

SVEUČILIŠTE U ZAGREBU
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

DIPLOMSKI RAD br. 1044

ALAT ZA ANALIZU POVRŠINA PROTEINA I
MJESTA PROTEINSKIH INTERAKCIJA

Josip Mihel

Zagreb, rujan 2006.

Zahvala

Zahvaljujem se svojoj obitelji: ocu Ivanu, majci Veri i bratu Ivanu na potpori i razumijevanju tijekom cijelog studija (razumijevanja je trebalo); svim novim prijateljima koje sam stekao tokom studija, neki od vas su doživotni; djevojci Sanji koja me bodrila kad je bilo najteže i pomogla oko slika sadržanih u ovom radu; Mili Šikiću na izboru zanimljive teme rada, konstantnom poticaju na rad i stjecanje novih znanja, te njegovoj otvorenosti prema svim mojim idejama.

Sadržaj

1	Uvod	4
2	Analiza i razrada zadatka	5
3	Analiza geometrijske strukture proteina.....	8
3.1	Otapalu dostupno područje površine.....	8
3.1.1	Otapalu dostupno područje površine atoma.....	8
3.1.2	Otapalu dostupno područje površine aminokiselina	11
3.2	Izbočenost atoma i aminokiselina u proteinima	12
3.2.1	Izbočenost atoma – CX algoritam	13
3.2.2	Izbočenost aminokiselina	15
3.3	Ukopanost atoma i aminokiselina u proteinima	15
3.3.1	Ukopanost atoma – DPX algoritam	16
3.3.2	Ukopanost aminokiselina	18
3.4	Prikaz proteina pomoću 3D mreže atoma	19
3.4.1	Stvaranje i pretraživanje 3D mreže	19
3.4.2	Rezultati provjere ubrzanja algoritama	23
4	Analiza proteinskih interakcija	25
4.1	Veze između aminokiselinskih residuuma	25
4.1.1	Ionske veze	26
4.1.2	Vodikove veze.....	27
4.1.3	Van der Waalsove veze	27
4.1.4	Hidrofobne veze	27
4.2	Interakcije na osnovu udaljenosti između residuuma	28
4.2.1	Interakcije na osnovu udaljenosti između jezgri atoma	28
4.2.2	Interakcije na osnovu udaljenosti Van der Waalsovih radijusa	29
4.3	Mjesta proteinskih interakcija na osnovu promjene ASA.....	29
4.4	iPFAM baza proteinskih interakcija	30
4.5	LSS&Ruđer algoritam za definiranje interakcija	31
4.5.1	Ionske veze	31
4.5.2	Vodikove veze.....	32
4.5.3	Van der Waalsove veze	32

4.5.4	Hidrofobne veze	33
4.5.5	Rezultati provjere rada algoritma	33
4.6	Usporedba rezultata različitih algoritama	34
4.6.1	Usporedba pronađenih interakcijskih parova	34
4.6.2	Usporedba pronađenih vezujućih residuuma	37
4.7	Predstavljanje residuuma pomoću koordinata jezgre C _α atoma ..	40
5	PSAIA – Protein Structure And Interaction Analyzer	42
5.1	Modul za analizu svojstva geometrijske strukture	43
5.1.1	Podešavanje rada modula	43
5.1.2	Upravljanje radom modula	47
5.1.3	Upravljanje učitanim datotekama	48
5.2	Modul za analizu interakcija	48
5.2.1	Podešavanje rada modula	49
5.2.2	Upravljanje radom modula	51
5.2.3	Upravljanje učitanim datotekama	51
5.3	Usporedba sa postojećim programima	51
6	Zaključak	53
7	Dodatak A: analiza nukleinskih kiselina	54
8	Dodatak B: XML sheme	56
8.1	Svojstva geometrijske strukture	56
8.2	Interakcijski parovi	58
8.3	Mjesta interakcija (vezujući residuumi)	59
8.4	Vezujuće stanje residuuma	60
9	Literatura	62

1 Uvod

Međusobne interakcije proteina jedno su od najvažnijih svojstava funkcioniranja živih organizama pa njihovo proučavanje danas predstavlja jedno od važnijih područja proučavanja u biokemiji i bioinformatici. Proučavanje interakcija između proteina usko je vezano uz područje proučavanja geometrijskih svojstava proteina.

Područje bioinformatike koje se danas znatno razvija je pokušaj predviđanja interakcija između proteina (čija mjesta i načini interakcije još nisu poznati), tj. pronalaženje onih mjesta na kojima može doći do interakcije između različitih proteina i onih parova aminokiselina koje mogu međusobno sudjelovati u interakciji.

Do danas razvijeni alati za proračun geometrijskih svojstava proteina imaju sposobnost izračuna samo jednog od važnijih svojstava, a alati za analizu interakcija između proteina gotovo i da ne postoje (izuzev baza podataka). Drugi veliki nedostatak postojećih softvera je nemogućnost istovremene analize više proteinskih struktura. Iz tih razloga poprilično je komplicirano pripremiti podatke na kojima bi se moglo zasnivati predviđanje proteinskih interakcija ili sama analiza svojstava proteina.

U svrhu olakšavanja pripreme podataka za danju analizu, u sklopu ovog rada potrebno je izraditi softverski alat koji će objediniti izvršavanje analize geometrijske strukture proteina i analize mjesta interakcije između proteina čije su strukture poznate. Potrebno je istražiti, razviti i implementirati vlastiti algoritam za identifikaciju mjesta interakcije i aminokiselinskih parova koji sudjeluju u proteinskim interakcijama.

U ovom radu biti će opisani algoritmi za analizu geometrijske strukture proteina i analizu proteinskih interakcija implementirani u razvijenom softverskom alatu. Te će biti dane usporedbe sa već poznatim alatima.

2 Analiza i razrada zadatka

Postupcima kristalografije X-zrakama i NMR-a *in vivo* danas je pronađen i zapisan veliki broj struktura proteina koji se nalaze u živim organizmima. Strukture samostalnih proteina i peptida stvorenih interakcijom različitih proteina nalaze se u bazama podataka koje su svima dostupne putem interneta. Jedna od najvažnijih i najvećih takvih baza podataka je *Protein Data Bank* (PDB) (<http://www.rcsb.org>), koja danas sadrži 38320 struktura (22.08.2006.). Postoje i baze podataka koje sadrže interakcije između proteina, proračunate prema različitim geometrijskim i kemijskim pravilima. Primjer takve baze je iPFAM baza proteinskih interakcija (<http://www.sanger.ac.uk/Software/Pfam/iPfam/>) koja sadrži interakcije između poznatih proteinskih domena.

Do sada se većina istraživanja proteinskih interakcija temeljila na istraživanju sekvence aminokiselina i dijelova površine proteina koje sudjeluju u interakciji. Pokušavalo se predvidjeti mjesta interakcije iz same sekvence (Ofra, Rost 2003.) ili koristeći dijelove površine kojima se protein veže (Jones, Thornton 1997.) . Pošto geometrijske osobine proteina kao što su otapalno dostupno područje površine, ukopanost, izbočenost i hidrofobnost aminokiselina usko vezane uz međusobne interakcije proteina logično bi bilo pokušati predvidjeti proteinske interakcije uz pomoć tih parametara.

Softverski alat koji je potrebno razviti mora se sastojati od dva modula:

- a) modul za analizu geometrijske strukture proteina
- b) modul za analizu interakcija između proteina

Struktura proteina, tj. peptida, mora se učitavati iz datoteka PDB baze podataka. Iako je format PDB datoteka definiran (službena definicija može se pronaći na: http://www.rcsb.org/pdb/static.do?p=file_formats/pdb/index.html), on nije u svim aspektima strogo određen i može sadržavati određene pogreške. Za vrijeme razvoja alata potrebno je detaljno proanalizirati sadržaje što više PDB datoteka, identificirati što više njihovih mogućih pogrešaka i pronaći načine kako ih ispraviti tako da nemaju značajniji utjecaj na analizu.

Prije samog kretanja u razvoj modula za analizu geometrijskih svojstava bilo je potrebno odabrati algoritme koji će se implementirati iz skupine već poznatih algoritama. Potrebno je bilo odrediti načine na koje se definiraju određena geometrijska svojstva proteina i kako se ona mogu protumačiti. Algoritmi koje je potrebno implementirati moraju izvršavati izračune otapalu dostupnog područja površine (engl. *Solvent Accessible Surface Area*), ukopanosti (engl. *Depth*) i izbočenosti (engl. *Protrusion*) atoma i aminokiselina koji grade proteine. Važne odlike odabranih algoritama moraju biti brzina izračuna vrijednosti i korisnost podataka dobivenih njihovom upotrebom. Brzina izvođenja svakog algoritma usko je povezana sa njezinom implementacijom, stoga je jako važno osmisliti na koji način prikazati i kako pretraživati geometrijske strukture proteina u memoriji računala. Nakon proučavanja različitih algoritama za analizu geometrijskih svojstava odlučeno je u ovome radu implementirati slijedeće:

- a) algoritam za izračun otapalu dostupnog područja površine atoma prema Lee i Richards algoritmu (Lee, Richards 1971.)
- b) DPX algoritam za izračun ukopanosti atoma (Pintar, Carugo, Pongor, 2003.)
- c) CX algoritam za izračuna izbočenosti atoma (Pintar, Carugo, Pongor, 2002.)

Svi odabrani algoritmi za analizu geometrijske strukture izvršavaju se na razini atoma, te se onda tako dobiveni rezultati preslikavaju na aminokiseline izgrađene od tih atoma. Analiza na atomskoj razini odabrana je zbog veće točnosti dobivenih podataka.

U modul za izračun interakcijskih parova i mjesta interakcije potrebno je implementirati nekoliko već postojećih algoritama:

- a) algoritam baziran na mjerenju udaljenosti između atoma aminokiselina (Ofra, Rost 2003.)
- b) algoritam baziran na mjerenju udaljenosti Van der Waalsovih radijusa atoma (Aytuna 2005.)

- c) algoritam baziran na mjerenju promjene otapalu dostupnog područja površine prilikom interakcije (Jones, Thornton 1997.)

Razvijanje vlastitog algoritma za analizu interakcija između proteina zahtjeva određivanje geometrijskih, kemijskih i elektrostatskih pravila na temelju kojih će se zaključivati o postojanju interakcije. Nakon odabira pravila osmišljen je algoritam, te su na temelju pravila definirani mogući interakcijski parovi aminokiselina, tj. atoma koji ih grade.

Nakon njihove implementacije, potrebno je izvršiti usporedbu podataka dobivenih pojedinim algoritmom za analizu interakcija sa podacima koji se nalaze u PFAM bazi podataka.

Softverski alat mora imati svoju inačicu za Windows i Linux operativne sustave, biti razvijen u C++ programskom jeziku i imati mogućnost pokretanja sa grafičkim sučeljem ili iz konzole. Kao izlazni formati datoteka alata odabrani su tablični format zbog njegove preglednosti i XML format zbog njegove široke rasprostranjenosti i jednostavnosti njegove kasnije analize.

3 Analiza geometrijske strukture proteina

Proteini su polipeptidi građeni od dvije do nekoliko tisuća aminokiselinskih residuuma vezanih peptidnom vezom. Iako postoji samo dvadeset standardnih aminokiselina koje grade proteine, svojim različitim slijedom vezanja i prostornim rasporedom (konformacijom) one tvore tisuće različitih proteina s različitim biološkim funkcijama.

Primarna, sekundarna i tercijarna struktura proteina igraju važnu ulogu u definiranju biološke funkcije proteina i njihovo proučavanje je ključ do razumijevanja funkcioniranja živog organizma.

Baze podataka koje sadrže građe proteina već sadrže informacije o primarnoj i sekundarnoj strukturi. Cilj ovoga rada je izraditi alat koji će služiti za analizu svojstava tercijarne strukture kao što su otapalu dostupno područje površine, ukopanost i izbočenost aminokiselina i na taj način pripremiti podatke za danju analizu.

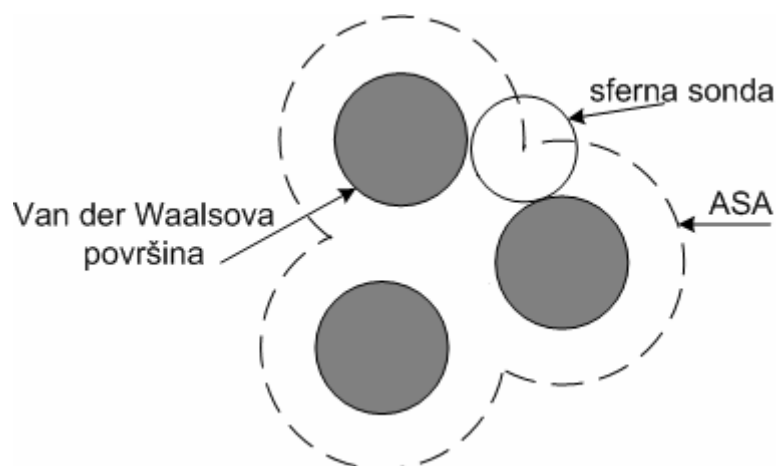
3.1 Otapalu dostupno područje površine

Ideju o otapalu dostupnom području površine prvi put je predstavljena u svrhu kvantifikacije hidrofobnog ukopavanja aminokiselina. Otapalu dostupno područje površine opisuje područje površine proteina na kojem može doći do kontakta između proteina i otapala u kojem se nalazi (najčešće voda).

Poznavanje topologije površine proteina važno je pri proučavanju interakcija između proteina i drugih molekula jer neki dijelovi površine direktno sudjeluju u interakcijama. Topologija površine proteina također je usko povezana i sa samom funkcijom proteina u organizmu.

3.1.1 Otapalu dostupno područje površine atoma

U ovome radu implementiran je originalni algoritam koji su predložili Lee i Richards. Otapalu dostupna površina definirana je kao pozicija centra sferne sonde, koja predstavlja molekulu otapala, dok se ona pomiče po Van der Waalsovoj površini proteina.



Slika 1: otapalu dostupno područje površine

Algoritam započinje tako da se svakom atomu ili grupi atoma unutar promatranog proteina pridruži njegov Van der Waalsov radijus. Izbor Van der Waalsovih radijusa nije određen algoritmom, te su u ovom radu korišteni radijusi prema Chothia (1976.). Iz razloga što nije običaj uključivati atome vodika u datoteke koje sadrže 3D strukturu proteina, Van der Waalsovi radijusi teških atoma koji na sebi imaju vezane atome vodika uključuju i radijuse vodika.

Tablica 1: Van der Waalsovi radijusi korišteni u radu

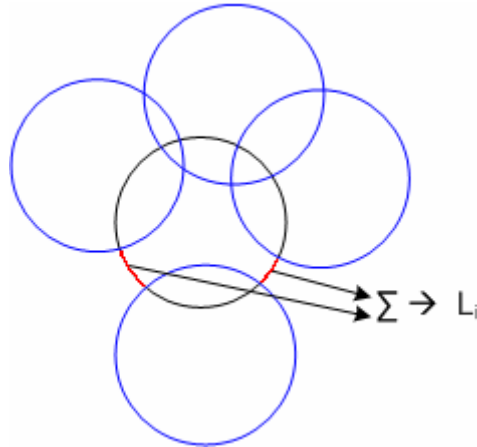
Atom	Van der Waalsov radijus
Kisik	1,40 Å
Sumpor	1,85 Å
Trigonalni dušik	1,65 Å
Tetrahedralni dušik	1,50 Å
Tetrahedralni ugljik	1,87 Å
Trigonalni ugljik	1,76 Å

Kada svi atomi imaju pridružen Van der Waalsov radijus taj radijus se proširuje za radijus sferne sonde (*RSolvent*) te se time dobiva prošireni radijus atoma R . Uobičajeno je koristiti radijus sonde u iznosu 1,40 Å, što odgovara iznosu Van der Waalsovog radijusa molekule vode. Iznos dostupnog područja površine pojedinog atoma ovisiti će o izboru Van der Waalsovog radijusa samog atoma i radijusa sonde.

Svaki atom smatra se sferom sa centrom određenim koordinatama atoma i radijusa jednakog njemu pripadajućeg proširenog radijusa i struktura proteina se može smatrati skupom ispresijecanih sfera.

Dostupno područje površine svakoga atoma računa se tako da se prošireni volumen atoma „izreže“ po jednoj osi na 2-dimenzionalne ravnine (kružnice), udaljene za neku određenu vrijednost (*zslice*). U svakoj ravnini sada postoje ispresijecane kružnice. Ako za promatrani atom postoji dio kružnog luka koji se ne siječe niti sa jednom kružnicom susjednih atoma, kažemo da je taj atom dostupan otapalu. Dužina slobodnog luka (L_i) određuje dostupnost atoma u i-toj ravnini, a ukupna dostupna površina atoma proporcionalna je zbroju dužina slobodnih lukova na svim ravninama.

Slikom 2 ilustriran je princip računanja dužine slobodnoga luka na i-toj ravnini; dužina slobodnoga luka prikazana je u crvenoj, a zauzetog u crnoj boji; kružnice koje pripadaju susjednim atomima obojane su u plavo.



