Masarykova univerzita
Fakulta informatiky

Detekcia pórov v molekulách proteínov

Diplomová práca

Bc. Matej Ružôň

Brno, jar 2016

PREHLÁSENIE

Prehlasujem, že táto diplomová práca je mojím pôvodným autorským dielom, ktoré som vypracoval samostatne. Všetky zdroje, pramene a literatúru, ktoré som pri vypracovaní používal alebo z nich čerpal, v práci riadne citujem s uvedením úplného odkazu na príslušný zdroj.

Bc. Matej Ružôň

Vedúci práce: RNDr. Barbora Kozlíková, Ph.D.

POĎAKOVANIE

Týmto sa chcem poďakovať vedúcemu práce, RNDr. Barbore Kozlíkovej, Ph.D., a RNDr. Ondřejovi Strnadovi, Ph.D. za trpezlivosť, cenné rady, odbornú pomoc a čas venovaný konzultáciám. Ďalej sa chcem poďakovať mojej rodine a partnerke za podporu počas obdobia vzniku tejto práce.

ZHRNUTIE

Cieľom tejto práce je štúdium algoritmu PoreWalker, ktorý slúži na detekciu pórov v proteínoch a jeho implementácia do aplikácie CAVER Analyst. V teoretickej časti táto práca definuje proteíny, biologické membrány, ich štruktúry a vzájomné prepojenie. Ďalej popisuje algoritmus PoreWalker a nástroje používané pri jeho implementácii do aplikácie CAVER Analyst.

V praktickej časti potom popisuje konkrétnu implementáciu a jej výsledky porovnáva s výsledkami PoreWalkeru.

KĽÚČOVÉ SLOVÁ

Proteín, biologická membrána, pór, Java, PoreWalker, CAVER, CAVER Analyst.

OBSAH

1. Úvod **1**

2. Motivácia **3**

3. Biologická membrána **4**

4. Proteín **6**

4.1. Štruktúra proteínu 6

5. Membránové proteíny **9**

5.1. Integrálne monotopické proteíny 11

5.2. Transmembránové proteíny 11

6. Nástroje **13**

6.1. Protein Data Bank 13

6.2. PDB formát 14

6.3. Java 14

6.4. NetBeans IDE 15

6.5. CAVER 16

7. PoreWalker **17**

7.1. Predbežná definícia osi póru a reorientácia proteínu 18

7.2. Definícia stredu póru 19

7.3. Spresnenie osi póru a detekcia dutiny 21

7.4. Analýza a predikcia charakteristík póru 22

7.4.1. Identifikácia atómov a rezíduí obklopujúcich pór 22

7.4.2. Popis tvaru dutiny póru 23

7.4.3. Analýza pravidelnosti dutiny póru 24

8. Implementácia **25**

8.1. Predbežný odhad osi póru 25

8.2. Rotácia štruktúry 27
8.3. Výpočet póru 30
8.4. Vizualizácia póru 32
8.5. Ukladanie informácií o póre 33
8.6. Integrácia algoritmu do aplikácie CAVER Analyst 34

9. Porovnanie s algoritmom PoreWalker **37**
9.1. Rýchlosť výpočtu póru 43
9.2. Zhodnotenie 44

10. Záver **47**

11. Literatúra **48**

1. ÚVOD

Proteíny patria medzi najrozšírenejšie a najvšestrannejšie molekuly v živých organizmoch. Zastávajú veľké množstvo pre život dôležitých funkcií, ako sú prenos a ukladanie iných molekúl, prenos nervových impulzov, katalyzácia, stavba buniek, imunitná ochrana [1], a preto sa o nich (spolu s nukleovými kyselinami a lipidmi) hovorí aj ako o molekulách umožňujúcich život [8].

Táto diplomová práca sa zmeriava na skupinu proteínov, ktoré sú neoddeliteľnou súčasťou biologickej membrány (Kapitola 3.), nazývaných membránové proteíny (Kapitola 5). A o konkrétnu ich skupinu, ktorou sú transmembránové proteíny (Kapitola 5.2). Transmembránové proteíny sú polypeptidové reťazce, ktoré prechádzajú cez obe časti fosfolipidovej dvojvrstvy tvoriacej biologickú membránu. Vďaka tejto polohe zaobstarávajú niekoľko špecifických funkcií ako sú komunikácia bunky s jej okolitým prostredím alebo s inými bunkami, prenos iónov do bunky a z nej, reakcia na nervové impulzy a spúšťanie biologických procesov v bunke. Väčšinu z týchto funkcií umožňujú dutiny v štruktúre transmembránových pórov, ktoré sú vytvorené ich sekundárnymi štruktúrami. Tieto dutiny nazývame póry

V tejto dobe je momentálne najpoužívanejším algoritmom pre detekciu pórov algoritmus PoreWalker. Tento algoritmus je založený na postupnosti niekoľkých krokov, pomocou ktorých najskôr definuje predbežnú os póru, prechádzajúcu cez stred póru a smerujúcu kolmo na rovinu membrány. Tú potom optimalizuje a pomocou rezov o veľkosti 1Å (Ångström[1]) postupne počíta pór samotný.

Cieľom tejto diplomovej práce je štúdium PoreWalkeru, jeho následná implementácia do aplikácie CAVER Analyst a porovnanie

[1] Jednotka dĺžky, ktorá sa používa pri určovaní veľkostí atómov, dĺžok chemických väzieb alebo vlnových dĺžok. Platí, že 1Å sa rovná 1nm alebo 10^{-10} m.

výsledkov mojej implementácie a s výsledkami PoreWalkeru na niekoľkých vybraných molekulách proteínov získaných z databázy Protein Data Bank.

CAVER Analyst je nástroj slúžiaci na analýzu a vizualizáciu molekúl proteínov, ktoré môžu byť statické ale aj dynamické. Projekt CAVER začal v roku 2006 ako akademický vývoj, na ktorom spolupracovali tými z Laboratoře interakce člověka s počítačem (Fakulta informatiky Masarykovej Univerzity) a Loschmidt laboratories (Prírodovedecká fakulta Masarykovej univerzity)[2].

2. MOTIVÁCIA

V posledných desaťročiach vzniklo veľké množstvo nových vedných oborov, ktoré v sebe spájajú biologický a informatický smer. Ich častým spoločným cieľom je práve štúdium proteínov. Sú nimi napríklad bioinformatika, počítačová chémia, systémová biológia, proteomika a chemoinformatika.

Jedným z hlavných dôvodov, je veľký význam proteínov a možnosti, ktoré nám detailné vedomosti o nich poskytujú. Vďaka týmto znalostiam môžeme neustále vyvíjať lepšie a účinnejšie lieky, ktoré nemajú vážne vedľajšie účinky. Taktiež budeme možno v budúcnosti schopný, vďaka týmto znalostiam, niektoré choroby alebo zdravotné problémy úplne odstrániť.

Na to, aby sme mali k týmto možnostiam prístup, musíme najskôr poznať štruktúru proteínov. Pomocou nej môžeme ďalej skúmať závislosti medzi danou štruktúrou a jej funkciou a až potom budeme schopný túto funkciu využiť alebo odstrániť.

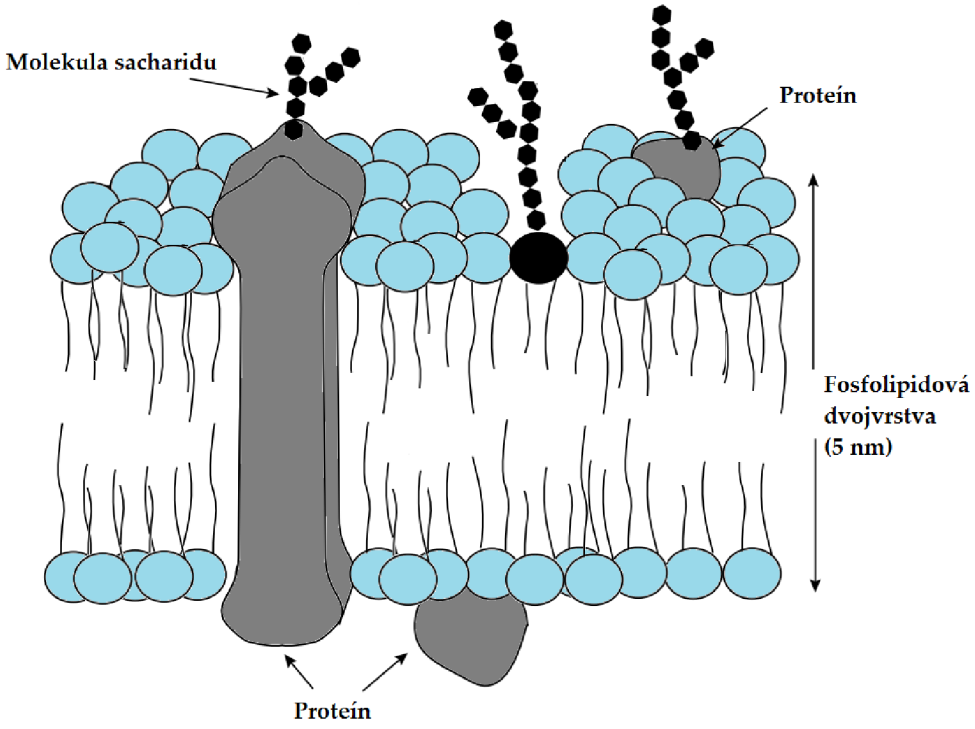
3. BIOLOGICKÁ MEMBRÁNA

Membrána je bariéra, ktorá dáva bunkám ich vonkajšie hranice (plazmatická membrána) a ich vnútorné oddelenia (organely). Je semipermeabilná a má tak schopnosť kontrolovať tok substancií z bunky a do nej, vďaka čomu reguluje zloženie vnútrobunkovej tekutiny. Kontrolujú taktiež tok informácií medzi bunkami, a to buď pomocou rozpoznania signálnych molekúl odoslaných inými bunkami alebo posielaním chemických alebo elektrických signálov iným bunkám. Membrány sú tiež zapojené v ukladaní a uvoľňovaní energie – prebieha tu fotosyntéza a oxidačná fosforylácia [2] [3]. Biologické membrány sú teda viac ako len internými bariérami alebo obalom, sú aktívnou súčasťou života bunky.

Biologické membrány majú v priereze 5 – 10 nm a sú zložené z troch komponent – lipidov, proteínov a cukrov (Obrázok 1). Všetky membrány majú rovnakú základnú štruktúru, membránové proteíny sú osadené v lipidovej dvojvrstve. Štruktúra je vysoko tekutá a tak sa v nej môže väčšina lipidových a proteínových molekúl laterálne pohybovať [4]. Tieto dva druhy molekúl sú ku sebe viazané najmä nekovalentnými väzbami. Ale molekuly cukrov, ktoré môžeme nájsť len na jednej strane membrány, sú k niektorým lipidovým a proteínovým molekulám viazané väzbami kovalentnými.

Lipidová dvojvrsta je tvorená prevažne fosfolipidami a to tak, že ich hydrofilné časti (tzv. hlavičky, tvorené napríklad zvyškami fosfátov) sú orientované smerom von a hydrofóbne časti (chvosty tvorené 2 mastnými kyselinami) smerom dovnútra membrány [5].

[2] Proces syntézy ATP pomocou elektrochemického protónového gradientu.



OBRÁZOK 1: SCHÉMATICKÁ REPREZENTÁCIA BIOLOGICKEJ MEMBRÁNY

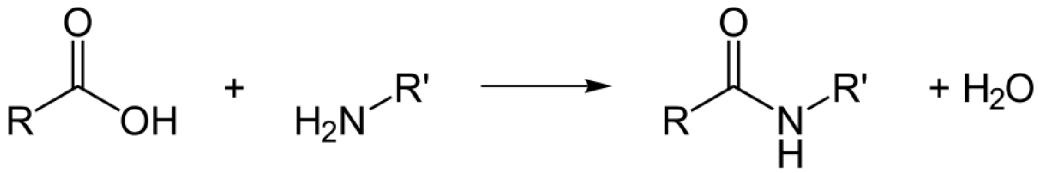
4. PROTEÍN

Proteíny sú veľké biologické molekuly, alebo inak makromolekuly, ktoré v živých organizmoch vykonávajú veľké množstvo vnútrobunkových aj mimobunkových funkcií a sú, spolu s ostatnými makromolekulami (polysacharidmi a nukleovými kyselinami), pre ne nepostrádateľnou súčasťou. Medzi tieto funkcie patrí replikácia DNA, odpoveď na podnety a prenos nervových impulzov, kontrola rastu a diferenciácie buniek, katalyzácia metabolických reakcií, transport molekúl z jedného miesta na druhé a sú taktiež stavebnou jednotkou cytoskeletu (bunkovej steny)[6].

4.1. ŠTRUKTÚRA PROTEÍNU

Proteíny sú zložené prevažne z aminokyselín, ktoré sú usporiadané do jedného alebo viacerých reťazcov, pričom z hľadiska vlastností daného proteínu je dôležité nie len to z akých aminokyselín je zložený, ale aj ich počet a usporiadanie v reťazci. Jednotlivé aminokyselinové zvyšky sú spojené peptidovými väzbami [3] (Obrázok 2), vďaka čomu sú proteíny nazývané aj polypeptidy. Konce každého polypeptidového reťazca potom rozlišujeme na koniec s voľnou (nezreagovanou) aminoskupinou – N-koniec, a koniec s voľnou karboxylovou skupinou – C-koniec [7].

[3] Peptidová väzba – druh chemickej väzby, ktorý vzniká medzi karboxylovou skupinou (–COOH) jednej molekuly a aminoskupinou (–NH_2) druhej molekuly. Peptidová väzba má teda tvar –CO–NH–.



