej molekuly sa podieľajú najmä slabé nekovalentné väzby. Sú to predovšetkým vodíkové mostíky, iontové väzby a van der Waalsove sily. Stabilita celého tvaru závisí na celkovej sile veľkého počtu týchto väzieb. Zásadný vplyv na výsledný tvar molekuly však má rozloženie polárnych a nepolárnych aminokyselín. Nepolárne postranné reťazce sa snažia zhlukovať vo vnútri proteínu, čo im umožňuje vyhnúť sa kontaktu s vodou, ktorá sa v bunke nachádza. Tieto reťazce sa najčastejšie viažu vodíkovými mostíkmi s polárnymi aminokyselinami alebo polypeptidovou kostrou. Polárne postranné reťazce sa snažia zdržovať na povrchu molekuly, kde sa môžu pomocou vodíkových mostíkov viazať s molekulami vody alebo inými polárnymi látkami.

Vďaka neustálemu vývoju lepších a rýchlejších metód sekvenovania DNA je v súčasnosti pomerne jednoduché určiť poradie nukleotidov DNA kódujúcich daný proteín a preložiť ich na postupnosť aminokyselín. Typická dĺžka proteínu je 50-2000 aminokyselín. Existujú však výnimky, ktorých dĺžka je menšia, približne 30 aminokyselín, a zároveň oveľa väčšia až okolo 10000 aminokyselín. [1]



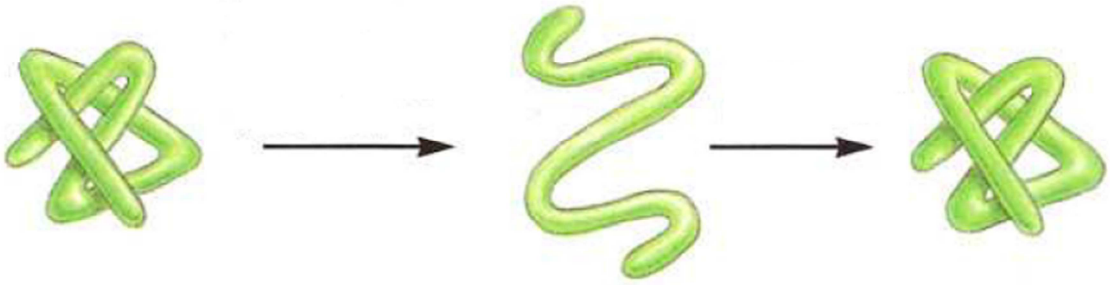
Obrázok 2.1: molekula proteínu vyjadrená štrukturálne a ako reťazec aminokyselín. Na výslednú štruktúru má kľúčový vplyv rozmiestnenie polárnych a nepolárnych reťazcov. [1]

2.2 Štruktúra

Každý druh proteínu má svoju vlastnú trojrozmernú štruktúru, ktorá je určená poradím aminokyselín v jeho reťazci. Po ukončení procesu translácie zaujme proteín energeticky najvýhodnejšiu priestorovú štruktúru.

2.2.1 Tvar molekuly

Pri denaturácii molekuly vhodným rozpúšťadlom sa prerušia nekovalentné väzby a vznikne voľný polypeptidový reťazec, ktorý úplne stratí svoju prirodzenú podobu. Naopak pri renaturácii, teda vysušení, sa molekula spontánne zvinie späť do svojej prirodzenej podoby. Tento jav sa nazýva skladanie proteínu resp. *protein folding*. Z toho vyplýva, že všetky potrebné informácie o tvare proteínu sú obsiahnuté v obsahu a poradí aminokyselín. Každý proteín sa skladá do svojho prirodzeného tvaru, ktorý je jednoznačne daný týmto poradím. Tvar sa však môže čiastočne zmeniť, ak proteín interaguje s ostatnými molekulami v bunke, pričom aj nepatrná zmena je často kľúčová pre výslednú funkciu proteínu.



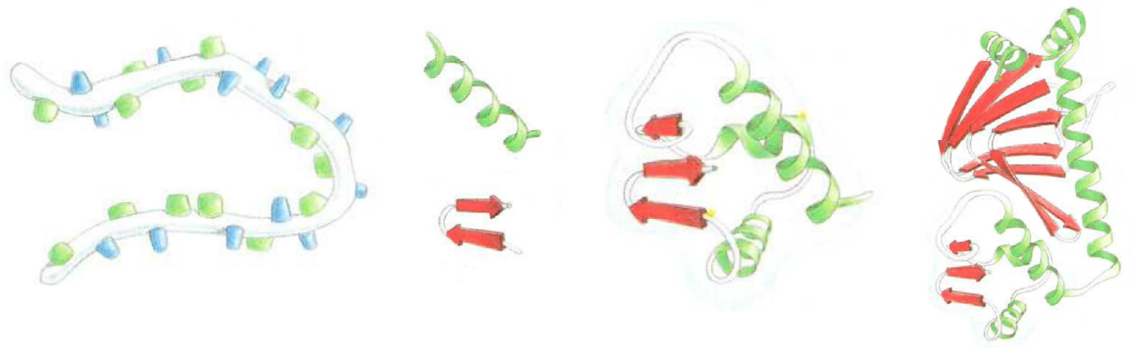
Obrázok 2.2: molekula sa pri denaturácii rozpadne na voľný polypeptidový reťazec, ktorý často reprezentuje postupnosť aminokyselín. Pri renaturácii sa proteín zloží do pôvodnej podoby. [1]

2.2.2 Hierarchické rozdelenie štruktúry proteínov

Na štruktúru proteínov sa možno pozerať hierarchicky. Jednotlivé úrovne štúdia sú nasledovné:

- Spomínaná sekvencia aminokyselín, ktorú čítame od N-konca ku C-koncu s nazýva primárna štruktúra. V tejto konformácii sa však molekuly v prírode bežne nevyskytujú, ale skladajú sa do veľkého množstva komplikovaných tvarov.
- Napriek tomu, že tvar každého proteínu je unikátny, možno tu pozorovať často sa opakujúce motívy. Tvar proteínu z pohľadu týchto opakujúcich sa elementov sa nazýva sekundárna štruktúra. Týchto motívov existuje viacero, avšak medzi najčastejšie patria α-šrôbovica (α-helix) a β-list (β-sheet). Štúdiom sekundárnych štruktúr sa zaoberá celá táto práca, preto budú detailne rozobrané a je im venovaná celá nasledujúca kapitola.
- Znalosť sekundárnej štruktúry je nevyhnutná pre určenie výsledného tvaru molekuly. Finálna trojrozmerná štruktúra sa nazýva terciárna.
- V prírode často nevystupujú molekuly proteínov osamote ale spájajú sa do zložitejších celkov prostredníctvom väzobných miest. Vznikajú tak komplexy, ktoré sa nazývajú kvartérna štruktúra. V bunkách tak často vznikajú symetrické útvary, ktoré sú komplexom molekúl rovnakých proteínov.

Špeciálnym prípadom usporiadania je tzv. proteínová doména. Je tvorená ľubovoľnou časťou polypeptidového reťazca, ktorá sa môže nezávisle zvinúť do kompaktnej stálej štruktúry. Domény spravidla vykonávajú unikátnu funkciu a jeden proteín môže mať niekoľko rozličných domén.



Obrázok 2.3: Hierarchia proteínových štruktúr. Primárna – sekvencia aminokyselín, sekundárna - α-helix a β-sheet, terciárna – jednoduchá doména, kvartérna – doména tvorená dvoma molekulami. [1]

2.2.3 Proteínové rodiny

V priebehu evolúcie sa štruktúra niektorých proteínov mierne pozmenila s účelom vzniku nových vlastností a funkcií. Mnoho dnešných proteínov možno zaradiť do skupín nazývaných rodiny. Sekvencia aminokyselín v rámci rovnakej rodiny je veľmi podobná a na úrovni primárnej štruktúry obsahuje iba malé zmeny. Rovnako na úrovni sekundárnej a terciárnej štruktúry je tvar molekuly takmer identický. Naopak proteíny patriace do rôznych rodín majú štruktúru výrazne odlišnú. Keďže funkcia proteínov sa odvíja výlučne od ich štruktúry, proteíny patriace do rovnakej rodiny často vykonávajú podobnú alebo rovnakú funkciu. Nie je to však podmienkou, preto aj proteíny patriace do rovnakej rodiny môžu v organizme vykonávať rôzne funkcie.

Skúmanie štruktúry a zaraďovanie do proteínových rodín má teda zásadný význam pri určovaní vlastností a funkcií proteínov.

3 Predikcia sekundárnej štruktúry

Z predchádzajúcej kapitoly vyplýva, že znalosť sekundárnej štruktúry je neodmysliteľnou súčasťou znalosti celkovej trojrozmernej štruktúry proteínu a určovania približnej funkcie proteínu. Vo rámci tejto práce sú cieľom predikcie proteíny globulárne, avšak existujú aj predikcie pre určenie tvaru proteínov pri prechode membránou. Existuje viacero spôsobov a metód, ako sekundárnu štruktúru určovať. Každá z nich využíva rozličné princípy, má svoje výhody aj nevýhody.

3.1 Problémy spojené s predikciou

Sekundárnu štruktúru molekuly proteínu možno jednoznačne určiť experimentálne v laboratórnych podmienkach použitím metód röntgenovej kryštalografie alebo NMR-spektroskopie. Tieto techniky sú však často nepresné, sú vysoko náročné na čas a prostriedky.

Pravdepodobne najzávažnejším problémom, ktorý je spojený s analýzou sekundárnej štruktúry proteínov je ich enormne vysoký počet. Teoreticky sa dá zostaviť nekonečné množstvo rôznych polypeptidových reťazcov. Rozoznávame 21 rôznych aminokyselín, z ktorých každá je chemicky odlišná a potenciálne sa môže vyskytovať na ktoromkoľvek mieste v proteínovom reťazci. Vzniká nám teda 21^n jedinečných možností, ako vytvoriť reťazec zložený z n aminokyselín. V prípade typického proteínu zostaveného z asi 300 aminokyselín dostávame 21^{300} (10^{397}) možných reťazcov.

Z toho vyplýva, že v reálnom čase nie je možné explicitne efektívne analyzovať molekuly všetkých proteínov. Bolo teda potrebné vymyslieť výpočtové metódy, ktoré umožňujú na základe znalosti štruktúry a vlastností známych proteínov vytvoriť isté štruktúrne rysy a sekundárnu štruktúru ľubovoľného proteínu odvodiť.

3.2 Sekundárna štruktúra

Pre účely ďalšieho textu je potrebné zaviesť dôležitý a často sa opakujúci pojem residua. Jedná sa o výrastky polypeptidovej kostry, ktoré vytvárajú väzby s postrannými reťazcami, inými molekulami alebo medzi sebou. Sekundárnu štruktúru potom možno definovať na základe nasledovných vlastností:

- Torzný uhol residuí,
- Vzory vodíkových mostíkov,
- Dĺžka vodíkovej väzby,
- Zakrivenie polypeptidovej kostry,
- Energetické interakcie.

Najvýznamnejší vplyv na vznik sekundárnych štruktúr majú práve vodíkové väzby. Tie vytvárajú opakujúce sa vzory tak, že spájajú skupiny N-H a C=O v polypeptidovej kostre a nezahŕňajú postranné reťazce. Rozoznávame nasledujúci trojstavový model často sa opakujúcich štruktúr.

3.2.1 α-helix

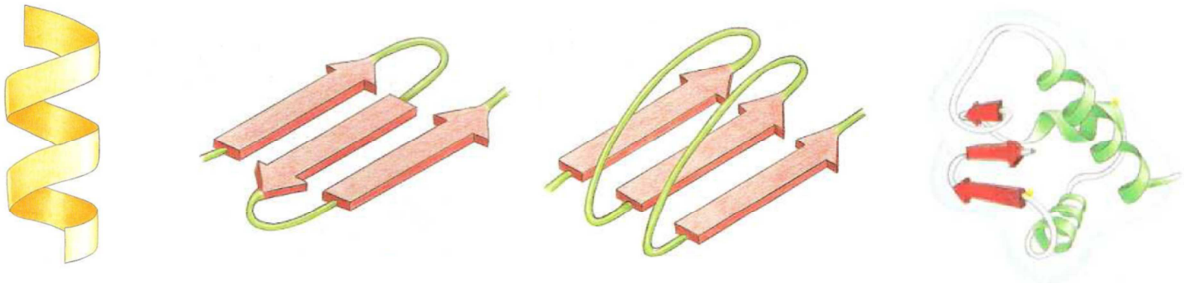
Objavený v proteíne α-keratín, ktorý je zložkou kože a kožných derivátov. Polypeptidový reťazec sa ovíja okolo seba a vzniká tak tuhý valec. Vzniká tak, že medzi každým štvrtým residuom (i,i+3) vzniká vodíkový mostík medzi skupinami C=O a N-H. Najmenší možný počet residuí, ktoré sú potrebné na vytvorenie takejto štruktúry je teda dva. V molekule tak vznikajú pravidelné závitnice s 3,6 aminokyselinovými zvyškami na jednu otáčku.

3.2.2 β-sheet

Objavený v proteíne fibroín, ktorý je hlavnou zložkou hodvábu. Definícia je podobná ako pre helix, avšak namiesto osamotených residuí prichádza ku spojeniu celých úsekov, ktoré sa môžu vyskytovať na ľubovoľnom mieste proteínu. β-štruktúra býva spravidla rádovo dlhšia než α-helix. Rozoznávame dva typy týchto štruktúrnych elementov v závislosti na spôsobe ich vzniku. Prvým je paralelný, v ktorom sa skladané listy tvoria zo susediacich polypeptidových reťazcov s rovnakou orientáciou. Druhý je antiparalelný, kde má každý úsek vzhľadom k najbližším úsekom opačnú orientáciu.

3.2.3 Coil

Jedná sa o náhodné úseky a klbká vznikajúce v miestach, kde sa nenachádza α ani β štruktúra. Zároveň sa však môže jednať sa o špeciálny prípad α-helix. Tu sa niektoré páry závitníc ovíjajú okolo seba a vytvárajú tak stabilnú štruktúru. Takáto situácia môže nastať, ak dva helixy majú väčšinu svojich nepolárnych postranných reťazcov na jednej strane, takže sa môžu ovíjať okolo seba a ich nepolárne reťazce sú obrátené smerom dnu.



Obrázok. 3.1 – sekundárna štruktúra proteínov. Zľava *α-helix*, antiparalelný a paralelný *β-sheet*, krátky segment molekuly proteínu. Úseky medzi *α* a *β* štruktúrou sú *random coil*. [1]

3.2.4 Priradenie sekundárnej štruktúry

Je nutné spomenúť, že na základe znalostí o sekundárnej štruktúry proteínov v minulosti vznikli metódy, ktoré ju analyzujú podrobnejšie. Boli to prvé pokusy o automatizované určenie sekundárnej štruktúry na základe heuristík získaných zo štruktúry známych molekúl.

Významnou metódou, ktorá skúmala existenciu vodíkových väzieb medzi blízkymi resiudami je DSSP (Define Secondary Structure of Proteins)[1]. Táto metóda využíva rozšírenie zaužívaného trojstavového modelu na model osemstavový. Takýmto spôsobom identifikuje 50-60% všetkých residuí ako štruktúrne elementy. Získané výsledky sú následne filtrované a elementy, ktoré sú príliš krátke alebo dlhé sú odstránené. Na záver prevádza získané elementy na trojstavový model.

Ďalšou dôležitou metódou je metóda PALSSE (Predictive Assignment of Linear Secondary Structure Elements)[2]. Štruktúru analyzuje na základe metriky definovanej pomocou tzv. C_α atómov, teda atomických súradníc. Táto metrika pracuje s dvoma geometrickými vlastnosťami, ktorými sú vzdialenosť medzi dvojicou residuí (i, i+3) a torzný uhol štvorice úspešných C_α atómov. V prípade, že vzdialenosť residuí je dostatočne krátka alebo torzný uhol splňuje požadovanú veľkosť, je možné danú časť sekvencie prehlásiť za štruktúrny element.

[1] http://swift.cmbi.ru.nl/gv/dssp/
[2] http://www.biomedcentral.com/1471-2105/6/202

3.3 Hodnotenie úspešnosti predikcie

Od žiadnej z metód predikcie nemožno očakávať 100% úspešnosť. Nedá sa teda povedať, ktorá metóda je najlepšia a žiadnu z nich nemožno považovať za referenčnú . Keďže rôzne metódy využívajú odlišné techniky a postupy, výsledky získané z rozličných metód sú často veľmi odlišné. Niektoré metódy sú schopné odhaliť všetky štruktúrne elementy, môžu však predikovať aj štruktúry, ktoré neexistujú. Naopak existujú metódy, ktoré sa špecializujú na zistenie menšieho počtu elementov avšak s vyššou presnosťou. V prípade, že sa predikcia realizuje na neznámej molekule proteínu, je potrebné poznať jej presnosť.

Bolo teda potrebné navrhnúť techniky, ktoré určia kvalitu predikcie a celkovú kvalitu používaných metód. Využitím týchto techník je možné zhodnotiť presnosť predikcie jednak pozície residuí v sekvencii a zároveň počtu a umiestnenia sekundárnych elementov. Pre takéto hodnotenie je potrebné poznať sekundárnu štruktúru skúmaného proteínu a porovnať ju z výsledkom predikcie. Pri dostatočne veľkom počte vzoriek je možné pomerne presne určiť jej presnosť.