Slika 2: dužina slobodnog luka na i-toj ravnini

Nakon izračuna slobodnog luka u svim ravninama, ukupno dostupno područje površine računa se pomoću slijedećih formula:

$$ASA = \sum_{i=1}^n \frac{R \cdot D \cdot L_i}{\sqrt{R^2 - Z_i^2}} \quad (3.1)$$

$$D = \frac{zslice}{2} + \Delta Z \quad (3.2)$$

$$\Delta Z = \min \left(\frac{zslice}{2}, R - Z_i \right) \quad (3.3)$$

gdje je Z_i duljina okomice spuštene iz centra sfere na i -tu ravninu.

Iz opisa algoritma za izračun dostupnog područja površine može se zaključiti da je vrijednost ASA atoma zapravo područje površine atoma na kojem može doći do kontakta sa otapalom (ili nekom drugom molekulom).

3.1.2 Otapalu dostupno područje površine aminokiselina

Ukupna ASA svake aminokiseline računa se zbrajanjem ASA svih atoma koji grade tu aminokiselinu. Osim **ukupne ASA (engl. *Total*)** postoje još četiri vrijednosti koje su zanimljive pri proučavanju topologije površine aminokiseline:

- a) **ASA glavnog lanca (engl. *Backbone*)** – suma ASA svih atoma koji grade glavni lanac aminokiselina
- b) **ASA bočnog lanca (engl. *Side-chain*)** – suma ASA svih atoma koji grade bočni lanac aminokiseline
- c) **ASA polarnog dijela (engl. *Polar*)** – suma ASA svih polarnih atoma (atomi kisika i dušika) koji grade aminokiselinu
- d) **ASA nepolarnog dijela (engl. *Non-polar*)** – suma ASA svih nepolarnih atoma (svi atomi osim kisika i dušika) koji grade aminokiselinu

U nekim primjenama nije dovoljno poznavati samo apsolutne vrijednosti ASA aminokiselina već vrijednosti apsolutne vrijednosti u odnosu na standardnu vrijednost ASA koja pripada toj aminokiselini dok ona nije sastavni dio proteina, tj. relativne vrijednosti. Pošto se aminokiseline nikada ne nalaze same u prostoru standardne vrijednosti ASA se proračunavaju tako da se promatra aminokiselina okružena (po sekvenci) sa još dvije druge aminokiseline (npr. ALA-X-ALA trojke).

Za potrebe ovoga rada za svaku aminokiselinu uzeto je po tisuću slučajno odabranih trojki iz PDB datoteka, te je za svaku od njih izračunata vrijednost ASA. Dobivene vrijednosti ASA izračunate su sa vrijednostima $zsl_{ice} = 0,25$ Å i $RSolvent = 1,40$ Å, a vrijednosti su prikazane u tablici 2 (sve vrijednosti u tablici izražene su u Å²).

Tablica 2: standardne vrijednosti ASA za aminokiseline

Residuum	Ukupno	Glavni lanac	Bočni lanac	Nepolarno	Polarno
Alanine	107,24	43,32	63,92	76,06	31,17
Arginine	233,01	36,86	196,15	86,30	146,71
Asparagine	150,85	36,46	114,39	42,77	108,08
Aspartate	144,06	36,15	107,91	49,57	94,49
Cysteine	131,46	36,12	95,34	104,07	27,40
Glutamine	177,99	34,24	142,76	62,78	115,21
Glutamate	171,53	35,75	135,77	70,72	100,81
Glycine	80,54	80,54	0,00	42,63	37,92
Histidine	180,93	34,84	146,09	106,27	74,66
Isoleucine	173,40	31,08	142,33	147,36	26,04
Leucine	177,87	32,68	145,19	150,10	27,76
Lysine	196,14	35,55	160,59	123,55	72,59
Methionine	186,80	34,06	152,74	159,15	27,65
Phenylalanine	200,93	33,67	167,26	174,16	26,77
Proline	133,78	32,90	100,88	112,82	20,96
Serine	115,30	40,86	74,44	52,93	62,37
Threonine	136,59	34,14	102,45	80,39	56,20
Tryptophan	240,12	32,51	207,61	186,22	53,90
Tyrosine	213,21	33,59	179,62	143,97	69,25
Valine	149,34	31,33	118,01	123,43	25,91

Kao i kod apsolutnih vrijednosti, postoji pet relativnih vrijednosti (ukupna, glavnog lanca, bočnog lanca, polarna i nepolarna) koje se računaju omjerom apsolutne i standardne vrijednosti pomnožene sa sto. Relativne vrijednosti ASA aminokiselina opisuju kolikim dijelom svoje površine, izraženim u postotcima, je aminokiselina dostupna otapalu.

3.2 Izbočenost atoma i aminokiselina u proteinima

Identifikacija izbočenih ili jako konveksnih regija proteina pokazuje se važnom pri proučavanju proteinskih interakcija i proteinsko-proteinskih kompleksa. Kroz povijest koristili su se različiti pristupi za identifikaciju

izbočenih i konveksnih regija unutar proteina. Većina tih algoritama računalno su zahtjevni i jako ovisni o parametrima korištenim u analizi.

3.2.1 Izbočenost atoma – CX algoritam

Algoritam za proračun izbočenosti pojedinog atoma u proteinu bazira se na uspoređivanju volumena koji protein zauzima u okolini atoma i slobodnog volumena oko istog atoma.

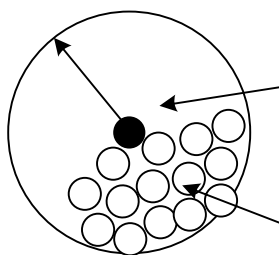
Izbočenost pojedinog atoma računa se tako da se izbroji koliko se njemu susjednih atoma N_{atom} nalazi unutar sfere sa centrom u promatranom atomu i polumjera R . Broj atoma unutar sfere nakon toga pomnoži se sa srednjim volumenom atoma koji se nalaze u proteinima ($V_{atom} = 20,1 \pm 0,9 \text{ \AA}^3$; Richards 1974.), te na taj način dobivamo volumen koji zauzima protein unutar sfere V_{int} . Ostatak volumena sfere nalazi se u slobodnom prostoru te se računa kao razlika volumena sfere i volumena zauzetog proteinom:

$$V_{ext} = V_{sfera} - V_{int} = \frac{4}{3} \cdot R \cdot \pi - V_{atom} \cdot N_{atom} \quad (3.4)$$

Izbočenost promatranog atoma određena je omjerom slobodnog volumena i volumena zauzetog proteinom:

$$CX = \frac{V_{ext}}{V_{int}} = \frac{V_{sfera}}{V_{int}} - 1 = \frac{4 \cdot R \cdot \pi}{3 \cdot V_{atom} \cdot N_{atom}} - 1 \quad (3.5)$$

Algoritam za izračun izbočenosti atoma ilustriran je slikom 3:

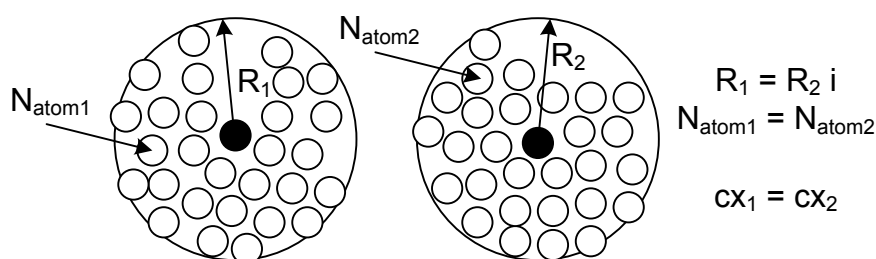


Slika 3: ilustracija CX algoritma

Vrijednosti izbočenosti atoma u proteinu prema ovom algoritmu kreću se u granicama od 0 do 15, tako da izbočenijim atomima propada veća vrijednost cx . Vrijednosti cx ovise o izboru polumjera sfere unutar koje prebrojavamo susjedne atome. Predložena vrijednost polumjera je $10 \text{ \AA}$ koja bi trebala biti

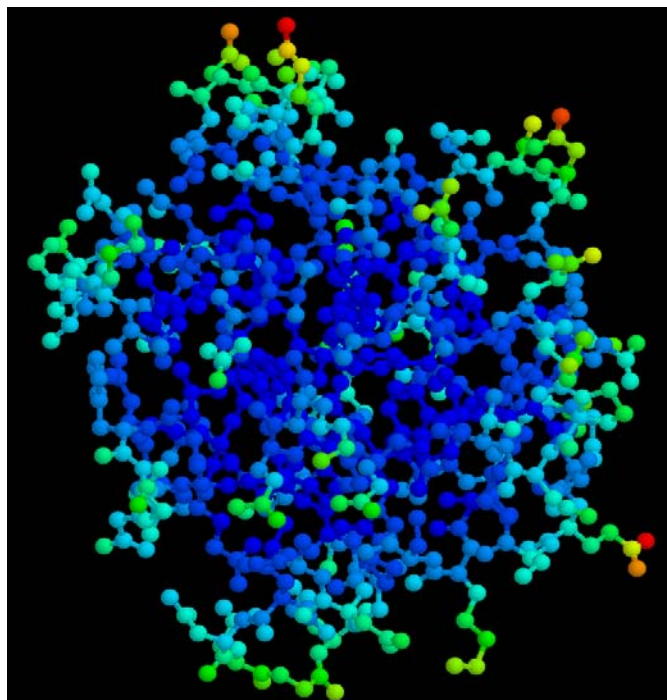
zadovoljavajuća u većini primjena. Manje vrijednosti polumjera prouzročiti će osjetljivost vrijednosti cx na bližu (lokalnu) okolinu atoma, dok će veće pri odabiru većih vrijednosti R vrijednost cx biti više osjetljiva na širu (globalnu) strukturu samog proteina.

Jedina mana ovoga algoritma je što se iz njime dobivenih rezultata ne može razlikovati da li se atom nalazi u konkavnom dijelu proteina ili je samo ukopan ispod površine. Način rješavanja problema identifikacije konkavnih regija je povezivanjem vrijednosti izbočenosti i vrijednosti otapalu dostupne površine, tj. atomi koji se nalaze u konkavnim regijama proteina imaju male vrijednosti izbočenosti, ali su istovremeno i dostupni otapalu. Primjer kako može doći do krive interpretacije vrijednosti cx prikazan je slikom 4; lijevi atom na slici nalazi se u konkavnoj regiji proteina i nalazi se na površini, a desni atom ukopan je ispod površine proteina.



Slika 4: Primjer moguće krive interpretacije rezultata

Na slici 5 prikazan je 3D model molekule PDB identifikatora 1RRO nakon provođenja CX algoritma. Najmanje izbočeni atomi obojani su tamno plavom, a najviše izbočeni žarko crvenom bojom. Vrijednosti cx proračunate su uz parametre $R = 10 \text{ \AA}$ i $V_{atom} = 20,1 \text{ \AA}^3$, a model i slika stvoreni su pomoću programa za molekularnu grafiku RASMOL.



Slika 5: 3D prikaz rezultata dobivenih CX algoritmom

3.2.2 Izbočenost aminokiselina

Vrijednosti izbočenosti pojedine aminokiseline računaju se iz izbočenosti atoma koji tvore tu aminokiselinu. Za pojedinu aminokiselinu računaju se po šest vrijednosti kojima se opisuje izbočenost aminokiseline:

- a) srednja vrijednost izbočenosti svih atoma koji tvore aminokiselinu
- b) standardna devijacija srednje vrijednosti izbočenosti svih atoma
- c) srednja vrijednost izbočenosti atoma bočnog lanca aminokiseline
- d) standardna devijacija srednje vrijednosti izbočenosti atoma bočnog lanca
- e) najmanja vrijednost izbočenosti atoma koji tvore aminokiselinu
- f) najveća vrijednost izbočenosti atoma koji tvore aminokiselinu

3.3 Ukopanost atoma i aminokiselina u proteinima

Ukupanost atoma i aminokiselina u proteinu geometrijski je parametar koji se koristi u svrhu davanja uvida u unutrašnjost proteina. To je jednostavan, ali vrijedan način geometrijskog opisa strukture proteina.

Ukopanost atoma veže se uz nekoliko osobina proteina, kao što su stabilnost proteina, prosječna veličina domena, slobodna energija stvaranja proteinskih kompleksa, hidrofobnost aminokiselina, i druge.

Sposobnost proteina da se pravilno savija (engl. *fold*) i da zadrži stabilnost savijanja, koja određuje njegovu funkciju, određena je osobinama unutrašnjosti proteina. Aminokiseline koje su duboko ukopane u unutrašnjosti proteina mogu imati ključnu ulogu u procesu savijanja. Također, duboko ukopane aminokiseline mogu više doprinijeti termodinamičkoj stabilnosti proteina.

Ukopanost nam pruža mogućnost razlikovanja aminokiselina koje su ukopane, ali se nalaze blizu površine proteina, od onih koje su ukopane duboko u proteinu. Ovako nam ukopanost pruža dodatne informacije koje nam nisu dostupne metodama određivanja otapalu dostupne površine ili ukopane površine (te metode daju nam samo informaciju o površini ukopane aminokiseline, ali ne i o dubini na kojoj se ona nalazi unutar proteina).

Atomi koji se nalaze u prvom ukopanom sloju (atomi ukopani 0,75 – 1,75 Å od površine proteina) mogu postati dostupni otapalu kroz unutarnju dinamiku ili male konformacijske promjene; takvi atomi, tj. aminokiseline kojima ti atomi pripadaju, nadalje mogu sudjelovati u interakcijama između proteina.

Ukopanost svakog pojedinog atoma ili aminokiseline može koristiti za njihovu klasifikaciju u proteinu nezavisno o njihovim fizičko-kemijskim svojstvima.

3.3.1 Ukopanost atoma – DPX algoritam

Prvi način definiranja ukopanosti atoma, je kao udaljenost atoma od najbliže molekule vode (otapala) koja okružuje protein.

Druga definicija, određuje ukopanost atoma kao najmanju udaljenost od otapalu dostupne površine.

Noviji pristup, ujedno i pristup korišten u ovom radu, ukopanost atoma definira kao udaljenost atoma od najbližeg atoma na površini istog proteina.

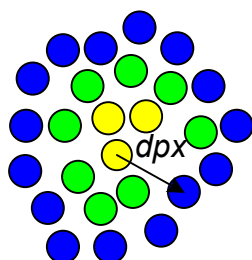
Algoritam za ovakav proračun ukopanosti zahtjeva prethodno određivanje atoma koji se nalaze na površini. Površinski atomi proteina u ovom radu određuju se prethodnim računanjem površine dostupne otapalu; svaki atom koji je dostupan otapalu smatra se površinskim i neukopanim atomom.

Ukopanost atoma određena je kao udaljenost atoma do najbližeg atoma koji se nalazi na površini proteina, te se može izraziti slijedećom formulom:

$$dpx = \min(d_1, d_2, \dots, d_n) \quad (3.6)$$

gdje je d_j za $j = 1, 2, \dots, n$ udaljenost promatranog atoma od j -tog atoma na površini proteina.

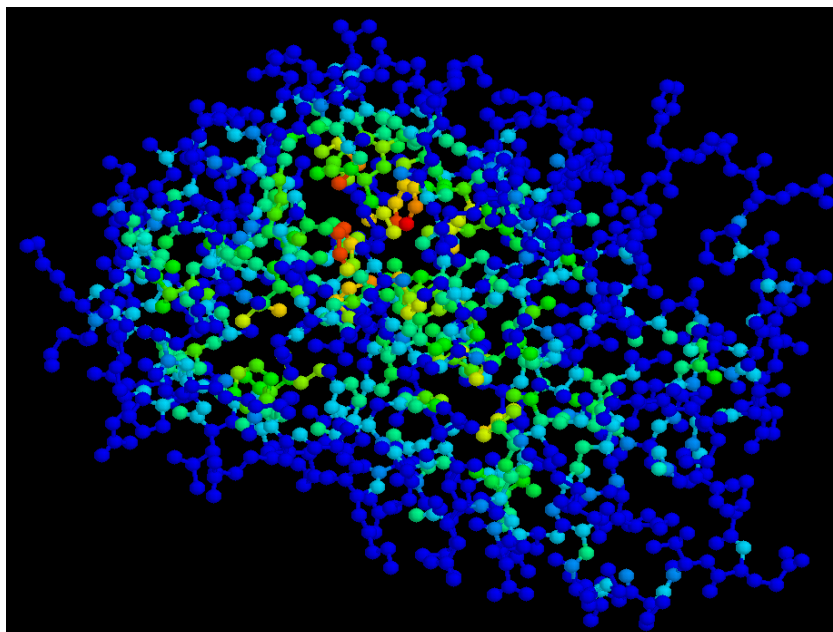
Slikom 6 prikazan je način određivanja ukopanosti atom DPX algoritmom; površinski atomi prikazani su plavom bojom, atomi prvog sloja ukopanosti zelenom, a duboko ukopani atomi žutom bojom.



Slika 6: ilustracija DPX algoritma

Prema tome vrijednost dpx je jednaka nuli za sve one atome koji se nalaze na površini (tj. za atome dostupne otapalu), a veća od nula za atome ukopane u unutrašnjosti proteina, tako da dublje ukopani atomi imaju veće vrijednosti dpx .