OBRÁZOK 2: VZNIK PEPTIDOVEJ VÄZBY, KDE R A R' PREDSTAVUJÚ ĽUBOVOLNÚ AMINOKYSELINU

Úrovne štruktúry proteínu rozdeľujeme na [8]:

- Primárnu –je definovaná len presným poradím aminokyselín v reťazci v smere od N-konca po C-koniec.
- Sekundárnu – určuje lokálne usporiadanie určitej časti polypeptidového reťazca, ktoré je stabilizované vodíkovými mostíkmi. Najbežnejšími druhmi sú α-helix a β-skladaný list.
- Terciárnu – zahŕňa oveľa dlhší úsek ako sekundárna, pretože aj aminokyseliny, ktoré sú v sekvencii od seba veľmi vzdialené, môžu v priestore navzájom interagovať. Určuje priestorové usporiadanie jednotlivých α-helixov, β-skladaných listov a bočných reťazcov. Táto štruktúra je stabilizovaná veľkým množstvom nekovalentných väzieb (iónové interakcie, van der Waalsove sily, hydrofóbne väzby) a tiež niektorými špecifickými väzbami (disulfidové mostíky).
- Kvartérnu – majú len proteíny skladajúce sa z dvoch alebo viacerých polypeptidových reťazcov. Kvartérna štruktúra potom popisuje ich vzájomné priestorové usporiadanie.

Podľa jedného z tradičných delení, proteíny môžu byť rozdelené do troch hlavných kategórií, ktoré korelujú s typickou terciárnou štruktúrou [1]:

1) Globulárne – je guľovité usporiadanie, ktoré je tvorené striedaním α-helixu a β-skladaného listu s ostatnými úsekmi proteínu. Väčšina týchto proteínov je rozpustná vo vode alebo soľných roztokoch. Najznámejším zástupcom je krvný proteín – hemoglobín.

2) Fibrilárne – je vláknovité usporiadanie, v ktorom prevládajú α-helixi alebo len β-skladané listy. Väčšinou sú nerozpustné a medzi zástupcov patria napríklad kolagény a keratíny.

3) Membránové – sú naviazané na membránu a sú jej funkčnou súčasťou.

5. MEMBRÁNOVÉ PROTEÍNY

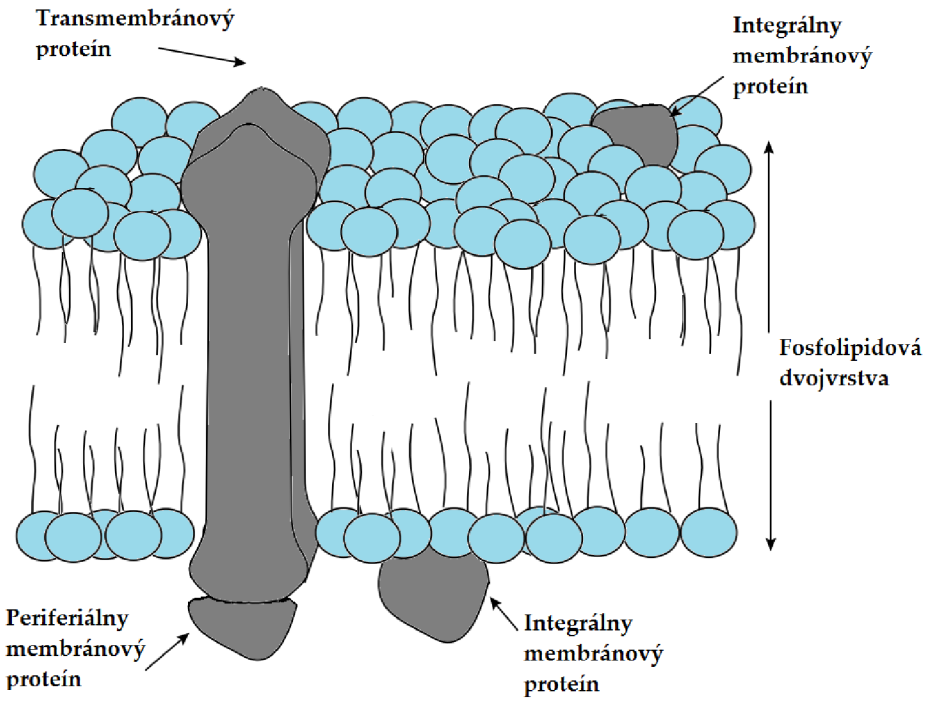
Takto nazývame proteíny, ktoré sa nachádzajú v biologickej membráne. Patria medzi biologicky najdôležitejšie proteíny a kvôli množstvu funkcií, ktoré vykonávajú, vyskytujú sa na povrchu membrány vo vysokej koncentrácii. Umožňujú vzájomnú komunikáciu medzi bunkami a ich okolitým prostredím, rozhodujú o tom, či sa imunitný systém bude k danej bunke správať ako k cudzej a teda potenciálne nebezpečnej alebo nie [9]. Tieto proteíny sú taktiež cieľom väčšiny, ak nie všetkých liečiv, kontrolujú adhéziu pri vytváraní tkanív, riadia vývoj rastlín a živočíchov a kontrolujú dôležité metabolické procesy vrátanie udržovania rovnováhy solí, výrobu a prenos energie a fotosyntézu. Taktiež ale môžu slúžiť ako kanále, ktoré prenášajú špecifické molekuly cez membránu z bunky alebo do nej.

Tieto proteíny môžeme, v závislosti na sile ich interakcie s membránou, rozdeliť do dvoch hlavných kategórii (Obrázok 3):

1) Integrálne membránové proteíny – táto skupina je charakteristická veľmi silnou interakciou s membránou, ktorá môže byť zlomená len pridaním detergentu alebo iného nepolárneho rozpúšťadla. Väčšina integrálnych membránových proteínov obsahuje rezídua s hydrofóbnymi postrannými reťazcami, ktoré interagujú s mastnými acylovými skupinami membránových fosfolipidov a tým ukotvujú proteín na membráne. Toto ukotvenie môže byť realizované dvoma spôsobmi a tak integrálne membránové proteíny ďalej delíme na transmembránové (bielkovinové reťazce prekračujú celú membránu) a integrálne monotopické proteíny (prekračujú len jeden list membrány).

2) Periférne membránové proteíny – skupina, ktorá má oveľa slabšiu interakciu s membránou ako integrálne membránové proteíny. Pripojenie k membráne alebo k iným integrálnym membránovým

proteínom zabezpečujú hydrofóbne, elektrostatické alebo iné nekovalentné väzby. Ich polárna interakcia môže byť narušená zmenou pH.



OBRÁZOK 3: MEMBRÁNOVÉ PROTEÍNY - MEMBRÁNOVÉ INTEGRÁLNE (MONOTOPICKÉ), TRANSMEMBRÁNOVÉ A PERIFÉRNE

Formálne sa môžu membránové proteíny deliť nasledovne [10]:

I. typ – membránou prechádza iba raz, C koniec polypeptidu je na vnútornej strane membrány

II. typ – membránou prechádza iba raz, N koniec polypeptidu je na vnútornej strane membrány

III. typ – membránou prechádza niekoľko krát

IV. typ – v membráne pomocou niekoľkých domén proteínu vytvára transmembránový kanál

V. typ – naviazané k fosfolipidom membrány pomocou kovalentných väzieb

VI. typ – prechádza membránou a zároveň je naviazaný k fosfolipidom

5.1. INTEGRÁLNE MONOTOPICKÉ PROTEÍNY

Skupina integrálnych membránových proteínov, ktorá je kotvená len v jednom (vnútornom alebo vonkajšom) liste membrány. Membránou teda neprechádzajú a ich veľká časť sa teda nachádza mimo nej. Mnohými zástupcami tejto skupiny sú enzýmy. Príkladom je cyklooxygenáza. Pomocou jej inhibície protizápalové lieky (aspirín, ibuprofén) zmierňujú bolesť a príznaky zápalu.

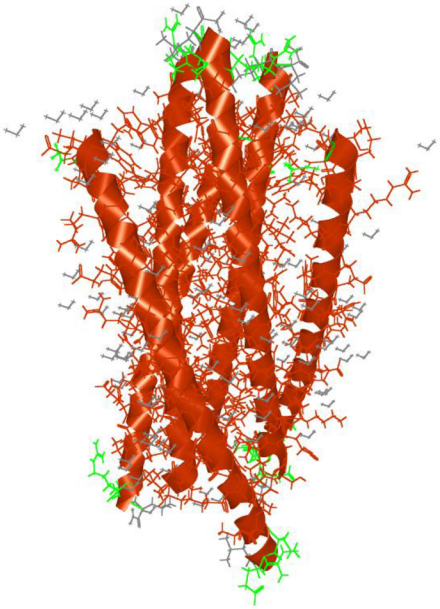
5.2. TRANSMEMBRÁNOVÉ PROTEÍNY

Zástupcovia tejto skupiny prechádzajú celou membránou, majú teda časť mimo bunky, časť prechádzajúcu membránou a časť vo vnútri bunky. Časť v membráne je zabudovaná medzi fosfolipidy a poskytuje tak kanál, cez ktorý môžu molekuly a ióny prechádzať do bunky a von z nej. Tento kanál nazývame pór [5]. Transmembránové proteíny tiež uľahčujú komunikáciu medzi bunkami pomocou chemických poslov a aktiváciou niektorých konkrétnych zástupcov môže byť spustených mnoho biologických procesov, ako napríklad metabolizmus glukózy alebo produkcia mastných kyselín.

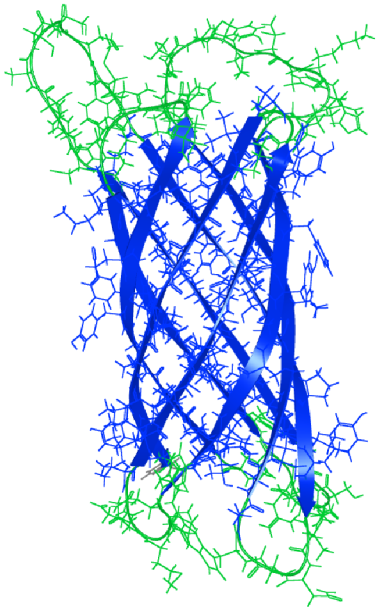
Ďalšou úlohou týchto proteínov je kyvadlová preprava iónov, ako je sodík a draslík, ktoré napomáhajú udržaniu správneho chemického prostredia. Pri nesprávnej funkcii pórov, nemôžu niektoré bunky plniť svoje špecifické úlohy. Dôležitým príkladom sú napäťovo riadené kanály nervových buniek. V kľudovom stave je pór uzavretý a bráni prechodu iónov cez membránu. Ako náhle je detekovaný stimul, napríklad popálenie alebo porezanie sa, nervový impulz je odoslaný z jedného konca bunky na druhý [5]. K tomu môže dôjsť len ak sú póry otvorené a umožňujú priechod iónov cez bunkovú membránu.

Transmembránové proteíny môžu byť rozdelené podľa štruktúry na:

- α-helix – sú zložené z jedného reťazca a obsahujú 20 – 30 aminokyselín (Obrázok 4)
- β-barel – niekoľko β-skladaných listov, ktoré tvoria valec prenikajúci membránou (Obrázok 5).



OBRÁZOK 4: ALFA-HELIX (ČERVENÝ)[4]



OBRÁZOK 5: BETA-BAREL (MODRÝ)[4]

[4] Vytvorené pomocou nástroja CAVER Analyst

6. NÁSTROJE

V nasledujúcich sekciách trochu priblížim nástroje, ktoré sú používané pri implementácii v praktickej časti tejto práce.

6.1. PROTEIN DATA BANK

Protein Data Bank (PDB) [11] je celosvetové vysoko anotované a udržiavané úložisko experimentálne stanovených trojrozmerných dát o štruktúrach biologických makromolekúl, napríklad proteínov alebo nukleových kyselín. Pre odvetvia využívajúce dáta a znalosti o týchto štruktúrach je PDB kľúčovým zdrojom informácií.

PDB bola založená v roku 1971v Brookhaven National Laboratory pod vedením Waltera Hamiltona a na začiatku obsahovala 7 štruktúr[5]. Od roku 2003 je udržiavaná organizáciou Worldwide Protein Data Bank (wwPDB), ktorá formalizuje úložisko a vkladateľom z celého sveta zabezpečuje jeho transparentnosť. Fyzické úložisko je pod správou RCSB PDB[6], ktorý je jeden zo zakladateľov wwPDB. Ďalšími členmi sú PDBe[7], PDBj[8] a BMRB[9].

Dáta v PDB sú uchovávané v PDB a mm-CIF[10] formáte. Každý záznam má svoje unikátne PDB ID reprezentované štvormiestnym alfanumerickým kódom a nové záznamy sú pridávané raz za týždeň prostredníctvom členov wwPDB.

[5] Pre porovnanie k 14.5.2015 archív obsahuje 108789 štruktúr [11].
[6] Research Collaboratory for Structural Bioinformatics
[7] Protein Data Bank in Europe
[8] Protein Data Bank Japan
[9] Biological Magnetic Resonance Data Bank
[10] macromolecular Crystallographic Information File

6.2. PDB FORMÁT

PDB formát (.pdb) popisuje trojdimenzionálnu štruktúru uloženú v Protein Data Bank databáze. Poskytuje najmä informácie o atómoch, z ktorých sa štruktúra skladá, ich súradniciach, a údaje o primárnej a sekundárnej štruktúre. Často ale obsahuje aj dodatočné údaje ako sú kryštalografická štruktúra, referencie v databáze, bibliografické citácie, detaily o zbere dát a podobne.