3.3.1 Metóda Q_3

Táto metóda udáva presnosť predikcie jednotlivých residuí v rámci vstupnej sekvencie. Jedná sa o triviálnu metódu, ktorá používa nasledovný výpočet:

$$Q_3 = \frac{počet\ koretne\ predikovaných\ residuí}{celkový\ počet\ residuí}$$

Hodnoty Q_3 sa pohybujú v rozmedzí 0 až 1, kde 0 značí absolútne nepresnú predikciu a 1 naopak predikciu bezchybnú. Keďže žiadna z metód neumožňuje predikovať výskyt 100% všetkých residuí, hodnota Q_3 bude vždy menšia než 1. Pri jej používaní však treba brať ohľad na jej jednoduchosť a používať ju len za účelom vytvorenia si predstavy o presnosti metód predikcie. Môže totiž nastať prípad, že je predikcia takmer nepoužiteľná, napriek tomu má vysokú hodnotu Q_3. [1]

3.3.2 Metóda Sov

Slúži na hodnotenie úspešnosti predikcie celých štruktúrnych elementov a je založená na čiastočnom prekrývaní segmentov. Je oveľa výhodnejšie zistiť ich korektný počet, typ a poradie ako predikovať neexistujúce, prípadne niektoré vynechať. Vzorec na výpočet je vyzerá takto:

$$Sov = \frac{100}{N_{Sov}} \sum_{S_0} \left[\frac{minov(s_{obs}, s_{pred}) + \delta(s_{obs}, s_{pred})}{maxov(s_{obs}, s_{pred})} len(s_{obs}) \right]$$

Význam jednotlivých symbolov je nasledovný. Ak všetky pozorované segmenty označíme ako s_{0bs} a všetky predikované segmenty označíme ako s_{pred}, potom S_o je množina všetkých prekrývajúcich sa dvojíc s_{0bs} a s_{pred}, ktoré sa nachádzajú v rovnakom stave. Dĺžka pozorovaného segmentu značíme ako $len(s_{obs})$. Pre ľubovoľnú dvojicu z S_o v danom stave, je dĺžka prekrývajúcej sa časti definovaná ako $minov(s_{obs}, s_{pred})$ a úplný rozsah residuí segmentov tejto dvojice sa nazýva $maxov(s_{obs}, s_{pred})$. Hodnota prekrytia je rozšírená o hranice residuí pomocou faktoru $\delta(s_{obs}, s_{pred})$. Táto hodnota je definovaná ako:

$$\delta(s_{obs}, s_{pred}) = min\left\{\begin{matrix}\left(maxov(s_{obs}, s_{pred}) - minov(s_{obs}, s_{pred})\right) \\ minov(s_{obs}, s_{pred}) \\ int\left[len(s_{obs})/2\right] \\ int\left[len(s_{pred})/2\right]\end{matrix}\right\}$$

Operátor $\mathrm{int}[x]$ značí celočíselné zaokrúhľovanie smerom nadol. Na záver je suma všetkých dvojíc normalizovaná počtom pozorovaných residuí. Na rozdiel od Q3 tak možno získať rádovo odlišné hodnoty a odhaliť mnohé nepresné predikcie. [1]

3.4 Štatistické metódy

Kľúčovým aspektom pre určovanie sekundárnej štruktúry pomocou štatistických metód bolo zavedenie pojmu propensity. Táto veličina vyjadruje preferenciu jednotlivých aminokyselín vytvárať elementy sekundárnej štruktúry. Pre určenie týchto preferencií bolo potrebné poznať aspoň približnú sekundárnu štruktúru skúmaných segmentov. Existuje však množstvo proteínov, ktorých štruktúra nebola určená experimentálne. Riešením tohto problému sa stalo použitie odhadu štruktúry na základe homologických sekvencií. Za týmto účelom možno použiť napríklad techniku *leave one out*. Tá z vybraných sekvencií vytvorí databázu štruktúr, u ktorý je zhoda sekvencií väčšia ako prahová hodnota (typicky 25%). Z pôvodnej množiny je tak vybraná podmnožina typicky obsahujúca typicky 1-4% prvkov. Na základe takejto databázy možno hodnotu propensity získať pomerne jednoducho pomocou nasledovného výpočtu:

$$P_{s,a} = \frac{P_{s,a}}{p_a}$$

Tu je najprv potrebné zistiť hodnotu p_a, teda podmnožinu všetkých residuí, ktoré sú typu *a*. Následne sa vypočíta hodnota $p_{s,a}$, teda podmnožina všetkých residuí typu *a*, ktoré sa nachádzajú v štrukturálnom stave *s*.

3.4.1 Windowing

Predikciu sekundárnej štruktúry na základe hodnoty propensity možno upraviť tak, že nepracujeme s preferenciami jednotlivých residuí, ale preferencie počítame ako priemer v rámci menších pod reťazcov. Ukázalo sa, že z praktického hľadiska je takáto úprava výhodná a má výrazný vplyv na zvýšenie kvality predikcie.

Počet residuí, z ktorých vytvárame priemer sa nazýva window (okno) a jeho dĺžka závisí na predpovedanej štruktúre. Pre získanie správnych výstupov je nevyhnutné zvoliť vhodnú dĺžku okna. Okno musí byť dostatočne krátke, aby v ňom bolo možné rozpoznať aj najkratšie štruktúry, avšak výber príliš krátkeho okna vedie k mnohým falošným výsledkom. β-štruktúry sú obvykle kratšie než α-štruktúry a vyžadujú kratšie okná, typicky 3-5 residuí. Priemerná dĺžka okna sa pohybuje v rozsahu 13-20 residuí.

3.4.2 Znalosť homologických sekvencií

Presnosť predikcie možno ďalej zvyšovať využitím informácií príbuzných sekvencií. Kľúčovým pozorovaním, ktoré podopiera tento prístup je skutočnosť, že zbalené molekuly s omnoho konzervovanejšie než sekvencie. Očakáva sa teda, že segmenty viacnásobného zarovnania majú korešpondujúce elementy sekundárnej štruktúry.

Dôležitým aspektom je správny výber a zarovnanie sekvencií. V opačnom prípade môže ich použitie presnosť naopak znižovať. Aby boli metódy predikcie sekundárnej štruktúry naozaj plne automatizované, je nutné sa vyvarovať chybnému zaradeniu nesúvisiacich sekvencií, ale zároveň sa pokúšať o výber všetkých sekvencií, ktoré sú naozaj homologické. Prevažná väčšina súčasných metód používa na výber týchto sekvencií algoritmus PSI-BLAST[3]. Tento systém prekonáva spomínané problémy vyhľadávaním v 3 iteráciách s použitím striktnej prahovej hodnoty.

Jednotlivé metódy využívajú túto znalosť tak, že najprv vytvoria predikciu pre všetky homologické sekvencie nezávisle. Následne sa vytvorí viacnásobné zarovnanie, v ktorom sú vyhľadávané podobné residuá a vypočíta sa hodnota skóre všetkých štruktúrnych stavov na každej pozícii. Finálna predikcia na každej pozícii je potom štruktúrny stav s najvyššou hodnotou skóre. Tá je spravidla modifikovaná vyfiltrovaním príliš krátkych elementov. Využitím znalostí z homologických sekvencií je možné zvýšiť úspešnosť Q_3 až o 9%.

3.4.3 C-F

Pochádza z roku 1974 a jej tvorcami sú Peter Chou a Gerald Fachman, podľa ktorých je aj nazvaná. Jedná sa o jednou z prvých metód využívajúcu propensity a predikcie sekundárnej štruktúry vôbec. Na základe preferencií vytvárať sekundárne štruktúry rozdelili aminokyseliny do nasledujúcich tried:

- Silne formujúce,
- Formujúce,
- Indiferentné,
- Prerušujúce,
- Silne prerušujúce.

Samotný výpočet je pomerne zložitý, využíva okná rôznych dĺžok a rôzne nastavenia prahových hodnôt aj pre odhadovanie rovnakých štruktúrnych elementov. Významnou vlastnosťou tejto metódy je existencia alternatív. Pri iniciálnom zistení štruktúrneho elementu sa metóda musí rozhodnúť o aký element sa jedná na základe toho, ktorý typ má väčšiu priemernú preferenciu v rámci okna.

Na svoje výpočty použili vstupnú množinu obsahujúcu približne 2500 vzoriek. Ich výsledky boli v roku 1998 vylepšené pomocou množiny obsahujúcej zhruba 33000 residuí. Hlavnou motiváciu tejto rekalkulácie bolo vylepšenie predikcie na koncoch α-helix.

Aj keď sa táto metóda v súčasnosti využíva iba zriedka, je považovanú za dogmu v oblasti predikcie sekundárnej štruktúry. Existujú rôzne derivácie a úpravy tejto metódy. Medzi najvýznamnejšie patrí jej využitie na zisťovanie vlastností sekundárnych elementov pri prechode proteínu membránou. Pôvodné metódy využívali hydrofóbnych vlastností bunky a predpovedali umiestnenie residuí vzhľadom k membráne. Na tieto predikcie sa používali metriky vytvorené výlučne experimentálne. V roku 1994 vznikla metóda MEMSAT[4], ktorá na základe metódy Chou-Fachman analyzovala residuá sekundárnej štruktúry pri prechode membránou. Na tento účel bola rovnica na výpočet upravená nasledovne:

[3] Position-Specific Iterative Basic Local Alignment Search Tool
[4] http://bioinf.cs.ucl.ac.uk/software_downloads/memsat/

$$P'_{s,a} = ln\left(\frac{P_{s,a}}{p_a}\right)$$

Na základe týchto hodnôt bola vytvorená nová metrika, ktorá predpovedá 5 stavov, z toho 3 medzi membránové (vo vnútri, uprostred, vonku) a 2 mimo membránové (vo vnútri, vonku).

3.4.4 GOR

Napriek úspešnosti predchádzajúcich postupov bolo zistené, že štruktúrny stav jednotlivých residuí je silno ovplyvnený susednými sekvenciami. Po rozpletení celého polypeptidového reťazca sa molekula vždy zvinie späť do pôvodnej podoby. Toto však neplatí pre veľmi krátke úseky a rovnaká sekvencia sa môže nadobúdať rôzny štrukturálny stav. Tieto tvrdenia sú podložené viacerými štúdiami, napr. štúdia, ktorú vypracovali Gavin Crooks a Steven Brenner.

Využitím tohto prístupu vznikli nové metódy, ktoré sú omnoho presnejšie než metódy založené na skúmaní individuálnych residuí. GOR predstavuje sériu metód založených na skutočnosti, že okrem lokálnej informácie ovplyvňuje sekundárnu štruktúru aj informácia z okolitých sekvencií. Dĺžka týchto okolitých sekvencií bola stanovená na 8 zľava aj sprava. Podobne ako ostatné metódy predikcie, aj GOR využíva databázu trénovacích vzoriek a znalosť heuristík. Centrálny koncept využitý v metóde GOR I a II možno formulovať nasledovne.

$$I\left(S_j = (s{:}\,\bar{s}); x_{j-8} \dots x_{j+8}\right) = \sum_{m=-8}^{8} I\left(S_j = (s{:}\,\bar{s}); x_{j+m}\right)$$

Residuum, ktoré je objektom skúmania možno označiť indexom *j*. Tento výrastok v rámci sekvencie *x* značíme ako x_j a jeho štrukturálny stav ako S_j. Pravdepodobnosť, že výrastok *j* sa nachádza v štrukturálnom stave *s* sa značí ako $P\left(S_j = s\right)$. Pravdepodobnosť, že výrastok *j* sa nachádza v štrukturálnom stave *s* práve ak sekvencia obsahuje residuá $\hat{x}$ je $P\left(S_j = s\middle|\hat{x}\right)$, kde $\hat{x}$ je špecifická časť sekvencie symetrickej dĺžky vzhľadom k x_j (napr. x_{j-2}, x_{j-1}, x_j, x_{j+1}, x_{j+2}). Tieto dve pravdepodobnosti sú identické iba v prípade, že residuá v $\hat{x}$ nemajú žiadny vplyv na štrukturálny stav S_j. Pomocou týchto dvoch pravdepodobností možno definovať spoločnú informáciu, ktorú $\hat{x}$ nesie o štrukturálnom stave *s* residua *j* nasledovne:

$$I\left(S_j = s; \hat{x}\right) = \log\left(\frac{P\left(S_j = s\middle|\hat{x}\right)}{P\left(S_j = s\right)}\right)$$

Skutočnosť, že sa skúmaný výrastok v stave *s* nenachádza značíme $\bar{s}$. Štrukturálne stavy *s* a $\bar{s}$ sú vždy vzájomne exkluzívne. Výpočet použitý v metóde GOR III a IV je vylepšený nasledovným spôsobom:

$$I\left(S_j = (s{:}\,\bar{s}); x_{j-8} \dots x_{j+8}\right) = I\left(S_j = (s{:}\,\bar{s}); x_j\right) + \sum_{m=-8}^{8} I\left(S_j = (s{:}\,\bar{s}); x_{j+m}\middle|x_j\right)$$

Týmto spôsobom sa postupne vypočíta hodnota $I\left(S_j = (s{:}\,\bar{s}); x_{j-8} \dots x_{j+8}\right)$ na všetkých pozíciách v skúmanej sekvencii a pre každú pozíciu je predpovedaný štruktúrny stav s najväčšou hodnotou.

Záverečným krokom nielen týchto metód filtrovanie výsledkov na základe určenia minimálnej dĺžky predikcie, ktorá má pre α-helix hodnotu 4 a pre β-sheet hodnotu 2. Poslednou zo série je metóda GOR V, ktorá na rozdiel od predchádzajúcich verzií používa znalosť z homologických sekvencií. Z GOR V vychádza ďalšia často používaná metóda s názvom ZPred.

3.5 Nearest-neighbor metódy

Centrálnym konceptom týchto metód je využitie podobnosti krátkych segmentov sekvencií v prípade, že aspoň jeden zo segmentov je známou štruktúrou. Dĺžka segmentov zodpovedá dĺžke okna a volí sa zvyčajne 17-19 residuí, ale častokrát aj oveľa menej. Na rozdiel od predchádzajúcich metód teda nie je podmienkou, aby boli sekvencie v zarovnaní homologické. Aj keď existuje súvislosť medzi sekvenciou a jej štruktúrou, je náročné ju definovať pre krátke úseky sekvencií.

Pre každý segment sekvencie vyhľadáme v databáze sekvencií, ktorých štruktúra je známa, najlepšie zarovnania bez medzier. Nasleduje výber skórovacej schémy, ktorá je obvykle navrhovaná a upravená špecificky pre danú metódu. Zarovnania s najvyšším skóre sú potom vybrané a použité na predikciu štrukturálneho stavu, ktorý je daný buď stavmi s najväčším výskytom, alebo sú jednotlivé stavy váhované pomocou skóre. Hlavným problémom týchto metód je kvantitatívne hľadisko a napriek dostupnosti rôznych známych skórovacích matíc si mnohé metódy predikcie odvodili schémy vlastné. Pre vylepšenie predikcie využívajú aj tieto metódy informácie z homologických sekvencií.

3.5.1 Vážený priemer segmentov

Najjednoduchší spôsob získania predikcie založenej na hľadaní najbližších susedov je použiť iba jeden najbližší segment a priradiť jeho štruktúru ekvivalentnému segmentu dotazovanej sekvencie. Spočiatku vznikali prevažne metódy a postupy, ktoré predpovedali štrukturálny stav centrálneho residua pre zvolené okno. Výsledok predikcie každého okna je takto nezávislý na výsledkoch z ostatných okien. Preto stačí po získaní množiny najbližších susedov pre zadané okno definovať metódu, ktorá potenciálny štrukturálny stav vyberie spomedzi predikcií rôznych segmentov.

Jedným zo spôsobov je jednoducho spočítať počet výskytov štrukturálnych stavov centrálneho výrastku všetkých najbližších segmentov a následne použiť najčastejšie sa vyskytujúci stav ako predpoveď skúmaného výrastku dotazovanej sekvencie. Váhy susedných sekvencií sú pri použití tohto prístupu vždy rovnaké bez možnosti rozlišovať medzi nimi. Napriek jednoducosti sa tento spôsob osvedčil v nasledujúcich metódach, ktoré na základe výpočtu podobnosti váhujú predikcie.

3.5.2 SIMPA96

Táto metóda ako jedna z mála využíva štandardnú skórovaciu maticu, v tomto prípade BLOSUM-62. Vybrané sú segmenty, ktorých skóre prekračuje zvolenú prahovú hodnotu. Prahová hodnota bola určená na 13 residuí pri okne o veľkosti 15. Za účelom výberu vhodných segmentov je prehľadávaná komplexná množina známych proteínových štruktúr, pričom pre predikciu stavu každého residua je použitých priemerne 3000 segmentov.

3.5.3 NNSSP

Pri predpovedaní stavu centrálneho residua využíva 50 segmentov s najlepším skóre o celkovej dĺžke 19 residuí. Pri predikcii rozoznáva 12 možných štrukturálnych stavov rozdelením sekundárnych štruktúr na N a C konce a vnútorný úsek. Navyše používa 6 možných stavov prostredia, čo spolu činí 72 rôznych stavov.

Hlavným rozdielom oproti ostatným metódam je skutočnosť, že množina proteínových sekvencií prehľadávaných na najbližších susedov je značne obmedzená. Na tento účel boli navrhnuté dve techniky, pričom obe pracujú s hodnotou propensity. Prvá z nich vypočítava rozdiel v skladbe aminokyselín a je definovaná ako rozdiel frekvencií výskytu residua typu *a* v dotazovanej a známej sekvencii:

$$D_{composition} = \sum_{a=1}^{20} \left(P_a^{query} - P_a^{database}\right)^2$$

Druhý pístup za založený na Chou-Fachman propensity, ktorý sa zistí ako suma rozdielov propensity dotazovanej a známej sekvencie cez potenciálne štrukturálne stavy. Suma hodnôt propensity je normalizovaná počtom residuí, pričom aktuálne skúmaný výrastok značíme x_j prípadne y_j. Vzorec na výpočet potom vyzerá nasledovne.

$$D_{CF} = \sum_{S} \left(\frac{\sum_{j=1}^{N_{query}} P_{s,x_j}}{N_{query}} - \frac{\sum_{j=1}^{N_{database}} P_{s,y_j}}{N_{database}} \right)^2$$

Týmto spôsobom sa získa priemerný rozdiel propensity pre každý štrukturálny stav v rámci celej sekvencie. Do hľadania najbližších susedov je potom začlenených 90 najbližších sekvencií vzhľadom na obe uvedené metódy.