Na slici 7 prikazan je 3D model molekule PDB identifikatora 1ITF (model 1) nakon provođenja DPX algoritma. Model je obojan u skladu sa vrijednostima dpx atoma, od tamno plave boje za površinske atome ($dpx = 0$) do crvene boje za najdublje ukopane atome. Vrijednosti dpx proračunate su uz parametre za proračun otapalu dostupne površine $RSolvent = 1,40 \text{ \AA}$ i $zslice = 0,5\text{\AA}$, a model i slika stvoreni su pomoću programa za molekularnu grafiku RASMOL.



Slika 7: 3D prikaz rezultata dobivenih DPX algoritmom

3.3.2 Ukopanost aminokiselina

Vrijednosti ukopanosti pojedine aminokiseline računaju se iz vrijednosti ukopanosti (dobivenih DPX algoritmom) atoma koji tvore tu aminokiselinu. Za pojedinu aminokiselinu računaju se po šest vrijednosti kojima se opisuje njezina ukopanost:

- a) srednja vrijednost ukopanosti svih atoma koji tvore aminokiselinu
- b) standardna devijacija srednje vrijednosti ukopanosti svih atoma
- c) srednja vrijednost ukopanosti atoma bočnog lanca aminokiseline
- d) standardna devijacija srednje vrijednosti ukopanosti atoma bočnog lanca
- e) najmanja vrijednost ukopanosti atoma koji tvore aminokiselinu
- f) najveća vrijednost ukopanosti atoma koji tvore aminokiselinu

Pri proračunu ukopanosti aminokiselina koristeći DPX algoritam na razini atoma može se primijetiti da su srednje vrijednosti ukopanosti relativno male. Male vrijednosti ukopanosti aminokiselina proizlaze iz činjenice da postoji jako malo aminokiselina koje su u potpunosti duboko ukopane u proteinu (to ponajviše vrijedi za proteine male i srednje veličine), te zbog pridruživanja

vrijednosti nula za d_{px} svakog atoma koji ima površinu dostupnu otapalu $ASA > 0 \text{ \AA}^2$.

3.4 Prikaz proteina pomoću 3D mreže atoma

Svi ranije navedeni algoritmi za analizu strukture proteina dobivene iz PDB datoteka služe za analizu 3D građe proteina. Podatci o proteinu dobiveni iz PDB datoteka sadrže informacije o 3D strukturi, ali su atomi, tj. aminokiseline, poredane po sekvenci u kojoj su povezane. Udaljenost po sekvenci između susjednih atoma i susjednih aminokiselina najčešće nije komplementarna njihovoj udaljenosti u prostoru, te pretraživanje prostora pomoću 1-dimenzionalne sekvence zahtjeva puno računalnog vremena (što se najviše ističe kod analize velikih proteina kompleksne strukture).

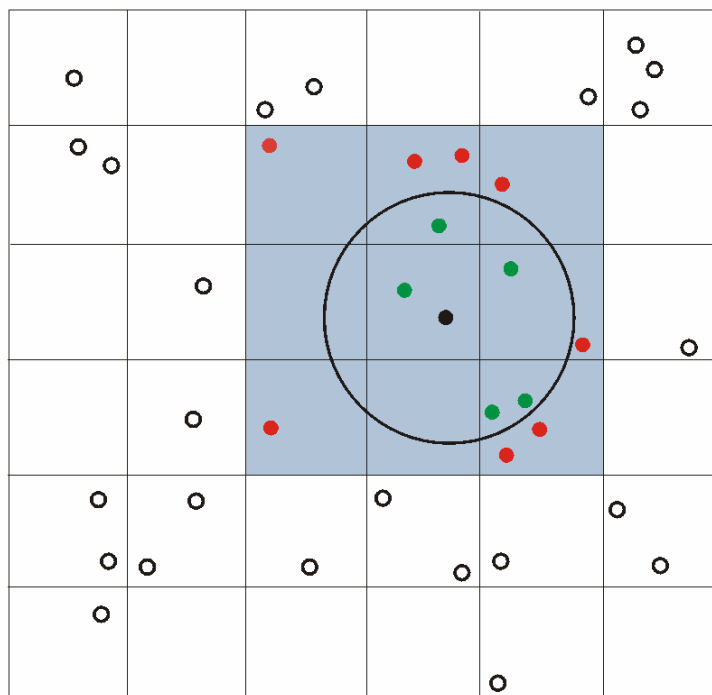
Zbog potrebe za bržim pretraživanjem prostora koji okružuje pojedini atom čiji se strukturni atributi analiziraju, tj. pronalaženja njegovih susjeda koji se nalaze unutar određenih granica (npr. svih atoma koji se nalaze unutar sfere radijusa $10 \text{ \AA}$), u ovom radu koristi se računalni prikaz proteina kao 3D mreže (*3D grid*) atoma. Ovakav prikaz proteina znatno ubrzava izvođenje algoritama za analizu strukture.

3.4.1 Stvaranje i pretraživanje 3D mreže

Ideja ovoga algoritma je stvaranje umjetne mreže koja dijeli područje, namijenjeno za pretraživanje, na polja u obliku kocke određene veličine i pamti liste točaka koji po svojim koordinatama upadaju u određeno polje (Sedgewick 1990.). Prilikom pretraživanja prostora koji okružuje određenu točku, ispituju se samo ona polja koja se nalaze unutar traženog opsega.

Slikom 8 ilustrirana je, u 2D prostoru, ideja pronalaženja susjednih točaka koje se nalaze unutar kružnice koja okružuje promatranu točku. Pretraživanje se vrši na svim poljima mreže koje kružnica barem malo zahvaća (sivo obojana polja), tj. ispituje se udaljenost od svih točaka koje se nalaze u tim poljima. Točke koje se nalaze unutar kružnice (unutar određenog radijusa) smatraju se susjednim kružnicama (obojane zelenom bojom). Točke obojane

crvenom bojom predstavljaju točke koje nisu susjedne, tj. nalaze se izvan traženog opsega.



Slika 8: 2D prikaz pretraživanja mreže točaka

Glavna odluka koju je potrebno donijeti pri stvaranju mreže je veličina polja koja ju sačinjavaju. Ako je mreža previše gruba svako njeno polje sadržavati će previše atoma, a ako je previše fino izgrađena sadržavati će velik broj praznih polja koja će se bespotrebno pretraživati. Najbolji način za određivanje veličine polja je tako da očekivani broj atoma u svakom polju bude približno jednak nekoj maloj konstanti. Takav raspored po poljima može se postići tako da broj polja koji sačinjava mrežu bude jednak određenom konstantnom broju podijeljenom sa brojem točaka koji se trebaju nalaziti unutar mreže.

Dolje se nalazi dio koda za implementaciju klase cAtomGrid koji služi za stvaranje 3D mreže atoma.

```
struct sAtomGridNode
{
    cResidueAtom a;
    sAtomGridNode *next;
};

cAtomGrid::cAtomGrid(cResidueAtom* atoms, unsigned int full count, sPoint max point,
sPoint min point, unsigned int fill count)
{
```

```

    unsigned int i, count;
    if(fill count == -1)
        count = full count;
    else
        count = fill count;

    dummy = NULL;

    offset.x = -1 * min point.x;
    offset.y = -1 * min point.y;
    offset.z = -1 * min point.z;

    double max x, max y, max z;
    max x = fabs(max point.x - min point.x);
    max y = fabs(max point.y - min point.y);
    max z = fabs(max point.z - min point.z);

    for(size = 1; pow(size, 3) < max x * max y * max z / full count; size*=2);

    cnt x = (unsigned int)ceil(max x / size) + 1;
    cnt y = (unsigned int)ceil(max y / size) + 1;
    cnt z = (unsigned int)ceil(max z / size) + 1;

    gridArraySize = cnt x * cnt y * cnt z;

    grid = new sAtomGridNode*[gridArraySize];

    for(i = 0; i < gridArraySize; i++)
        grid[i] = NULL;
    for(i = 0; i < count; i++)
        insertAtom(atoms[i]);
}

void cAtomGrid::insertAtom(cResidueAtom &atom)
{
    sAtomGridNode *n = new sAtomGridNode;
    n->a = atom;
    unsigned int i = getGridIndex(atom.getX() + offset.x, atom.getY() + offset.y,
        atom.getZ() + offset.z);

    n->next = grid[i];
    grid[i] = n;
}

unsigned int cAtomGrid::getGridIndex(double x, double y, double z)
{
    unsigned int index;

    unsigned int i, j, k;

    i = (unsigned int)floor(x / size);
    j = (unsigned int)floor(y / size);
    k = (unsigned int)floor(z / size);

    index = i * cnt y * cnt z + j * cnt z + k;

    return index;
}

```

Postoji nekoliko važnih varijabli koje su bitne za ispravan rad ovog algoritma:

- a) offset - sadrži informaciju o tome koliko je potrebno pomaknuti svaki atom unutar mreže kako bi mreža mogla imati ishodište u točki (0, 0, 0), tj. da svi atomi unutar mreže nalaze samo u prvom kvadrantu 3D kartezijevog sustava
- b) size – veličina stranice polja mreže

- c) `cnt_x`, `cnt_y`, `cnt_z` – broj polja koja sačinjavaju mrežu po x, y i z osi; broj polja po jednoj osi dobiva se omjerom najveće koordinate atoma na toj osi i veličine stranice polja (npr. `cnt_x = (unsigned int)ceil(max_x / size) + 1`)

Mreža se na kraju sastoji od `cnt_x•cnt_y•cnt_z` polja, a polje koje pripada određenom atomu može se pronaći dijeljenjem njegovih koordinata sa vrijednosti varijable `size`. Ova implementacija koristi reprezentaciju liste atoma koji se nalaze unutar jednog polja pomoću povezane liste, sa repnom točkom `dummy` koja je zapravo NULL pokazivač.

Nakon što je mreža izgrađena, pretraživanje prostora oko određenog atoma sastoji se od jednostavnog indeksiranja u polje koje sadrži 3D mrežu.

```
unsigned int cAtomGrid::getNeighboursInRangeByCenter(cResidueAtom &a, double range,
cResidueAtom *neighbours)
{
    sAtomGridNode *n;
    cResidueAtom a_pnt;
    unsigned int i, j, k, neighbour count = 0;
    unsigned int max i, max j, max k;

    max i = (unsigned int) ceil((a.getX() + offset.x + range + 1)/size) < cnt x ?
(unsigned int)ceil((a.getX() + offset.x + range + 1)/size) : cnt x;

    max j = (unsigned int) ceil((a.getY() + offset.y + range + 1)/size) < cnt y ?
(unsigned int)ceil((a.getY() + offset.y + range + 1)/size) : cnt y;

    max k = (unsigned int) ceil((a.getZ() + offset.z + range + 1)/size) < cnt z ?
(unsigned int)ceil((a.getZ() + offset.z + range + 1)/size) : cnt z;

    for(i = (unsigned int)((a.getX() + offset.x - range - 1)/size >= 0 ? (a.getX() +
offset.x - range - 1)/size : 0); i < max i; i++)
    {
        for(j = (unsigned int)((a.getY() + offset.y - range - 1)/size >= 0 ?
(a.getY() + offset.y - range - 1)/size : 0); j < max j; j++)
        {
            for(k = (unsigned int)((a.getZ() + offset.z - range - 1)/size >= 0 ?
(a.getZ() + offset.z - range - 1)/size : 0); k < max k; k++)
            {
                for(n = grid[i * cnt y * cnt z + j * cnt z + k]; n != dummy; n = n-
>next)
                {
                    a_pnt = n->a;
                    if(a.isInRangeByCenter(a_pnt, range))
                    {
                        neighbours[neighbour count] = a_pnt;
                        neighbour count ++;
                    }
                }
            }
        }
    }
    return neighbour count;
}
```

Vrijeme izvođenja ovoga algoritma proporcionalno je broju polja mreže koja se pretražuju. Broj polja koja s pretražuju u prosjeku proporcionalan je

broju ispitanih točaka (atoma), a to je postignuto namještanjem veličine polja tako da u prosjeku sadrži stalan broj točaka.

3.4.2 Rezultati provjere ubrzanja algoritama

Za potrebe provjere ubrzanja algoritama za analizu strukture proteina pomoću 3D mreže atoma, algoritmi za izračun otapalu dostupnog područja površine, ukopanosti i izbočenosti atoma i aminokiselina izvršeni su na pet PDB datoteka, tj. peptida, različitih veličina. Peptidi korišteni pri izvršavanju testova imaju slijedeće identifikatore: 1DFN (470 atoma; 60 aminokiselina), 1KBA (1.002 atoma; 132 aminokiseline), 1LRH (5.076 atoma; 640 aminokiselina), 1JI2 (9.552 atoma; 1.170 aminokiselina) i 1K32 (49.062 atoma; 6.138 aminokiselina).

Algoritmi su izvršeni pretraživanjem po sekvenci (slijedno) i pretraživanjem pomoću 3D mreže atoma. Testiranje brzine izvršavanja algoritama izvedeno je na tri različite konfiguracije računala, operativnog sustava i kompajlera:

- a) Intel® Pentium® 4 CPU 3.00Ghz; 1GB RAM; Windows XP Service Pack 2; Microsoft Visual Studio .NET 2003 Visual C++ Compiler
- b) Intel® Pentium® 4 CPU 3.00Ghz; 1GB RAM; Windows XP Service Pack 2; GNU GCC Compiler
- c) Intel Celeron CPU 2.60Ghz; 512MB RAM; Linux Ubuntu; GNU GCC Compiler

Izvođenje testova za računanje otapalu dostupnog područja površine izvršeno je sa parametrima $zslice = 0.5\text{\AA}$ i $RSolvent = 1.4\text{\AA}$.

Tablica 3: vremena izvođenja izračuna vrijednosti ASA

PDB id	sekvenca (a)	3D mreža (a)	sekvenca (b)	3D mreža (b)	sekvenca (c)	3D mreža (c)
1DFN	0,203 s	0,188 s	0,328 s	0,312 s	0,12 s	0,15 s
1KBA	0,625 s	0,437 s	1,047 s	0,719 s	0,33 s	0,41 s
1LRH	10,797 s	2,453 s	18,125 s	4,000 s	2,86 s	2,40 s
1JI2	36,686 s	5,016 s	61,203 s	8,282 s	7,48 s	4,68 s
1K32	893,125 s	25,984 s	1.516,39 s	39,344 s	132,88 s	23,21 s

Pri izvođenju testova za brzinu računanja ukopanosti atoma i aminokiselina prethodno su izračunate vrijednosti ASA, tako da se trajanja izvršenja algoritma obuhvaćaju samo trajanje izračuna ukopanosti.

Tablica 4: vremena izvođenja DPX algoritma

PDB id	sekvenca (a)	3D mreža (a)	sekvenca (b)	3D mreža (b)	sekvenca (c)	3D mreža (c)
1DFN	0,026 s	0,015 s	0,032 s	0,00 s	0,00 s	0,00 s
1KBA	0,094 s	0,016 s	0,156 s	0,015 s	0,03 s	0,02 s
1LRH	2,313 s	0,062 s	3,828 s	0,141 s	0,4 s	0,09 s
1JI2	8,344 s	0,140 s	13,766 s	0,312 s	7,48 s	0,22 s
1K32	220,422 s	0,688 s	363,922 s	1,562 s	30,12 s	1,09 s

Izvođenje testova za računanje izbočenosti atoma i aminokiselina izvršeno je sa parametrima $R = 10\text{\AA}$ i $V_{atom} = 20.1\text{\AA}^3$.

Tablica 5: vremena izvođenja CX algoritma

PDB id	sekvenca (a)	3D mreža (a)	sekvenca (b)	3D mreža (b)	sekvenca (c)	3D mreža (c)
1DFN	0,094 s	0,016 s	0,156 s	0,078 s	0,02 s	0,06 s
1KBA	0,391 s	0,078 s	0,656 s	0,188 s	0,07 s	0,14 s
1LRH	9,500 s	0,578 s	16,016 s	1,484 s	1,49 s	1,35 s
1JI2	33,797 s	1,282 s	56,937 s	3,297 s	4,91 s	3,15 s
1K32	876,344 s	5,828 s	1.498,3 s	14,391 s	119,93 s	13,18 s

Iz dobivenih rezultata testova može se vidjeti da je, unatoč razlikama performansi na različitim sustavima, optimizacija pretraživanja u većini slučajeva efikasnija od sekvencijalnog pretraživanja.

4 Analiza proteinskih interakcija

Velik broj bioloških funkcija organizma uključuju formaciju proteinsko-proteinskih kompleksa (proteina građenih od dva ili više polipeptidnih lanaca), kako trajnih tako i prijelaznih. Sposobnost vezanja proteina za druge molekule omogućava proteinima da djeluju kao antitijela koja se vežu na viruse ili bakterije, te kao katalizatori, signalni receptori, motori, prekidači, itd. Tvar koja se veže za protein naziva se ligandom, neovisno da li je to ion, mala ili velika molekula.