PDB formát je čisto textový formát vytvorený tak, aby bol čitateľný aj pre človeka. Stĺpce majú fixné poradie a maximálnu dĺžku 80 znakov (stĺpcov), pričom posledný znak je identifikátor konca riadku[11]. Každý riadok začína šesť znakovým názvom záznamu (TITLE, COMPND, ATOM...), ktorý daný záznam identifikuje a špecifikuje typ informácií, ktoré sa na danom riadku (zázname) nachádzajú. Sada znakov je obmedzená na nekontrolné ASCII znaky a neodporúča sa používať diakritiku [11].

6.3. JAVA

Na implementáciu praktickej časti tejto diplomovej práce bol použitý programovací jazyk Java. Java, navrhnutá Jamesom Goslingom pracujúcim pre Sun Microsystems, patrí medzi objektovo orientované jazyky. Platforma Java sa skladá z Java API a JVM (Java virtual machine) [12].

Je navrhnutá tak, aby mala čo najmenšiu implementačnú závislosť, čo by sa dalo popísať mottom „write once, run anywhere“ [12] [13], a znamená to, že skompilovaný Java kód môže byť spustený na akejkoľvek platforme bez nutnosti novej kompilácie. Práve pre svoju prenositeľnosť je tento programovací jazyk často používaný pre aplikácie, ktoré majú pracovať na rôznych systémoch, a to od čipových

[11] Ten je závislí od daného systému a tak je prípustný line feed ale aj carriage return nasledovaný line feed .

[12] Vo voľnom preklade : napíš raz, spusť kdekoľvek.

kariet (Java Card), cez mobilné telefóny (Java ME – Micro Edition),televízie (Java TV) desktopové aplikácie (Java SE – Standart Edition) až po rozsiahle distribuované systémy (Java EE – Enterprise Edition) [12].

Táto prenositeľnosť je zabezpečená kompiláciou do Java bytecode, ktorý následne JVM prekladá do strojového kódu operačného systému, pod ktorým práve beží.

6.4. NETBEANS IDE

Na vývoj algoritmu bolo použité vývojové prostredie NetBeans IDE postavené na jazyku Java. Práve pre vývoj aplikácií v tomto jazyku je aj primárne určené, podporuje ale aj niekoľko ďalších jazykov (PHP, Javascript, C/C++, ...).

Funkcie NetBeans IDE sú poskytované modulmi, pomocou ktorých je možné toto prostredie rozšíriť. Základnými integrovanými modulmi sú NetBeans Profiler, slúžiaci na monitorovanie aplikácie a zjednodušenie jej optimalizácie, nástroj na návrh GUI a NetBeans Javascript editor, ktorý rozširuje podporu pre Javascript a CSS.

NetBeans je čiastočne vyvíjané komunitou (registrovaných je viac ako 800 000 zúčastnených vývojárov) a tak je možné dodatočne stiahnuť rozširujúce moduly, prípadne ich vyvinúť a zdieľať s ostatnými užívateľmi [14].

6.5. CAVER

CAVER je softwarový nástroj slúžiaci na analýzu a vizualizáciu tunelov, kanálov a transportných ciest v štruktúrach proteínov a iných makromolekúl a to ako statických, tak aj dynamických. Na jeho vývoj je využívaná platforma NetBeans a je napísaný v programovacom jazyku Java. Je ho teda možné spustiť na akejkoľvek platforme, kde je nainštalované Java Runtime Enviroment verzie 6.0 alebo vyššej [15].

CAVER (v súčasnej verzii 3.02) je konzolový program, ktorého výstupom sú čisto textové súbory, analýza ktorých nie je užívateľsky príliš príjemná. Na túto analýzu a taktiež na vizualizáciu je potom použitý CAVER Analyst[13][16], ktorý CAVER využíva a jeho výsledky vizualizuje a prevádza do užívateľsky príjemnej a ľahko zrozumiteľnej a čitateľnej podoby. S jeho pomocou sa dajú jednoducho nastaviť jednotlivé parametre výpočtov a taktiež podoba výslednej vizualizácie.

[13] Predchádzajúci názov CAVER Viewer

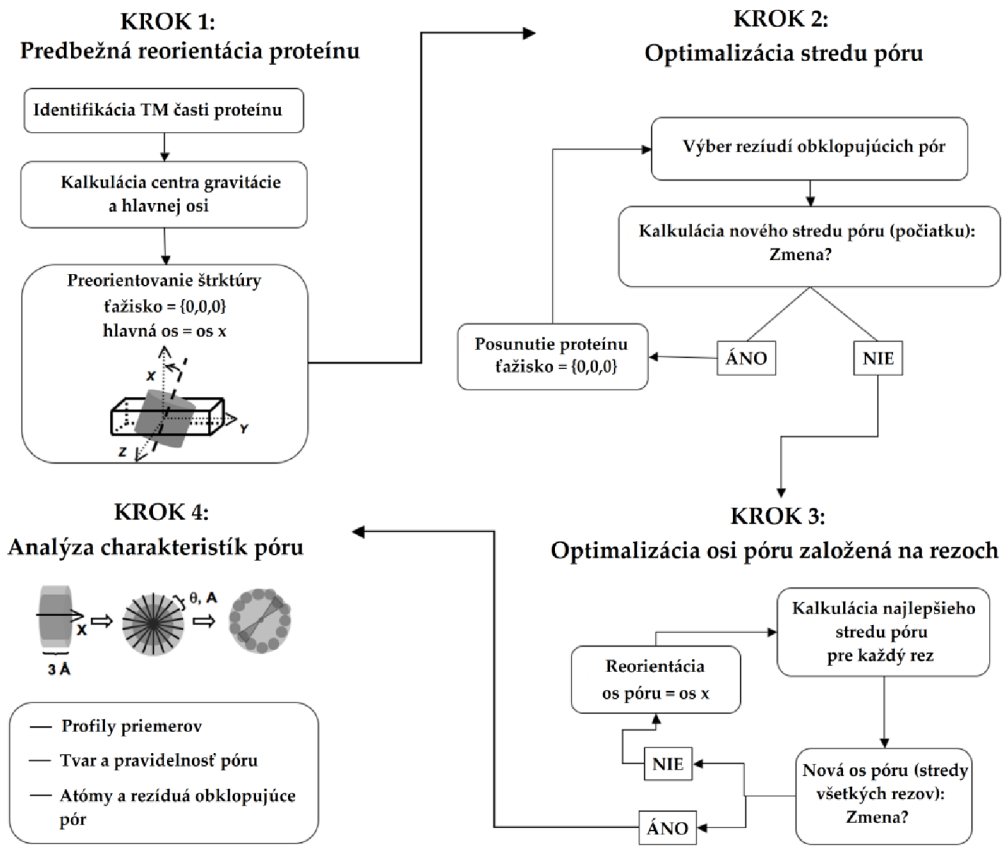
7. PoreWalker

PoreWalker [17], vyvinutý týmom Marialuisa Pellegrini-Calace, Tim Maiwald a Janet M. Thornton v jazykoch C a PYTHON, je plne automatizovaná metóda, ktorá vyhľadáva a plne charakterizuje kanály v transmembránových proteínoch z ich 3D štruktúry. Postupne identifikuje stred a os póru, ktoré sú následne optimalizované pomocou geometrických kritérií. Následne nájde najväčšiu a najdlhšiu dutinu, kopírujúcu túto osu. Následne sa vypočítajú vlastnosti póru vrátane priemeru, veľkosti, tvaru a rezíduí, ktoré ho obklopujú. Tieto vlastnosti poskytujú kvantitatívnu a vizuálnu charakteristiku póru.

Algoritmus je heuristický a iteratívny. Zahŕňa v sebe nasledujúce kroky (Obrázok 6):

1. Definícia osi kanálu, ktorá sa počíta ako priemer vektorov sekundárnych štruktúr a prechádza gravitačným centrom póru.
2. Identifikácia a maximalizácia počtu predpokladaných rezíduí, obklopujúcich proteín nasledujúcim spôsobom :
   a) Použitie aktuálnej osi póru na identifikáciu rezíduí obklopujúcich proteín ako tie, ktoré sú v dostupnej blízkosti póru a ich C-alfa-C-beta vektor smeruje k tejto osi.
   b) Použitie nájdených rezíduí na predefinovanie stredu póru.
   c) Opakovanie krokov *a* a *b*, pokiaľ počet rezíduí obklopujúcich pór nedosiahne maximum.

   To definuje nový geometrický „stred“ póru a predbežnú osu póru, ktorá je kolmá na rovinu membrány a prechádza cez stred póru.
3. Opakované upresňovanie osi póru. Štruktúra proteínu je rozdelená na rezy (kolmé na aktuálnu os) a pre každý rez sa počíta lokálny „stred“ póru, ktorý je definovaný tak, aby mal čo najväčší priemer. Tieto stredy definujú optimalizovanú „cestu“ dutinou kanálu a sú použité tiež na definíciu konečnej osi póru.
4. Výpočet 3D deskriptorov popisujúcich geometriu a chémiu póru.



OBRÁZOK 6: ZÁKLADNÉ KROKY POREWALKERU [17]

V nasledujúcich sekciách detailnejšie popíšem jednotlivé kroky algoritmu PoreWalker.

7.1. PREDBEŽNÁ DEFINÍCIA OSI PÓRU A REORIENTÁCIA PROTEÍNU

Cez transmembránové proteíny póry prechádzajú približne kolmo na rovinu membrány a paralelne s β-barelom alebo balíkom α-helixov, ktoré transmembránovú časť proteínu vytvárajú.

Prvá časť programu teda spočíva v preorientovaní proteínovej štruktúry tak, aby počiatok ležal v gravitačnom centre transmembránovej časti proteínu a balík α-helixov prípadne β-barel

ležali kolmo na rovinu membrány. Hlavná os transmembránového balíku/barelu sa vypočíta podľa polohy sekundárnych štruktúr, ktoré ich vytvárajú. Tie musia byť dlhšie ako 15 – 10 aminokyselín, čo je v priemere dĺžka potrebná na prechod membránou. Sekundárne štruktúry, ktoré sú kratšie alebo sú mimo balíku/barelu a ležia viac kolmo ako paralelne s ním sú vynechané z dôvodu, aby výsledná poloha proteínu (vypočítaná z priemeru vektorov sekundárnych štruktúr) bola čo najlepšia pre budúci výpočet.

Celá proteínová 3D štruktúra je následne preorientovaná tak, aby vypočítaná hlavná os splynula s osou x daného 3D systému a gravitačné centrum transmembránovej časti ležalo v počiatku tohto systému. Tento počiatočný predpoklad, aj napriek svojej hrubosti, zjednodušuje a urýchľuje nasledujúce kroky algoritmu.

7.2. DEFINÍCIA STREDU PÓRU

Stred póru je definovaný pomocou iteratívnej maximalizácie počtu rezíduí, ktoré podľa nášho predpokladu obklopujú pór – vode prístupné aminokyseliny, ktoré smerujú k ose póru.

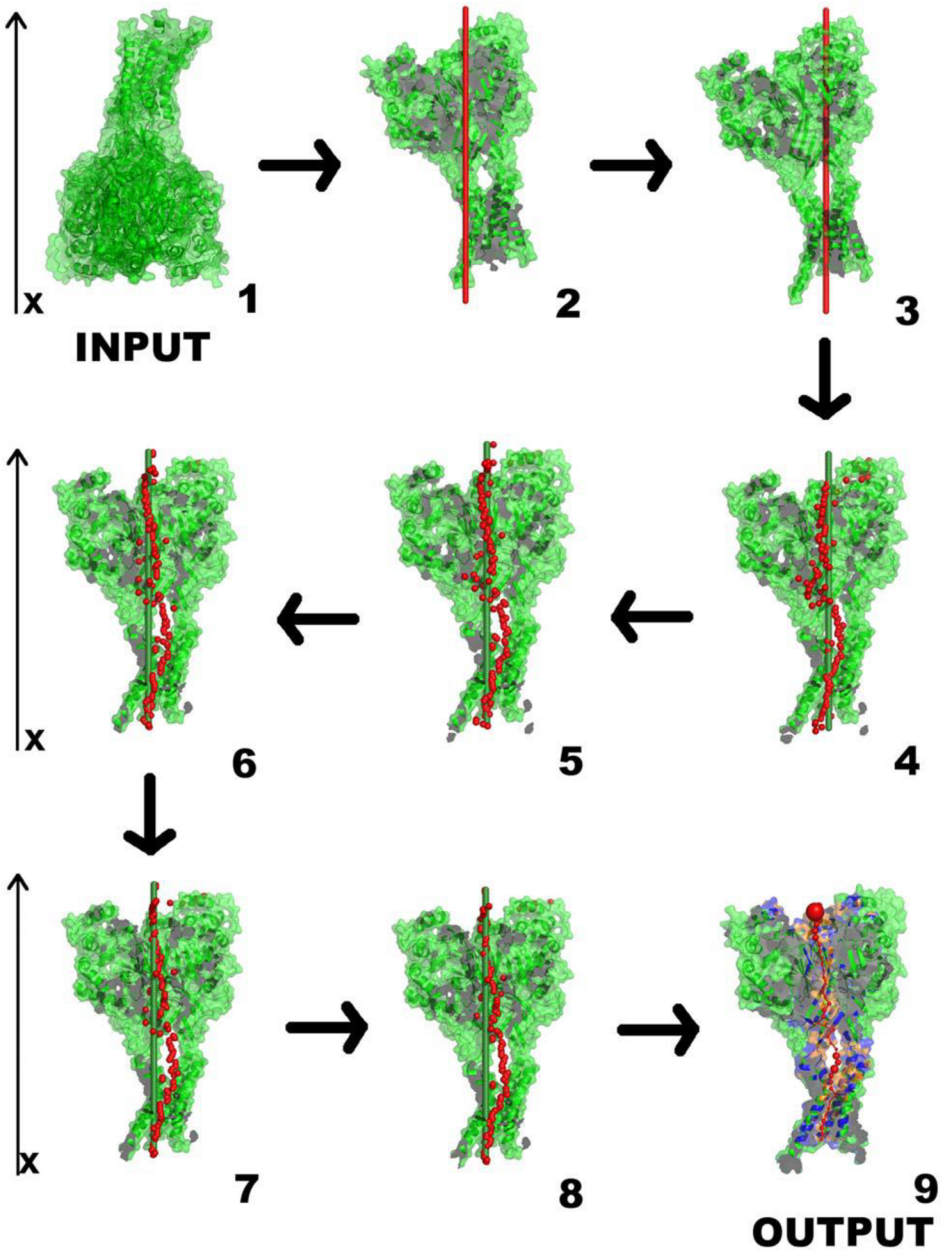
Na začiatku procedúry predpokladáme, že stred póru os póru (t.j. lineárny vektor prechádzajúci stredom póru) zodpovedajú stredu hmotnosti proteínu respektíve ose x.