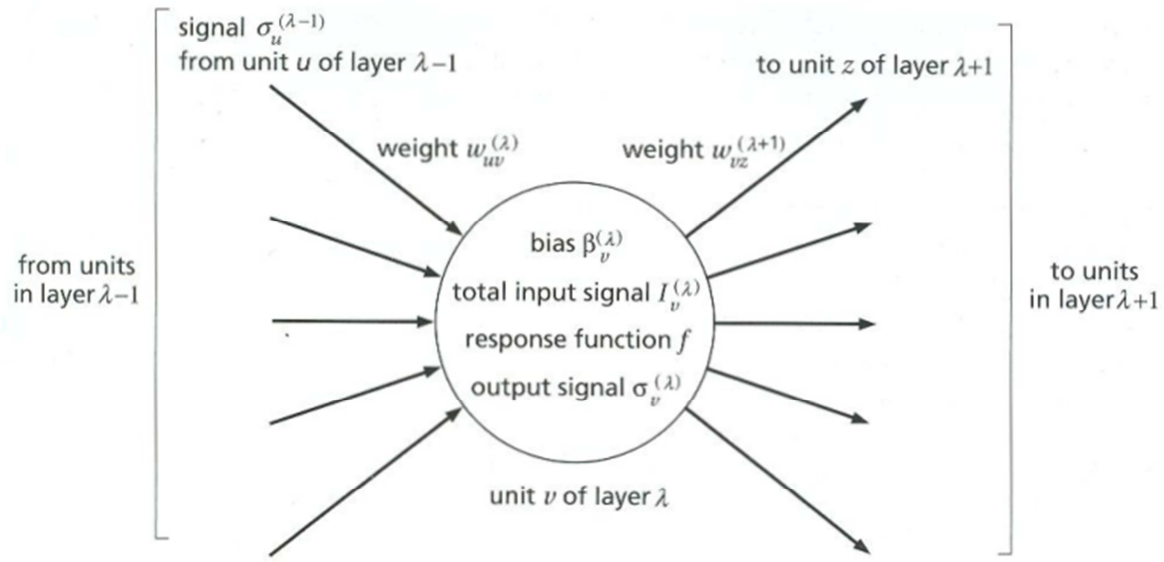
3.5.4 SSPAL

Všetky predchádzajúce metódy počítali s pevnou dĺžkou okna. Inovácia tejto metódy spočíva v použití viacerých okien rôznej dĺžky, nad ktorými vykonáme separované predikcie a tie priemerujeme do výslednej predikcie. Potenciálnou výhodou tohto prístupu je indikácia dlhších zarovnaní, ktoré možno jednoducho prehliadnuť výberom príliš malého okna. Postup výpočtu korešponduje s metódou NNSSP. Medzi najbližších susedov sa opäť dostane iba 90 najbližších sekvencií a predikciu vytvoríme tak, že pomocou spomínanej skórovacej funkcie vyberáme 50 segmentov.

3.6 Neurónové siete

Aplikácia neurónových sietí sa pri analýze a predikcii sekundárnej štruktúry ukázala ako vysoko výkonný nástroj používajúci odlišný a efektívny prístup. Nemožno napísať presný vzorec, ktorý prevádza proteínovú sekvenciu na sekundárnu štruktúru, pretože reprezentácia dát je omnoho komplexnejšia ako v prípade predchádzajúcich metód. Z časového hľadiska sa ukázalo, že neurónové siete majú zásadný vplyv na zvýšenie presnosti predikcie a metódy pracujúce na tomto princípe dosahujú jedny z najlepších výsledkov.

Najčastejšie používaným typom sú siete dopredné. Uzly sú v nich usporiadané do vrstiev, v rámci ktorých spolu neinteragujú. Komunikácia prebieha jednosmerne a do vstupnej vrstvy vchádzajú vstupné dáta v podobe proteínovej sekvencie. Signály postupne prechádzajú cez skryté vrstvy, ktorých počet sa volí v rozmedzí 1 až 3. Výsledok v podobe predikcie sekundárnej štruktúry získame z vrstvy výstupnej. Siete sú spravidla plne prepojené, teda všetky uzly jednej vrstvy sú prepojené so všetkými uzlami vo vrstve nasledujúcej. Prenášané signály nadobúdajú hodnoty od 0 do 1 a závisia výlučne na signále prijatom z predchádzajúcej vrstvy. Komunikáciu vybraného uzlu skrytej vrstvy možno vidieť na nasledovnom obrázku.



Obrázok 3.2 – ľubovoľný uzol skrytej vrstvy neurónovej siete. [2]

Zobrazený uzol značíme ako uzol na indexe *v* vo vrstve λ. Táto vrstva má N_λ uzlov, z ktorých každý získava signály od všetkých $N_{\lambda-1}$ uzlov predchádzajúcej vrstvy. Uzol *u* posiela z predchádzajúcej vrstvy signál $\sigma_u^{(\lambda-1)}$, pričom uzol *v* modifikuje túto hodnotu prostredníctvom váhy $w_{uv}^{(\lambda)}$. Celkový vstupný signál uzla *v* potom zodpovedá nasledovnej hodnote:

$$I_v^{(\lambda)} = \sum_{u=1}^{N_{\lambda-1}} w_{uv}^{(\lambda)} \sigma_u^{(\lambda-1)}$$

Všetky vrstvy okrem vstupnej majú prídavnú prahovú hodnotu zvanú *bias* $\beta_v^{(\lambda)}$, prostredníctvom ktorej sa posúvajú hodnoty vstupného signálu. Transformáciu vstupného signálu na výstup zabezpečuje prenosová resp. transferová funkcia. V prípade predikcie sekundárnej štruktúry používajú všetky uzly rovnakú funkciu *f*, pomocou ktorej sa určuje výstupná hodnota nasledovne:

$$\sigma_v^{(\lambda)} = f\left[I_v^{(\lambda)} + \beta_v^{(\lambda)}\right]$$

Najčastejšie sa na transformáciu používa logistická funkcia zadaná ako:

$$f(x) = \frac{1}{1 + e^{-sx}}$$

Veličina *x* je vstupná hodnota posunutá o *bias* a *s* je konštanta, najčastejšie 1.

3.6.1 Predikcia prostredníctvom neurónovej siete

Sekvencia proteínu je v sieti reprezentovaná vstupnou vrstvou. Počet uzlov vstupnej vrstvy zodpovedá počtu residuí v okne sekvencie. Na každej pozícii sekvencie sa môže nachádzať ktorákoľvek aminokyselina, teda počet uzlov vstupnej vrstvy zodpovedá počtu 20 pre každý výrastok. Uzol korešpondujúci daným typom aminokyseliny má výstup 1, ostatné uzly majú výstup 0. Mnohé siete navyše používajú prídavný uzol nazývaný *spacer*. Tento uzol má na výstupe 1 v prípade absencie residua na skúmanej pozícii, ku ktorej prichádza na N a C koncoch. Tento prístup sa nazýva *ortogonálne kódovanie*.

Signál sa postupne šíry sieťou smerom od vstupnej vrstvy k výstupnej. Výstupná vrstva obvykle obsahuje toľko uzlov, koľko je predpovedaných štrukturálnych stavov. Počet stavov je najčastejšie 3 (helix, sheet, coil) a ich hodnota sa pohybuje v rozmedzí od 0 do 1. Výsledok predikcie získame použitím metódy *winner takes all*, teda vyberáme stav s najvyššou hodnotou signálu.

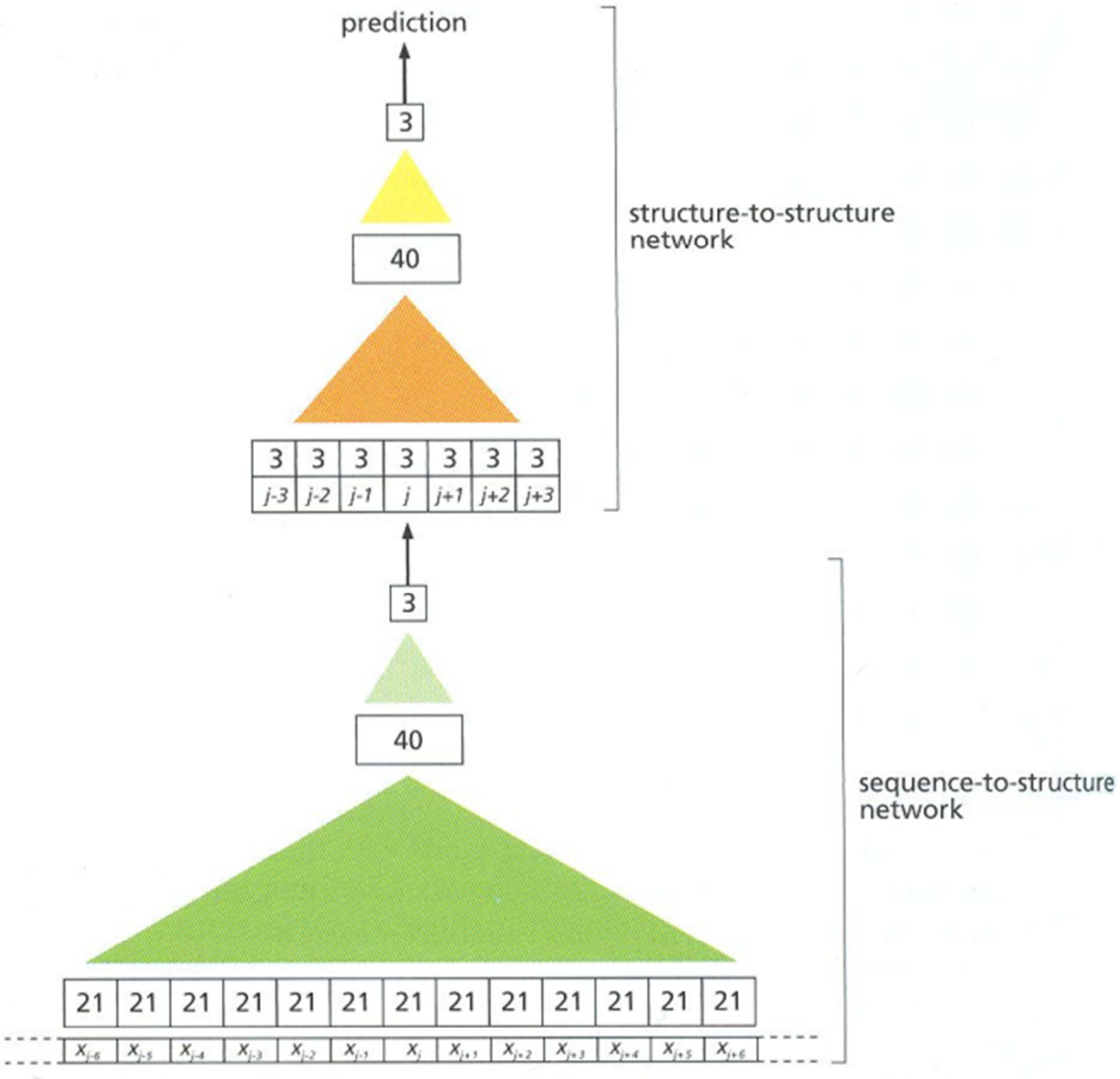
Sieť je trénovaná na množine sekvencií so známou sekundárnou štruktúru a váhy siete upravujeme metódou spätného šírenia chyby. Na nastavenie parametrov siete teda teoreticky postačuje zadať tieto priamočiare informácie.

3.6.2 Quian-Sejnowsky

Jednu z prvých aplikácií neurónových sietí na predikciu sekundárnej štruktúry zostavila táto dvojica. Vo svojom koncepte použili pomerne jednoduchú jedno vrstvovú doprednú sieť bez skrytých vrstiev. Sieť používala okno o veľkosti 13 residuí, teda vo vstupnej vrstve obsahovala 273 uzlov, ktoré boli priamo pripojené na vstup 3 uzlov vo výstupnej vrstve. V princípe takáto funkcionalita iba reprodukovala činnosť prvých metód GOR. Napriek tomu, že tento výpočtový mechanizmus dosiahol iba približne rovnakú úspešnosť ako metódy CF alebo GOR, vzniklo z nej množstvo modifikácií, ktorých výsledky boli porovnateľne lepšie.

Pridanie skrytých vrstiev neprinieslo očakávané zlepšenie úspešnosti, preto sa ďalší vývoj zameral na využitie štruktúry susedných residuí a jej vplyv na aktuálnu sekvenciu. Výstup pôvodnej siete, ktorá sa nazýva *sequence-to-structure* bol privedený na vstup ďalšej siete s názvom *structure-to-structure*. Jedná sa o dopredné plne prepojené siete s jednou skrytou vrstvou, obsahujúcou 40 skrytých uzlov. Táto architektúra obsahuje rozšírenie priebežnej predikcie pridaním siedmych pozícií na vstupnú vrstvu druhej siete, ktorá produkuje finálnu predikciu residua. Takáto architektúra sa ukázala ako úspešná a opäť vzniklo množstvo modifikácií.

Medzi najznámejšie patrí napríklad technika *output expansion*. Počet uzlov vo výstupnej vrstve prvej siete je stanovený na 9 a korešponduje s predikciou centrálneho residua a jemu priľahlých 2 residuí. Ďalším vylepšením bolo pridanie znalostí, ktoré silno korelujú so sekundárnou štruktúrou. Príkladom môže byť nahradenie uzlov vstupnej vrstvy jediným uzlom, ktorý obsahuje vhodne zakódovanú hodnotu propensity. Takáto drastická redukcia umožňuje jednoduchšie trénovanie siete, pretože sieť nemusí identifikovať kódovanie.



Obrázok 3.2: schematická reprezentácia architektúry používajúcej dve zreťazené siete. Farebné trojuholníky značia prepojenia uzlov, čísla v štvorcoch značia počet uzlov vo vrstvách. [2]

3.6.3 Riis-Krogh

Tejto dvojici sa podarilo zostaviť efektívne a účinné vylepšenie spomínanej architektúry. Ich metóda používa pre každú množinu 20 vstupných uzlov vo vstupnej vrstve 3 spoločné uzly vo vrstve skrytej. Celkový počet parametrov siete sa tak zredukuje na 63, z toho 60 váh a 3 hodnoty *bias*. Táto štruktúra sa následne opakuje pre všetky pozície okna s použitím rovnakých hodnôt všetkých 63 parametrov. Pre okno o veľkosti 13 residuí potom skrytá vrstva obsahuje celkovo 39 skrytých uzlov.

Technika používania rovnakých váh pre ekvivalentné časti siete sa nazýva *weight sharing* a bola natívne navrhnutá na predpovedanie štruktúry α-helix. Na predpovedanie jednotlivých štruktúrnych elementov boli použité oddelené siete, každá z nich používa odlišný spôsob trénovania a počet uzlov v skrytej vrstve. Výsledky z viacerých sietí sú potom priemerované a privedené na vstup *structure-to-structure* siete.

Spomeňme dva postupy, ako tento priemer vypočítať. Prvou možnosťou je použiť tzv. *jury decision*, kde sa z každej siete vypočíta aritmetický priemer predpokladaného štrukturálneho stavu a výsledný stav residua potom zodpovedá stavu s najvyšším priemerom. Alternatívne možno priemer

získať metódou *balloting probabilities*, ktorá používa rozdiel medzi dvoma najvyššími hodnotami signálov z uzlov výstupnej vrstvy druhej siete, s účelom určiť dôvernosť predikcie. Týmto spôsobom sa do výslednej predikcie vyberá iba podmnožina predikcií jednotlivých sietí s najvyššou hodnotou dôvery. Výsledná predikcia je potom vážený priemer tejto podmnožiny predikcií, kde váha je priemerná dôvernosť v rámci siete pre danú sekvenciu.

Heuristiky dodané použitím znalosti štruktúry homologických sekvencií boli použité podobne ako v metódach GOR V a ZPred, teda pre každú podobnú sekvenciu sa určí sekundárna štruktúra separovane a následne sa na báze zarovnaní jednotlivé výsledky priemerujú do výslednej predikcie. Záverečným krokom je aplikácia filtra, ktorým je v tomto prípade 3 vrstvová sieť používajúca okno o dĺžke 15. Presnosť predikcie Q_3 je použitím týchto techník približne 71%, čo je približne o 5% viac, než pri predchádzajúcich metódach.

3.6.4 PHDsec

Táto metóda na rozdiel od predchádzajúcich používa pre každú pozíciu 32 uzlov vo vstupnej vrstve pri použití okna štandardnej dĺžky 13 residuí. Z toho 20 popisuje aminokyselinové zloženie, 4 udávajú približnú dĺžku sekvencie a 8 indikuje vzdialenosť aktuálneho okna od N prípadne C konca. Dĺžka sekvencie sa pri indikácii pomocou 0 / 1 nachádza v rozmedzí ≤ 60, ≤ 120, 240 alebo > 240.

Informácie z homologických sekvencií získava táto metóda odlišným spôsobom a to mnohonásobným zarovnaním sekvencií, ktorý modifikuje signály vo vstupnej vrstve. Výrastky na všetkých pozíciách sekvencie sú reprezentované frekvenciou ich výskytu. Dosiahnutá úspešnosť Q_3 je priemerne 72%.