Različite regije proteinskih lanaca se uglavnom vežu za različite ligande i na taj način omogućavaju regulaciju aktivnosti i funkcije proteina. Poznavanje načina na koji različiti proteinski lanci međusobno interagiraju, te poznavanje svojstava regija kojima dolaze u kontakt jako je važno za razumijevanje načina na koji proteini djeluju.

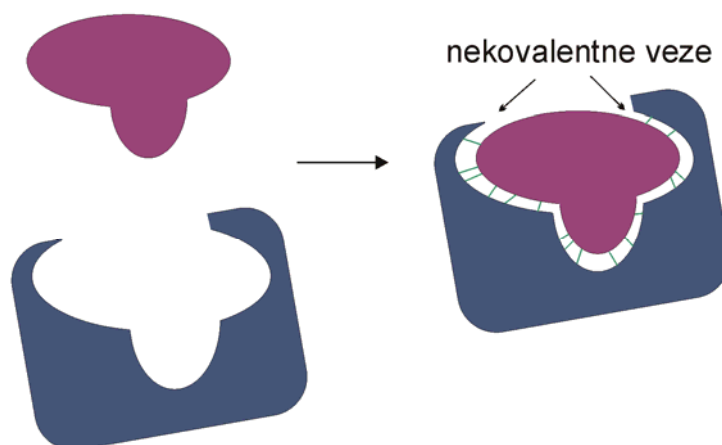
Predikcija proteinskih interakcija pronalazi široku primjenu na područjima biologije, kemije i biokemije, a najveću važnost uživa u farmaceutskoj industriji (ponajviše pri razvoju lijekova). Kako bi se razvili alati za predikciju interakcija i naučili predikcijski algoritmi da prepoznaju mjesta interakcije i interakcijske parove potrebno je analizirati već poznate proteinske komplekse i na taj način pripremiti podatke koji bi pomogli pri razumijevanju proteinskih interakcija.

4.1 *Veze između aminokiselinskih residuuma*

Sposobnost vezanja proteinskih lanaca za ligande sa velikom sklonošću ovisi o stvaranju nekoliko vrsta slabih nekovalentnih veza između aminokiselinskih residuuma:

- a) ionske veze
- b) vodikove veze
- c) Van der Waalsove veze
- d) hidrofobne veze

Nekovalentne veze između residuuma puno (30 do 300 puta) su slabije od kovalentnih veza koje tipično tvore biološke molekule. Unatoč njihovoj slabosti veliki broj istodobno stvorenih nekovalentnih veza mogu čvrsto držati različite polipeptidne lance spojenima. Da bi takvo spajanje bilo moguće jako je važno da su geometrijske strukture polipeptidnih lanaca međusobno komplementarne, tj. da lanci dobro prijanjaju jedan uz drugi.

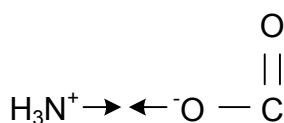


Slika 9: vezanje polipeptidnih lanaca nekovalentnim vezama

4.1.1 Ionske veze

Ionske veze nastaju između suprotno nabijenih atoma zbog elektrostatskih privlačnih sila. Ovakva vrsta veza može imati veliku snagu ako se nalazi izvan vode. Ako su polarne molekule vode nakupljene oko nabijenih iona, zbog velike dielektrične konstante vode dolazi do iznimnog smanjenja njihove međusobne privlačnosti. Snaga, odnosno sila F , ionskih veza između polarnih atoma ovisi o udaljenosti njihovih jezgara d (snaga opada sa kvadratom udaljenosti), jačini naboja Q i dielektričnoj konstanti ε otapala u kojem se nalaze, te se može izraziti idućom formulom 4.1.

$$F = \frac{Q_1 \cdot Q_2}{\varepsilon \cdot d^2} \quad (4.1)$$

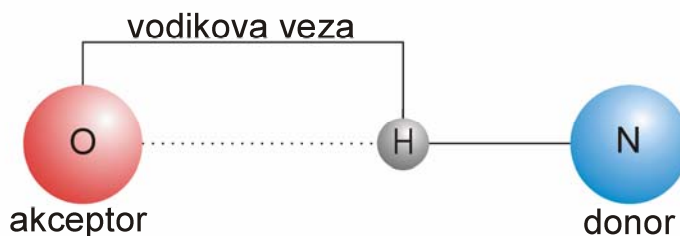


Slika 10: tipičan primjer ionske veze (privlačnosti)

4.1.2 Vodikove veze

Vodikove veze su specijalni oblik polarnih interakcija pri kojima dva elektro-negativna atoma međusobno dijele jedan elektro-pozitivan atom vodika. Taj vodik se može promatrati kao proton kojeg proton donor dijeli sa akseptorom. Za razliku od tipičnih elektrostatskih veza, vodikove veze jako su usmjerene i najjače su kada postoji ravna linija između sva tri atoma koji sudjeluju u interakciji. Prisutnost vode slabi i snagu ovih veza stvarajući vodikove veze sa molekulama koje okružuje.

Najčešće vodikove veze u organizmu su: $\text{OH} - \text{O}$; $\text{OH} - \text{O}^-$; $\text{OH} - \text{N}$; $\text{NH} - \text{O}$; $\text{NH}^+ - \text{O}$; $\text{NH} - \text{N}$.



Slika 11: tipičan primjer vodikove veze

4.1.3 Van der Waalsove veze

Van der Waalsove veze nastaju pri fluktuacijama elektronskog oblaka nepolarnog atoma koje stvaraju trepereći dipol. Takav dipol inducirati će stvaranje suprotno polariziranog treperećeg dipola u susjednom atomu i nastati će slaba privlačnost među tim atomima. Van der Waalsove veze najslabije su od nekovalentnih veza, ali obično imaju najveći domet i nisu oslabljene u prisutnosti vode. Prilikom prianjanja površina različitih molekula može doći do istovremenog kontakta između velikog broja atoma pa rezultat Van der Waalsovih privlačenja može biti od velikog značaja za snagu i stabilnost interakcije.

4.1.4 Hidrofobne veze

Voda koja okružuje molekulu tjera grupe nepolarnih aminokiselina što dalje od površine molekule. Između tako grupiranih nepolarnih aminokiselina

kaže se da postoje hidrofobne veze, iako je njihova privlačnost u biti stvorena zbog njihove odbojnosti prema vode. Nepolarnih aminokiseline koje se grupiraju i tako stvaraju hidrofobne veze su *Alanine*, *Isoleucine*, *Leucine*, *Methionine*, *Phenylalanine*, *Proline*, *Valine*. Grupi nepolarnih aminokiselina još pripada i *Glycine*, ali on zbog svojeg malog bočnog lanca ne pridonosi hidrofobnim interakcijama.

4.2 Interakcije na osnovu udaljenosti između residuuma

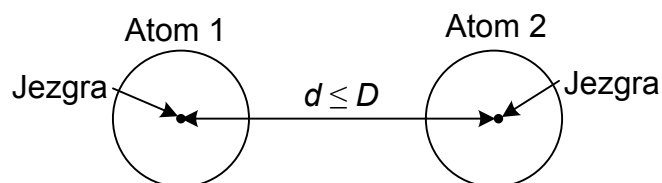
Najbrži i najjednostavnije, ali i najmanje precizan način pronalaženja veza između aminokiselinskih residuuma već poznatih proteinskih kompleksa je na osnovu udaljenosti između residuuma. Ovakav način identifikacije interakcijskih parova i mjesta interakcija koristan je ako se želi na brz način pronaći zadovoljavajući broj interakcija ne obazirući se na svojstva samih veza.

Poznata su dva načina kojima se na osnovu udaljenosti pronalaze interakcijski parovi residuuma različitih proteinskih lanaca.

4.2.1 Interakcije na osnovu udaljenosti između jezgri atoma

Interakcijski parovi između proteinskih lanaca pronalaze se na osnovu udaljenosti između jezgri atoma interagirajućih residuuma. Koordinate jezgri svakoga atoma nalaze se u PDB datotekama.

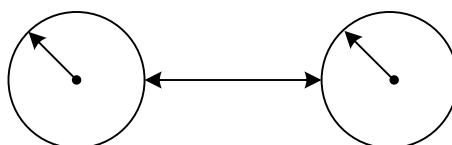
Ako postoji barem jedan par atoma, od kojih svaki pripada jednoj aminokiselini različitih lanaca, udaljen manje od neke određene vrijednosti D smatra se da su ta dva residuuma vezana, tj. da interagiraju. Najčešće korištene vrijednosti udaljenosti D , koje daju realne rezultate, kreću se između 4,0Å (najčešće nisu sve veze pronađene) i 6,0Å (najčešće se pronađe više nego stvaran broj veza).



Slika 12: definiranje veze na temelju udaljenosti jezgri atoma

4.2.2 Interakcije na osnovu udaljenosti Van der Waalsovih radijusa

Ovakvim definiranjem interakcija između residuuma smatra se da dva residuuma interagiraju ako postoji barem jedan par atoma, od kojih svaki pripada jednoj aminokiselini različitih lanaca, čiji su Van der Waalsovi radijusi udaljeni manje od neke određene vrijednosti D .



Slika 13: definiranje veze na temelju udaljenosti Van der Waalsovih radijusa atoma

Iako je običaj na ovakav način definirati Van der Waalsove veze između residuuma, ovaj algoritam ipak ne uzima u obzir svojstva samih veza.

Najčešće korištene vrijednosti udaljenosti D , koje daju realne rezultate, kreću se između $0,5\text{\AA}$ (najčešće nisu sve veze pronađene) i $1,5\text{\AA}$ (najčešće se pronađe više nego stvaran broj veza).

4.3 Mjesta proteinskih interakcija na osnovu promjene ASA

Prilikom interakcije proteinskih lanaca dolazi do kontakta dijelova njihovih površina koji tada postaju nedostupni otapalu. Računanjem promijene otapalu dostupnog područja površine residuuma prije i poslije stvaranja kompleksa, može se zaključiti koji residuumi sudjeluju u interakciji.

Smatra se da se residuum nalazi u regiji kojom proteinski lanac interagira sa drugim lancem, ako se njegovo otapalu dostupno područje površine promijeni za više od neke određene vrijednosti A , što se može izraziti formulom 4.2.

$$\Delta ASA = ASA_{poslije} - ASA_{prije} \geq A \quad (4.2)$$

Ovakvim pristupom analize mogu se samo odrediti residuumi koji sudjeluju prilikom interakcije, a ne i interakcijski parovi residuuma različitih lanaca.

Ovaj algoritam je više vremenski zahtjevan od definiranja na osnovu udaljenosti jer je za njegovo izvođenje potrebno prethodno izračunati vrijednosti ASA za pojedini lanac prije i poslije interakcije.

4.4 iPFAM baza proteinskih interakcija

iPFAM je baza podataka koja sadrži domensko-domenske interakcije pronađene u zapisima PDB baze podataka. iPFAM analizira interakcije, tj. veze između različitih domena, ako se unutar jedne strukture nalaze dvije ili više domena koje su na dovoljno maloj udaljenosti.

iPfam procjenjuje stvaranje veza između residuuma na temelju skupine pravila temeljenih na udaljenosti i kemijskim svojstvima:

- a) **ionske veze** – mogu se stvoriti između suprotno nabijenih atoma različitih residuuma ako je udaljenost tih atoma manja od zbroja njihovih Van der Waalsovih radijusa.
- b) **vodikove veze** – mogu se stvoriti između dva atoma različitih residuuma, ako je jedan atom proton donor, a drugi proton akceptor. Udaljenost atoma između kojih se stvara veza mora biti manja od 3,6 Å i njihovi kovalentni radijusi moraju biti udaljeni za najmanje 1 Å.
- c) **Van der Waalsove veze** – mogu se stvoriti između atoma različitih residuuma ako su oni udaljeni više od zbroja njihovih Van der Waalsovih radijusa i ako im udaljenost Van der Waalsovih radijusa ne prijelazi 1,5Å.

U iPFAM bazi najveći broj veza čine Van der Waalsove veze što se podudara i sa stvarnom situacijom u organizmima jer prilikom interakcija proteinskih lanaca dolazi do kontakta velikog broja atoma.

U tablici 6 zapisani su kovalentni i Van der Waalsovi radijusi korišteni u iPFAM algoritmu (iste vrijednosti koristi i program za molekularnu grafiku RASMOL).

Tablica 6: kovalentni i Van der Waalsovi radijusi korišteni u iPFAM-u

Atom	Kovalentni radijus	Van der Waalsov radijus
N	0,680 Å	1,507 Å
C	0,720 Å	1,872 Å
O	0,680 Å	1,400 Å
S	1,020 Å	1,848 Å

4.5 LSS&Ruđer algoritam za definiranje interakcija

Iako iPFAM baza podataka sadrži na zadovoljavajući način pronađene veze između aminokiselina te veze su definirane samo između poznatih proteinskih domena pa na taj način nisu obuhvaćene sve veze koje mogu nastati između proteinskih lanaca i nisu obuhvaćene sve datoteke koje sadrže komplekse proteinskih lanaca. Još jedna mana iPFAM baze je što ne sadrži informacije o jako važnim hidrofobnim vezama koje mogu nastati i o jakim ionskim vezama koje se mogu stvoriti na većim udaljenostima.

Zbog potrebe za preciznim pronalaženjem što većeg broja veza između proteinskih lanaca u ovome radu izrađen je algoritam koji na osnovu kemijskih svojstava i udaljenosti definira interakcijske parove residuuma. Algoritam za pronalaženje interakcijskih parova te sami mogući interakcijski parovi i njihove udaljenosti definirani su uz pomoć Dr. sc. Sanje Tomić zaposlene na Institutu Ruđer Bošković kao viša znanstvena suradnica.

Definirani su uvjeti za stvaranje sve četiri vrste nekovalentnih veza koje mogu nastati između residuuma (ionske, vodikove, Van der Waalsove i hidrofobne) i definirano je **2.911** različitih interakcijskih parova (ne ubrajajući Van der Waalsove veze).

4.5.1 Ionske veze

Od svih definiranih veza ionske veze imaju najveći doseg. Veze se mogu stvoriti između negativno (akceptor) i pozitivno (donor) nabijenih residuuma.

Ionske veze imaju domet (najveću dozvoljenu udaljenost između dva suprotno nabijena atoma) od 6,0 Å, definirano je **968** različitih parova, a potencijalni sudionici su slijedeći: **DONORI** – Lys N(NH₃⁺), Arg N(NH₂⁺),

terminalni $\text{N}(\text{NH}_3^+)$; **AKCEPTORI** – Asp $\text{O}(\text{COO}^-)$, Glu $\text{O}(\text{COO}^-)$, terminalni O (COO^-)

4.5.2 Vodikove veze

Najveća dozvoljena udaljenost između donora i akceptora pri stvaranju vodikovih veza iznosi 4,7 Å. Definirano je **1.894** različita para u tri skupine potencijalnih parova donor-akceptor.

- a) **DONORI** – Asn $\text{N}(\text{H}_2)$, Cys $\text{S}(\text{H})$, Gln $\text{N}(\text{H}_2)$, His $\text{N}(\text{H})$, Ser $\text{O}(\text{H})$, Thr $\text{O}(\text{H})$, Trp $\text{N}(\text{H})$, Tyr $\text{O}(\text{H})$, N glavnog lanca; **AKCEPTORI** – Asp $\text{O}(\text{COO}^-)$, Glu $\text{O}(\text{COO}^-)$
- b) **DONORI** – Lys $\text{N}(\text{NH}_3^+)$, Arg $\text{N}(\text{NH}_2^+)$; **AKCEPTORI** – Asn O $(\text{C}=\text{O})$, Cys $\text{S}(\text{H})$, Gln O $(\text{C}=\text{O})$, His N, Ser $\text{O}(\text{H})$, Thr $\text{O}(\text{H})$, Tyr $\text{O}(\text{H})$, O glavnog lanca
- c) **DONORI** – Asn $\text{N}(\text{H}_2)$, Cys $\text{S}(\text{H})$, Gln $\text{N}(\text{H}_2)$, His $\text{N}(\text{H})$, Ser $\text{O}(\text{H})$, Thr $\text{O}(\text{H})$, Trp $\text{N}(\text{H})$, Tyr $\text{O}(\text{H})$, N glavnog lanca; **AKCEPTORI** – Asn O $(\text{C}=\text{O})$, Cys $\text{S}(\text{H})$, Gln O $(\text{C}=\text{O})$, His N, Ser $\text{O}(\text{H})$, Thr $\text{O}(\text{H})$, Tyr $\text{O}(\text{H})$, O glavnog lanca

4.5.3 Van der Waalsove veze

Van der Waalsove veze nastaju uslijed fluktuacija elektronskog oblaka, i mogu nastati između bilo kojeg para residuuma, stoga je definirano da su dva residuuma povezana Van der Waalsovom vezom ako postoji barem jedan par atoma čije se jezgre nalaze na udaljenosti d manjoj od zbroja njihovih Van der Waalsovih radijusa i vrijednosti u iznosu 1,125 Å.

$$d < R_1 + R_2 + 1,125 \quad (4.3)$$

Vrijednost od 1,125 Å dobivena je aproksimacijom uzimajući u obzir moguće fluktuacije elektronskih oblaka i empirijskim putem uspoređujući dobivene vrijednosti s onima iz iPFAM baze. Koristeći Van der Waalsove radijuse navedene u poglavlju 3.1.1 najveći doseg Van der Waalsove veze je pri interakciji dva tetrahedralna ugljika: $1,87 + 1,87 + 1,125 = 4,865$ Å.