Na výber predpokladaných pór obklopujúcich aminokyselín sú použité tri kritériá:

- Relatívna prístupnosť postranného reťazca pre rozpúšťadlo počítaná pomocou NACCESS[14] musí byť väčšia ako 5%.
- Vektor definovaný pomocou C-alfa-C-beta väzby musí smerovať k ose póru.
- Vzdialenosť C-alfa musí byť menšia ako daná hraničná hodnota.

[14] NACCESS je program, ktorý počíta prístupnú oblasť molekuly z PDB formátu [19]

Z vybraných aminokyselín obklopujúcich pór je následne vypočítaný nový stred póru a proteínová štruktúra je premiestnená tak, aby zodpovedala rovnakým podmienkam ako na začiatku procedúry.



OBRÁZOK 7: POSTUPNÉ DEFINOVANIE STREDU PÓRU [17]

Tieto kroky sú vykonávané iteratívne (Obrázok 7) a procedúra je zastavená, keď počet nových aminokyselín obklopujúcich pór konverguje na jeho maximum, čo indikuje, že stred póru dosiahol jeho optimálnu polohu.

7.3. SPRESNENIE OSI PÓRU A DETEKCIA DUTINY

Na odvodenie čo najpresnejšej osi a dutiny póru je použitý prístup na báze rezov, pri ktorom je stred póru systematicky optimalizovaný pre každý rez a tak môžu byť detekované prípadné nepresnosti. Pri každej iterácii je proteínová štruktúra mapovaná na 3D sieť s krokmi veľkosti 1Å a rezaná pozdĺž osi x (teda terajšej osi póru) na 1Å hrubé rezy. Stred póru pre každý rez je potom vypočítaný ako stred gule s maximálnym priemerom, ktorý sa do dutiny daného rezu vojde.

Na konci iterácie sú súradnice stredov, posledných štyroch za sebou idúcich gúľ na každom konci póru, zaznamenané a pomocou ich priemeru sú generované dva body, ktoré definujú novú os póru. Štruktúra je potom preorientovaná tak, aby bola zarovnaná s novým vektorom. Posledné štyri za sebou idúce stredy sú použité preto, že konce pórov môžu byť veľmi nepravidelné čo sa týka tvaru a tak pri použití dvoch stredov (jedného pre každý koniec póru) môžu byť výsledky veľmi skreslené a os z nich vypočítaná nemusí pretínať jeden alebo ani jeden z koncových bodov.

Spresňovací proces sa zastaví vo chvíli, keď nový vektor póru „prekryje“ starý vektor (čo znamená, že uhol, ktorý zvierajú bude menší ako 0,5°) a aktuálna os póru a maximálny polomer gule sú uchované ako optimálne a sú použité pri ďalšej analýze tvaru póru.

7.4. ANALÝZA A PREDIKCIA CHARAKTERISTÍK PÓRU

Posledným krokom metódy je analýza a výpočet troch hlavných deskriptorov póru:

- atómov a rezíduí, ktoré obklopujú pór
- tvar dutiny póru
- a pravidelnosť tejto dutiny

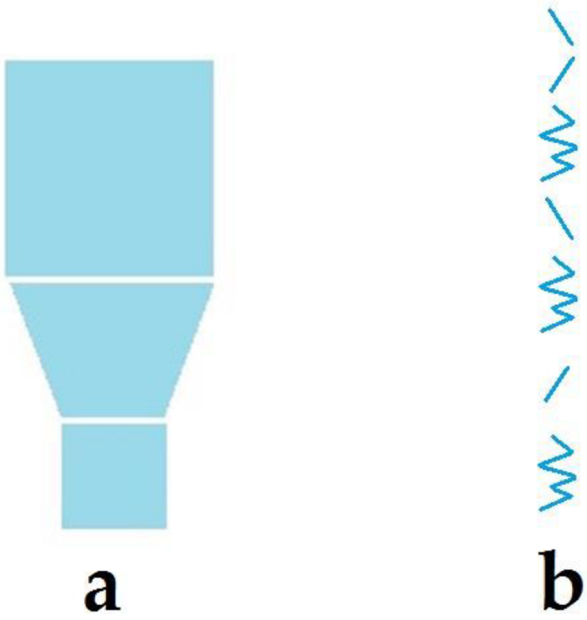
7.4.1. IDENTIFIKÁCIA ATÓMOV A REZÍDUÍ OBKLOPUJÚCICH PÓR

Ako rezíduá obklopujúce pór môžu byť definované aminokyseliny, ktoré prispievajú k vytvoreniu povrchu póru aspoň jedným atómom. Za povrch póru môže byť považované kontinuum, tvorené horizontálnymi a vertikálnymi vrstvami atómov. Na preskúmanie každej vrstvy atómov je použitý rovnaký prístup ako v predchádzajúcom kroku – rozrezanie štruktúry proteínu horizontálne cez os póru na rezy s hrúbkou porovnateľnou s priemerom ťažkých atómov v proteíne (tými sú uhlík, dusík alebo kyslík), takže teoreticky bude každý rez obsahovať len jednu vrstvu atómov. Keďže priemerný priemer ťažkých proteínových atómov je v AMBER silovom poli ~3,6Å, ako hrúbka rezov bolo zvolené najbližšie nižšie celé číslo (3Å), aby bola čo najmenšia šanca, že sa v jednom reze budú nachádzať dva atómy zo susedných vrstiev. Každý rez môže byť ďalej rozdelený na výseky (opäť s hrúbkou porovnateľnou s priemerom ťažkých atómov v proteínoch) takže, pre každý výsek môže byť nájdený atóm, ktorý leží najbližšie k stredu póru a ten je považovaný za atóm obklopujúci pór pre daný výsek.

7.4.2. POPIS TVARU DUTINY PÓRU

3D tvar póru je zjednodušený na stoh dvoch základných stavebných blokov – skosených kužeľov a cylindrov, takže môže byť vytvorený schematický „kreslený“ diagram póru (Obrázok 8(a)). Stavebné bloky sú identifikované pomocou analýzy trendu najmenších vypočítaných priemerov, pozdĺž celého póru, pre každý rez o hrúbke 3Å. Táto analýza prebieha počas detekcie rezíduí obklopujúcich pór. Skosený kužeľ sa používa na reprezentáciu častí, kde sa priemer mení lineárne a cylinder pre časti s konštantným priemerom.

Variácia priemerov je analyzovaná naprieč všetkými rezmi póru. Pre každý rez sa vypočíta štandardná odchýlka (SO) hodnôt v rámci tohto úseku a na jej základe je vybratý stavebný blok, ktorý bude reprezentovať daný rez. Ak je SO > 1Å, stavebný blok bude skosený kužeľ, ak je SO < 1Å, bude ním cylinder.



OBRÁZOK 8: SCHEMATICKÝ DIAGRAM PÓRU (A) A 2D REPREZENTÁCIA DUTINY PÓRU

7.4.3. ANALÝZA PRAVIDELNOSTI DUTINY PÓRU

Pravidelnosť dutiny póru je vyvodená z pozícií stredov póru, ktoré sú počítané v treťom kroku programu, ktorý je zameraný na optimalizáciu osi póru. Ak sú pozície stredov kolineárne, pór je lineárny a dutina musí byť veľmi pravidelná a symetrická s ohľadom na os póru. V opačnom prípade, ak sú iba stredy niektorých častí póru kolineárne, pór môže byť symetrický ale nie lineárny a dutina bude čiastočne pravidelná. Posledný prípad nastáva, ak je len veľmi málo stredom kolineárnych, vtedy musí byť dutina nesymetrická a pór bude nepravidelný.

Tu je zobrazenie póru taktiež zjednodušené na dva 2D stavebné bloky, ktorými sú rovná čiara a kľukatá čiara (Obrázok 8(b)). Rovná čiara reprezentuje kolineárne stredy póru, kľukatá nekolineárne.

8. IMPLEMENTÁCIA

Cieľom mojej diplomovej práce je návrh implementácie algoritmu PoreWalker v jazyku Java a jeho následná integrácia do aplikácie CAVER Analyst. Ako už bolo povedané, implementácia je v jazyku Java s pomocou vývojového prostredia NetBeans. Molekuly proteínov sú získané z databázy PDB.

8.1. PREDBEŽNÝ ODHAD OSI PÓRU

Prvým krokom, zásadným pre výpočet póru, je stanovenie približného predpokladaného začiatku a konca póru (Obrázok 9). Postup je podobný tomu, aký využíva PoreWalker.

Na začiatku sú zo všetkých typov sekundárnych štruktúr vybraté len tie, ktoré sú pre výpočet podstatné – α-helixi a β-skladané listy (Algoritmus 1). Z nich sú následne odstránené tie, ktoré sú príliš krátke a tak sa nemusia nutne podieľať na tvorbe dutiny póru, čo znamená, že ich smerový vektor nemusí byť približne kolmý na rovinu membrány. Ak by boli do výpočtu zahrnuté aj tieto sekundárne štruktúry, body začiatku a konca by mohli byť veľmi vychýlené a celkový výpočet by tak bo nepresný. Minimálna dĺžka sekundárnych štruktúr je 15 aminokyselinových zvyškov.

Po odstránení pre výpočet nepodstatných sekundárnych štruktúr, metóda pokračuje definíciou smerového vektoru, začiatku a konca pre každú z nich. Smerový vektor je vypočítaný z polohy α-uhlíkov prvého a posledného rezídua danej sekundárnej štruktúry a následne uložený do zoznamu smerových vektorov molekuly - `List<Vector3f> ssVectors`. Z týchto vektorov, sú následne vytvorené zoznamy pre ich začiatočné body - `List<Vector3f> ssStartPoints` a pre ich koncové body - `List<Vector3f> ssEndPoints`, z ktorých priemerov je neskôr vypočítaný výstup celej metódy – odhadovaný začiatok a koniec póru danej molekuly. Tu sa môže vyskytnúť problém

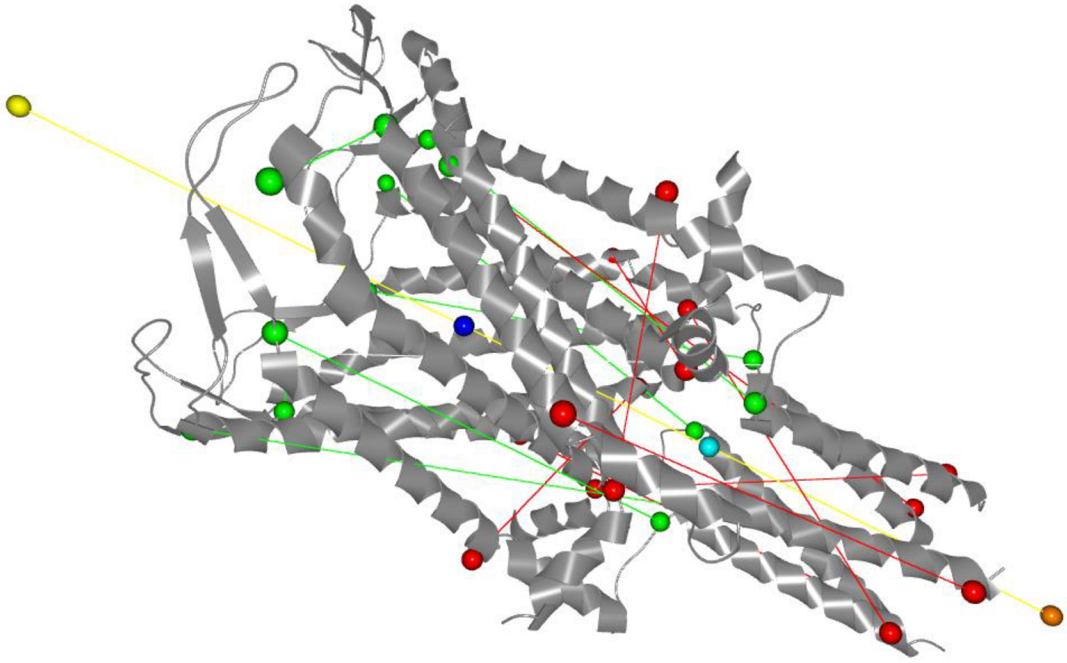
v orientácií týchto štruktúr a teda aj ich smerových vektorov, pretože aj štruktúry, ktoré pri lineárnom pohľade na reťazec nasledujú priamo za sebou, môžu mať opačnú orientáciu smerových vektorov. Pre výsledný výpočet nie je dôležité, na ktorej strane molekuly sa bude začiatok a koniec nachádzať a tak sa za „správny" smer považuje orientácia smerového vektoru prvej sekundárnej štruktúry vo výpočte. Pre všetky nasledujúce sa potom počíta uhol s týmto vektorom.

```
for (SecondaryStructureElement sse:
   SecondaryStructureElements()){
  if ((sse == HELIX) || (sse == STRAND)) {
   if (sse.getLenght() >= ELEMENT_ALLOWED_LENGTH){
      Residue firstResidue = molecule.getResidues().
         get(sse.getStartIndex());
      Residue lastResidue = molecule.getResidues().
         get(sse.getEndIndex());
      ssVectors.add(createVector(
      firstResidue.getAlphaCarbonPosition(),
      lastResidue.getAlphaCarbonPosition()));

      float angle = ssVectors.get(0).
                     angle(ssVectors.get(i));

         if (angle < PI/2) {
             ssStartPoints.add(firstResidue.
                  getAlphaCarbonPosition());
             ssEndPoints.add(lastResidue.
                  getAlphaCarbonPosition());
         } else {
             ssStartPoints.add(lastResidue.
                  getAlphaCarbonPosition());
             ssEndPoints.add(firstResidue.
                  getAlphaCarbonPosition());
         }
      i++;
}}}
```

ALGORITMUS 1: ALGORITMUS NA FILTRÁCIU SEKUNDÁRNYCH ŠTRUKTÚR A VÝPOČET SMEROVÝCH VEKTOROV



OBRÁZOK 9: PREDBEŽNÝ ZAČIATOK(ORANŽOVÝ) A KONIEC(ŽLTÝ) PÓRU

Z výstupu vyššie spomenutej metódy je získaný predbežný začiatok a koniec osi proteínu, ktorý môže byť (aj s pomocnými vektormi) samostatne zobrazený alebo ďalej použitý pre nasledujúci výpočet.