3.6.5 PSIPRED

Ako zdroj informácií z homologických sekvencií používa skórovaciu maticu PSSM metódy PSI-BLAST. Použitím skóre sa do výsledku dostávajú rovnako pozitívne aj negatívne hodnoty, ktoré je potrebné konvertovať do podoby signálu vstupnej vrstvy. Na tento účel používa logistickú funkciu, kde člen e^{-x} je nahradený členom e^{-mx} s konštantou m. Úspešnosť Q_3 dosiahnutá týmto spôsobom je približne 75% a skórovacia matica PSSM je v súčasnosti považovaná za optimálny spôsob vloženia homologických sekvencií do predikcie sekundárnej štruktúry.

3.6.6 DESTRUCT

Táto metóda využíva iteratívnu architektúru zvanú *kaskádová korelácia*. Okrem predikcie troch štrukturálnych stavov predpovedá aj veľkosť torzného uhla ψ, ktorý slúži na rozlišovanie medzi α a β štruktúrami. Používa okno o veľkosti 15 residuí a pomocou dvoch oddelených sietí s jednou skrytou vrstvou vytvára predbežnú predikciu štrukturálneho stavu a torzného uhla centrálneho residua. Tieto hodnoty sú privedené na vstup iteratívnej siete, ktorá v 3 iteráciách vytvára finálnu predikciu. Trénovanie siete realizuje výpočtom druhej derivácie a metódou kontrolovaného pridávania uzlov do skrytej vrstvy. Tieto techniky umožnili zvýšenie úspešnosti Q_3 na hodnotu 80%.

3.7 Konsenzuálne metódy

Existujú metódy, ktoré implicitne nezavádzaj nové techniky predikcie sekundárnej štruktúry, ale pracujú s už existujúcimi metódami. Na základe výsledkov získaných z iných metód potom predpovedá štruktúru na základe konsenzu, teda žiadna z použitých metód nie je zanedbaná a priamo ovplyvňuje celkový výsledok. Štrukturálny stav s najčastejším priradením sa stáva finálnym.

3.7.1 NPS

Patrí medzi typických zástupcov metód využívajúcich konsenzus. Radí sa medzi staršie techniky a pracuje s dovtedy známymi technikami. Preto tu nájdeme aj dnes už takmer nepoužívané metódy s úspešnosťou nižšou ako 70%, ako napríklad GOR, SIMPA96, PHD a mnohé iné, ktoré nie sú v tejto práce spomínané. V prípade, že výsledok predikcie je na základe konsenzu nerozhodný, ako referenčný sa použije výsledok metódy PHD.

4 Štúdia dostupných nástrojov

Aby bolo možné predikciu sekundárnej štruktúry získať v reálnom čase, sú teoretické koncepty jednotlivých metód realizované prostredníctvom výpočtových nástrojov. Takýchto nástrojov je dostupných viacero, vzhľadom na unikátnosť použitej technológie a algoritmických postupov ich v súčasnosti existuje približne 60. Mnohé z nich využívajú nielen konvenčné postupy spomínané v predchádzajúcej kapitole, ale poskytujú odlišný spôsob reprezentácie dát a rozdielne poradie aplikácie konvenčných metód. Dôležité je spomenúť, že jednotlivé nástroje sú špecializované a okrem predikcie sekundárnej štruktúry poskytujú aj rôzne ďalšie informácie získané počas výpočtu, napr. zoznam homologických sekvencií alebo dodatočné znalosti o proteíne v prípade, že bol nájdený v referenčnej databáze, prípadne poskytujú inú prídavnú znalosť. Nás však v tejto štúdii budú zaujímať výlučne informácie týkajúce sa predikcie sekundárnej štruktúry.

Nástroje bývajú spravidla realizované formou webových aplikácií, ktoré sú obvyklým spôsobom prezentácie metód z oblasti bioinformatiky. Výhodou týchto aplikácií je predovšetkým fakt, že výpočty sú často náročné a zdĺhavé, preto je využitie vzdialených serverov efektívne z hľadiska použitia výpočtových zdrojov. Detailná technológia aplikácií je známa iba výnimočne a programy pracujú ako *black box*, z ktorého užívateľ získava výstupy na základe zadaných vstupov. Najväčšou prednosťou používania webových aplikácií je ich dostupnosť priamo v okne prehliadača bez nutnosti inštalácie ďalšieho softwaru, intuitívne a rýchlo osvojiteľné ovládanie, jednoduchá manipulácia s dátami a možnosť zdieľania výsledkov prostredníctvom URL.

4.1 Analýza vybraných nástrojov

V tejto časti sú predstavené vybrané webové aplikácie predikcie sekundárnej štruktúry. V prvej fáze návrhu bolo potrebné jednotlivé nástroje dôkladne preskúmať a otestovať. Na testovanie bola použitá pomerne jednoduchá množina sekvencií, ktoré boli jednak náhodné ale aj také, ktorých sekundárna štruktúra je známa. Z väčšieho množstva bolo nakoniec vybraných 6 najlepších. Konkrétne sa jedná o nástroje PCI-SS, NetSurfP, PSIPRED, PSSPred, Jpred a SymPred. Hlavné kritériá výberu boli vysoká presnosť predikcie, možnosť pristupovať k nástroju aspoň z časti vzdialene a v neposlednom rade dostupnosť a prehľadnosť výsledku.

Užívateľské rozhranie jednotlivých nástrojov obsahuje na úvodnej podstránke jednoduchý webový formulár, do ktorého užívateľ zadá vstupnú sekvenciu aminokyselín spolu s potrebnými dátami. Každej predikčnej úlohe je vygenerovaný jednoznačný identifikátor, prostredníctvom ktorého možno k úlohe pristupovať aj dodatočne. Po odoslaní formuláru sa spustí samotný výpočet a užívateľ je presmerovaný na podstránku, kde môže kontrolovať aktuálny stav predikcie. Po skončení výpočtu sa zobrazí podstránka obsahujúca formátovaný výstup.

Jednotlivé nástroje sa líšia jednak v množstve a povahe informácií, ktoré musí užívateľ zadať na vstup a spôsobe, akým sú podstránky presmerované, ale najmä vo výstupnom formáte predikcie. Ten je pre každý nástroj odlišný, treba mu preto venovať zvýšenú pozornosť. Dĺžka sekvencie určenej na analýzu je často obmedzená a pohybuje sa v rozmedzí minimálne 30-40 znakov a maximálne 1000-4000 znakov. Detailný rozbor jednotlivých nástrojoch je spolu s krátkym popisom a hodnotou úspešnosti predikcie Q_3 je pre zvýšenie prehľadnosti realizovaný v nasledovných bodoch:

- Vstup – vstupné polia formuláru, ktoré musí užívateľ zadať, aby mohol formulár odoslať
- Komunikácia – spôsob vzdialeného ovládania nástroja bez nutnosti interakcie užívateľa
- Výstup – dostupnosť výsledku a formát výstupných dát

Niektoré nástroje poskytujú viacero úrovní výstupov. Budeme voliť výstup obsahujúci komplexné informácie, ktoré možno ďalej využiť vo vlastnej predikcii. Často požadovaným vstupom je email, na ktorý je informácia o úspešnom ukončení výpočtu zaslaná užívateľovi spolu s výsledkami predikcie. Ďalším spoločným vstupom býva názov sekvencie. Niektoré nástroje tento názov používajú v rámci FASTA výstupu alebo je priamo zakomponovaný v identifikátore predikcie, prípadne je úplným identifikátorom.

K nástrojom možno s ohľadom na automatické spracovanie pristupovať v podstate troma rôznymi prístupmi. U niektorých nástrojov je pre akademické alebo vývojárske účely možné získať lokálnu kópiu nástroja, tzv. *portable* verziu, ktorá obsahuje spustiteľný súbor pre operačný systém UNIX. Pre stiahnutie je potrebné sa zaregistrovať, prípadne tvorcov oboznámiť s účelom, na ktorý bude program použitý. V rámci tejto aplikácie je však použitie lokálnych kópií spojené s viacerými ťažkosťami. Hlavnou nevýhodou je fakt, že použitie takýchto súborov na serveri vyžaduje povolenie štandardne zablokovaných funkcií umožňujúcich spúšťanie skriptov tretích strán v rámci terminálu. Takéto nastavenie je z pohľadu bezpečnosti veľmi riskantné a teda nepoužiteľné. Tieto lokálne verzie ďalej využívajú lokálne databázy neopakujúcich sa sekvencií, tzv. *non-redundant sequence database*, ktorých veľkosť je enormná a po rozbalení zaberá približne 5 GB miesta na disku. Napriek tomu, že *portable* verzie môžu uľahčiť prácu s jednotlivými nástrojmi, pre potreby servera sú neefektívne a ďalej sa im teda nebudeme venovať.

4.1.1 Stručný úvod do webových služieb

Druhou možnosťou je použiť webové služby tzv. *web services*, ktoré sú dostupné pre 2 z vybraných nástrojov. Webovú službu možno stručne charakterizovať ako prostriedok komunikácie dvoch zariadení prostredníctvom siete, v tomto prípade po internete. Jedno zo zariadení vystupuje ako server, druhé ako klient. Spôsob komunikácie klienta so serverom je realizovaný prostredníctvom API[5] a rozoznávame nasledujúce dva typy implementácie:

- REST (*Representational State Transfer)* – jednoduchšia forma komunikácie založená na základných metódach protokolu HTTP, ktorými sú GET, POST a iné. K premenným a metódam možno pristupovať priamo prostredníctvom URI protokolu a prenos dát je napr. vo formáte *Json*.
- SOAP (*Simple Object Access Protocol)* – komunikácia založená na použití správ jazyka XML. Tieto systémy poskytujú popis dostupných metód, resp. operácií v jazyku WSDL *(Web Services Description Language).* Webovú službu možno ovládať volaním týchto metód, teda operácií, s vhodnými parametrami. Jedná sa prevažne o dátové operácie typu *get* a *set*.

Detailnejšie informácie ohľadne webových služieb, technológií REST, SOAP a WSDL možno nájsť v literatúre. [4]

Spomínané nástroje predikcie sekundárnej štruktúry poskytujú SOAP WSDL server spolu s API, ktorým možno nástroj vzdialene ovládať. Nespornou výhodou použitia SOAP je nezávislosť na použitej verzii špecifikácie SOAP a použitom programovacom jazyku. Na vytvorenie klienta teda možno použiť ľubovoľný *framework* podporujúci SOAP, pomocou ktorého je implementácia v podstate triviálna.

[5] Application programming interface

V štúdii jednotlivých nástrojov som sa teda zameral na použitie webových služieb a spôsob, akým komunikácia klienta so serverom prebieha. Napriek tomu, že niektoré nástroje implicitne webové služby neponúkajú, pri správnej manipulácii s dátami možno vytvoriť simuláciu takéhoto klienta. Z pohľadu webových služieb je spôsob akým jednotlivé nástroje pracujú veľmi podobný, preto je možné vytvoriť pomerne univerzálny systém na komunikáciu priamo s webovými formulármi a spracovať výsledky na úrovni jazyka HTML.

4.1.2 PCI-SS

Tento nástroj je založený na použití metódy PCI *(Parallel Cascade Identification)*, ktorá slúži na identifikáciu nelineárnych systémov. Na vstup PCI modelu sú privádzané dynamické nelineárne dáta vytvorené z rôznych profilov PSI-BLAST. Model pracuje ako *black box*, pričom jeho úlohou je analýza tvaru molekuly. Na optimalizáciu parametrov modelu sa používajú genetické algoritmy. Výstup modelu je v kombinácii s výsledkami metódy PSIPRED privádzaný na vstup *sequence-to-structure* neurónovej siete. Dostupné na [5].

- **Vstup:** názov sekvencie, sekvencia.
- **Komunikácia:** realizovaná pomocou dostupného SOAP API, ktoré je pomerne jednoduché a obsahuje iba jedinú metódu **getPrediction(string seqName, string seq)**. Návratové hodnoty tejto metódy sú buď XML dokument obsahujúci výsledok predikcie alebo vracia chybu neplatných dát.
- **Výstup:** výstupný XML dokument sa skladá z postupnosti elementov *record* s atribútom poradového čísla residua *position*. Každý *record* obsahuje ďalej vstupný znak *aminoAcid*, výsledok predikcie *prediction* a prídavné dáta v podobe tzv. vzdialenosti jednotlivých štrukturálnych stavov *hDistance*, *eDistance*, *tDistance*. Štrukturálny stav s najmenšou vzdialenosťou je považovaný za najpravdepodobnejší a je použitý ako výsledok predikcie. Metóda používa trojstavový model a štrukturálne stavy sú *H* (helix), *E* (sheet) a *other*.

4.1.3 NetSurfP

Nástrojov využívajúci neurónové siete. Sieť je natrénovaná množinou experimentálne zvolených proteínov, ktorým je už v dobe trénovania priradené skóre podobnosti. Za účelom vytvorenia skóre používa techniku Z-score. Samotná predikcia prebieha tak, že vstupná sekvencia je najprv zarovnaná s podobnými sekvenciami využitím techniky BLAST, a na toto zarovnanie je aplikovaný algoritmus predikcie. Okrem predikcie sekundárnej štruktúry poskytuje nástroj informácie o povrchovej prístupnosti molekuly, tzv. *surface accessibility*. Úspešnosť Q_3 je stanovená na 79%. Dostupné na [6].

- **Vstup:** sekvencia.
- **Komunikácia:** dostupné nie sú žiadne webové služby ani možnosti vzdialeného prístupu. Pre odoslanie webového formuláru je potrebné zahrnúť konfiguračný súbor prostredníctvom URL. Formulár je po odoslaní automaticky presmerovaný na podstránku, ktorá odkazuje na podstránku s výsledkom. Komunikácia je preto riadená pomocou navrhnutého komunikačného protokolu.
- **Výstup:** výstupné hodnoty sú zoradené do stĺpcov a pre nás zaujímavé sú položky vstupnej sekvencie *aminoAcid*, poradového čísla znaku *aminoAcidNumber* a pravdepodobnosť jednotlivých štrukturálnych stavov *probH*, *probE* a *probC*. Stav s najvyššou hodnotou pravdepodobnosti je vybraný ako výsledný. Metóda používa trojstavový model a možné štrukturálne stavy sú *H* (helix), *E* (strand) a *C* (coil).

4.1.4 PSIPRED

Je typickým zástupcom metód využívajúcich neurónové siete. Jedná sa o obľúbený, rozšírený a pravidelne aktualizovaný výpočtový nástroj. Použitím striktnej validácie výstupných dát možno použitím tohto nástroja dosiahnuť úspešnosť Q_3 až 82%, čím sa radí medzi aktuálne najlepšie nástroje. Dostupné na [7].

- **Vstup:** email, názov sekvencie, sekvencia.
- **Komunikácia:** prostredníctvom SOAP API pozostávajúceho z dvoch metód:
  - **submit(string sequence, string email, string name)**: metóda, ktorá slúži na zaslanie žiadosti na server. Okrem 3 povinných parametrov akceptuje metóda voliteľné parametre slúžiace na prídavné výpočty. Server odpovedá na túto žiadosť prostredníctvom objektu, ktorý obsahuje položky *message* **–** správa o úspechu alebo zlyhaní odoslania, *job_id* - identifikátor úlohy na serveri a *state* – stav odoslania, kde **0** značí zlyhanie, **1** úspech.
  - **result(string job_id)**: požiadavka na získanie výsledku predikcie, pričom jediným parametrom je id predikčnej úlohy. Odporúčaná perióda opakovania volania tejto metódy je približne 5 minút, pretože časté opakujúce sa dotazy na server môžu byť považované za útok. Server odpovedá opäť objektom, ktorý obsahuje *message* – textová správa o dostupnosti výsledku, *job_id* – id predikčnej úlohy, *state* – aktuálny stav predikcie, kde **0** je chyba pri výpočte, **1** značí úspešné ukončenie predikcie, v prípade **2** sa úloha stále vykonáva a **3** signalizuje neexistujúce id. V prípade úspechu je súčasťou objektu premenná *result*, ktorej hodnota je URL výstupného súboru s výsledkom.
- **Výstup:** dokument výsledku je formátovaný do štruktúry podobnej FASTA formátu. Po úvodných textoch nasleduje trojica riadkov, ktoré postupne nesú informácie o vstupnej sekvencii, výsledku predikcie a hodnotu miery dôvery tzv. *confidence*, teda spoľahlivosti predikcie pre danú pozíciu. Dĺžka znakov v jednom riadku je stanovená na 100. Metóda používa trojstavový model a možné štrukturálne stavy sú *H* (helix), *E* (sheet) a *C* (coil).

4.1.5 PSSpred

Je ďalším nástrojom, ktorý využíva neurónové siete. Viacnásobné zarovnania získava prostredníctvom PSI-BLAST. Frekvencie výskytu aminokyselín sú vážené *Henikoff* váhami a jednotlivo použité na trénovanie siete metódou *Rumelhart error backpropagation*. Predikcia je výsledkom kombinácie predikcií zo 7 neurónových sietí s odlišnými nastaveniami a parametrami. Dostupné na [8].

- **Vstup:** email, sekvencia, názov sekvencie.
- **Komunikácia:** dostupné nie sú žiadne webové služby ani možnosti vzdialeného prístupu. Pre odoslanie webového formuláru je potrebné vyplniť všetky spomínané vstupné polia. Po úspešnom odoslaní je nástroj automaticky presmerovaný na podstránku, ktorá odkazuje na podstránku s výsledkom. Podstránka s výsledkom obsahuje buď hlásenie o stave predikcie alebo samotný dokument s výsledkom. Komunikácia je riadená pomocou navrhnutého komunikačného protokolu.
- **Výstup:** výstup je vo formáte, ktorý je dostatočne univerzálny a v implementácii je použitý ako referenčný. Na začiatku a konci riadka obsahuje poradové číslo, ďalej vstupnú sekvenciu, výsledok predikcie a mieru dôvery v daný výsledok v rozsahu 0 až 9. Metóda používa trojstavový model a možné štrukturálne stavy sú *H* (helix), *E* (strand) a *C* (coil).