4.5.4 Hidrofobne veze

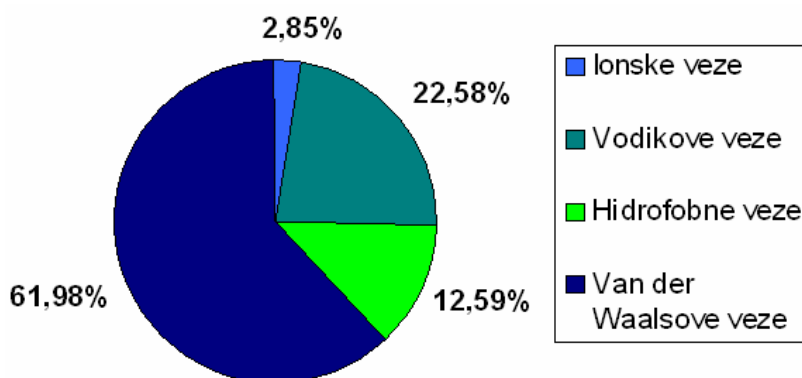
Hidrofobne veze definirane su između sedam nepolarnih residuuma navedenih u poglavlju 4.1.4, ako postoji barem jedan par atoma, od kojih svaki pripada po jednom nepolarnom residuumu, čija je udaljenost između jezgara manja od 4,7 Å. Vrijednost za najveću dopuštenu udaljenost određena je uzimajući u obzir mogućnost pojavljivanja vodika između atoma koji tvore vezu.

Definirano je **49** mogućih parova residuuma koji mogu tvoriti hidrofobne veze.

4.5.5 Rezultati provjere rada algoritma

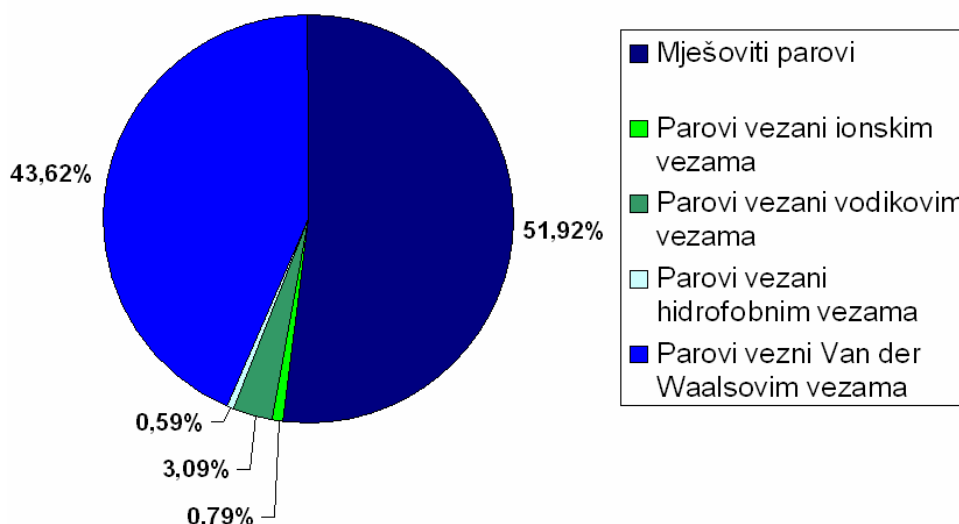
Provjera rada algoritma izvedena je na **2.848** različitih kompleksa (**10.484** lanca) pronađenih u *Protein Data Bank* bazi proteinskih struktura. Provjera rada je izvršena tako da je izbrojano koliko je veza svake od četiri vrste veza pronađeno između residuuma i koliko je interakcijskih parova povezano sa više od jedne vrste veza, a koliko samo sa jednom od četiri vrste veza.

Pronađeno je **ukupno 932.213** veza od čega je **26.544 (2,85%) ionskih veza**, **210.499 (22,58%) vodikovih veza**, **117.364 (12,59%) hidrofobnih veza** i **577.806 (61,98%) Van der Waalsovih veza**. Potrebno je napomenuti da svaki interakcijski par ovdje može imati najviše jednu vezu određene vrste iako postoji mogućnost da su više atoma istog interakcijskog para povezana istom vrstom veze.



Slika 14: usporedba vrsta pronađenih veza Lss&Ruđer algoritmom

Pronađeno je **ukupno 605.268** interakcijskih parova od čega je **314.229 (51,92%) mješovitih** (između residuuma postoji više od jedne vrste veze), **4.776 (0,79%) ionskih**, **18.712 (3,09%) vodikovih**, **3.551 (0,59%) hidrofobnih** i **264.000 (43,62%) Van der Waalsovih**.



Slika 15: usporedba vrsta pronađenih interakcijskih parova Lss&Ruđer algoritmom

4.6 Usporedba rezultata različitih algoritama

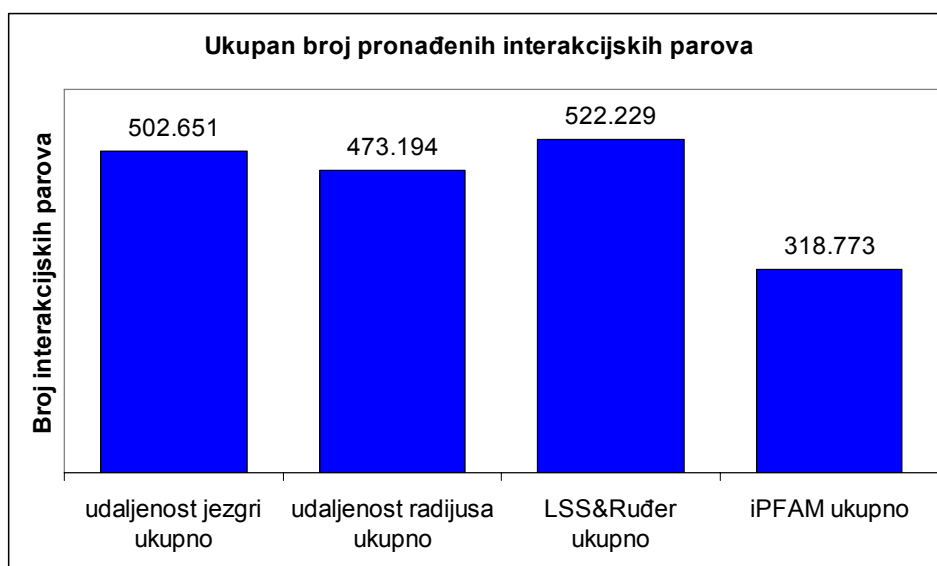
Usporedba različitih algoritama za pronalaženje interakcijskih parova i mjesta interakcija izvedena je na **2.848** različitih kompleksa (**10.484** lanca) pronađenih u *Protein Data Bank* bazi proteinskih struktura. Od ukupnog broja kompleksa u *iPFAM* bazi interakcija nalazile su se informacije o interakcijama **2.445 (85,85%)** traženih kompleksa, tako da je usporedba izvršena na tolikom broju kompleksa.

Usporedba algoritama je napravljena prema broju interakcijskih parova i prema broju mjesta interakcija koje pojedini algoritam pronalazi.

4.6.1 Usporedba pronađenih interakcijskih parova

Prema broju pronađenih interakcijskih parova uspoređen je LSS&Ruđer algoritam sa algoritmima udaljenosti jezgri atoma od 4,5 Å, udaljenosti Van der Waalsovih radijusa atoma od 1,0 Å i podacima dobivenim iz *iPFAM* baze interakcija.

Ukupan broj pronađenih interakcijskih parova: udaljenost jezgri – **502.651** par; udaljenost radijusa – **473.194** para; LSS&Ruđer – **522.229** parova; *i*PFAM – **318.773** parova. LSS&Ruđer algoritmom pronađeno je najviše interakcijskih parova jer dopušta najveće udaljenosti između pojedinih atoma, dok najmanje parova postoji u *i*PFAM bazi jer ona prikuplja samo parove unutar poznatih proteinskih domena. Unutar sva četiri skupa pronađeno je **300.502** zajednička interakcijska para.



Slika 16: ukupan broj pronađenih interakcijskih parova

U tablici 7 zapisani su rezultati usporedbe algoritama. Uspoređeno je koliko algoritmi pronalaze jednakih interakcijskih parova, koliko novih parova pronalazi LSS&Ruđer algoritam koji drugi algoritmi ne pronalaze i koliko parova LSS&Ruđer algoritam nije pronašao, a drugi algoritam jest. Usporedba sa *i*PFAM bazom napravljena je na onim parovima koji ulaze unutar poznatih proteinskih domena i na onima koji ne ulaze (tj. oni parovi koje je LSS&Ruđer algoritam pronašao, a u *i*PFAM bazi ne postoje jer se u njoj nalaze samo parovi onih residuuma koji se nalaze unutar domena).

Tablica 7: usporedba dobivenih interakcijskih parova

LSS&Ruđer		jednako	različito	nije pronašao
udaljenost jezgri		486.781	35.448	15.870
udaljenost radijusa		473.194	49.035	0
iPFAM	unutar domena	314.420	49.529	4.353
	van domena	X	316.578	X

Uspoređujući LSS&Ruđer algoritam i algoritam udaljenosti jezgri sa vrijednosti praga udaljenosti 4,5 Å zaključuje se da su sve veze koje LSS&Ruđer algoritam nije pronašao one veze koje prelaze dozvoljene udaljenosti Van der Waalsovih veza, a ne pripadaju niti jednoj drugoj grupi veza čiji su potencijalni parovi pojedinačno definirani. Svi novi pronađeni parovi prelaze prag od 4,5 Å, te uključuju sve četiri moguće vrste parova.

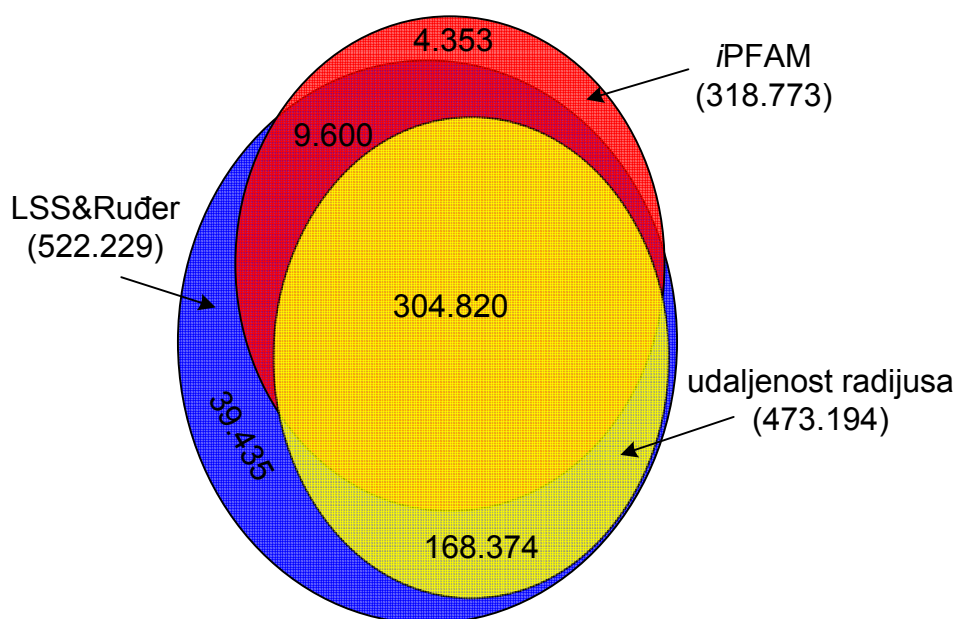
Svi parovi pronađeni algoritmom udaljenosti Van der Waalsovih radijusa sa pragom 1,0 Å pronađeni su i LSS&Ruđer algoritmom jer su u biti ti parovi Van der Waalsove veze LSS&Ruđer algoritma sa manjim pragom (prag Van der Waalsovih veza je 1,125 Å).

Iako se prema rezultatima čini da su najveće razlike između iPFAM baze i LSS&Ruđer algoritma, to nije nužno točno jer se mora uzeti u obzir da iPFAM ne sadrži sve proteinske lance i sadrži samo interakcijske parove između poznatih proteinskih domena. Novi parovi pronađeni LSS&Ruđer algoritmom su uzrok uzimanja većih pragova udaljenosti kod vodikovih i ionskih interakcija i uvođenje hidrofobnih interakcija. Razlike kod novo nađenih parova i ne pronađenih parova rezultat su različitog praga Van der Waalsovih veza (1,125 Å LSS&Ruđer; 1,5 Å iPFAM) i korištenih različitih Van der Waalsovih radijusa.

Prilikom uspoređivanja LSS&Ruđer algoritma sa drugim poznatim algoritmima izvedena je i međusobna usporedba rezultata dva danas najčešće korištena načina prikupljanja interakcijskih parova, iPFAM baza

interakcija i algoritam udaljenosti radijusa, i LSS&Ruđer algoritma. Slikom 17 ilustriran je presjek skupova pronađenih interakcijskih parova ova tri načina.

Iz ove usporedbe može se vidjeti da se većina interakcijskih parova nalazi u sva tri skupa (95,62% *i*PFAM parova, 64,41% parova algoritma udaljenosti radijusa i 58,37% parova LSS&Ruđer algoritma), a taj broj bio bi još veći da *i*PFAM sadrži interakcijske parove izvan poznatih proteinskih domena.

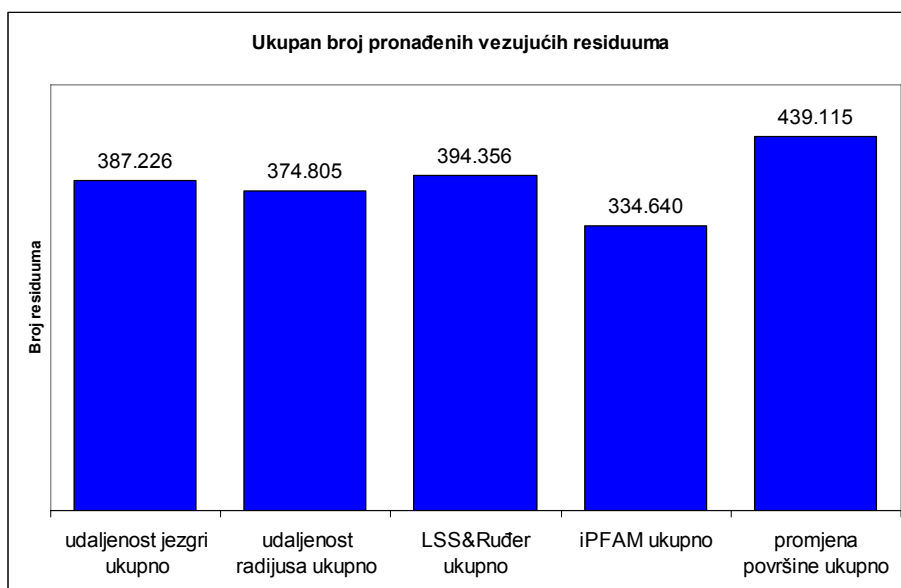


Slika 17: presjek skupova pronađenih interakcijskih parova

4.6.2 Usporedba pronađenih vezujućih residuuma

Prema broju pronađenih vezujućih residuuma uspoređen je LSS&Ruđer algoritam sa algoritmima udaljenosti jezgri atoma od 4,5 Å, udaljenosti Van der Waalsovih radijusa atoma od 1,0 Å, promjena dostupnog područja površine od 1,0 Å² uz radijus sonde 1,40 Å i podacima dobivenim iz *i*PFAM baze interakcija.

Ukupan broj pronađenih vezujućih residuuma: udaljenost jezgri – **387.226** residuuma; udaljenost radijusa – **374.805** residuuma; promjena površine – **439.115** residuuma; LSS&Ruđer – **394.356** residuuma; *i*PFAM – **334.640** residuuma.



Slika 18: ukupan broj pronađenih vezujućih residuuma

U tablici 8 zapisani su rezultati usporedbe algoritama. Uspoređeno je koliko algoritmi pronalaze jednakih vezujućih residuuma, koliko različitih residuuma pronalazi LSS&Ruđer algoritam od drugih algoritama i koliko residuuma LSS&Ruđer algoritam nije pronašao, a drugi algoritam jest. Usporedba sa iPFAM bazom napravljena je na onim residuumima koji ulaze unutar poznatih proteinskih domena i na onima koji ne ulaze (tj. oni residuumi koje je LSS&Ruđer algoritam pronašao, a u iPFAM bazi ne postoje jer se u njoj nalaze samo vezujući residuumi koji se nalaze unutar domena).