8.2. ROTÁCIA ŠTRUKTÚRY

Ďalším krokom je rotácia celej štruktúry tak, aby bola táto os zhodná s osou x, pozdĺž ktorej bude prebiehať ďalší výpočet.

Na rotáciu je potrebné poznať normálový vektor a uhly, ktoré zviera s osami x a y. Pomocou týchto uhlov sú následne vytvorené transformačné matice pre otočenie (Obrázok 10), okolo ktorých je daná štruktúra následne otočená.

$$\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos\theta & -\sin\theta & 0 \\ 0 & \sin\theta & \cos\theta & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} \cos\theta & -\sin\theta & 0 & 0 \\ \sin\theta & \cos\theta & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

OBRÁZOK 10: MATICA OTOČENIA OKOLO OSI X (V ĽAVO) A OKOLO OSI Z (V PRAVO) O UHOL Θ

Pre vektor (x,y,z) sa potom nové súradnice vypočítajú ako:

1. Otočenie okolo osi x o uhol θ:
   $x' = x$
   $y' = y * \cos\theta - z * \sin\theta$
   $z' = y * \sin\theta + z * \cos\theta$

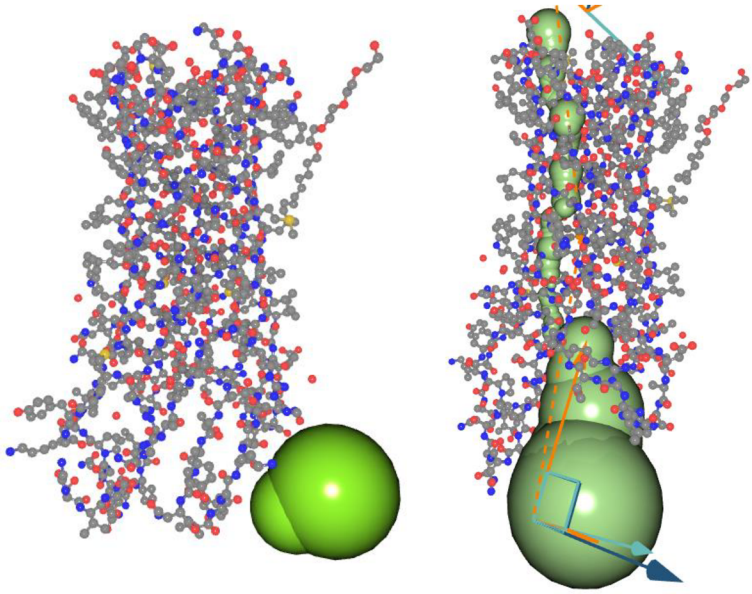
2. Otočenie okolo osi z o uhol ϑ:
   $x'' = x' * \cos\vartheta - y' * \sin\vartheta$
   $y'' = x' * \sin\vartheta + y' * \cos\vartheta$
   $z'' = z'$

A výsledný vektor bude mať súradnice (x'', y'', z'').

Po otočení štruktúry sa teda os póru zhoduje s osou x a môže nastať následná úprava koncových bodov tejto osi. Tie sú zmenené tak, aby boli na úrovni prvého respektíve posledného atómu z pohľadu jeho súradnice x.

To je zabezpečené pomocou dvoch jednoduchých metód `getMinX` a `getMaxX`, ktoré majú ako vstup kolekciu všetkých atómov a ich výstupom je najmenšia alebo najväčšia súradnica x. Táto zmena má dva dôvody. Prvým je snaha o odstránenie niekoľkých zbytočných krokov výpočtu, ktoré by počítali stred a priemer póru ležiaceho mimo molekuly proteínu a mohli by viesť k zacykleniu alebo nesprávnemu výsledku s pórom, ktorý by mal príliš veľký priemer na jeho koncoch.

Druhým je nasmerovanie výpočtu správnym smerom, kde je podstatný hlavne začiatočný bod. Čím ďalej sa tento bod od hlavnej časti štruktúry nachádza, tým ľahšie sa môže výpočet póru odchýliť od odhadnutej osi. V tomto prípade môže na povrchu nájsť iný otvor, ktorý ale nie je pór a tak sa celý výpočet vydá zlým smerom. Výsledkom toho môže byť buď len niekoľko počiatočných gúľ, ktoré namiesto pokračovanie správným smerom končia v slepej uličke (obrázok 11 a 12) alebo nájdenie neexistujúceho póru, ktorý vznikol vďaka tejto odchýlke. Týmto neexistujúcim pórom je myslená dutina, ktorá v skutočnosti neexistuje, ale výpočet si ju vytvoril tak, že prerušil pór v mieste väzby a pokračoval za ňou.



OBRÁZOK 11 A OBRÁZOK 12: UVIAZNUTIE VÝPOČTU V SLEPEJ ULIČKE (VĽAVO) A JEHO SPRÁVNY VÝSLEDOK (VPRAVO)

8.3. VÝPOČET PÓRU

Výpočet póru potom prebieha podľa vzoru PoreWalkeru. Najskôr sa určí hrúbka rezov. V základnom nastavení je to 1, čiže hrúbka rezov je 1Å. Rozdielom oproti algoritmu PoreWalker je, že v mojej implementácia si užívateľ pomocou parametru *Samplig* môže túto hodnotu upraviť, čo môže mať vplyv (dobrý, ale aj zlý) na výsledok výpočtu (viac v Kapitole 8.6).

Algoritmus teda nadväzuje na výsledky z kapitoly 8.2, ktoré určili a následne upravili odhadovanú os póru a vykonali rotáciu štruktúry, na ktorej celý výpočet prebieha. Prvý bod tejto osi sa považuje za počiatočný stred dutiny póru. Tento stred je následne optimalizovaný tak, aby sa do dutiny póru vošla guľa s čo najväčším priemerom (Algoritmus 2).

Optimalizácia prebieha tak, že sa pomocou funkcie *closestAtom* nájde atóm, ktorý je najbližšie danému stredu. Tento bod je následne posunutý o 0,1Å v smere od stredu tohto najbližšieho atómu. Pre zabránenie zacyklenia výpočtu je ale najskôr určený nový najbližší atóm a vypočítaný nový priemer dutiny. Len ak je nový priemer väčší ako súčasný, vykoná sa posun a postup sa opakuje s novým atómov. Ak je rovnaký alebo menší, výpočet sa zastaví a pokračuje sa s ďalším rezom. Keďže os póru zodpovedá osi x, menia sa pri tomto výpočte iba súradnice y a z. Táto optimalizácia prebieha do chvíle, kedy je priemer dutiny najväčší. V tej chvíli sa zastaví a uloží sa pozícia tohto stredu a jeho priemeru. Podľa jeho polohy sa pozmení aj os, po ktorej výpočet postupuje. A to tak, aby súradnice y a z bodu, ktorý predstavuje stred nasledujúceho rezu zodpovedali súradniciam y a z práve optimalizovaného bodu, čo zabezpečí, aby výsledok bol čo najsúvislejší a neobsahoval prerušenia, ktoré by mohli viesť k odchýleniu sa od osi, prípadne k ignorovaniu nejakej chemickej väzby alebo atómu.

Tieto kroky sú opakované až kým sa výpočet nedostane do koncového bodu, kde sa algoritmus zastaví a výsledný zoznam stredov a priemerov je predaný metóde pre vizualizáciu.

Súčasťou tohto algoritmu je aj detekcia resíduí obklopujúcich pór. Tu sa taktiež využíva výstup metódy *closestAtom*, kde sa pre každý atóm vždy zistí jeho rezíduum a uloží sa do zoznamu spolu so súradnicami, ktoré predstavujú stred rezídua a sú vypočítané z priemeru polôh všetkých atómov, ktoré ho tvoria.

```
Float add                    // hodnota o ktorú má byť vykonaný
                                posun,pôvodne nastavená na 0.1
Float ad            //vzdialenosť medzi daným atómom a stredom
                     dutiny rezu zodpovedajúca hodnote Sampling
while (true) {
   Vector3f vect    //vektor udávajúci polohu najbližšieho atómu
   if ((newStart.y - vect.y) < 0) {
        y = -add;
   } else {
        y = add;
   }
   if ((newStart.z - vect.z) < 0) {
        z = -add;
   } else {
        z = add;
   }

   newStart = new Vector3f(newStart.x, (newStart.y
           + y), (newStart.z + z));      //nový stred rezu
   Atom atom1                   //atóm najbližšie novému stredu
   Float ad1              //vzdialenosť atom1 od nového stredu
   if (ad1 > ad) {
        ad = ad1;
        atom = atom1;

   } else {
        break;
}}
```

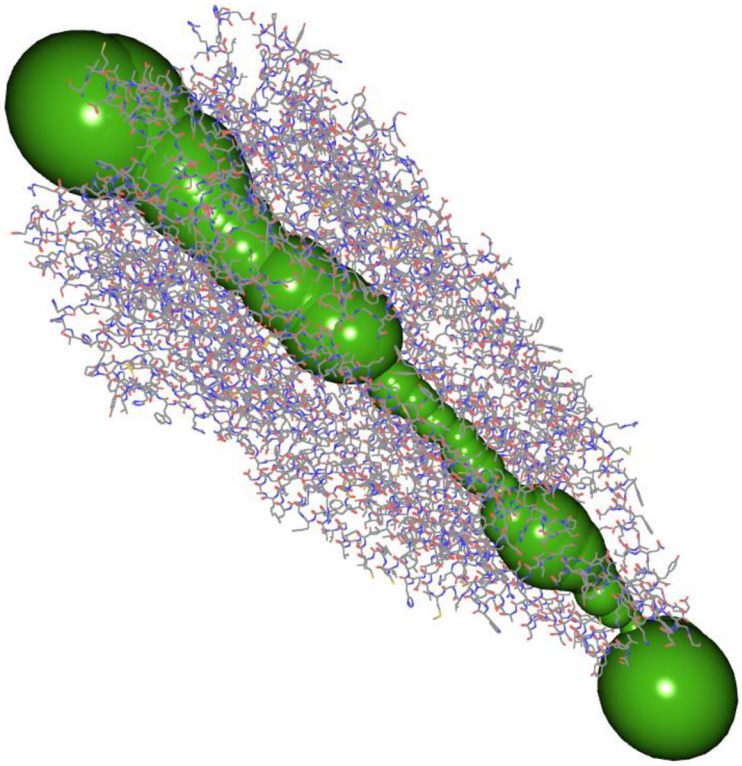
ALGORITMUS 2: OPTIMALIZÁCIA STREDU PÓRU PRE DANÝ REZ

8.4. VIZUALIZÁCIA PÓRU

Po ukončení optimalizácie je posledným krokom mojej implementácie vizualizácia vypočítaného póru (Obrázok 13) proteínu pomocou grafického rozhrania CAVER Analyst.

Pre každý rez proteínovou štruktúrou je vytvorený bod, v ktorom je pomocou vyššie popísaného algoritmu vypočítaný stred póru a jeho priemer. Tieto údaje sú následne uložené a ďalej využité pre vygenerovanie gule s daným stredom a priemerom tak, aby bola dutina póru v danom reze maximálne vyplnená. Farba je pre celý pór zhodná a z počiatku náhodná. Užívateľ ju ale môže ľubovoľne zmeniť pomocou rozhrania, ktoré CAVER ponúka.

Vizualizácia je posledným významným krokom algoritmu. Po jej dokončení nasleduje už len spätná rotácia štruktúry do jej pôvodne polohy pomocou inverznej rotácie.