4.1.6 Jpred

Tento nástroj, pretože patrí medzi najpoužívanejšie a najobľúbenejšie nástroje vôbec. Tvorcovia zverejnenie zdrojového kódu stále zvažujú, teda neexistuje implicitný spôsob ako nástroj použiť bez využitia webového formulára. Princíp metódy je vo využití neurónovej siete s názvom *Jnet*. Homologické sekvencie získava opäť z PSI-BLAST, z ktorých následne vyfiltruje tie s najvyššou podobnosťou. Sekvencie následne vstupujú do siete *Jnet*. Okrem predikcie sekundárnej štruktúry nástroj poskytuje detailnú analýzu *coiled* oblastí proteínov. Pre analýzu a editáciu výsledku nástroj ponúka vlastnú Java aplikáciu s názvom *Jalview*. Dostupné na [9].

- **Vstup:** sekvencia.
- **Komunikácia:** dostupné nie sú žiadne webové služby ani možnosti vzdialeného prístupu. Po odoslaní formuláru sú dostupné 2 možnosti odpovede servera. Prvou z nich je zhoda analyzovanej sekvencie s určitými sekvenciami v databáze *PDB*. Nástroj je presmerovaný na túto podstránku a užívateľ sa môže zvoliť zobrazenie výsledkov zhody, alebo môže v predikčnej úlohe pokračovať. Jedná sa v podstate o odoslanie rovnakého formulára, v ktorom sa nastaví hodnota skrytého vstupu *pdb* na 1. Následne je nástroj presmerovaný na podstránku zobrazujúcu výsledok formou *progress baru*. Po dokončení úlohy je k dispozícii dostupných hneď niekoľko spôsobov zobrazenia výsledku, pričom ideálna voľba je zvoliť komplexné informácie, tzv. *full HTML view*.
- **Výstup:** výstup dokument obsahuje v prípade zhody s *PDB* odkazy na homologické sekvencie. Ďalej výstup ponúka pomerne širokú škálu informácií, z ktorých s podstatné vstupná sekvencia *orgiSeq*, výsledok predikcie *jnet* a miera dôvery v daný výsledok *jnetRel* v rozsahu 0 až 9. Metóda používa trojstavový model a možné štrukturálne stavy sú *H* (helix), *E* (strand) a niekoľko typov *"-", c, C* (coil) vzhľadom na vierohodnosť výsledku.

4.1.7 SymPred

Táto metóda reprezentuje metódu nearest-neighbor a rozhodol som sa juj použiť pre jej unikátny prístup. Na zvýšenie presnosti predikcie používa znalosť synoným jazyka, aplikovanú na podobnosť krátkych segmentov proteínov. Na postupnosť aminokyselín sa pozerá ako na jazyk nad abecedou 20 unikátnych symbolov a vstupnú sekvenciu berie ako neznámy text v tomto jazyku. Pomocou BLAST vytvorí derivácie skúmanej sekvencie, ktoré spolu so skúmanou sekvenciou ukladá ako synonymický slovník. Na synonymá sa následne pozerá z hľadiska miery podobnosti a frekvencie výskytu v slovníku. Slovník sa počtom používaní zväčšuje a na porovnávanie oproti slovám v slovníku používa vlastnú skórovaciu funkciu. Týmto inovatívnym prístupom dosahuje hodnotu Q_3 81% a v kombinácii s výstupom PSIPRED až 84%. Dostupné na [10].

- **Vstup:** sekvencia, názov sekvencie.
- **Komunikácia:** dostupné nie sú žiadne webové služby ani možnosti vzdialeného prístupu. Po odoslaní formulára je nástroj presmerovaný na podstránku s odkazom na výsledok predikcie. V prípade nedostupnosti výsledku vracia neexistujúcu podstránku, inak vracia dokument s výsledkom.
- **Výstup:** výstupný formát je štruktúrovaný na 60 položiek v jednom riadku. Výstup sa skladá z postupnosti trojíc riadkov v poradí: vstupná sekvencia, výsledok predikcie a miery dôvery v daný výsledok v rozsahu 0 až 9. Metóda používa trojstavový model a možné štrukturálne stavy sú *H* (helix), *E* (sheet) a niekoľko typov *C* (coil) vzhľadom na vierohodnosť výsledku.

4.2 Zhodnotenie výsledkov štúdie

Po analýze dostupných nástrojov predikcie sekundárnej štruktúry a následnom výbere vhodných nástrojov použiteľných pre zhotovenie servera možno konštatovať nasledovné závery.

Nástroje sú vyvíjané skupinami bioinformatikov na univerzitách po celom svete, pričom medzi najzastúpenejšie krajiny patria USA a Francúzsko. Všetky nástroje pracujú ako autonómne webové aplikácie, na ktorých je predikcia realizovaná vo forme webových formulárov. Užívateľ zadá povinné a prípadne voliteľné informácie, ktorými sú najčastejšie email, názov výstupu a samotná sekvencia a po odoslaní formuláru sa na pozadí spustí samotný výpočet a užívateľ je presmerovaný na podstránku s výsledkom. Výsledok predikcie je v závislosti na vyťaženosti servera dostupný do niekoľkých minút, najviac však do dvoch hodín.

Kľúčovou znalosťou vyplývajúcou z tejto štúdie je možnosť využitia webových služieb, pomocou ktorých možno k jednotlivým nástrojom pristupovať vzdialene. Napriek tomu, že iba veľmi malá časť nástrojov poskytuje možnosť takejto komunikácie, z dostupných služieb možno výborne pochopiť spôsob akým nástroje pracujú a následne navrhnúť univerzálny komunikačný protokol.

Po vložení vstupných dát a odoslaní formuláru je pre každý nástroj rozhodujúca znalosť identifikátora predikčnej úlohy. Toto ID je pre každý nástroj špecifické, avšak vždy sa jedná o alfanumerickú postupnosť znakov rôznej dĺžky, v niektorých prípadoch oddelenú oddeľovačom. Spoločnou črtou všetkých nástrojov je výskyt tohto ID na podstránke, na ktorú je formulár presmerovaný po odoslaní. Jednou z najdôležitejších častí celej implementácie je teda ID z príslušnej podstránky získať. Znalosť ID predikčnej úlohy následne umožňuje sledovať stav predikcie a pristupovať k výsledkom prostredníctvom URL.

Výstupný formát jednotlivých nástrojov ja často diametrálne odlišný. Niektoré nástroje poskytujú výsledok v XML alebo inom štruktúrovanom dokumente, iné výsledok zobrazujú priamo na podstránke s výsledkom. Tu je výsledkom zobrazený vertikálne, kde každý stĺpec značí jednu zobrazovanú veličinu, alebo horizontálne. Horizontálne sú veličiny formátované do jednej n-tice riadkov, ktorých dĺžka zodpovedá dĺžke vstupu (napr. trojica riadkov *input*, *output*, *confidence* dĺžky vstupu 437 znakov), alebo opakujúcich sa n-tíc pevnej dĺžky oddelených prázdnym riadkom (napr. pätica riadkov *input*, *output*, *probH*, *probE*, *probC* dĺžky 352 znakov zarovnaná na 100 znakov na riadok). Z toho plynie nutný individuálny prístup k výsledkom jednotlivých nástrojov.

V prípade záujmu možno prehľad nástrojov, ktoré poskytujú webové služby, nájsť na stránkach *BioCatalogue* [11]. Neoficiálne zoznamy nástrojov predikcie sekundárnej štruktúry možno nájsť napríklad na stránkach *ExPASy* [12] alebo *molbiol* [13].

5 Špecifikácia servera

V tejto kapitole sú popísané prevažne skutočnosti, ktoré je nevyhnuté špecifikovať pred samotnou implementáciou. Text sa skladá z motivácie návrhu, stanovenia hlavných cieľov projektu a popisu použitých programových prostriedkov.

5.1 Motivácia návrhu

Cieľovou skupinou navrhovanej aplikácie sú prevažne osoby pracujúce v oblasti bioinformatiky, biológovia a študenti, teda osoby, pre ktoré je znalosť sekundárnej štruktúry proteínov istým spôsobom zaujímavá. Cieľom projektu je zjednodušiť a urýchliť prácu týchto osôb, prehľadne zobraziť výsledky a ponúknuť vlastnú predikciu sekundárnej štruktúry zostavenú nad výsledkami zvolených nástrojov.

V súčasnosti tieto osoby pristupujú k jednotlivým nástrojom autonómne. Užívateľ, ktorý si nechá požadovanú sekvenciu zanalyzovať niekoľkými dostupnými nástrojmi môže na tento úkon vynaložiť množstvo času a energie, pretože nástroje spúšťa postupne. Porovnanie výsledkov musí navyše realizovať svojpomocne. Výsledok je užívateľovi zobrazený formou natívnou pre daný nástroj, ktorá je takmer pre každý nástroj odlišná.

5.1.1 Ciele projektu

- Extrahovať spoločné rysy vybraných nástrojov a vytvoriť jednoduché a funkčné užívateľské prostredie, ktoré minimalizuje interakciu užívateľa s nástrojom.
- Navrhnúť a implementovať univerzálny komunikačný protokol, ktorý pre každý zo zvolených nástrojov umožňuje zadať predikčnú úlohu, otestovať dostupnosť výsledku, výsledok spracovať a vhodným spôsobom uložiť.
- Takto získané dáta prehľadne zobraziť užívateľovi v unifikovanom formáte.
- Nad výsledkami jednotlivých nástrojov vytvoriť vlastnú predikciu, založenú na použití získaných heuristík.

5.2 Použité technológie

Keďže sa jedná o webovú aplikáciu typu klient–server, pre implementáciu nástroja som sa rozhodol použiť *framework* programovacieho jazyka PHP s názvom *CodeIgniter* [14] (ďalej len CI). Pre tento *framework* so sa rozhodol z dôvodu, že s ním mám predchádzajúce skúsenosti pri tvorbe webu. Medzi prednosti CI patrí množstvo integrovaných knižníc a jednoduchá manipulácia s nimi, intuitívna konfigurácia aplikácie, vysoká miera bezpečnosti a iné. Rovnako ako iné PHP *frameworky* pracuje CI s dizajnom *Model-View-Controller* (ďalej len MVC), ktorého základnou ideou je separovať logickú časť aplikácie od časti užívateľskej. Architektúra pozostáva z nasledujúcich troch vrstiev:

- **Model** – reprezentuje jadro aplikácie. V tejto vrstve prebiehajú všetky výpočty, dátové operácie, práca s databázou a pod.
- **View** – vrstva zabezpečujúca interakciu s užívateľom, ktorý zadáva vstupy a pozoruje výstupy. S pravidla býva realizovaná pomocou *HTML*, *CSS* a *Javascriptu*.
- **Controller** – zabezpečuje presun dát od užívateľa do modelu a naopak, výstup z modelu prezentuje užívateľovi.

6 Implementácia

Po dôkladnej teoretickej štúdii, analýze dostupných nástrojov, stanovení cieľov a výbere technológie prichádza na rad samotná realizácia servera. Na rozdiel od špecifikácie popísanej v semestrálnom projekte je v tejto časti detailne popísaný *workflow* celého projektu. Tvorba aplikácie prebiehala v nasledovných fázach, ktoré sú analyzované v rámci tejto kapitoly:

1. Konfigurácia a testovanie štartovacieho balíka CI.
2. Extrakcia dát z jednotlivých nástrojov.
3. Zapracovanie nástrojov do aplikácie.
4. Nad dátami z jednotlivých nástrojov zostaviť vlastnú predikciu.
5. Tvorba užívateľského rozhrania.
6. Formátovanie a zobrazenie výsledkov užívateľovi.
7. Validácia vstupných dát.
8. Testovanie a optimalizácia

6.1 Konfigurácia CI

Štartovací balík CI je dostupný na odkaze uvedenom v literatúre [14]. Na úvod je však potrebná iniciálna konfigurácia, ktorá zabezpečí korektné smerovanie podstránok v celom projekte a pripojenie databázy. Kompletný návod, ako CI nastaviť je dostupný v prílohe č. 1.

6.2 Extrakcia dát z vybraných nástrojov

Rozhodujúcou časťou celého projektu je postupne implementovať vybrané nástroje. Zo štúdie vyplýva, že k nástrojom je nutné pristupovať individuálne, no napriek tomu je možné vypozorovať ich spoločné rysy a vytvoriť tak univerzálny komunikačný protokol.

Komunikačný protokol je založený na použití webových služieb, ktoré sú implicitne dostupné iba pre 2 zo 6 vybraných nástrojov. Pre potreby týchto nástrojov je implementovaný SOAP klient. Pre ostatné nástroje je potrebné vytvoriť klienta tak, aby pracoval podobným spôsobom, ako keby komunikuje so serverom webovej služby. Pre tieto účely je vytvorený klient používajúci techniku cURL.

6.2.1 Implementácia SOAP klienta

V jazyku PHP je dostupných hneď niekoľko rozšírení SOAP. Po štúdii dostupných možností som sa rozhodol použiť *open source* knižnicu NuSOAP, dostupnú na [15]. Pre použitie knižnice v CI je potrebné v adresári *libraries* vytvoriť novú triedu, ktorá obsahuje volanie štandardnej knižnice. Takto pripravenú knižnicu už možno jednoducho použiť na vytvorenie klienta vložením URL k príslušnému WSDL súboru. Odkaz na kompletný manuál ku knižnici SOAP možno nájsť v literatúre. [16]

Na vytvorenie klienta server reaguje návratom objektu, ktorého operácie umožňujú manipuláciu s premennými nástroja. Funkčnosť klienta si možno jednoducho overiť volaním metódy *__getFunctions()*, ktorá vracia zoznam dostupných operácií s príslušnými parametrami a návratovou hodnotou.

6.2.2 Implementácia cURL klienta

Pre nástroje, ktoré webové služby neposkytujú je alternatívou použitie štandardnej knižnice PHP s názvom cURL (*client URL*), ktorá slúži na pripojenie a komunikáciu so rôznymi servermi a podporuje širokú škálu protokolov a metód. Pri volaní cURL s povinným parametrom *CURLOPT_URL* funkcia vráti obsah stránky na danom odkaze rovnakým spôsobom, akým sa dá zdrojový kód stránky zobraziť v prehliadači. Odkaz na kompletný manuál ku knižnici cURL je dostupný v literatúre. [17]

Pre každý nástroj je funkcia cURL zapracovaná 2 spôsobmi. Obsah vrátených podstránok je nutné starostlivo analyzovať s cieľom separovať pozitívnu a negatívnu odpoveď servera. Pre každý nástroj a každú použitú podstránku je preto potrebné vybrať referenčný text nachádzajúci sa unikátne iba na skúmanej podstránke. Prvý raz je funkcia použitá na odosielanie vstupného formulára, nastavením parametra *CURLOP_POST* na *TRUE*. Funkcia následne očakáva parameter *CURLOPT_POSTFIELDS*, ktorého hodnotou je pole, prípadne objekt, obsahujúci vstupné polia formulára s príslušnými hodnotami. V tomto prípade volíme URL, ktorá zodpovedá hodnote parametru *action* vstupného formulára, teda odkaz kam formulár smeruje. Text nachádzajúci sa na podstránke signalizuje buď úspešné prijatie úlohy, alebo zlyhanie, najčastejšie z dôvodu neplatných dát. V prípade pozitívneho výsledku sa na podstránke nachádza ID predikčnej úlohy. Rozhodujúcim krokom spracovania podstránky je získanie ID, ktoré je esenciálnou znalosťou ďalšej komunikácie s nástrojom. Druhý raz slúži funkcia na testovanie dostupnosti výsledku úlohy s príslušným ID. Textový obsah podstránky je opäť potrebné analyzovať a zistiť, či je výpočet dokončený alebo stále prebieha. V prípade dostupnosti výsledku si výsledok uložíme, v opačnom prípade otestujeme dostupnosť výsledku neskôr.