Tablica 8: usporedba dobivenih vezujućih residuuma

LSS&Ruđer		jednako	različito	nije pronašao
udaljenost jezgri		382.175	12.181	5.051
udaljenost radijusa		374.805	19.551	0
iPFAM	unutar domena	259.649	67.958	74.991
	van domena	X	66.749	X
promjena površine		387.187	7.169	51.928

Razlike između algoritama manje su pri traženju vezujućih residuuma nego kod traženja interakcijskih parova. Razlog tome je što svi algoritmi većinom pronađu iste vezujuće residue, ali ne pronađu jednak broj njima pripadajućih interakcijskih parova.

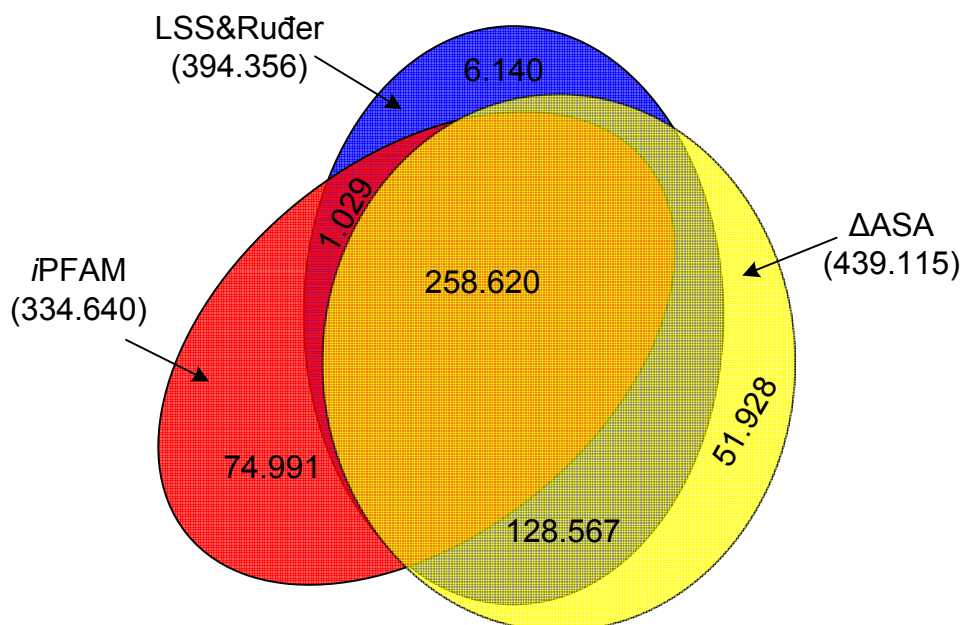
Velike razlike LSS&Ruđer algoritma i iPFAM baze podataka su, kao i kod pronalaženja interakcijskih parova, uzrokovane pronalaženjem van domenskih parova LSS&Ruđer algoritmom, razlikama u korištenim Van der Waalsovima radijusima, te nedostatku hidrofobnih i daleko dosežućih ionskih i vodikovih veza.

Algoritmom kojim se ipak pronađe najviše vezujućih residuuma je algoritmom promjena otapalu dostupnog područja površine. Razlog tome je i sam izvor ove metode koja se koristila za pronalaženje većih dijelova površine (engl. *surface patch*) proteina koja sudjeluje u interakciji kako bi postojalo više podataka za predikciju vezujućih residuuma. Iz rezultata može se zaključiti da se tom metodom obično pronađe većina residuuma koje pronađe LSS&Ruđer algoritam (98,18 %), a razlog pronađenog većeg broja residuuma je mali prag promjene površine (1,0 Å) pri pronalaženju vezujućih residuuma. Bolji omjer pronađenih i nepronađenih residuuma oba algoritma mogao bi se dobiti povećanjem praga promjene površine kod algoritma promjene otapalu dostupnog područja površine. Točnu vrijednost koja bi donijela tu ravnotežu nije moguće jednoznačno odrediti jer ona izrazito ovisi o samoj strukturi pojedinog kompleksa koji se analizira.

Prilikom uspoređivanja LSS&Ruđer algoritma sa drugim poznatim algoritmima izvedena je i međusobna usporedba rezultata dva danas najčešće korištena načina prikupljanja vezujućih residuuma, iPFAM baza interakcija i algoritam promjene otapalu dostupnog područja površine (Δ ASA algoritam), i LSS&Ruđer algoritma. Slikom 19 ilustriran je presjek skupova pronađenih interakcijskih parova ova tri načina.

Iz rezultata ove usporedbe može se vidjeti da se više od polovice pronađenih vezujućih residuuma svakom metodom nalazi u sva tri skupa (77,28% iPFAM residuuma, 65,58% residuuma Δ ASA algoritma i 58,37% residuuma LSS&Ruđer algoritma). Ovi rezultati iako pokazuju velik stupanj

sličnosti različitih metoda pokazuju i koliko njihovi rezultati ovise o strukturi proteinskih lanaca i fizio-kemijskim kriterijima koje koriste.



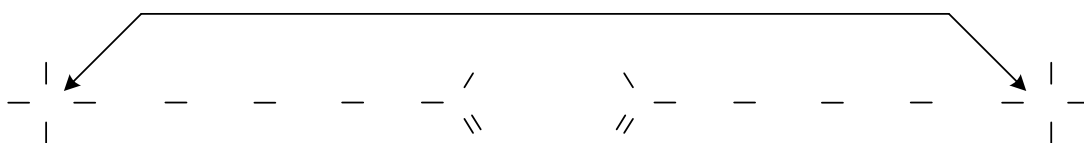
Slika 19: presjek skupova pronađenih vezujućih residuuma

4.7 Predstavljanje residuuma pomoću koordinata jezgre C_α atoma

Svi algoritmi za pronalaženje interakcijskih parova residuuma između dva proteinska lanca bazirani su na međusobnoj udaljenosti residuuma, tj. atoma koji ih grade. Kao i kod algoritama za analizu geometrijskih svojstava residuuma i atoma ovdje se javlja potreba za pretraživanjem prostora. Pošto veze definiramo između residuuma potrebno je na neki način svaki residuum predstaviti jednom točkom u prostoru. Najjednostavnije rješenje pokazalo se predstavljajući svaku aminokiselinu jednim od atoma koji ju gradi i na taj način iskoristiti već razvijeni algoritam za pronalaženje susjednih atoma. Budući da C_α atom na neki način predstavlja srž svakog residuuma svaki residuum u prostoru predstavljen koordinatama jezgre C_α atoma.

Prvi korak pri pronalaženju susjednih residuuma je pretraživanje prostora za svim C_α atomima koji se nalaze unutar određenog praga udaljenosti. Pri ovakvom pretraživanju potrebno je bilo paziti da udaljenost između dva

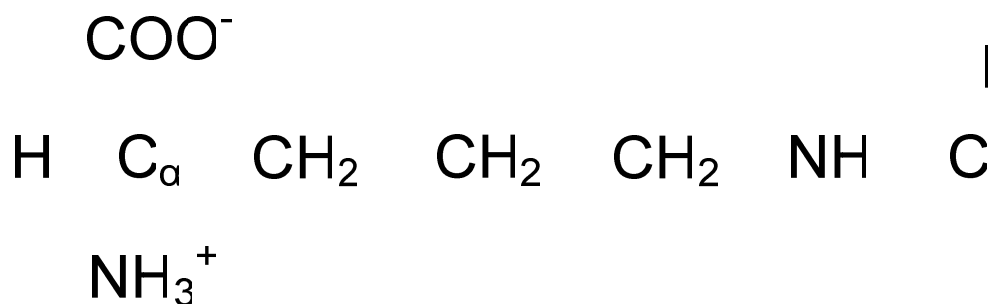
residuuma može biti znatno manja od udaljenosti između njihovih C_{α} atoma. Iz tog razloga je bilo potrebno na vrijednost dopuštene udaljenosti između residuuma dodati određenu vrijednost kako bi se eliminirala mogućnost za propustom neke od postojećih veza. Dodatna vrijednost koja se dodavala određena je izračunom najveće moguće udaljenosti između C_{α} atoma dva najduža residuuma kada su oni u potpunosti ispruženi. Dodatna vrijednost je udaljenost C_{α} atoma u potpunosti ispruženih *Arginine*-a i iznosi 14,3 Å.



Slika 20: udaljenost C_{α} atoma dva *Arginine*-a

Nakon što su na navedeni način prikupljeni svi residuumi koji se nalaze u susjedstvu promatranog, na svakom od njih provedena su ispitivanja da li zadovoljavaju zahtjeve za stvaranje veze.

Ovakav pristup pronalaženju interakcijskih parova također se pokazao bržim od pronalaženja veza uspoređujući udaljenosti svakog atoma svakog residuuma jednog lanca od svakog atoma svakog residuumom drugog lanca.



5 PSAIA – Protein Structure And Interaction Analyzer

PSAIA predstavlja završni produkt ovoga rada. To je softverski alat koji služi za proračun svojstava geometrijske strukture, pronalaženje interakcijskih parova i mjesta interakcija polipeptida čija je struktura zapisana u PDB datotekama.

PSAIA se sastoji od dva odvojena modula:

- a) **modul za analizu svojstva geometrijske strukture** – mogućnost proračuna svih svojstava navedenih u poglavlju 3
- b) **modul za analizu interakcija** – mogućnost pronalaženja interakcija na sve načine navedene u poglavlju 4

Oba modula kao ulazne podatke prihvaćaju jednu ili više standardnih datoteka iz PDB baze podataka, a kao format izlaznih podataka nude opciju XML (.xml ekstenzija) i/ili tabličnog zapisa (.tbl ekstenzija).

Zbog potrebe za radom alata pod više operativnih sustava (Linux i Windows), te potrebom za brzim radom implementiranih algoritama cijeli alat razvijen je u C/C++ programskom jeziku.

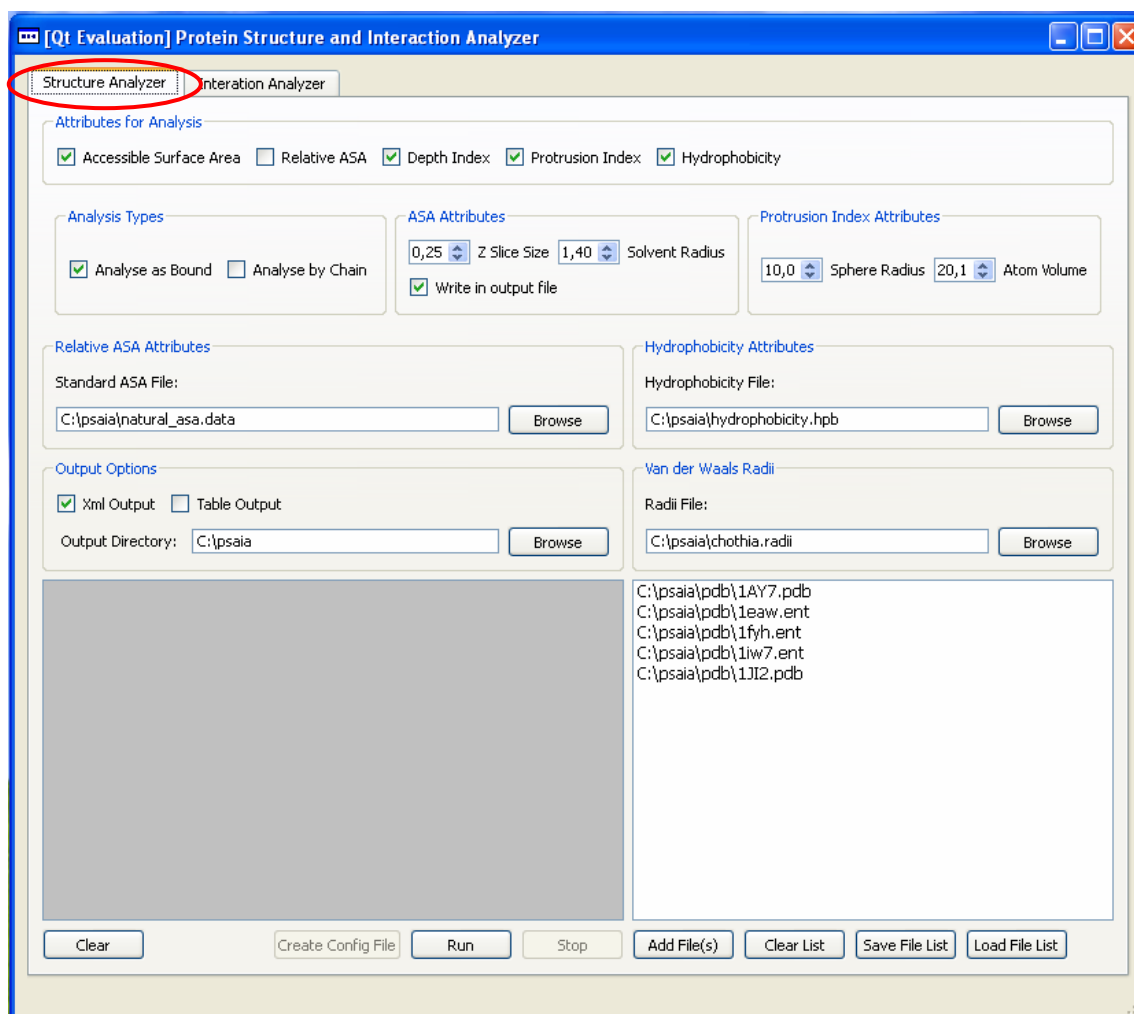
Alat sadrži i korisničko sučelje izrađeno u kros-platformnom GUI razvojnom alatu Qt (<http://www.trolltech.com/products/qt/>) izrađenom u C++ programskom jeziku.

Cijela aplikacija izrađena je u više-dretvenom (*multi-threaded*) dizajnu kako bi se korisniku omogućilo sigurno zaustavljanje rada programa u bilo kojem trenutku.

Uz samu aplikaciju izrađena je i statička biblioteka (za Windows OS i Linux OS) namijenjena za razvoj daljnjih aplikacija. Biblioteka sadrži niz klasa i funkcija razvijenih u C/C++ programskom jeziku koje služe za učitavanje i analizu strukture peptida koji se nalaze u PDB datotekama.

5.1 Modul za analizu svojstva geometrijske strukture

Nakon pokretanja programa modul se može odabrati pritiskom miša na tabulator „Structure Analyzer“. Ovaj modul koristi se za istovremenu analizu jednog ili više geometrijskih svojstava proteina. Analiza strukture se provesti na strukturi u cijelosti ili za svaki lanac unutar strukture zasebno.



Slika 21: modul za analizu svojstva geometrijske strukture

Prozor modula sastoji se od više grupa atributa koji služe za podešavanje rada modula, od dijela za upravljanje radom modula i dijela za upravljanje učitanim datotekama

5.1.1 Podešavanje rada modula

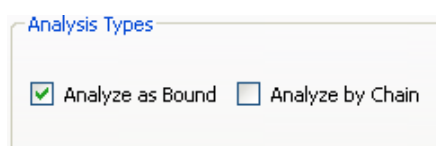
Svojstava residuuma koja se mogu izabrati za analizu – dostupno područje površine (engl. *Accessible Surface Area*), relativno dostupno

područje površine (engl. *Relative ASA*), ukopanost (engl. *Depth Indeks*), izbočenost (engl. *Protrusion Indeks*) i hidrofobnost (engl. *Hydrophobicity*). Korisnik može izabrati proizvoljan broj (1 – 5) svojstava koja će se analizirati na učitanim datotekama, uz napomenu da će se izborom ukopanosti i/ili relativne ASA morati uključiti opcija za računanje ASA. Korisnik se tada neće morati brinuti za uključivanje opcije ASA, jer će se ta opcija sama uključiti.



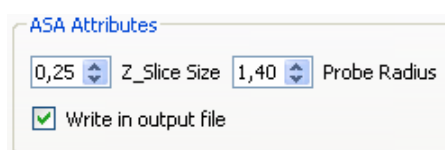
Slika 22: izbor svojstva strukture za analizu

Korisniku su na izbor dane dvije **vrste analize peptida** – analiza se može vršiti na svakom lancu pojedinačno (engl. *Analyze by Chain*) i/ili na peptidu u cijelosti (engl. *Analyze as Bound*). Rezultat odabira obje opcije biti će dvije različite datoteke.



Slika 23: izbor analize peptida

Atributi za izračun ASA – rezolucija podjele sfere na ravnine (engl. *Z_Slice Size*) , radijus sferne sonde (engl. *Probe Radius*). U slučaju da korisnik želi proračun ukopanosti i/ili relativne ASA što zahtjeva izračun ASA, ali ne želi da se vrijednosti izračunate za ASA ispišu u izlaznoj datoteci ima opciju isključivanja ispisa ASA (engl. *Write in output file*).



Slika 24: izbor atributa za izračun ASA

Ne postoje posebni **atributi za izračun ukopanosti**, nego korisnik samo mora odabrati vrijednosti atributa za izračun ASA.

Atributi za izračun izbočenosti – radijus sfere unutar koje se broji količina susjednih atoma (engl. *Sphere Radius*), prosječni volumen atoma (engl. *Atom Volume*).