OBRÁZOK 13: MOLEKULA NIKOTINOVÉHO ACETYLCHOLINOVÉHO RECEPTORU (PDB ID: 2BG9) A VIZUALIZÁCIA JEJ PÓRU

8.5. UKLADANIE INFORMÁCIÍ O PÓRE

Keďže samotná vizualizácia neponúka veľmi presné a pre ďalšie výpočty hodnotné dáta, ponúka výsledná implementácia možnosť exportu dát o vypočítanom póre do .csv súborov. Tieto súbory sú dva a ukladajú sa do priečinku "C:\\Caver\\Output":

1. Súbor so zoznamom rezíduí obklopujúcich pór - tento súbor obsahuje zoznam rezíduí obklopujúcich pór. Konkrétne v sebe nesie údaje o názve aminokyseliny (*AA identifier*), číslo rezídua (*AA number*) a jeho súradnice x,y,z. Ukladá sa vo formáte „poreLiningResidues_xxxx.csv“, kde xxxx predstavuje kód proteínu. Vzor:

   AA identifier;AA number;x;y;z
   ARG;41;-1.1677085;-0.3358334;0.29587492
   ARG;41;-1.251875;-0.04949997;0.24516658
   GLU;37;-1.3820002;-0.05193329;0.54993325
   ...

2. Súbor so zoznamom priemerov póru – obsahuje x-ové súradnice bodov, v ktorých prebieha výpočet priemeru póru a priemer póru v danom mieste. Taktiež je na konci riadok udávajúci slovný popis tvaru dutiny póru (viz Kapitola 7.4.2). Tento popis predstavujú 3 písmená – S pre cylinder, U pre rozširujúci sa kužeľ a D pre zužujúci sa kužeľ [15]. Súbor je uložený vo formáte „poreDiameters_xxxx.csv“, kde xxxx opäť predstavuje kód proteínu. Vzor:

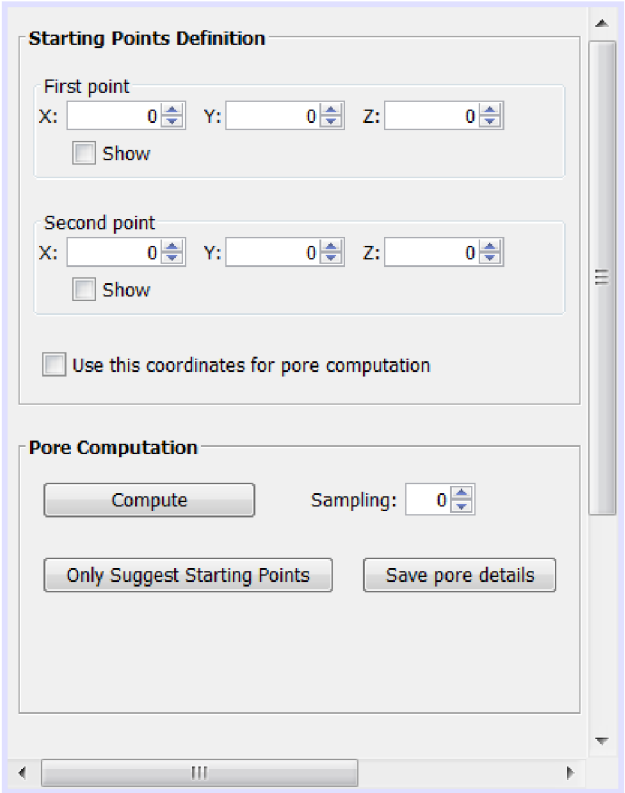
   coordinate x;diameter
   -31.887026;14.766039
   -30.887026;14.926449
   ...
   Identified shape: SDSDSUSUSUSUSDSUSDSUSUSU

[15] Napríklad pre štruktúru z obrázku 8 z kapitoly 7.4.2 by popis vyzeral takto – SDS.

8.6. INTEGRÁCIA ALGORITMU DO APLIKÁCIE CAVER ANALYST

Súčasťou zadania mojej diplomovej práce bola aj integrácia naimplementovaného algoritmu a jeho integrácia do grafického rozhrania aplikácie CAVER Analyst. Toto rozhranie sa skladá z komponentov, ktoré majú rozdielnu funkcionalitu, ako je napríklad výpočet tunelov, zmena zobrazenia štruktúry, interakcia so štruktúrou, zoznam reťazcov a aminokyselín a podobne.

Pre moju implementáciu bol vytvorený komponent (Obrázok 14) – `MoreSitesPoreDetectionTopComponent`, v grafickom rozhraní pod názvom Pore Detection.



OBRÁZOK 14: KOMPONENT PRE SPÚŠŤANIE ALGORITMU NA DETEKCIU PÓROV

Tento komponent poskytuje následujúce hlavné funkcie:

(1) Compute – spúšťa celkový výpočet póru, v ktorom sú zakomponované kroky popísané v kapitolách 8.1 až 8.4. Jeho súčasťou je teda aj odhad osi póru, ktorý je spustiteľný aj samostatne pomocou druhej funkcie tohto komponentu. Jeho výstupom je vizualizovaný pór v danej proteínovej štruktúre (Obrázok 13). Pred samotným spustením výpočtu si užívateľ môže vybrať čo všetko bude vo výsledku zobrazené. Výber sa skladá zo štyroch možností – *pore only* (čistá vizualizácia póru), *add axis* (vizualizácia póru aj s predbežnou osou, ktorá bola pri jeho výpočte použitá), *add sets of atoms* (vizualizácia póru, pri ktorej sú farebne rozlíšené skupiny atómov použité pre výpočet póru v jednotlivých rezoch) a *add axis and sets of atoms* (kombinácia predošlých dvoch možností).

    (a) Sampling – parameter nastavujúci hrúbku rezov pri výpočte rezov. Pomocou zmeny tohto parametru môže mať algoritmus v niektorých prípadoch o niečo lepšie výsledky. Implicitná hodnota je 1 a odporúčaný rozsah nastavení je $0{,}5 \leq \text{sampling} \leq 3$.

(2) Only Suggest Starting Points – spúšťa iba algoritmus na predbežný odhad osi póru spolu s jeho začiatkom a koncom, ktorý je popísaný v kapitole 8.1. Pred samotným spustením je pomocou dialógového okna prezentovaná možnosť výberu zobrazenia prípadne nezobrazenia pomocných vektorov a odhadovaných bodov začiatku a konca póru (*suggested helpers*).

(3) Starting point definition – je časť komponenty, ktorá poskytuje nie len informačnú, ale aj funkčnú časť. V základnom stave len zobrazuje súradnice predpovedaného začiatku a konca póru. Tie môžu byť výsledkom oboch predchádzajúcich funkcií. Niektoré proteíny sú ale veľmi nepravidelné a tak môže byť výstup algoritmu pre odhad osy póru nepresný. Práve pre tieto prípady je možné využiť manuálne určovanie súradníc

začiatku a konca póru, ktoré pri tomto druhu proteínov môže viesť k lepšiemu výpočtu celého póru. Pôvodne určené súradnice môžu teda byť užívateľom upravené a pri zaškrtnutí checkboxu *Use this coordinates for pore computation* následne použité výpočte samotného póru. Tieto hodnoty sa dajú upraviť priamym zadaním súradníc x,y,z pre začiatočný a/alebo koncový bod. Druhou možnosťou pre túto úpravu je zobraziť si nástroj pre definíciu súradníc daného bodu pomocou checkboxu *Show*, ktorý je prítomný pre oba konce póru. Týmto nástrojom je možné pohybovať a umiestniť tak daný bod do požadovaného miesta.

(4) Save pore details – ponúka možnosť uloženia informácií o póre. Na výber je generácia buď len jedného zo súborov alebo oboch.

9. POROVNANIE S ALGORITMOM POREWALKER

Pre porovnanie výsledkov vlastnej implementácie (ďalej VI) s výsledkami PoreWalkeru použijem dáta dostupné z jeho oficiálnej stránky [18]. Bohužiaľ náročnosť algoritmu alebo dĺžka výpočtu nie sú poskytované a tak sú na porovnanie vhodné tieto 3 druhy dát:

1) Vizualizácia póru.

2) Graf zmien priemeru póru podľa osi x.

3) Popis tvaru dutiny póru

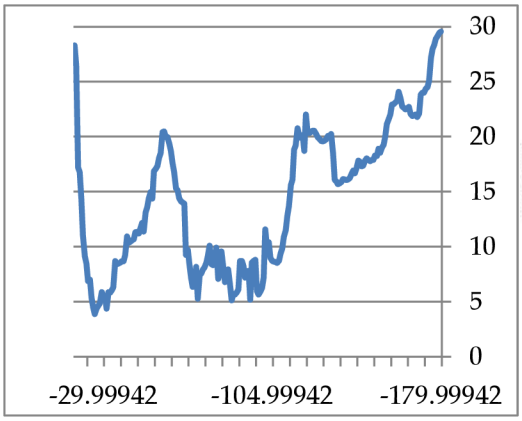
Pre toto porovnanie som vybral 4 molekuly (z celkových 30 testovaných), v zátvorkách sú uvedené PDB ID kódy - nikotinový acetylcholinový receptor (2bg9), MscS kanál (2oau), transportér pre amónny ión Amt-1 (2b2f) a NA^+ komplex sodno-draselného kanálu (2ahy). Pomocou aplikácie CAVER Analyst (CA) s už implementovaným algoritmom (sampling = 0,5) som sa snažil zachytiť štruktúry s pórmi tak, aby sa dali čo najlepšie porovnať s vizualizáciami vytvorenými PoreWalkerom (PW) a taktiež pre každú štruktúru vytvoriť graf s rovnakými údajmi – osa x zodpovedá súradnici x osi póru a os y zase priemeru póru. Smer údajov na ose x zľava doprava na obrázkoch zodpovedá smeru výpočtu zdola nahor.

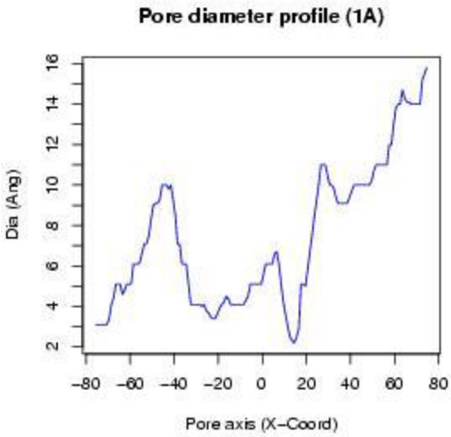
Prvým príkladom je molekula s PDB ID 2bg9. Jej štruktúra je relatívne pravidelná a uzavretá a obsahuje jedinú väčšiu dutinu, ktorú je vidieť v prvej tretine grafu 1 medzi súradnicami -21 až – 71. Graf 2 má rovnakú výchylku v oblasti -60 až -30. Na obrázku 15 získaného pomocou PW je táto odchýlka viditeľná o niečo menej ako z vizualizácie pomocou CV (Obrázok 16).

<u>Popis tvaru dutiny póru:</u>

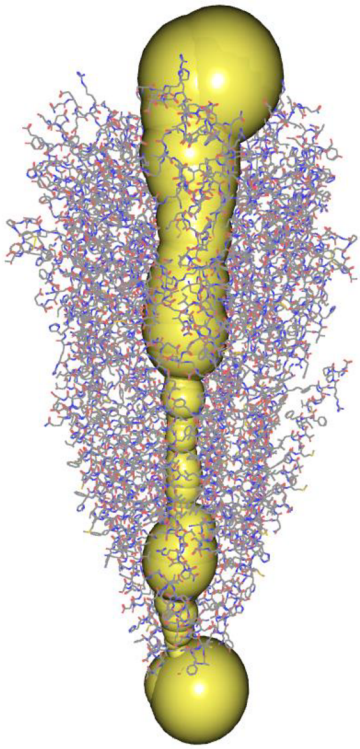
PoreWalker – UDSUDUD

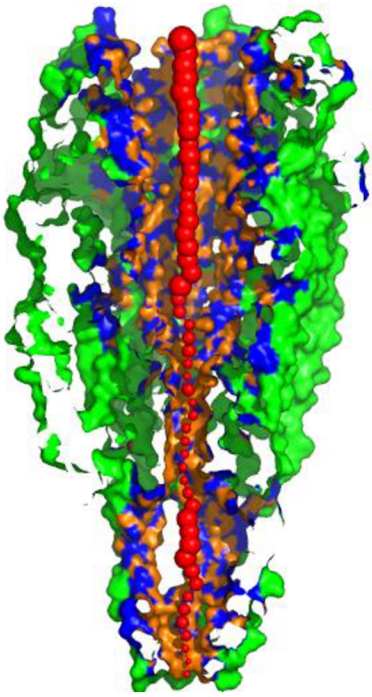
VI – SDSDSDSUSDUSUDSDSDSUDUDSUSUSDSUSUSDSUDSDSUD USUSUSDSDSDSUDSDSUSU





GRAF 1(CA) A GRAF 2(PW): ZMENA PRIEMERU PÓRU MOLEKULY 2BG9





OBRÁZOK 15(CA) A OBRÁZOK 16(PW) : VIZUALIZÁCIA MOLEKULY 2BG9

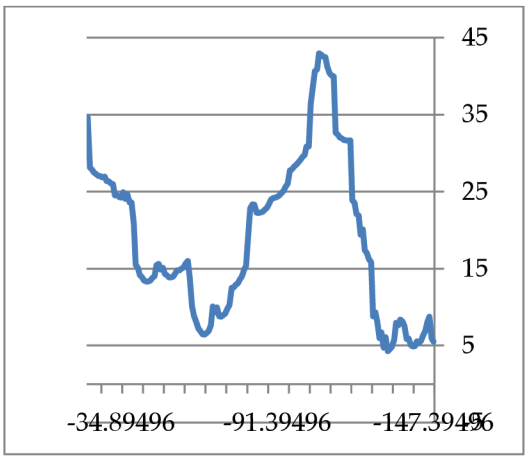
Príklad s molekulou s PDB ID 2oau je podobný, ale sú tam o niečo viac vidieť rozdiely. Štruktúra je taktiež pravidelná a dutina póru uzavretá. Obsahuje širšie miesto na začiatku, za ním nasleduje zúžené miesto a potom ďalšia, väčšia, dutina. Grafy 2 a 3 pre túto molekulu majú opäť porovnateľný vývoj zachytávajúci obe rozšírenia dutiny, prvé hneď na začiatku a druhé v druhej polovici grafu. Rozdiel je opäť

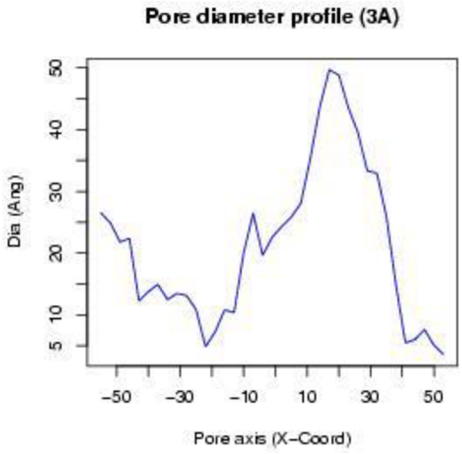
viditeľný na obrázkoch (Obrázky 17 a 18). CV algoritmus vypĺňa dutiny guľami s maximálnym priemerom, v prípade PW nie je toto rozšírenie až tak veľké.