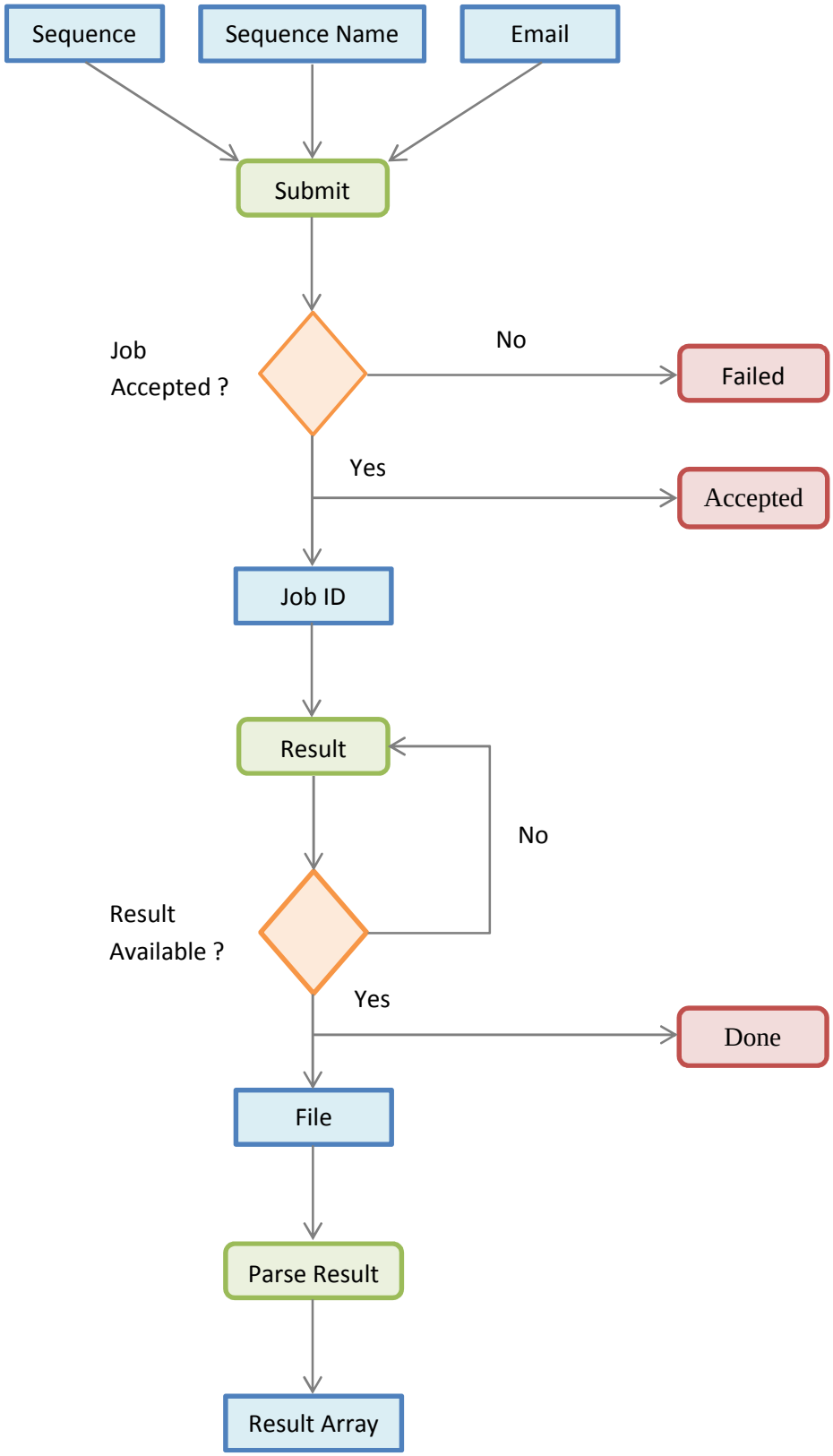
6.2.3 Komunikačný protokol

Napriek tomu, že vybrané nástroje používajú vytvorených SOAP alebo cURL klientov, každý z nástrojov je jedinečný a jeho implementácia zahŕňa drobné odchýlky. Práve táto skutočnosť vedie k vytvoreniu samostatných modelov pre jednotlivé nástroje. Komunikácia vytvorených klientov je zapracovaná do nasledovných troch metód:

- **Submit:** zadávanie úlohy na server. Vo výslednom texte je vyhľadaný reťazec signalizujúci úspešné spracovanie požiadavky. V prípade úspechu je z obsahu vyextrahované ID úlohy.
    - Parametre: sekvencia (povinný), názov (voliteľný) sekvencie, email (voliteľný).
    - Návratová hodnota: ID úlohy, v prípade zlyhania prázdny reťazec.
- **Result:** kontrola dostupnosti výsledku úlohy. Obsah výsledku je prehľadávaný na prítomnosť reťazca o dokončení výpočtu. Výsledok predikcie je dočasne zapísaný do textového súboru.
    - Parametre: ID úlohy (povinný).
    - Návratová hodnota: 1 v prípade úspechu, 0 inak.
- **ParseResult:** analýza textového súboru s cieľom získať výsledok predikcie. Zo súboru je najprv odstránených niekoľko riadkov zhora a zdola. V Závislosti na formáte výstupu je výsledok postupne spracovaný a vhodným spôsobom uložený.
    - Parametre: názov súboru (povinný).
    - Návratová hodnota: pole výsledných hodnôt.

Vzhľadom na odpoveď servera sú pre stav úlohy definované nasledovné tri stavy:

- **Accepted:** úloha úspešne prijatá, prebieha výpočet.
- **Failed:** server nie je dostupný alebo úloha nebola prijatá.
- **Done:** výpočet úlohy je dokončený, značí koniec komunikácie s nástrojom.



Obrázok 6.1: diagram komunikačného protokolu použitého univerzálne pre každý zo zvolených nástrojov. Legenda: dátové premenné, metódy, stavy

6.3 Zapracovanie nástrojov

Ďalším krokom po dokončení komunikácie s nástrojmi je zapracovanie jednotlivých častí do spoločného celku. Vlastná aplikácia funguje veľmi podobne ako komunikačný protokol popísaný v predchádzajúcej kapitole.

Rovnako ako jednotlivé nástroje, používa aplikácia na identifikáciu svojich úloh jednoznačné ID. Východiskové požiadavky na ID sú jeho úplná unikátnosť a pevná dĺžka. Ako generátor ID je použitá štandardná funkcia *uniqid()*, v ktorej ako *prefix* použijeme názov sekvencie zadaný užívateľom. Výsledkom je alfanumerická postupnosť 23 znakov. Následne je táto postupnosť privedená na vstup štandardnej funkcie *md5()*, ktorej použitie zabezpečí vytvorenie spoľahlivo unikátnej postupnosti 32 znakov. Nevýhodou tohto ID je jeho dĺžka, avšak užívateľ s ním prichádza do styku iba v rámci URL, teda žiadnym spôsobom neovplyvňuje jeho prácu s aplikáciou.

Tok aplikácie je založený na použití nasledovných metód:

- **Submit**: vytvorenie novej úlohy a pridelenie unikátneho ID. Hoci validáciu vstupných dát spomenieme neskôr, je potrebné si uvedomiť, že dáta nachádzajúce sa v tejto metóde sú vždy správne. Formulár sa podarí totiž odoslať iba v prípade, že vstupné dáta spĺňajú všetky validačné požiadavky. Úloha je teda vytvorená vždy pri volaní tejto metódy. Vstupné dáta sú následne rozposlané jednotlivým nástrojom. Návratová hodnota nástrojov je stav zadania úlohy a v prípade úspechu ID. V tomto kroku je potrebné definovať rozdiel medzi spomínanými identifikátormi. ID úlohy vytvorenej na serveri budeme od teraz značiť **JID** (Job ID), pričom sa jedná o unikátnu alfanumerickú postupnosť 32 znakov. ID jednotlivých nástrojov budeme značiť **myID**, ktorého dĺžka a formát sú rôzne.
    - Parametre: pole potrebných informácií o predikčnej úlohe (povinný).
    - Návratová hodnota: JID – ID úlohy.
- **CheckResult**: kontrola dostupnosti výsledku jednotlivých nástrojov. V prípade pozitívnej odpovede je obsah podstránky uložený do dočasného textového súboru s názvom *myID.txt* a stav nástroja je upravený.
    - Parametre: JID – ID úlohy (povinný).
    - Návratová hodnota: pole stavov jednotlivých nástrojov.
- **CheckState**: kontrola stavového pola nástrojov. V prípade, že niektorý z nástrojov je v stave *done*, je možné ďalej s výsledkom pracovať a prezentovať ho užívateľovi. Ak je stav všetkých nástrojov *done*, komunikácia s nástrojmi je dokončená.
    - Parametre: pole stavov jednotlivých nástrojov (povinný).
    - Návratová hodnota: stav predikčnej úlohy.

Pre predikčnú úlohu je definovaná nasledovná trojica stavov. Úloha sa reálne do iného stavu dostať nemôže, pretože je vytvorená iba pre správne dáta a testovaná môže byť v podstate neobmedzene dlho. Stačí nám teda definovať stavy:

- **InProgress**: úloha bola vytvorená a prebieha výpočet jednotlivých nástrojov. Stav výsledku úlohy sa vzhľadom na výsledky nástrojov môže ďalej deliť na:
    - NotAvailable: žiadny z nástrojov neposkytuje výsledok.
    - Available: dostupný je výsledok aspoň jedného z nástrojov.
- **Done**: dostupný je výsledok zo všetkých nástrojov.
- **Failed**: zadávanie úlohy všetkých nástrojov bolo neúspešné.

6.3.1 Implementácia databázy

Jednou z východiskových požiadaviek na server je dostupnosť výsledku s časovým odstupom. V prípade vyťaženosti serverov jednotlivých nástrojov sa môže čas výpočtu výrazne zvýšiť. Užívateľovi sa teda môže jednoducho stať, že okno prehliadača cielene alebo náhodne zavrie a k predikčnej úlohe sa chce vrátiť neskôr, prípadne chce výsledok predikcie zdieľať. V podstate jediným uspokojivým riešením tejto požiadavky je pripojenie relačnej databázy (ďalej len DB).

Použitá DB je pomerne jednoduchá a pozostáva z hlavnej tabuľky a tabuliek pre jednotlivé nástroje. Hodnoty konkrétnej úlohy sú v tabuľkách upravované pri zmene stavov.

job	
ID	*int (10)*
JID	*varchar (32)*
title	*varchar (255)*
email	*varchar (255)*
sequence	*longtext*
jpred	*enum('T','F')*
netsurfp	*enum('T','F')*
pciss	*enum('T','F')*
psipred	*enum('T','F')*
psspred	*enum('T','F')*
sympred	*enum('T','F')*
result	*longtext*
conf	*longtext*
state	*enum('inProgress','done','failed')*

Tabuľka 6.1: štruktúra centrálnej tabuľky.

Pre každú úlohu je možné zvoliť 1-6 použitých nástrojov, pričom každý z nástrojov má pridelenú autonómnu tabuľku, v ktorej si uchováva informácie o vlastnej úlohe. Štruktúra tabuľky je pre jednotlivé nástroje takmer rovnaká.

tool_name	
ID	*int (10)*
JID	*varchar (32)*
myID	*varchar (255)*
result	*longtext*
conf	*longtext*
state	*enum('accepted','done','failed')*

Tabuľka 6.2: vzorová štruktúra tabuľky jednotlivých nástrojov.

6.4 Tvorba vlastnej predikcie

Po zapracovaní jednotlivých nástrojov je možné ďalej výsledky analyzovať a vyvodiť tak isté závery v podobe vlastnej predikcie. Okrem reťazca štrukturálnych stavov, ktorý je požadovaným výstupom predikčnej úlohy, je spolu s výsledkom dostupná aj informácia o správnosti a kvalite výsledku. Tieto prídavné informácie sa pre jednotlivé nástroje opäť rôznia a rozoznávame 3 spôsoby, akým sú tieto heuristiky dodávané. Nad dátami je preto potrebné vytvoriť spoločnú metriku, ktorá je realizovaná nasledovným spôsobom:

- **Pravdepodobnosť štrukturálnych stavov (probability):** dodatočná informácia je dodávaná v podobe percentuálnej pravdepodobnosti výskytu štrukturálnych stavov a stav s najvyššou pravdepodobnosťou je zvolený ako výsledný. Suma jednotlivých stavov pre danú pozíciu je 1. Tento spôsob možno považovať za dostatočne univerzálny a stáva sa teda vhodnou metrikou, na ktorú sú konvertované ostatné spôsoby.
- **Miera dôvery v daný stav (confidence):** tá je k dispozícii pre výsledný stav ako celé číslo v rozsahu 0 až 9. Informácie o pravdepodobnosti výskytu ostatných štrukturálnych stavoch sú úplne zanedbané. Položme teda pravdepodobnosť výsledného stavu na 50%, pričom každá jednotka v hodnote miery dôvery zvýši pravdepodobnosť o 5%. Rozdiel 100% a výslednej pravdepodobnosti je rozdelený na polovicu medzi zvyšné dva potenciálne stavy.
- **Vzdialenosť od štrukturálnych stavov (distance):** v podstate sa jedná o obrátenú hodnotu pravdepodobnosti vynásobenú číslom 10. Suma jednotlivých vzdialeností pre danú pozíciu je rovná približne 10 s odchýlkou približne 5 - 10%. Výsledným stavom sa stáva stav, ktorého vzdialenosť je najmenšia. Položme teda hodnotu pravdepodobnosti daného stavu rovnú 1 – vzdialenosť * 0.1.

Na základe zadanej metriky možno jednoducho vypočítať pravdepodobnosť výskytu jednotlivých štrukturálnych stavov. Výpočet predikcie zabezpečuje nasledovná metóda:

- **Calculate**: pre každý stav i z množiny štrukturálnych stavov $\{H, E, C\}$ možno zaviesť pomocnú veličinu r_i, ktorá určí pravdepodobnosť jednotlivých stavov na základe nasledovného výpočtu:

$$r_i = \frac{\sum_{j=1}^{n} p_{i,j}}{n}$$

  Hodnota r_i pre štrukturálny stav i je sumou pravdepodobností všetkých nástrojov, podelená počtom nástrojov, v našom prípade 6. Takýmto spôsobom vypočítame hodnoty r_H ,r_E ,r_C. Štrukturálny stav s najvyššou hodnotou r_i je považovaný za výsledný.
  - Parametre: pole hodnôt potrebných pre vytvorenie predikcie (povinný).
  - Návratová hodnota: pre každý zo zvolených nástrojov je to štvorica (*tool_name, probH, probE, probC*) a výstup vlastnej predikcie *result*.

6.5 Užívateľské rozhranie

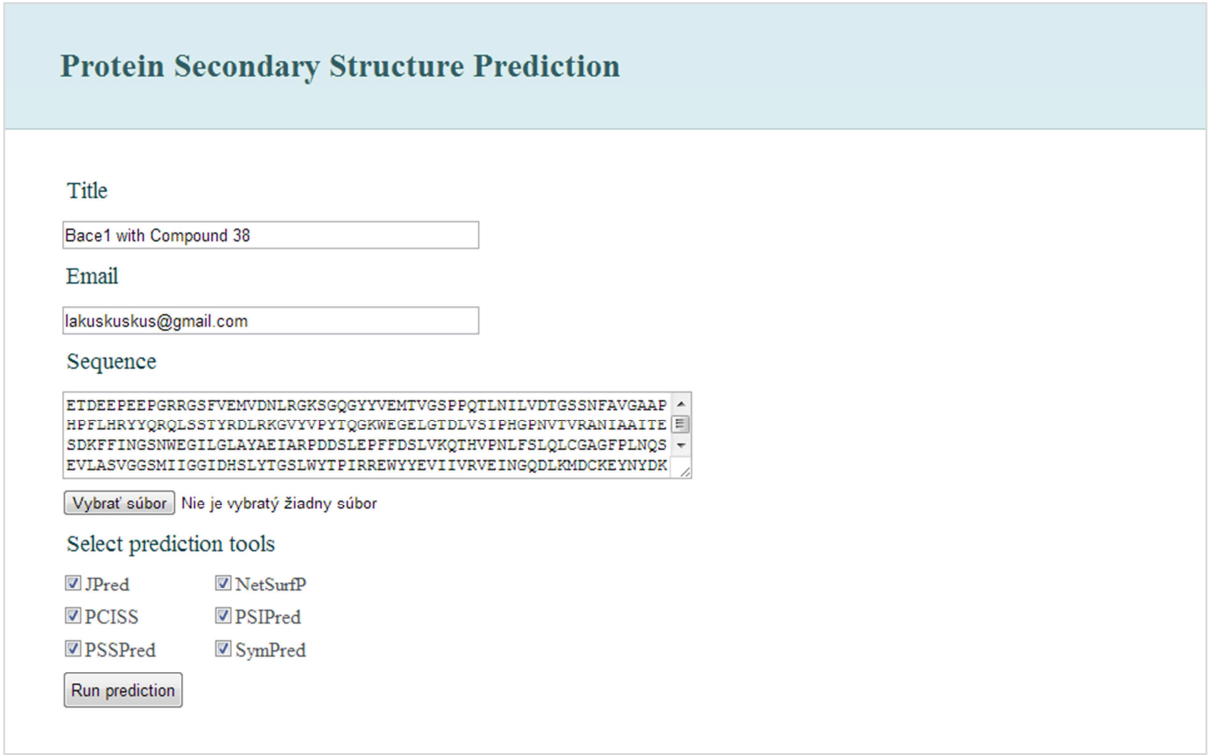
Pracovné prostredie je pomerne jednoduché a intuitívne, pričom pozostáva z 3 obrazoviek (*views*), na ktorých možno prezentovať pracovný tok aplikácie. Jedná sa o podstránky **index**, **submit** a **result**.

6.5.1 Index

Táto podstránka sa zobrazí iniciálne po spustení aplikácie. Slúži na interakciu s užívateľom a je realizovaná vo forme webového formuláru. Skladá sa z troch logicky odlišných častí:

- **Vstup od užívateľa:** vzhľadom na potreby jednotlivých nástrojov sa tu nachádzajú položky:
  - Email - *input*,
  - Názov sekvencie - *input* ,
  - Sekvencia – *text area*,
  - Súbor – *file*.

  Vstupy od užívateľa je potrebné pomerne striktne validovať. Bez validácie by sa mohlo stať, že potenciálny útočník získa prístup k dátovej časti aplikácie a prostredníctvom formulára zaútočí aj na použité nástroje. Spôsob akým kontrola dát prebieha je popísaný v časti venovanej validácii vstupných dát.
- **Voľba nástrojov:** obsahuje výber 1 až *n* nástrojov predikcie prostredníctvom *checkbox inputov*. Každý z nástrojov obsahuje *checkbox* a *label* s názvom nástroja.
- **Odoslanie:** po zadaní vstupov užívateľom sa vykoná validácia hodnôt. V prípade, že sú vstupné dáta irelevantné, je užívateľ presmerovaný späť na podstránku a upozornený adekvátnym chybovým hlásením. Chybových hlásení je opäť niekoľko a detailne sú analyzované v časti o validácii dát. V prípade úspešnej validácie je užívateľ presmerovaný na podstránku submit.