Slika 25: atributi za izračun izbočenosti

Za izračun **relativne ASA** potrebno je učitati datoteku koja sadrži zapise standardnih vrijednosti ASA za svaku poznatu aminokiselinu. Uz program dolazi datoteka (natural_asa.data) sa vrijednostima standardnih ASA izračunatim prema opisu iz poglavlja 3.1.2, ali se korisniku daje opcija izbora vlastite datoteke. Jedini zahtjev je da se poštuje format zapisa (stupci se počinju brojati od broja 1):

Tablica 9: format zapisa datoteke standardnih ASA

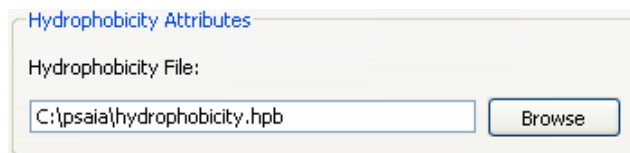
Stupci	Zapis
1 – 3	oznaka residuuma
5 – 10	ukupna ASA
12 – 17	ASA glavnog lance
19 – 24	ASA bočnog lanca
26 – 31	nepolarna ASA
33 – 38	polarna ASA

Primjer zapisa standardnih ASA:

ALA 107.24 43.32 63.92 76.06 31.17

Slika 26: učitavanje datoteke standardnih vrijednosti ASA

Za postavljanje **vrijednosti hidrofobnosti** potrebno je učitati datoteku sa vrijednostima hidrofobnosti pojedinih residuuma. Uz program dolazi datoteka (hydrophobicity.hpb) sa zapisanim vrijednostima hidrofobnosti. Korisniku se ostavlja izbor datoteke sa vrijednostima hidrofobnosti, a jedini je zahtjev da se poštuje format zapisa. Zapis započinje tro-slovnom oznakom residuuma koju slijedi jednim mjestom (engl. *space*) odvojena vrijednost hidrofobnosti.

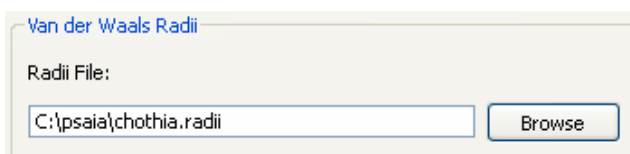


Slika 27: učitavanje datoteke sa vrijednostima hidrofobnosti

Pri analizi peptida izvađenih iz PDB datoteka svakome atomu pridružuje se odgovarajuća **vrijednost Van der Waalsovog radijusa** učitana iz datoteke koja sadrži vrijednosti za svaki atom. Uz program dolazi datoteka (chothia.radii) sa vrijednostima Van der Waalsovih radijusa navedenim u poglavlju 3.1.1. Korisniku daje mogućnost učitavanja vlastite datoteke sa vrijednostima radijusa, a jedini je zahtjev da se poštuje format zapisa. Zapis započinje oznakom residuuma i njemu pripadajućeg atoma (oznaka atoma mora odgovarati onoj iz PDB datoteke). Oznake residuuma i atoma spojene su podvlakom (_), a mogu zauzimati najviše sedam znakova. Nakon toga od devetog znaka na dalje mora biti zapisana vrijednost Van der Waalsovog radijusa.

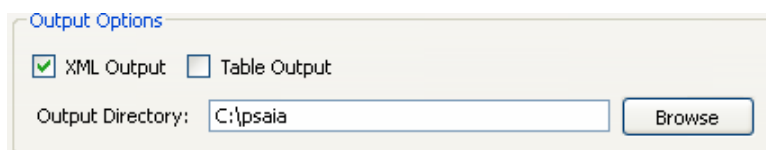
Primjer zapisa Van der Waalsovih radijusa za *Alanine*:

```
ALA_N  1.65
ALA_CA 1.87
ALA_C  1.76
ALA_O  1.40
ALA_CB 1.87
```



Slika 28: učitavanje datoteke sa vrijednostima Van der Waalsovih radijusa

Od **izlaznih opcija** korisnik može izabrati XML (engl. *XML Output*) i/ili tablični (engl. *Table Output*) format izlaznih datoteka te izlazni direktorij (engl. *Output Directory*) u koji se spremaju izlance datoteke.



Slika 29: izlazne opcije modula za analizu strukture

Imena izlaznih datoteka stvaraju se od PDB identifikatora peptida (npr. 1AY7) i podvlakom odvojenog identifikatora tipa analize: **bound** za analizu peptida u cijelosti ili **unbound** za analizu svakoga lanca zasebno. Postoje četiri moguća različita imena izlaznih datoteka za svaki peptid, npr. za peptid identifikatora 1ay7 moguća su slijedeća imena izlaznih datoteka: **1ay7_bound.xml**, **1ay7_unbound.xml**, **1ay7_bound.tbl**, **1ay7_unbound.tbl**.

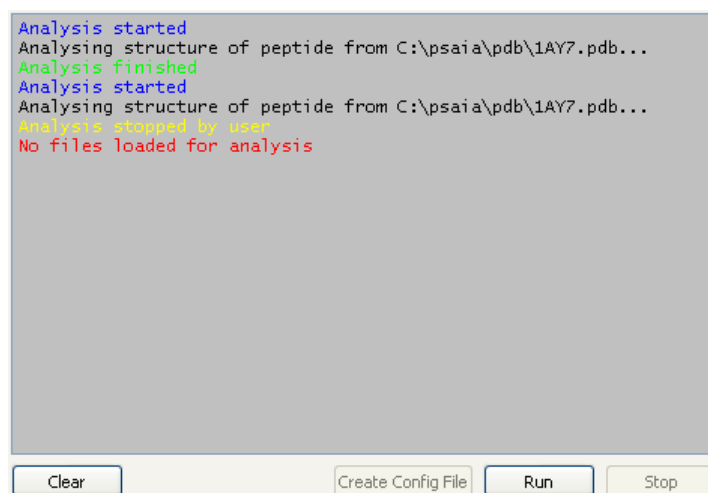
5.1.2 Upravljanje radom modula

Dio prozora za upravljanje radom modula sastoji se od ekrana za praćenje tijeka programa, te gumbi za pokretanje i zaustavljanje rada, brisanja ekrana i stvaranja konfiguracijske datoteke.

Na ekranu za praćenje tijeka programa ispisuju se poruke u trenucima početka, kraja ili prekida analize, početka analize pojedine učitane datoteke, poruke koje izvještavaju o određenim greškama pri pokretanju programa (npr. ako nije učitana niti jedna datoteka za analizu, ako jedna od specificiranih datoteka ne postoji itd.). Pritiskom tipke *Clear* briše se sav trenutni ispis na ekranu.

Analiza se pokreće pritiskom na gumb *Run*, a tokom njenog trajanja može se prekinuti u svakom trenutku pritiskom gumba *Stop*.

Pritiskom gumba *Create Config File* može se stvoriti konfiguracijska datoteka koja se može koristiti pri korištenju konzolne verzije ovog modula.

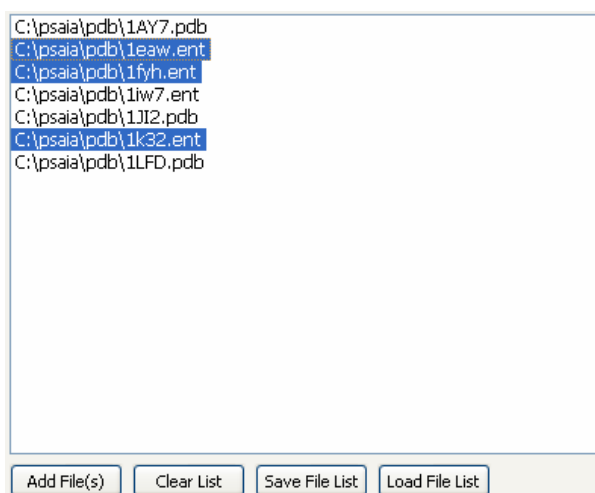


Slika 30: dio prozora za upravljanje radom modula

5.1.3 Upravljanje učitanim datotekama

Dio prozora za upravljanje učitanim datotekama sastoji se od ekrana sa listom učitanih datoteka, te gumbi za dodavanje datoteka, brisanje svih datoteka iz liste, spremanje i učitavanje liste datoteka.

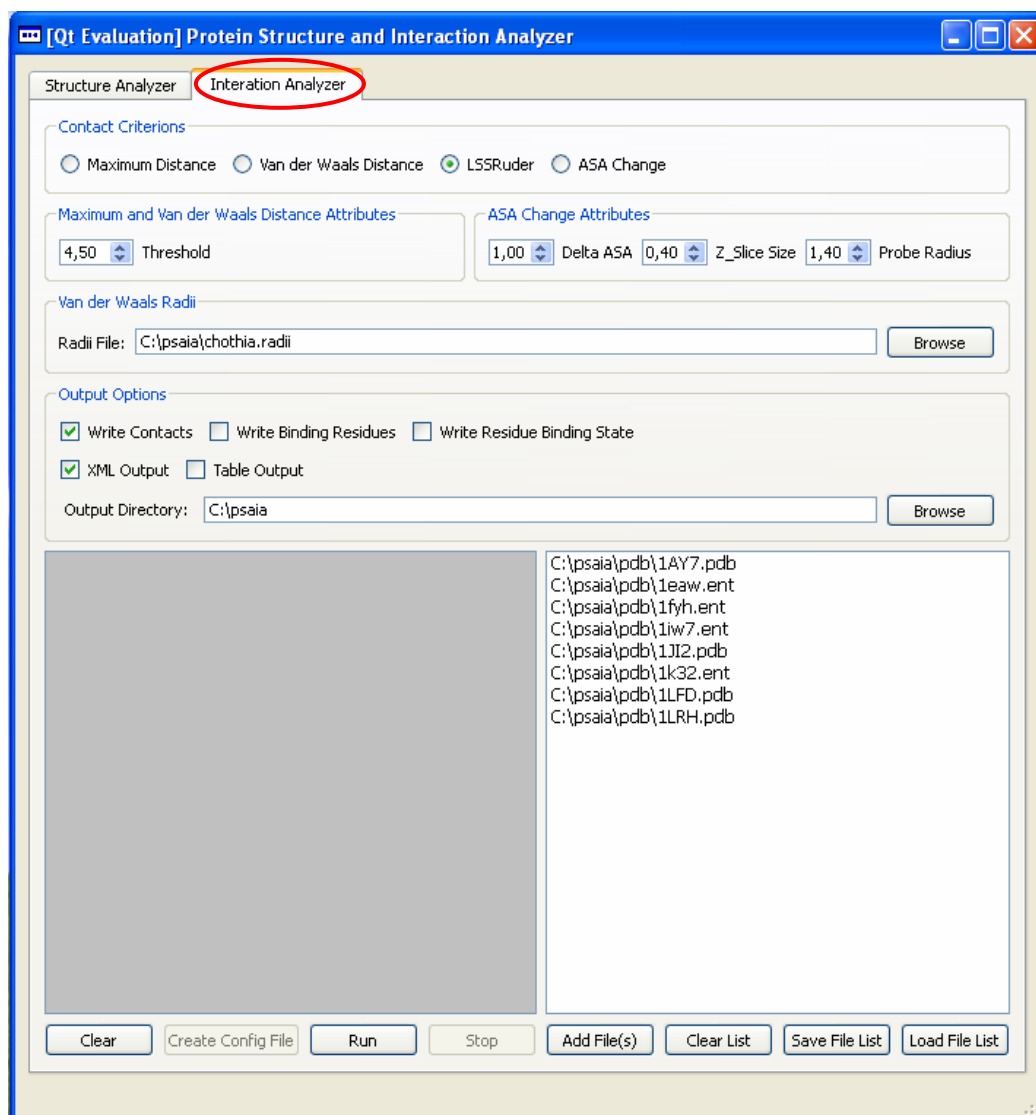
Datoteke se mogu dodati u listu pritiskom na gumb *Add File(s)* pri čemu se otvora prozor za izbor jedne ili više datoteka. Datoteke se mogu brisati iz liste pritiskom na gumb *Clear List*, što će rezultirati pražnjenjem liste, ili odabirom datoteka na ekranu te njihovim brisanjem pritiskom tipke *delete* na tipkovnici. Cijela lista učitanih datoteka može se spremiti pritiskom na gumb *Save File List*, a datoteka u koju se lista spremi može se koristiti i u konzolnoj verziji modula. Spremljena lista datoteka može se učitati pritiskom na gumb *Load File List*.



Slika 31: dio prozora za upravljanje učitanim datotekama

5.2 Modul za analizu interakcija

Nakon pokretanja programa modul se može odabrati pritiskom miša na tabulator „Interaction Analyzer“. Ovaj modul koristi se za pronalaženje interakcijskih parova između proteinskih lanaca i vezujućih residuuma na pojedinom lancu. Na izbor pružena su tri različita algoritma za pronalaženje interakcijskih parova i vezujućih residuuma i jedan algoritam za pronalaženje samo vezujućih residuuma.



Slika 32: modul za analizu interakcija

5.2.1 Podešavanje rada modula

Korisnik ima mogućnost podesiti modul da radi na jedan od **četiri kriterija definiranja interakcijskih parova i mjesta interakcija** (engl. *Contact Criteria*): interakcije na osnovu udaljenosti jezgri atoma (engl. *Maximum Distance*), interakcije na osnovu udaljenosti Van der Waalsovih radijusa (engl. *Van der Waals Distance*), LSS&Ruđer algoritam (LSS&Ruder), mjesta interakcije na osnovu promjene otapalu dostupnog područja površine (engl. *ASA Change*).

Contact Criteria

☐ Maximum Distance
 ☐ Van der Waals Distance
 ☒ LSS Ruder
 ☐ Thornton

Slika 33: izbor kriterija definiranja interakcija

Željeni **prag** (engl. *Threshold*) potreban za pronalaženje interakcija na osnovu udaljenosti residuuma podešava se istim atributom za oba algoritma (kriterij udaljenost jezgre i kriterij udaljenosti Van der Waalsovih radijusa). Korisnik ne mora paziti na značajne razlike udaljenosti koje ovi algoritmi koriste jer se izborom bilo kojeg od ta dva algoritma vrijednost samostalno namješta na prikladnu, tj. onu koja se zadnji put koristila pri analizi odabranim algoritmom.

Maximum and Van der Waals Distance Attributes

4,50 Threshold

Slika 34: podešavanje atributa za algoritme bazirane na udaljenosti residuuma

Za **pronalaženje mjesta interakcija na osnovu promjene ASA** potrebno je namjestiti vrijednost promjene ASA (engl. *Delta ASA*) na osnovu koje se identificira mjesto interakcije, te vrijednosti *z_slice* (engl. *Z_Slice Size*) i radijusa sonde (engl. *Probe Radius*) potrebne za izračun ASA.

ASA Change Attributes

1,00 Delta ASA 0,40 Z_Slice Size 1,40 Probe Radius

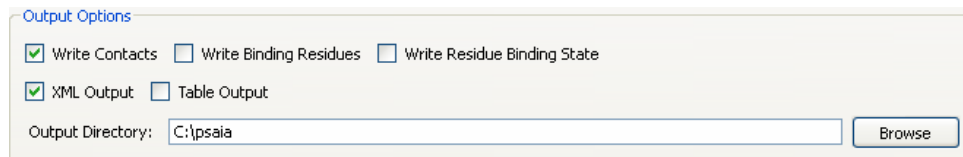
Slika 35: podešavanje atributa za algoritam baziran na promjeni ASA

Opis namještanja vrijednosti **Van der Waalsovih radijusa** odgovara onome opisanome u poglavlju 5.1.1.

Od **izlaznih opcija** korisnik može izabrati XML (engl. *XML Output*) i/ili tablični (engl. *Table Output*) format izlaznih datoteka te izlazni direktorij (engl. *Output Directory*) u koji se spremaju izlaze datoteke. Postoje tri različite vrste podataka koji se mogu ispisivati u izlazne datoteke:

- ispis interakcijskih parova, tj. kontakata residuuma (engl. *Write Contacts*)
- ispis mjesta interakcija, tj. vezujućih residuuma (engl. *Write Binding Residues*) – ispisani su samo residuumi koji sudjeluju u interakciji

- c) ispis vezujućeg stanja residuuma (engl. *Write Residue Binding State*) – ispisuje sve residuume i vrijednost *true/false* ovisno o tome da li residuum sudjeluje u interakciji ili ne



Slika 36: izlazne opcije modula za analizu interakcija

Imena izlaznih datoteka stvaraju se od PDB identifikatora peptida (npr. 1AY7) i podvlakom odvojenog identifikatora vrste podataka koji su zapisani u datoteku: **contacts** za ispis interakcijskih parova, **binding_residues** ispis mjesta interakcija i **binding_status** za ispis vezujućeg stanja residuuma. Postoji šest mogućih različitih imena izlaznih datoteka za svaki peptid, npr. za peptid identifikatora 1ay7 moguća su slijedeća imena izlaznih datoteka: **1ay7_contacts.xml**, **1ay7_binding_residues.xml**, **1ay7_binding_status.xml**, **1ay7_contacts.tbl**, **1ay7_binding_residues.tbl**, **1ay7_binding_status.tbl**.