<u>Popis tvaru dutiny póru:</u>

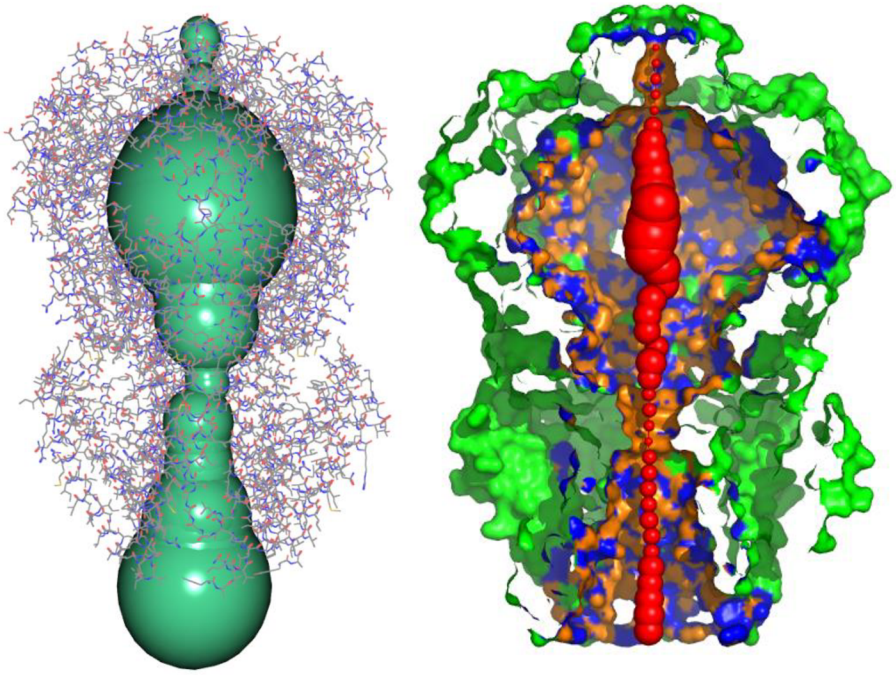
PoreWalker – DUD

VI – SUSUSDSUDUSUSUSUSUSDSDSDSDSUSDS DSUSDSUSDSUSU





GRAF 3(CV) A GRAF 4(PW): ZMENA PRIEMERU PÓRU MOLEKULY 2OAU



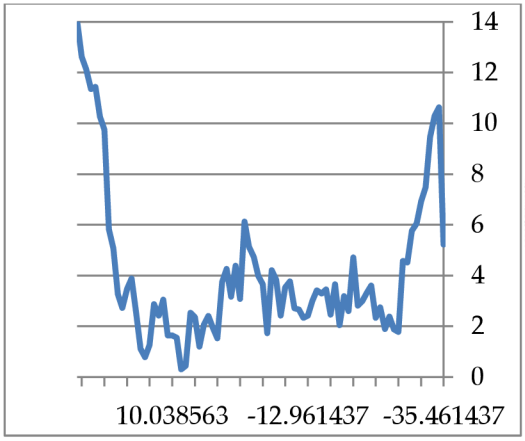
OBRÁZOK 17(CV) A OBRÁZOK 18(PW): VIZUALIZÁCIA MOLEKULY 2OAU

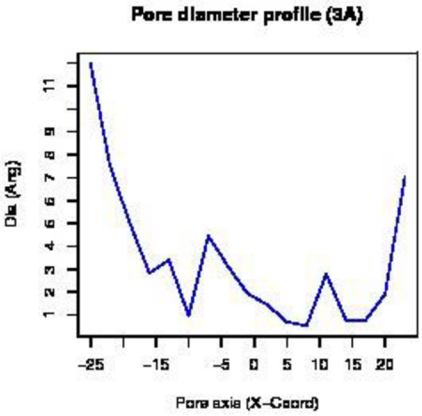
Ako tretí príklad slúži molekula s PDB ID 2b2f. Póru sa viac odchyľuje od priameho smeru, ale oproti prechádzajúcim príkladom neobsahuje žiadne väčšie dutiny a priemery sú teda veľmi podobné, čo potvrdzujú aj grafy 5 a 6. Hlavným rozdielom je opäť celkový priemer póru, ktorá je z obrázkov 19 a 20 dobre viditeľná.

<u>Popis tvaru dutiny póru:</u>

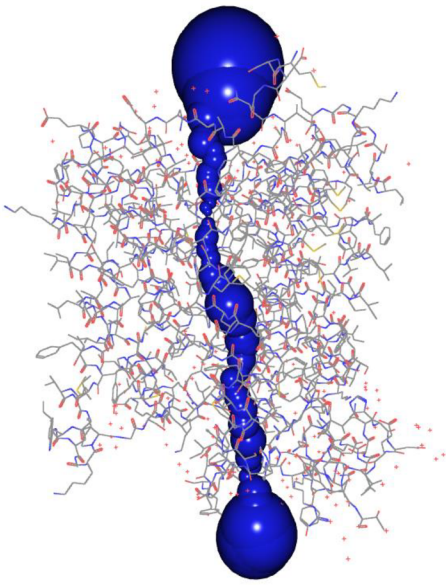
PoreWalker – DSU

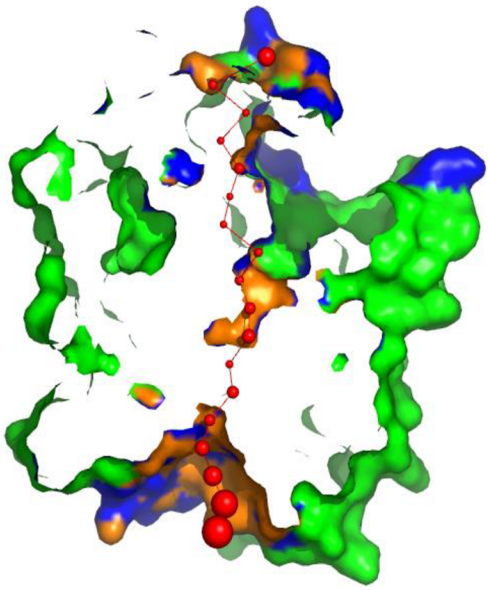
VI – SDSDSDSUSDUSUDUSDUSUDSUSDSUSDUSUSUSU



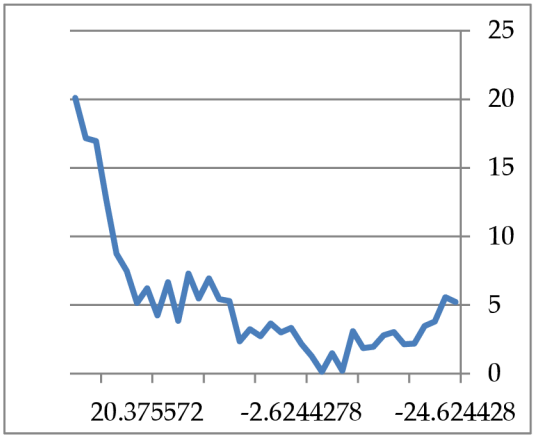


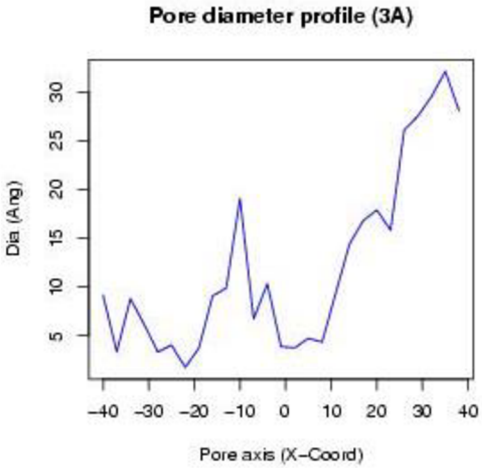
GRAF 5(CV) A GRAF 6(PW): ZMENA PRIEMERU PÓRU MOLEKULY 2B2F



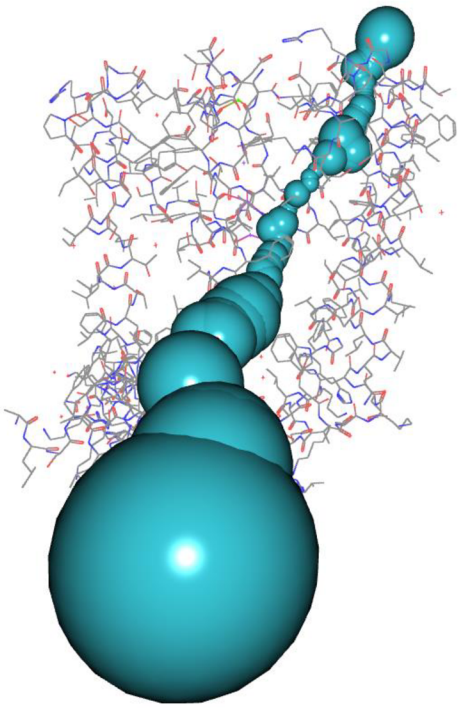


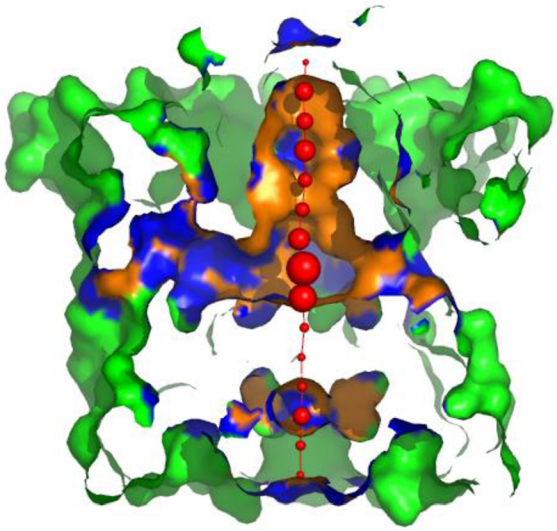
OBRÁZOK 19(CV) OBRÁZOK 20(PW): VIZUALIZÁCIA MOLEKULY 2B2F





GRAF 7(CV) A GRAF 8(PW): ZMENA PRIEMERU PÓRU MOLEKULY 2AHY





OBRÁZOK 21(CV) A OBRÁZOK 22(PW): VIZUALIZÁCIA MOLEKULY 2AHY

Posledným príkladom je molekula s PDB ID 2ahy (Obrázok 21 a 22). Molekula je nepravidelná s jednou relatívne veľkou dierou a niekoľkými menšími. Práve po dosiahnutí tejto dutiny sa výpočet veľmi odchýlil a výsledné grafy a vizualizácie sú teda odlišné. Ukazuje to na horší výkon mojej implementácie v prípade, že ide o podobný typ molekúl s nepravidelnou štruktúrou.

Popis tvaru dutiny póru:

PoreWalker – SUD

VI – SDSDUSDUSUSUSUDUSDUDUSDUSUSUSU

Z obrázkov sa môže často zdať, že aj keď by to mala byť implementácia v podstate rovnakého algoritmu, tak sú výsledky veľmi rozdielne. V prípade mojej implementácie póry pokračujú čiastočne aj mimo molekulu a sú taktiež viditeľne širšie.

Pre prvý rozdiel je vysvetlením rozdiel vo výpočte začiatku a konca póru a určenia prvého a posledného rezu pre výpočet póru. V mojom prípade je to súradnica x, ktorá je z pohľadu osi x najvyššie respektíve najnižšie. PoreWalker využíva kritérium, ktoré nie je zverejnené a tak nie je možné tento postup porovnať.

Rovnaká situácia platí aj v druhom prípade, kde ide o šírku pórov. Dokumentácia PoreWalkeru uvádza zhodný postup s mojou implementáciou a to, že priemer póru je počítaný tak, aby sa do dutiny vošla guľa s maximálnym priemerom. Nie je uvádzané žiadne ďalšie kritérium, ktoré by šírku póru na určitých miestach obmedzovalo a prekážalo by jeho rozšíreniu. Pri bližšom pohľade na molekulu sa ale dajú nájsť dutiny, kde by pór mohol mať väčší priemer než aký udáva algoritmus PoreWalker.