Obrázok 6.2: užívateľské rozhranie úvodnej podstránky.

6.5.2 Submit

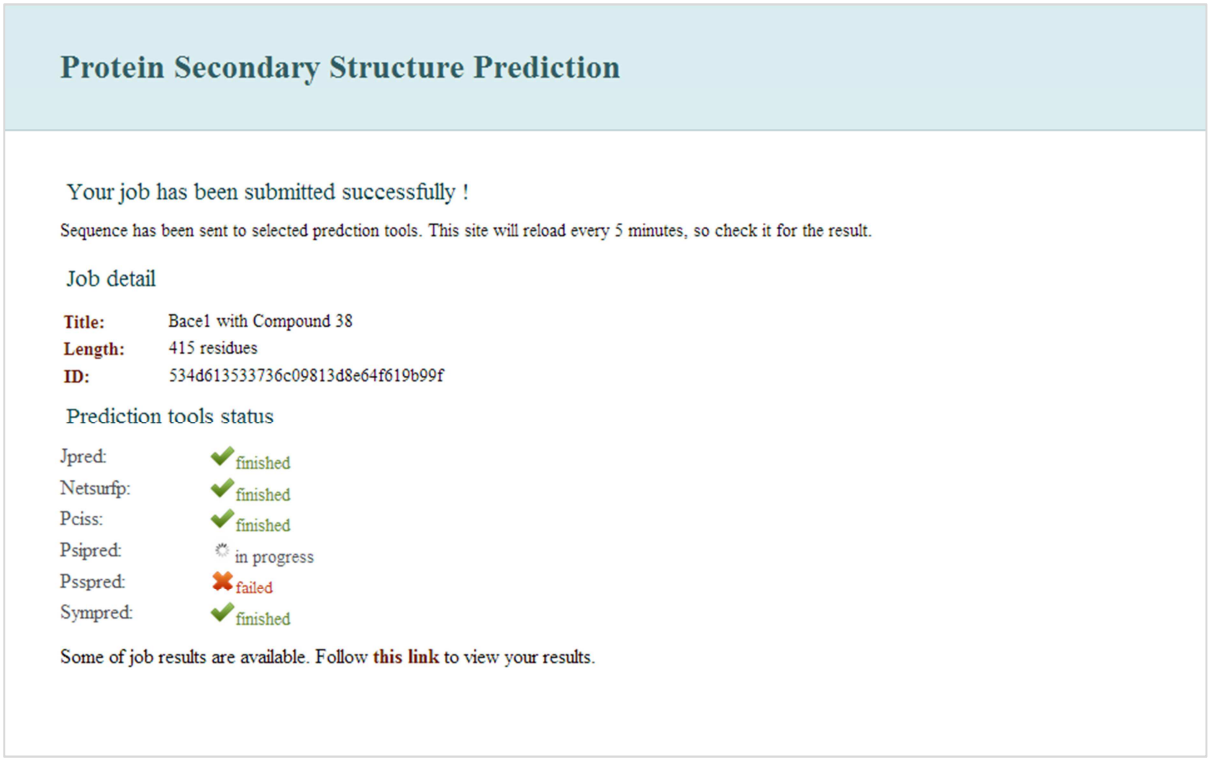
Po odoslaní a validácii dát zo vstupného formuláru sa spustí logika celej aplikácie. Dáta sú odoslané jednotlivým nástrojom, vytvorí sa nová úloha a jej stav sa nastaví na *inProgress*. Odpoveď nástrojov je animovaná nasledovným spôsobom:

- **Accepted:** 
- **Done:** 
- **Failed:** 

Ihneď po získaní odpovede od všetkých nástrojov je volaná metóda *checkResult* a nástroje, ktorých odpoveď je *accepted*, sú otestované na dostupnosť výsledku.

Po príchode na podstránku sa v pozadí spustí počítadlo a obsah sa znovu načíta každých 5 minút s cieľom získať aktuálny stav nástrojov. V prípade, že aspoň jeden z nástrojov vráti stav *done*, zobrazí sa odkaz na podstránku s výsledkom. V prípade, že stav všetkých nástrojov je *done*, stav úlohy sa zmení na *done* a dostupnosť výsledku daných nástrojov sa ďalej nekontroluje.

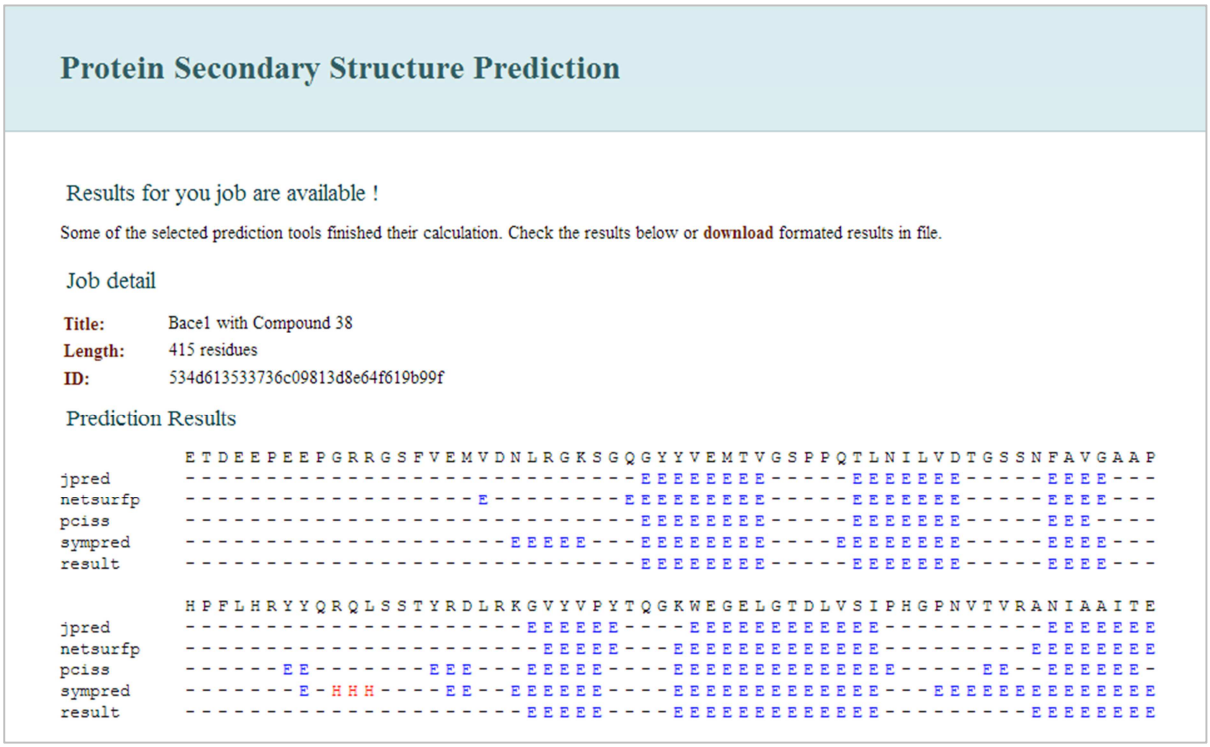
Okrem informácií o stave predikcie sú užívateľovi zobrazené aj detaily predikčnej úlohy. Konkrétne sa jedná o zadaný názov, dĺžku vstupnej sekvencie a ID úlohy.



Obrázok 6.3: užívateľské rozhranie podstránky submit.

6.5.3 Result

Táto podstránka slúži na zobrazenie výsledkov zadanej predikčnej úlohy. Súčasťou podstránky je opäť detail úlohy obsahujúci názov, dĺžku vstupu a ID. Ďalej už nasleduje výsledok zobrazovaný štruktúrovane vo zvolenom formáte. Na zobrazenie výsledku predikcie je použitý font *Courier New (monospace)*, ktorého znaky majú štandardnú šírku a je teda vhodný na formátovanie textu do mriežky. Aby bolo zobrazovanie ešte prehľadnejšie, sú na výsledok použité väčšie medzery medzi znakmi. Vo zvyšku aplikácie je použitý font *MyriadPro*. Na podstránke sa zároveň nachádza odkaz na stiahnutie výsledku v textovom súbore.



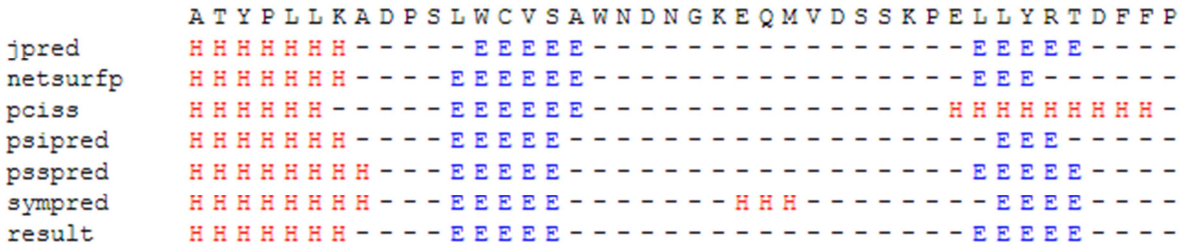
Obrázok 6.4: užívateľské rozhranie podstránky result.

6.6 Zvolený formát výstupu

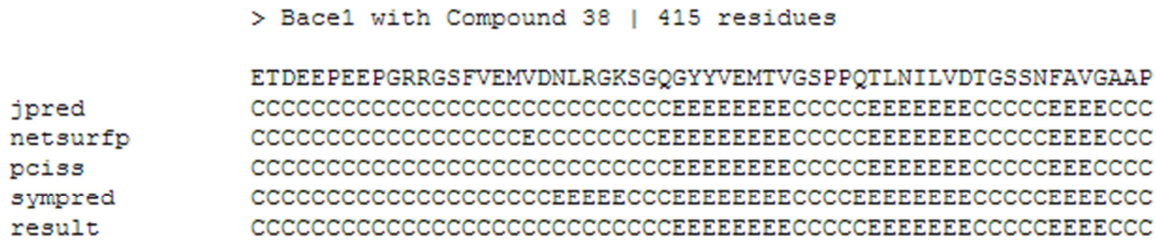
Napriek tomu, že niektoré nástroje predpovedajú viacero štrukturálnych stavov, vo výsledku je možné tieto stavy spojiť a výsledok tak previesť do zaužívaného trojstavového modelu. Najčastejšie sa jedná o predpoveď rôznych typov *coiled* oblastí. So zámerom sprehľadniť výsledok je výhodné jednotlivé štrukturálne stavy odlíšiť farebne značením, ktoré používajú aj niektoré nástroje. Na reprezentáciu štrukturálnych stavov som sa teda rozhodol pre nasledovné značenie:

1. **Helix (H):** červená
2. **Sheet (E):** modrá
3. **Coil (-):** čierna

Pri tvorbe výstupného formátu som sa inšpiroval štruktúrou formátu FASTA, ktorý je dostatočne univerzálny, rozšírený a zaužívaný. Ďalším dôvodom je fakt, že nástroje používajú buď tento formát alebo vlastné derivácie, najčastejšie 60 alebo 100 znakov na riadok. V aplikácii je použitý formát 60 znakov na riadok, ktorý sa pri medzerách medzi znakmi dá považovať za ideálny.



Obrázok 6.5: formát výstupu zodpovedá štandardu FASTA a je prehľadne zobrazený užívateľovi pomocou farebného rozlíšenia štrukturálnych stavov.



Obrázok 6.6: výstup predikcie je zároveň dostupný v textovom súbore, v ktorom je štrukturálny stav *Coil* reprezentovaný znakom *C* z dôvodu ďalšieho spracovania.

6.7 Validácia vstupných dát

V podstate každý webový formulár možno považovať za hrozbu, ktorú môže potenciálny útočník využiť na získanie dát alebo narušenie chodu aplikácie. Zároveň je potrebné myslieť na bezpečnosť celého systému a neprenášať tak možné hrozby na vstupy použitých nástrojov. Dôležitou súčasťou aplikácie je teda kontrola dát zadávaných užívateľom. Súčasťou CI je integrovaná knižnica *form validation*. Pre každé vstupné pole formuláru je možné definovať podmienky, ktoré musí vstup spĺňať, aby bola validácia úspešná. V opačnom prípade funkcia vráti *validation_error* a užívateľ je presmerovaný na úvodnú podstránku. Dôležitým krokom je príprava vloženej sekvencie pred validáciou. Z reťazca sú najprv odstránené špeciálne znaky a medzery. V prípade, že prvý riadok začína znakom „>“, užívateľ pravdepodobne vložil popis sekvencie podobne, ako je tomu vo formáte FASTA. Prvý riadok je teda odstránený a validácia sa ho netýka. Posledným krokom je prevod všetkých znakov na ich *uppercase* tvar. V rámci aplikácie sú použité 2 typy validačných pravidiel:

- **Štandardné:** integrované podmienky.
  - *Required*: pole je povinné
  - *Valid_email*: email vo formáte *name@example.com*.
- **Užívateľské:** možnosť nadefinovať vlastné pravidlá. Prvým z nich je prípad, že užívateľ vloží vstupnú sekvenciu v súbore. Textové pole potom nie je povinné a primárne je spracovaný vstup zo súboru. Ďalšie podmienky sú aplikované na samotnú sekvenciu a jedná sa o tieto pravidlá:
  - *Valid_length*: dĺžka minimálne 30 a maximálne 1000 znakov
  - *Valid_content*: reťazec musí vyhovovať regulárnemu výrazu, ktorý je vytvorený nad abecedou 20 znakov aminokyselín reprezentovaných veľkými písmenami.

6.8 Testovanie a optimalizácia

Po úspešnej implementácii je jednou z najdôležitejších častí celého projektu dôkladné otestovanie. Je totiž potrebné si uvedomiť, že vytvorená aplikácia slúži ako výpočtový nástroj pracujúci nad istými dátami, v tomto prípade molekulami proteínov. Výstupom nástroja je znalosť predikcie sekundárnej štruktúry danej molekuly, pričom kvalitu výstupu sa snažíme maximalizovať. V prípade, že súčasný stav aplikácie umožňuje tento výpočet, je nutné si overiť správnosť implementácie, zhodnotiť a optimalizovať navrhnutý systém. Z tejto úvahy možno sformulovať nasledovné ciele testovania:

- Otestovať dostupnosť jednotlivých nástrojov.
- Zhodnotiť kvalitu čiastkových výsledkov a na základe vhodnej kombinácie nástrojov zosumarizovať celkovú úspešnosť.
- Na základe úspešnosti implementovať systém váh, ktorý jednotlivým nástrojom pridelí zvýšený alebo znížený vplyv na celkový výsledok s cieľom maximalizovať hodnotu Q_3.

6.8.1 Výber množiny testovacích dát

Vhodným miestom pre výber testovacích dát je databáza s názvom *SCOP (Structural Classification of Proteins)* [18]. Tá obsahuje súbor proteínov, ktorých sekundárna štruktúra je známa a tento údaj preto možno považovať za referenčný. Okrem informácií o štruktúre poskytuje táto databáza klasifikáciu proteínov do 11 tzv. *SCOP* tried. Niektoré z nich však nemožno považovať za samostatné triedy, napr. veľmi krátke proteíny alebo proteíny s experimentálnou štruktúrou. Testovacie sekvencie sú preto zvolené z nasledovných 6 tried:

- **All aplha:** prevažne alfa štruktúry.
- **All beta:** prevažne beta štruktúry.
- **Alpha and beta (a/b):** prevažne paralelné beta štruktúry.
- **Alpha and beta (a+b):** prevažne anti paralelné beta štruktúry.
- **Coiled coil:** prevažne coiled oblasti.
- **Other:** krátke sekvencie, multi-doménové alebo experimentálne proteíny.

Výber dát teda nie je úplne náhodný a molekuly sú rovnomerne zvolené z týchto tried. Z každej triedy je vybraných 5 molekúl rôznej dĺžky a zložitosti, teda spolu testovacia množina pozostáva z 30 prvkov. Kompletný zoznam testovacích dát je dostupný v prílohe č. 2. Jednotlivé záznamy sú v databáze *SCOP* dostupné buď v interaktívnej alebo textovej forme. Textový výstup sekundárnej štruktúry je však pre účely testovania dostupný pomerne komplikovaným spôsobom a pre získanie referenčnej štruktúry by bolo potrebné vytvoriť vlastný *parser* tohto textu. Vhodnou alternatívou pre získanie štruktúry ako *plain text* je použiť špecializovaný textový súbor dostupný na stránkach *RCSB-PDB* [19], ktorý vo formáte FASTA obsahuje rozsiahly zoznam proteínov s ich známymi sekundárnymi štruktúrami. Sekundárna štruktúra je uvedená v 8-stavovom modeli, ktorý možno konvertovať na 3-stavový pomocou nasledovného kľúča:

Názov	8-stavov	3-stavy
β-bridge	B	E
Coil	C	C
β-strand	E	E
3_{10}-helix	G	H
α-helix	H	H
π-helix	I	H
bend	S	C
β-turn	T	C

Tabuľka 6.3: prevod 8-stavového modelu na 3-stavový. [1]

6.8.2 Priebeh testovania

Testovanie prebehlo na zvolenej sade proteínov, ktoré boli postupne privádzané na vstup aplikácie. Prvou z pozorovaných vlastností bola dostupnosť nástrojov a čas potrebný na úspešné dokončenie úlohy. Pre niektoré sekvencie bol náhodne vypnutý jeden z nástrojov s cieľom vypozorovať výraznejšie odchýlky vo výsledku predikcie. Samotný výpočet úspešnosti je realizovaný prostredníctvom jednoduchého skriptu:

- **Compare:** funkcia najprv prevedie hodnoty referenčného vstupu z 8-stavového modelu na model 3-stavový a následne porovná obsah reťazcov funkciou *similar_text*, ktorá vracia počet totožných znakov. Táto hodnota následne vstupuje do vzorca:

$$Q_3 = \frac{počet\ totožných\ znakov}{celková\ dĺžka\ vstupu}$$

  - Parametre: referenčná sekundárna štruktúra (povinný), výstup nástroja (povinný).
  - Návratová hodnota: úspešnosť Q_3.