5.2.2 Upravljanje radom modula

Upravljanje radom modula u potpunosti odgovara onome opisanome u poglavlju 5.1.2.

5.2.3 Upravljanje učitanim datotekama

Upravljanje učitanim datotekama u potpunosti odgovara onome opisanome u poglavlju 5.1.3.

5.3 Usporedba sa postojećim programima

PSAIA ima nekoliko prednosti nad već postojećim alatima. Glavna prednost PSAIA alata je mogućnost istovremene analize više od jednog geometrijskog svojstva i mogućnost izbora nekoliko različitih kriterija pronalaženja proteinskih interakcija i vezujućih residuuma.

Osnovni nedostaci postojećih alata je njihova nemogućnost analize više datoteka od jednom (svaki put kada se želi analizirati nova struktura program

se mora iznova pokretati) i nemogućnost analize pojedinog lanca (ako se želi analizirati pojedini lanac korisnik ga je potreban sam „zapisati“ u zasebnu datoteku i na njoj izvršiti analizu). Već postojeći alati nemaju opciju XML formata izlazne datoteke već tablični zapis koji je manje pogodan za danju analizu na računalu.

Već davno izrađen alat za analizu otapalu dostupnog područja površine NACCES (<http://wolf.bms.umist.ac.uk/naccess>), iako već ustaljen u akademskoj zajednici i odlikovan brzinom, ima svoje nedostatke. Osim već nabrojenih nedostataka koji dijele svi alati nedostaci kod NACCESS-a su slijedeći: inačica alata postoji samo za Linux platformu i C_{α} atom se smatra dijelom bočnog lanca kako bi *Glycine* imao bočni lanac (ta opcija se može isključiti, ali će onda relativna otapalu dostupna područja površina svih aminokiselina imati pogrešne vrijednosti).

DPX i CX alati dostupni su preko web sučelja (<http://hydra.icgeb.trieste.it/dpx>; <http://hydra.icgeb.trieste.it/cx>) što ih čini dostupnima sa svih platformi, ali je potrebno za njihovo korištenje imati pristup internetu (što je danas uobičajeno) tako da su njihovi nedostaci zapravo oni već prije navedeni.

Alati za proračun proteinskih interakcija danas i ne postoje (postoje samo algoritmi za njihovu analizu), ali postoje nekoliko baza podataka koje sadrže već pronađene interakcijske parove. Nedostaci baza podataka o interakcijama su što one ne pružaju namještanje vlastitih parametara i ne sadrže informacije o interakcijama svih zapisa u PDB bazi podataka.

Zbog svih ranije navedenih nedostatak dosadašnjih programa PSAIA predstavlja snažan softverski alat koji bi trebao koristiti znanstvenoj zajednici za lakše pripremanje podataka za daljnju analizu.

6 Zaključak

U sklopu ovoga rada izrađen je softverski alat koji služi za analizu svojstava geometrijske strukture proteina (i nukleinskih kiselina) i analizu interakcija između proteinskih lanaca. Ulazni podatci alata su datoteke sadržane u *Protein Data Bank* bazi proteinskih struktura.

Alat uključuje mogućnost analize svih važnijih geometrijskih svojstava kao što su otapalu dostupno područje površine, ukopanost i izbočenost aminokiselinskih residuuma. Uključena je i mogućnost izbora više vrsta kriterija na temelju kojih se pronalaze interakcijski parovi residuuma i vezujući residuumi. Osnovne prednosti ovoga alata su mogućnost istovremene analize više od jednog geometrijskog svojstva odjednom (dosada razvijeni alati nude analizu samo jednoga od svojstava), mogućnost istovremene analize više molekularnih struktura, tj. mogućnost učitavanja više datoteka odjednom (nije potrebno svaki puta iznova pokretati program kada se želi analizirati nova datoteka). Objedinjavanje analize geometrijskih svojstava proteina i proteinskih interakcija u jedan softverski alat omogućava korisniku jednostavno pripremanje podataka za danju analizu. Korisnički prijateljsko grafičko sučelje omogućava korisniku jednostavno podešavanje rada pojedinog modula. Izlazni podatci u XML formatu velika su prednost nad dosadašnjim programima (koji svoje rezultate ispisuju samo u tabličnom formatu) zbog jednostavnosti i brzine kasnijeg učitavanja podataka sadržanih u XML datoteci.

Razvijen je novi algoritam za pronalaženje interakcijskih parova između proteinskih lanaca, koji se bazira na skupini fizio-kemijskih svojstava residuuma i atoma koji ih grade. Algoritam je razvijen detaljno analizirajući interakcijske parove residuuma koji mogu nastati između proteinskih lanaca. Podudaranje novog algoritma sa već postojećima i rješavanje njihovih nedostataka čine ga jako korisnim sredstvom za analizu proteinskih interakcija.

7 Dodatak A: analiza nukleinskih kiselina

Proučavanjem PDB datoteka utvrđeno je da postoje slučajevi u kojima se uz proteinske lance mogu u strukturi javiti i za njih vezane nukleinske kiseline (lanci nukleotida) koji mogu uzrokovati neispravnost rezultata ako ih se pravilno ne uključi u analizu. Postoji i velik broj PDB datoteka koje sadrže samo strukture molekula DNA i/ili RNA. Iz tih razloga u izrađeni softver (prvotno namijenjen samo za analizu proteinskih lanaca) ugrađena je i podrška za pravilnu analizu geometrijskih svojstava pet osnovnih nukleotida – *Adenine*, *Cytosine*, *Guanine*, *Thymine* i *Uracil*.

Za pravilan izračun **otapalu dostupnog područja površine** nukleotida bilo je potrebno odrediti Van der Waalsove radijuse pojedinih atoma kiselina. Tijekom ovog rada korišteni su radijusi navedeni tablicom 10, a kao što je navedeno u poglavlju 5.1.1 korisniku se ostavlja mogućnost definiranja vlastitih radijusa.

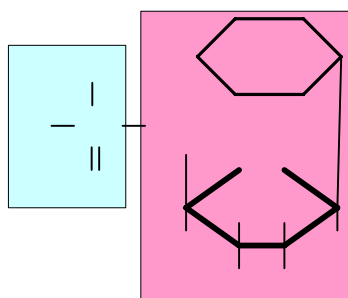
Tablica 10: Van der Waalsovi radijusi atoma nukleotida

Atom	Van der Waalsov radijus
Vodik	1,10 Å
Kisik	1,40 Å
Dušik	1,60 Å
Ugljik	1,80 Å
Fosfor	1,90 Å

Kako bi se zadržao algoritam za proračun ASA i format zapisa kao kod aminokiselinskih residuuma i za nukleotide se računa pet vrijednosti ASA:

- ukupna ASA (engl. *Total*)** – suma svih atoma koji grade nukleotid
- ASA glavnog lanca (engl. *Backbone*)** – suma ASA svih atoma fosfatne grupe nukleotida
- ASA bočnog lanca (engl. *Side-chain*)** – suma ASA svih atoma koji grade šećer pentozu i bazu nukleotida (*Nucleoside*)

- d) **ASA polarnog dijela (engl. *Polar*)** – suma ASA svih polarnih atoma (atomi kisika i dušika) koji grade nukleotid
- e) **ASA nepolarnog dijela (engl. *Non-polar*)** – suma ASA svih nepolarnih atoma (svi atomi osim kisika i dušika) koji grade nukleotid



Slika 37: građa osnovnih nukleotida

Koristeći princip trojki proračunate su i standardne vrijednosti ASA svakog od pet nukleotida koje se koriste za izračun **relativnog otapalu dostupnog područja površine**. Vrijednosti su izračunate su sa vrijednostima $z_{slice} = 0,25\text{\AA}$ i $RSolvent = 1,40\text{ \AA}$, a vrijednosti su prikazane tablicom 11 (sve vrijednosti u tablici izražene su u $\text{\AA}^2$).

Tablica 11: standardne vrijednosti ASA za nukleotide

Nukleotid	Ukupno	Glavni lanac	Bočni lanac	Nepolarno	Polarno
Adenine	291,12	37,18	253,94	177,86	113,26
Cytosine	269,20	37,30	231,90	164,30	104,90
Guanine	295,13	36,96	258,17	172,26	122,87
Thymine	285,84	38,04	247,80	179,67	106,17
Uracil	264,29	37,14	227,15	151,79	112,50

Za analizu **interakcija sa nukleinskim kiselinama** nisu definirani nikakvi kriteriji različiti od onih kod interakcija proteinskih lanaca, zato jer većina tih kriterija zanemaruje kemijske osobine. Potrebno je napomenuti da će sve veze pronađene LSS&Ruđer algoritmom biti Van der Waalsove veze jer nisu definirani dodatni parovi koji određuju ionske, vodikove i hidrofobne veze.

8 Dodatak B: XML sheme

8.1 Svojstva geometrijske strukture

```
<?xml version="1.0" encoding="us-ascii" ?>
<xs:schema targetNamespace="http://www.fer.hr/lss/namespaces/PDB"
  elementFormDefault="qualified"
  xmlns="http://www.fer.hr/lss/namespaces/PDB"
  xmlns:mstns="http://www.fer.hr/lss/namespaces/PDB"
  xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:element name="cPdbPeptide">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="pdb_id" type="xs:string" />
        <xs:element name="analysis_type" type="xs:string" />
        <xs:element name="chains">
          <xs:complexType>
            <xs:sequence>
              <xs:element name="chain" minOccurs="0" maxOccurs="unbounded">
                <xs:complexType>
                  <xs:sequence>
                    <xs:element name="serial_id" type="xs:string" />
                    <xs:element name="residues">
                      <xs:complexType>
                        <xs:sequence>
                          <xs:element name="residue" minOccurs="0" maxOccurs="unbounded">
                            <xs:complexType>
                              <xs:sequence>
                                <xs:element name="name" type="xs:string" />
                                <xs:element name="serial_id" type="xs:string" />
                                <xs:element name="asa" minOccurs="0" maxOccurs="1">
                                  <xs:complexType>
                                    <xs:sequence>
                                      <xs:element name="total" type="xs:double" />
                                      <xs:element name="backbone" type="xs:double" />
                                      <xs:element name="side_chain" type="xs:double" />

```

```

        <xs:element name="non_polar" type="xs:double" />
        <xs:element name="polar" type="xs:double" />
    </xs:sequence>
</xs:complexType>
</xs:element>
<xs:element name="relative_asa" minOccurs="0" maxOccurs="1">
    <xs:complexType>
        <xs:sequence>
            <xs:element name="total" type="xs:double" />
            <xs:element name="backbone" type="xs:double" />
            <xs:element name="side_chain" type="xs:double" />
            <xs:element name="non_polar" type="xs:double" />
            <xs:element name="polar" type="xs:double" />
        </xs:sequence>
    </xs:complexType>
</xs:element>
<xs:element name="depth_index" minOccurs="0" maxOccurs="1">
    <xs:complexType>
        <xs:sequence>
            <xs:element name="average" type="xs:double"/>
            <xs:element name="average_deviation" type="xs:double"/>
            <xs:element name="average_side_chain" type="xs:double" />
            <xs:element name="average_side_chain_deviation" type="xs:double" />
            <xs:element name="maximum" type="xs:double" />
            <xs:element name="minimum" type="xs:double" />
        </xs:sequence>
    </xs:complexType>
</xs:element>
<xs:element name="protrusion_index" minOccurs="0" maxOccurs="1">
    <xs:complexType>
        <xs:sequence>
            <xs:element name="average" type="xs:double" />
            <xs:element name="average_deviation" type="xs:double" />
            <xs:element name="average_side_chain" type="xs:double" />
            <xs:element name="average_side_chain_deviation" type="xs:double" />
            <xs:element name="maximum" type="xs:double" />
            <xs:element name="minimum" type="xs:double" />
        </xs:sequence>
    </xs:complexType>
</xs:element>

```

[illegible]

8.2 Interakcijski parovi

```
<?xml version="1.0" encoding="us-ascii" ?>
<xs:schema targetNamespace="http://www.fer.hr/lss/namespaces/PDB"
            elementFormDefault="qualified"
            xmlns="http://www.fer.hr/lss/namespaces/PDB"
            xmlns:mstns="http://www.fer.hr/lss/namespaces/PDB"
            xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:element name="peptideResidueContacts">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="pdb_id" type="xs:string" />
        <xs:element name="contacts">
          <xs:complexType>
            <xs:sequence>
              <xs:element name="contact" minOccurs="0" maxOccurs="unbounded">
                <xs:complexType>
                  <xs:sequence>
                    <xs:element name="interaction_criterion" type="xs:string" />

```

```

<xs:element name="chain_id_1" type="xs:string" />
<xs:element name="residue_id_1" type="xs:string" />
<xs:element name="residue_name_1" type="xs:string" />
<xs:element name="chain_id_2" type="xs:string" />
<xs:element name="residue_id_2" type="xs:string" />
<xs:element name="residue_name_2" type="xs:string" />
<xs:element name="contact_types" minOccurs="0">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="contact_type" type="xs:string" minOccurs="1" maxOccurs="4" />
    </xs:sequence>
  </xs:complexType>
</xs:element>
</xs:sequence>
</xs:complexType>
</xs:element>
</xs:sequence>
</xs:complexType>
</xs:element>
</xs:sequence>
</xs:complexType>
</xs:element>
</xs:schema>

```

8.3 Mjesta interakcija (vezujući residuumi)

```

<?xml version="1.0" encoding="us-ascii" ?>
<xs:schema targetNamespace="http://www.fer.hr/lss/namespaces/PDB"
  elementFormDefault="qualified"
  xmlns="http://www.fer.hr/lss/namespaces/PDB"
  xmlns:mstns="http://www.fer.hr/lss/namespaces/PDB"
  xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:element name="peptideBindingResidues">
    <xs:complexType>
      <xs:sequence>

```

```

<xs:element name="pdb_id" type="xs:string" />
<xs:element name="binding_residues">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="binding_residue" minOccurs="0" maxOccurs="unbounded">
        <xs:complexType>
          <xs:sequence>
            <xs:element name="interaction_criterion" type="xs:string" />
            <xs:element name="chain_id" type="xs:string" />
            <xs:element name="residue_id" type="xs:string" />
            <xs:element name="residue_name" type="xs:string" />
          </xs:sequence>
        </xs:complexType>
      </xs:element>
    </xs:sequence>
  </xs:complexType>
</xs:element>
</xs:sequence>
</xs:complexType>
</xs:element>
</xs:sequence>
</xs:complexType>
</xs:element>
</xs:schema>

```

8.4 Vezujuće stanje residuuma

```

<?xml version="1.0" encoding="us-ascii" ?>
<xs:schema targetNamespace="http://www.fer.hr/lss/namespaces/PDB"
  elementFormDefault="qualified"
  xmlns="http://www.fer.hr/lss/namespaces/PDB"
  xmlns:mstns="http://www.fer.hr/lss/namespaces/PDB"
  xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:element name="peptideResidueBindingStatus">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="pdb_id" type="xs:string" />
        <xs:element name="residues">

```

```
<xs:complexType>
  <xs:sequence>
    <xs:element name="residue" minOccurs="0" maxOccurs="unbounded">
      <xs:complexType>
        <xs:sequence>
          <xs:element name="interaction_criterion" type="xs:string" />
          <xs:element name="chain_id" type="xs:string" />
          <xs:element name="residue_id" type="xs:string" />
          <xs:element name="residue_name" type="xs:string" />
          <xs:element name="binding_status" type="xs:boolean" />
        </xs:sequence>
      </xs:complexType>
    </xs:element>
  </xs:sequence>
</xs:complexType>
</xs:element>
</xs:sequence>
</xs:complexType>
</xs:element>
</xs:schema>
```

9 Literatura

- [1] ALBERTS, JOHNSON, LEWIS, RAFF, ROBERTSON, WALTER, *Molecular Biology of the Cell*, Garland Science, New York, 2002.
- [2] D. L. NELSON, M. M. COX, *Lehninger Principles of Biochemistry 4/e*, Worth Publishers, 2005.
- [3] R. SEDGEWICK, *Algorithms in C*, Addison-Wesley, 1990.
- [4] A. S. Aytuna, Prediction of protein-protein interactions by combining structure and sequence conservation in protein interfaces, *Bioinformatics*, 12(2005) Vol. 21, pp 2850-2855.
- [5] B. LEE. F. M. RICHARDS, The Interpretation of Protein Structures: Estimation of Static Accessibility, *J. Mol. Biol.*, 55(1971), pp 379-400
- [6] S. JONES, J. M. THORNTON, Analysis of Protein-Protein Interaction Sites using Surface Patches, *J. Mol. Biol.*, 227(1997), pp 121-132.
- [7] Y. OFRAN, B. ROST, Predicted protein-protein interaction sites from local sequence information, *FEBS Letters*, 544(2003), pp 236-239.
- [8] A. PINTAR, O. CARUGO, S. PONGOR, CX, an algorithm that identifies protruding atoms in proteins, *Bioinformatics*, 7(2002) Vol. 18, pp 980-984.
- [9] A. PINTAR, O. CARUGO, S. PONGOR, DPX: for the analysis of the protein core, *Bioinformatics*, 2(2003) Vol.19, pp 313-314.