Veľký rozdiel je taktiež v popise tvaru dutiny póru. Keďže zmeny priemeru póru sú v prípade PoreWalkeru veľmi malé, je tento popis vždy veľmi krátky (<10 znakov). V prípadne vlastnej implementácie sa opäť opakuje rozdiel popísaný vyššie. Pór sa vždy rozťahuje do najväčšej možnej šírky a tak sú zmeny medzi jednotlivými bodmi relatívne veľké, čo sa nakoniec prejavuje vo veľkej dĺžke popisu dutiny póru (>20 znakov).

Podobnosť výpočtu ale ukazujú vývoje grafov 1 až 6. Z nich je vidieť, že aj keď majú póry odlišnú šírku, tendencia zmien týchto šírok

je podobná. Najväčším rozdielom sú opäť prvé a posledné body výpočtu.

Ďalším možným kritériom pre porovnanie týchto dvoch algoritmov by mohli byť rezíduá obklopujúce pór. Výsledky toho majú ale rovnakú tendenciu ako vyššie uvedené kritériá. Vlastná implementácia spravidla uvádza menší počet týchto rezíduí. Tento rozdiel (po odobraní duplicít) kolísal od veľmi malého - 21 oproti 34 (2ahy), až k veľkým rozdielom – 39 oproti 95 (2oau). Pričom spoločné rezíduá taktiež kolísali v rozpätí od 30% - 70%.

9.1. RÝCHLOSŤ VÝPOČTU PÓRU

PoreWalker ponúka len webové rozhranie, ktoré užívateľovi výsledky buď zobrazí z databázy už vypočítaných alebo po výpočte zašle email s odkazom na výsledky. Tieto výsledky však rýchlosť výpočtu póru neposkytujú a tak táto podkapitola len ponúkne prehľad o rýchlosti vlastnej implementácie.

Na vzorke 30 proteínov, ktoré boli vybrané náhodne dosiahol výsledky, ktoré popisuje Tabuľka 1. Hodnoty v zátvorkách boli dosiahnuté po manuálnej úprave začiatočného a koncového bodu výpočtu pomocou funkcie popísanej v kapitole 8.6. Tabuľka 2 zase ukazuje štatistiky dosiahnuté pri výpočte testovacej skupiny proteínov. Z uvedených výsledkov je vidieť, že veľkú časť výpočtov (v tomto prípade 73,3%) algoritmus vykoná v čase pod 1 sekundu. Problémové sú ale veľmi zložité a nepravidelné štruktúry, ktoré môžu výpočet výrazne predĺžiť (1ar1) alebo viesť k zlému výsledku (1brd, 1bxw).

Kód	ms	Kód	ms	Kód	ms
1ap9	30	1fi1	16170	2b2f	20
1ar1	70586	1fqy	20	2bg9	1937
1at9	10733	1fw2	25	2oau	35
1bl8	30	1h6i	20	3emo	30
1brd	282625 (35)	1h6s	20	4ltr	30
1bxw	40042 (30)	1i78	16327	4mee	10
1c17	148	1ih5	10075	4xnk	1818
1ehk	60	1j95	20	5eke	70
1ek9	129	1jvm	20	5ekp	70
1fcp	60	2ahy	28318	7ahl	240

TABUĽKA 1 – ČAS VÝPOČTU PRE JEDNOTLIVÉ PROTEÍNY

Pred úpravou		Po úprave	
Priemer	15990,9	Priemer	5237,2
Max	282625	Max	70586
Min	10	Min	10

TABUĽKA 2 – ŠTATISTIKY PRE DĹŽKU VÝPOČTU

9.2. ZHODNOTENIE

Vyhodnotenie porovnania týchto dvoch algoritmov je relatívne zložité. Väčšina výstupov je dostatočne podobná na to, aby sa dalo povedať, že moja implementácia dáva výsledky, ktoré sú blízko realite a tak nie je zlá. Na druhú stranu niektoré rozdiely sú dosť veľké a tak je vidieť, že nejde o totožnú implementáciu tohto algoritmu.

Hlavnými dôvodmi týchto rozdielov sú:

1. Chýbajúci výstup programu NACCESS, ktorý PoreWalker využíva. Jeho výstupom je relatívna prístupnosť postranného reťazca pre rozpúšťadlo, čo môže byť významným faktorom pri výpočte póru. Tento program je ale dostupný len pre Linux a tak ho nebolo možné využiť v mojej implementácií. Zakomponovanie tohto výpočtu by mohlo byť návrhom na ďalšiu prácu.

2. PoreWalker neuvádza dôvody, pre ktoré má jeho výstup veľmi malý priemer póru a nevypĺňa niektoré dutiny úplne, aj keď v popise stojí, že by priemer mal byť čo najväčší možný. Moja implementácia sa drží popisu uvedeného v oficiálnej dokumentácii a vedie tak k rozdielom vo výsledkoch. A to nie len v prípade vizualizácie, ale následne aj v rozdielnych priemeroch dutiny a tak aj popisu tejto dutiny.

	PoreWalker				Vlastná Implementácia			
	Priemer	Min	Max	Počet	Priemer	Min	Max	Počet
1ap9	4.379	0.607	8.778	17	3.971	0.536	11.439	60
1at9	2.684	0.167	5.214	17	3.293	0.315	8.202	61
1bl8	5.256	0.762	19.424	17	5.589	0.82	15.782	56
1bxw	4.439	0.156	10.117	15	4.224	0.796	9.234	61
1c17	21.43	15.007	26.061	19	20.47	12.241	41.012	70
1ehk	3.82	0.084	9.34	24	3.344	0.099	12.756	78
1ek9	19.793	4.764	28.297	46	14.241	0.978	23.131	151
1fcp	4.723	0.046	16.95	23	5.483	1.081	22.424	80
1fi1	5.101	0.653	17.642	22	5.472	0.139	24.349	81
1fgy	5.977	2.606	9.418	14	7.602	2.501	18.993	55
1fw2	4.25	1.692	11.24	13	4.085	0.282	12.16	53
1h6i	6.422	1.331	11.267	14	7.695	2.397	19.218	55
1h6s	16.046	7.095	21.842	13	16.591	8.781	28.294	47
1i78	3.635	0.319	13.761	20	5.178	0.055	20.673	73
1ih5	6.011	2.903	13.172	13	8.5	1.349	40.805	60
1j95	4.588	0.424	12.06	16	4.743	0.403	16.467	55
1jvm	4.801	1.151	14.262	18	4.966	0.011	16.469	58
2ahy	5.48	0.701	14.09	13	5.496	0.13	22.567	57
2b2f	3.362	0.536	11.98	17	4.953	0.202	15.032	60
2bg9	16.331	3.868	29.429	49	14.458	4.114	29.55	163
2oau	21.149	3.581	49.686	37	18.845	4.349	42.954	125
3emo	3.14	0.361	6.555	29	3.71	0.719	8.929	100
4ltr	6.283	1.745	15.024	34	7.02	1.772	16.58	112
4mee	5.68	1.26	12.52	20	6.051	0.503	21.368	70
4xnk	7.257	0.627	18.766	17	8.223	0.162	26.56	65

TABUĽKA 3 – ŠTATISTIKA PRIEMEROV DUTÍN PÓROV

Tabuľka 3 [16] ukazuje štatistiku pre výpočet pórov. Tu môžeme vidieť, že aj keď je rozdiel v počte rezov často veľmi veľký, ostatné údaje nie sú vo väčšine prípadov až také rozdielne.

Dá sa teda povedať, že aj napriek niektorým chýbajúcim údajom v dokumentácii PoreWalkeru sa moja implementácia blíži požadovanému výsledku a po menšom odladení a dodatočnej implementácii výpočtu relatívnej prístupnosti postranného reťazca pre rozpúšťadlo, môže dávať rovnako kvalitné výsledky.

[16] Štatistiky sú zobrazené len pre 25 z 30 proteínov. Pre chýbajúcich 5 PoreWalker ani po niekoľkých dňoch od zadania požiadavku stále nedoručil výsledok.

10. ZÁVER

Táto diplomová vo svojej teoretickej časti poskytuje náhľad do problematiky štruktúry proteínov zo zameraním na skupinu transmembránových proteínov. Práve táto skupina vytvára vo svojej štruktúre dutiny, nazývané póry, pomocou ktoré umožňujú napríklad komunikáciu bunky so svojím okolím, prenos iónov do bunky a z nej a aktiváciu niektorých vnútrobunkových procesov.

Jedným z cieľov, bolo štúdium podstaty aktuálne najpoužívanejšieho algoritmu na detekciu pórov – PoreWalkeru. Práca teda bližšie popisuje jednotlivé kroky, z ktorých je výpočet tohto algoritmu zložený a taktiež ukazuje, akou formou sú prezentované jeho výsledky.

Hlavným cieľom, a zároveň náplňou praktickej časti, bola implementácia algoritmu PoreWalker v programovacom jazyku Java a jeho integrácia do aplikácie CAVER Analyst. Vytvorená implementácia a aj jej následné zaintegrovanie sú vysvetlené. Spolu s nimi je tu aj popis grafického rozhrania na spúšťanie algoritmu a taktiež popis výslednej vizualizácie.

V poslednej časti boli z desiatok testovacích štruktúr vybraté 4, ktoré reprezentujú najčastejšie typy transmembránových proteínov. Na nich boli ukázané výsledky vytvorenej implementácie a tie boli porovnané s výsledkami algoritmu PoreWalker. Súčasťou porovnania je aj diskusia, ktorá popisuje rozdiely vo výsledkoch týchto dvoch algoritmov.

Do budúcna by bolo možné algoritmus optimalizovať tak, aby úspešne pokrýval aj veľmi netypické štruktúry transmembránových proteínov. Medzi ne patria napríklad veľmi nepravidelné alebo neuzavreté štruktúry, obsahujúce veľké dutiny, prípadne molekuly, ktorých pór má viac ako len jeden začiatok alebo koniec.

11. LITERATÚRA

[1] BERG, Jeremy M, John L TYMOCZKO a Lubert STRYER. Biochemistry. 7th ed. New York: W.H. Freeman and Company, c2012, xxxii, 1098 s., [78] s. příl. ISBN 9781429276351.

[2] About us. CaverSoft – software tool for protein analysis and visualisation . [online]. [cit. 2015-05-15]. Dostupné z: http://www.caversoft.com/index.php?sid=160.

[3] Biological Membranes [online]. 1996. [cit. 2015-05-07]. ISBN 0-904498-32-8. Dostupné z: http://www.biochemistry.org/portals/0/education/docs/basc08_full.pdf.

[4] Modern Biology. Biological Membranes. [online]. 2006 [cit. 2015-05-07]. Dostupné z: http://telstar.ote.cmu.edu/biology/animation/Membranes/membranes.html.

[5] ALLEN, James P. Biophysical chemistry. Oxford: Wiley-Blackwell Pub., 2008, xvi, 492 p. ISBN 9781405124362.

[6] VOET, Donald a Judith G VOET. Biochemistry. 4th ed. Hoboken, N.J.: John Wiley & Sons, c2011, xxv, 1428, 53 s. ISBN 9780470570951.

[7] VOET, Donald, Judith G VOET a Charlotte W PRATT. Principles of biochemistry. 4th ed. Hoboken, N.J.: John Wiley & Sons, c2013, xxxii, 1077, [88] s. příl. ISBN 9781118092446.

[8] Uic.edu. Chemistry and the Building Blocks of Life [online]. [cit. 2015-05-20]. Dostupné z: http://www.uic.edu/classes/bios/bios100/ lectures/ chemistry.htm

[9] LODISH, Harvey F. Molecular cell biology. 7th ed. New York: W. H. Freeman, c2013, xxxiii, 1154 s. ISBN 9781429234139.

[10] KUKOL, Andreas. Molecular modeling of proteins. Totowa, N.J.: Humana Press, c2008, xi, 390 p. ISBN 9781588298645.

[11] RCSB PDB. WWPDB. RCSB PDB [online]. 2014-04-30 [cit. 2015-05-03]. Dostupné z: http://www.rcsb.org/pdb/home/ home.do

[12] Oracle. Oracle. [online]. [cit. 2015-05-13]. Dostupné z: http://www.oracle.com/index.html

[13] Write once, run anywhere?. ComputerWeekly.com. [online]. 2002 [cit. 2015-05-12]. Dostupné z: http://www.computerweekly.com/ feature/Write-once-run-anywhere

[14] Welcome to the NetBeans Community. NetBeans. [online]. 2015 [cit. 2015-05-18]. Dostupné z: https://netbeans.org/about/ index.html

[15] Eva Chovancova et al. "CAVER 3.0: A Tool for the Analysis of Transport Pathways in Dynamic Protein Structures". In: PLOS Computational Biology (2012). DOI: 10.1371/journal.pcbi.1002708. URL: http://www.ploscompbiol.org/article/info:doi/10. 1371/ journal.pcbi.1002708..

[16] Kozlikova B et al. "CAVER Analyst 1.0: graphic tool for interactive visualization and analysis of tunnels and channels in protein structures." In: Bioinformatics 30 (2014), s. 2684–2685.

[17] Pellegrini-Calace M, Maiwald T, Thornton JM (2009) PoreWalker: A Novel Tool for the Identification and Characterization of Channels in Transmembrane Proteins from Their Three-Dimensional Structure. PLoS Comput Biol 5(7): e1000440. doi:10.1371/journal.pcbi.1000440

[18] PoreWalker. PoreWalker home page. [online]. [cit. 2015-05-16]. Dostupné z: http://www.ebi.ac.uk/thornton-srv/software/ PoreWalker/

[19] Welcome to the Naccess Home Page. Naccess homepage. [online]. 24.5.2015 [cit. 2015-05-22]. Dostupné z: http://www.bioinf. manchester.ac.uk/naccess/