6.8.3 Zhodnotenie testovania

Z pohľadu dostupnosti nástrojov sú výsledky testov vynikajúce. Pre zvolených 30 sekvencií prišlo ku zlyhaniu iba v 2 krát, pričom v oboch prípadoch nebolo možné zadať úlohu do jedného zo zvolených nástrojov. Po zadaní úlohy bol príslušný výsledok dostupný vo všetkých prípadoch. Z toho vyplýva, že užívateľ pristupujúci k nástroju takmer určite získa požadovaný výsledok. Zlyhanie zadávania úloh nástrojom môže byť spôsobené pomalým alebo kolísavým pripojením k internetu. Funkcie SOAP a CURL totiž testujú cieľovú URL niekoľko krát vo zvolenom časovom intervale niekoľko milisekúnd a v prípade neúspešnej komunikácie odpovedajú chybou.

Z časového hľadiska je najrýchlejším nástrojom PCI-SS, ktorý vracia výsledok po odoslaní vstupných dát s oneskorením približne 30 sekúnd. Výsledok je však často nepresný a z pomedzi vybraných nástrojov najhorší. Naopak najdlhšie trvá výpočet nástroju PSIPRED a výsledky boli počas testov dostupné v intervale 30 až 60 minút. Druhým najpomalším je PSSPred, ktorý poskytuje výsledok v intervale 20 až 30 minút. Výsledky PSIPRED a PSSPred možno považovať za najlepšie. Ostatné nástroje poskytujú výsledky už po prvom znovu načítaní stránky, ktoré sa spúšťa každých 5 minút. Platí tu nepísané pravidlo, čím dlhšia doba výpočtu, tým presnejší výsledok.

Výsledky testov úspešnosti predikcie možno považovať za rovnako uspokojivé a nástroj dosiahol hodnotu Q_3 79.4%. Pre získanie presnejších výsledkov by však bolo potrebné použiť oveľa väčšiu testovaciu množinu, ktorá by obsahovala nie niekoľko desiatok, ale niekoľko stoviek až tisícok proteínov. Medzi referenčnými sekvenciami sa totiž nachádzajú proteíny, ktorých štruktúra je komplikovaná a s jej predikciou majú problémy aj najlepšie nástroje. Výsledky testov vidno v nasledujúcej tabuľke:

SCOP trieda	Priemerná dĺžka [počet residuí]	Priemerná úspešnosť Q_3 [%]
All aplha	272	79.8
All beta	200	72.9
Alpha and beta (a/b)	355	79.9
Alpha and beta (a+b)	268	83.2
Coiled coil	453	78.5
Other	317	82.2
SPOLU	**311**	**79.4**

Tabuľka 6.4: výsledky testovania. Úspešnosť predikcie približne zodpovedá úspešnosti najvýkonnejších súčasných nástrojov. Kompletné testovacie dáta sú dostupné v prílohe č. 2.

6.8.4 Optimalizácia

Priebeh testovania bol starostlivo sledovaný a z tohto pozorovania možno vysloviť významné závery. Niektoré sekvencie sú príliš komplikované na to, aby ich štruktúra mohla byť predpovedaná ľubovoľnou technikou. V tomto prípade dĺžka sekvencie nehrá v podstate žiadnu rolu, pretože problematickou časťou sú komplikované krátke úseky, v ktorých sa štrukturálny stav mení každých napr. 2 až 5 residuí. Aj keď niektorý nástroj úspešne odhalí takéto úseky, vplyvom priemerovania s ostatnými nástrojmi sú jeho výsledky potlačené. Ďalšou problematickou časťou predikcie sú hraničné úseky, v ktorých sa mení štrukturálny stav molekuly. Tu sa opäť výsledky nástrojov rôznia a priemerovaním môže byť potenciálny začiatok a koniec α alebo β štruktúry posunutý aj o niekoľko residuií. Naopak štruktúru niektorých molekúl dokážu nástroje predpovedať s vysokou presnosťou, pričom dĺžka sekvencie opäť neovplyvňuje tento výsledok.

Spomínané anomálie by s dali z predikcie odstrániť pridaním ďalších heuristík. To však nemožno realizovať pridaním ďalších vstupných polí formulára, pretože nástroj slúži primárne na predpoveď štruktúry neznámych molekúl a užívateľ často o sekvencii nevie žiadne informácie. Vyhľadávanie homologických sekvencií metódou BLAST rovnako neprináša do predikcie novú znalosť, pretože tento krok realizujú použité nástroje nezávisle. Jedinou použiteľnou heuristikou sa javí znalosť úspešnosti predikcie jednotlivých nástrojov vypozorovaná počas testovania. V prvom kole testov totiž uvažujeme vplyv nástrojov na celkový výsledok úplne rovnaký. Toto nastavenie však nemusí byť úplne korektné. Z testov vyplýva, že najlepšie výsledky dosahujú nástroje PSIPRED a PSSPred, naopak najhoršie výsledky dosahuje nástroj PCI-SS. Rozdeľme teda nástroje do skupín podľa ich vplyvu nasledovným spôsobom:

- **Zvýšený:** PSIPRED, PSSPred
- **Priemerný:** JPred, NetSurfP, SymPred
- **Znížený:** PCI-SS

Váhy týchto tried sú stanovené tak, že zvýšený vplyv zodpovedá hodnote 120% (teda násobenie výsledku konštantou 1.2), priemerný vplyv 100% (nie je potrebné násobenie) a znížený vplyv 80% (násobenie konštantou 0.8). Takéto prestavenie váh vyžadovalo opätovné testovanie, ktorého výsledky vidno v nasledujúcej tabuľke:

SCOP trieda	Q_3 predtým [%]	Q_3 potom [%]
All aplha	79.8	80,1
All beta	72.9	74,5
Alpha and beta (a/b)	79.9	81,3
Alpha and beta (a+b)	83.2	83.7
Coiled coil	78.5	78.7
Other	82.2	83.3
SPOLU	**79.4**	**80.3**

Tabuľka 6.5: výsledky druhého kola testov. Úspešnosť sa po prestavení váh zvýšila o 0.9%.

Druhé kolo testov dopadlo podľa očakávaní a úspešnosť predikcie nástroja sa oproti prvému kolu zvýšila o 0.9% na celkových 80.3%. V testovaní by bolo samozrejme možné pokračovať ďalšou zmenou váh nástrojov a opätovným testovaním nad zvolenou množinou proteínov. Súčasný výsledok však možno považovať za dostatočne presný a optimalizáciu možno prehlásiť za úspešnú.

Záver

V rámci semestrálneho projektu som sa oboznámil s problematikou predikcie sekundárnej štruktúry proteínov. Do problematiky som detailne prenikol na základe štúdia literatúry. Získané znalosti o proteínoch a ich sekundárnej štruktúry som náležite prehodnotil a zdokumentoval.

Ďalším krokom bola analýza existujúcich nástrojov predikcie sekundárnej štruktúry. Referenčný zoznam týchto nástrojov v podstate neexistuje, preto som prehľadával rôzne neoficiálne zoznamy, v ktorých sa nachádzalo spolu približne 60 takýchto nástrojov. Nástroje som ručne testoval vzhľadom na možnosti automatickej komunikácie a spôsob zobrazovania výsledkov. Zo spomínaného množstva som vybral 6, ktoré som subjektívne považoval za najlepšie a najdostupnejšie. Jednotlivé nástroje pracujú ako autonómne, pomerne jednoduché, webové aplikácie. Snažil som sa preto vyextrahovať najlepšie spoločné črty a použiť ich pri návrhu vlastného nástroja. Na tento návrh sa možno pozerať ako na špecifikáciu webovej aplikácie vzhľadom na požiadavky zadávateľa. Špecifikáciu som realizoval v teoretickej rovine v rámci semestrálneho projektu, pričom niektoré časti návrhu sa pri implementácii mierne pozmenili.

V diplomovej práci som navrhnutý nástroj implementoval. Oproti semestrálnej časti projektu som sa po ďalšej analýze rozhodol jeden z nástrojov nahradiť iným a jeden nástroj úplne vypustiť. Na implementáciu som zvolil programovací jazyk PHP, konkrétne jeho rozšírenie CodeIgniter. Jednotlivé nástroje som zapracoval pomocou univerzálneho klienta webovej služby a u nástrojov, ktoré takúto možnosť neposkytujú, som vytvoril simuláciu webových služieb. Na implementáciu klienta som použil techniky SOAP a CURL. Implementácia prebehla úspešne a všetkých 6 zvolených nástrojov sa podarilo implementovať a optimalizovať tak, aby bolo možné získať výsledky predikcie v reálnom čase. Získané výsledky jednotlivých podsystémov sú dostupné v unifikovanom formáte buď priamo v prehliadači alebo v textovom súbore.

Výsledky nástrojov som podrobil ďalšej analýze a nad týmito dátami som realizoval vlastnú predikciu sekundárnej štruktúry. Okrem informácií o štrukturálnych stavoch sú vo výsledku dostupné aj niektoré ďalšie dáta. Tie som použil na výpočet pravdepodobnosti štrukturálnych stavov konkrétneho residua, pričom stav s najvyššou pravdepodobnosťou sa stáva výsledný.

Hotovú aplikáciu som podrobil dôkladnému testovaniu s cieľom otestovať dostupnosť nástrojov a zistiť úspešnosť predikcie Q_3. Výber testovacích dát nebol náhodný, ale použité boli proteíny patriace do rôznych štrukturálnych, tzv. SCOP, tried. Týmto spôsobom som zvolil 30 proteínov rôznej dĺžky, zložitosti a štruktúry. Výsledky testov možno považovať za viac než uspokojivé a úspešnosť predikcie pre zvolenú testovaciu množinu bola 79.4% a po úprave váh nástrojov dokonca 80.3%. Táto hodnota približne zodpovedá súčasnému štandardu a úspešnosti vybraných nástrojov.

V budúcnosti by sa tento nástroj mohol stať súčasťou väčšieho serveru slúžiaceho proteínovým inžinierom, ktorí by mohli v rámci serveru získavať rôzne druhy predikcií na jednom mieste. Z pohľadu implementácie je budúcnosť projektu v podstate otvorená a aplikácia by sa dala doplniť o viaceré vylepšenia. Pripojenie administrácie a užívateľských účtov by častým používateľom nástroja uľahčilo spätnú analýzu sekvencií. Pomocou navrhnutých klientov a komunikačného protokolu je pomerne jednoduché zapracovať ďalšie nástroje, ktoré by úspešnosť predikcie mohli ešte zvýšiť.

Literatúra

[1] Alberts, B., Bray, D., Johnson, A., Lewis, J., Raff, M., Roberts, K., Walter, P.: *Základy buněčné biologie*. Espero Publishing, 2. Vyd., 2005. ISBN: 80-902906-2-0

[2] Zvelebil, M., J., Baum, J., O.: *Understanding bioinformatics*. Garland Pub, 2008. ISBN: 0-8153-4024-9

[3] Webster, M., D.: *Protein structure prediction, methods and protocols*. Humana Press, 2000. ISBN: 0-8960-3637-5

[4] Web Services Activity [online]. 2011 [cit. 2013-5-4]. Dostupné na WWW: <http://www.w3.org/2002/ws/>

[5] PCI-SS – PCI-based protein secondary structure site prediction server [online]. 2007 [cit. 2013-5-4]. Dostupné na WWW: <http://bioinf.sce.carleton.ca/PCISS/start.php/>

[6] NetSurfP – Protein surface accessibility and secondary structure predictions [online]. 2013 [cit. 2013-5-4]. Dostupné na WWW: <http://www.cbs.dtu.dk/services/NetSurfP/>

[7] The PSIPRED Protein secondary structure prediction server [online]. 2010 [cit. 2013-5-4]. Dostupné na WWW: <http://bioinf.cs.ucl.ac.uk/psipred/>

[8] PSSPred – Protein secondary structure prediction [online]. 2012 [cit. 2013-5-4]. Dostupné na WWW: <http://zhanglab.ccmb.med.umich.edu/PSSpred/>

[9] Jpred – Incorporating Jnet [online]. 2012 [cit. 2013-5-4]. Dostupné na WWW: <http://www.compbio.dundee.ac.uk/www-jpred/advanced.html>

[10] SymPred – Protein secondary structure prediction [online]. 2012 [cit. 2013-5-4]. Dostupné na WWW: <http://bio-cluster.iis.sinica.edu.tw/SymPred/>

[11] BioCatalogue – Life science web services [online]. 2013 [cit. 2013-5-4]. Dostupné na WWW: <http://www.biocatalogue.org/>

[12] ExPASy – Bioinformatics resource portal [online]. 2010 [cit. 2013-5-4]. Dostupné na WWW: <http://www.expasy.org/tools/>

[13] MolBiol Tools – Online analysis tools [online]. 2012 [cit. 2013-5-4]. Dostupné na WWW: <http://www.molbiol-tools.ca/Protein_secondary_structure.htm/>

[14] CodeIgniter PHP Framework [online]. 2013 [cit. 2013-5-8]. Dostupné na WWW: <http://ellislab.com/codeigniter/>

[15] NuSOAP – SOAP toolkit for PHP [online]. 2011 [cit. 2013-5-8]. Dostupné na WWW: <http://sourceforge.net/projects/nusoap/>

[16] PHP SOAP – Manual [online]. 2013 [cit. 2013-5-8]. Dostupné na WWW: <http://php.net/manual/en/book.soap.php/>

[17] PHP Client URL library – Manual [online]. 2013 [cit. 2013-5-8]. Dostupné na WWW: <http://php.net/manual/en/book.curl.php/>

[18] SCOP – Structural classification of proteins [online]. 2009 [cit. 2013-5-15]. Dostupné na WWW: <http://scop.mrc-lmb.cam.ac.uk/scop/>

[19] RCSB-PDB – Biological macromolecular resource [online]. 2009 [cit. 2013-5-15]. Dostupné na WWW: <http://www.rcsb.org/pdb/static.do?p=help/ssHelp.html/>

[20] WAMP – a Windows web development environment [online]. 2013 [cit. 2013-5-18]. Dostupné na WWW: <http://www.wampserver.com/en/>

Zoznam príloh

Príloha 1. Konfigurácia CodeIgniter

1. Aplikácia je momentálne konfigurovaná na lokálne umiestnenie. Pre spustenie na systéme Windows je potrebné použiť lokálny server napr. balík *WAMP*, dostupný na stiahnutie zadarmo [20]. Projekt je potrebné rozbaliť a nakopírovať do adresára *wamp/www*. Iniciálne je potrebné v súbore *php.ini* povoliť import knižnice SOAP:

   ```
   extension=php_soap.dll
   ```

   Súbor *php.ini* možno nájsť po ľavom kliknutí na *tray* ikonu *WAMP*, adresár PHP. Po zmene ľubovoľných nastavení sa odporúča server reštartovať pomocou príkazu *Restart all services*.

2. Pre umiestnenie aplikácie na vzdialený server je potrebné korektne nastaviť východiskovú URL a upraviť obsah konfiguračného súboru *application/config/config.php* na:

   ```
   $config['base_url'] = 'http://example.com/pred/';
   ```

   v prípade, že sa projekt nachádza na odkaze *www.example.com*. Tento prefix sa bude ďalej pripájať pred každý použitý odkaz. V prípade, že sa prefix skladá z viacerých adresárov, napr. *www.example.com/example1/example2*, je potrebné upraviť súbor *.htaccess* v koreňovom adresári projektu a nastaviť *RewriteBase*:

   ```
   RewriteBase example1/example2/pred/
   ```

3. V prípade pripojenia externej databázy je potrebné upraviť obsah databázového súboru *application/config/database.php* a nastaviť korektné prístupy, napr. :

   ```
   $db['default']['hostname'] = 'www.example.com/database';
   $db['default']['username'] = 'exampleUsername';
   $db['default']['password'] = 'examplePassword';
   $db['default']['database'] = 'exampleDatabase';
   ```

   Štruktúru databázy je potrebné na server importovať napr. použitím prostredia *phpmyadmin*. Export databázy spolu s testovacími dátami je dostupný na priloženom CD v súbore *test.sql*.

Príloha 2. Zoznam testovacích dát

PDB záznam	SCOP trieda	Dĺžka [počet residuí]	Q_3 [%]
102m	alpha	154	95.5
1r4a	alpha	51	94.1
2a6i	alpha	222	66.7
2iy5	alpha	785	75.6
2q4f	alpha	149	67.2
1mxd	beta	435	72.6
1dif	beta	99	88.8
1wap	beta	75	68.7
1c39	beta	152	67.5
1uvp	beta	240	66.7
2bi7	a/b	384	78.6
2apc	a/b	342	87.1
1dx6	a/b	543	77.3
1dx9	a/b	169	65.1
1dxh	a/b	335	91.6
1inn	a+b	166	95.2
2boa	a+b	404	86.4
108l	a+b	164	75.0
2q4g	a+b	129	72.9
1wao	a+b	477	86.4
2vsg	coil	358	84.9
2fyz	coil	63	92.1
1qu1	coil	155	82.6
1epw	coil	1290	74.8
1rf1	coil	311	58.2
2wfp	other	394	85.6
3c4h	other	357	86.3
3in4	other	415	72.3
1dx5	other	36	91.7
2hty	other	387	74.9