UNIVERZITET U BEOGRADU
MATEMATIČKI FAKULTET

Milena R. Stojić

RAZVOJ APLIKACIJE ZA ANALIZU PORAVNANJA PROTEINA PREDSTAVLJENIH STRUKTURNIM ALFABETOM PROTEINSKI BLOKOVI

master rad

Beograd, 2023.

Mentor:

dr Mirjana Maljković Ružičić, docent
Univerzitet u Beogradu, Matematički fakultet

Članovi komisije:

prof. dr Nenad Mitić, redovni profesor
Univerzitet u Beogradu, Matematički fakultet

dr Jovana Kovačević, docent
Univerzitet u Beogradu, Matematički fakultet

Datum odbrane: _____

Svima koji su verovali u mene

Naslov master rada: Razvoj aplikacije za analizu poravnanja proteina predstavljenih strukturnim alfabetom Proteinski blokovi

Rezime: Zahvaljujući razvijenim strukturnim alfabetima je moguće opisivanje kompleksne trodimenzionalne strukture glavnog lanca proteina jednodimenzionim nizom strukturnih prototipova. *Proteinski blokovi* su jedan od najkorišćenijih strukturnih alfabeta. Predstavljanje strukture proteina pomoću strukturnih alfabeta omogućava jednostavnije poređenje proteina od poređenja trodimenzionalnih struktura proteina. Upoređivanje proteina se može svesti na poravnanje sekvenci. Cilj ovog rada je bio razvoj desktop aplikacije koja omogućava poravnanje sekvenci proteinskih blokova bez potrebe za mrežnom konekcijom, sa većim izborom mogućih zadavanja ulaza i većim izborom mogućih vizuelizacija. Dodatno, u razvijenoj aplikaciji je omogućeno i poravnanje aminokiselinskih sekvenci.

Ključne reči: bioinformatika, poravnanje, strukturni alfabet, *Proteinski blokovi*, proteini

Sadržaj

Glava 1

Uvod

Svi proteini u živom svetu su sačinjeni od samo 20 aminokiselina. Ipak, proteini imaju najrazličitije uloge jer je uloga proteina određena njegovom prostornom strukturom [14]. Da bi mogle da se uoče sličnosti i razlike između proteina i njihovih funkcija, neophodno je da se na neki način njihove trodimenzionalne strukture međusobno upoređuju. Direktno upoređivanje trodimenzionalnih struktura je veoma kompleksno. Strukturnim alfabetima je omogućeno opisivanje trodimenzionalne strukture glavnog lanca proteina jednodimenzionim nizovima strukturnih prototipova. Zahvaljujući takvoj reprezentaciji strukture proteina poređenje trodimenzionalnih struktura glavnog lanca proteina se može svesti na poređenje sekvenci za koje su razvijene metode poravnanja. Jedan od najpoznatijih i najrasprostranjenijih strukturnih alfabeta je strukturni alfabet *Proteinski blokovi*. U njemu je opisano 16 mogućih konformacija 5 uzastopnih aminokiselina koje se zovu proteinskim blokovima [27]. U ovom radu je rađeno na implementaciji desktop aplikacije koja omogućava poravnanje dve ili više sekvenci proteinskih blokova. Omogućeno je više načina ulaza, tekstualni izveštaji i zadavanje željenih vizuelizacija. Pored poravnanja sekvenci proteinskih blokova omogućene su i funkcionalnosti poravnanja aminokiselinskih sekvenci. Aplikacija je razvijena u programskom jeziku *Python* u radnom okviru *Qt*.

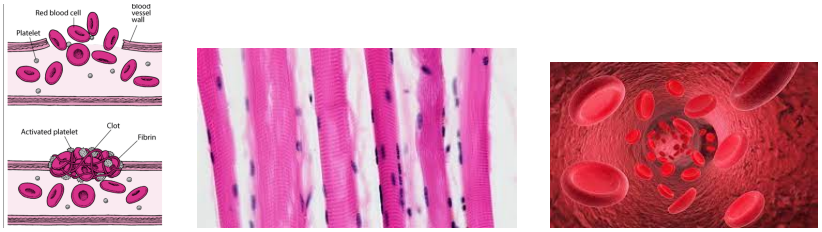
U drugoj glavi ovog rada su opisani proteini, struktura proteina i strukturni alfabet *Proteinski blokovi*. Trećom glavom je obuhvaćeno poravnanje i dat pregled metoda poravnanja sekvenci proteinskih blokova sa osvrtom na alate u okviru kojih su te metode implementirane. U četvrtoj glavi je dat opis aplikacije koja je razvijana u okviru master rada i opis svih vizuelizacija koje su omogućene u njoj. Peta glava sadrži zaključak i osvrt na ceo rad i dalja potencijalna unapređenja.

Glava 2

Proteini

2.1 Uloga proteina

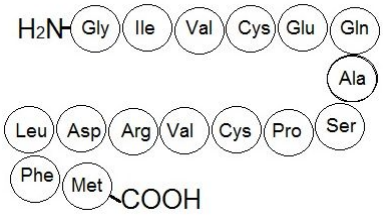
Proteini su biološka jedinjenja koja su neophodna za životno funkcionisanje. To su najvažnija jedinjenja koja izgrađuju žive organizme. Pored gradivne uloge, proteini imaju i mnoge druge važne uloge kao što su prenos kiseonika, zgrušavanje krvi, razmena materije ćelije sa spoljašnjom sredinom i mnoge druge. Neke od uloga proteina su prikazane na slici 2.1. Enzimi su važan deo metaboličkih procesa. Hemoglobin se nalazi u crvenim krvnim zrncima i zadužen je za prenos kiseonika. Fibrinogen je zaštitni protein koji je odgovoran za zgrušavanje krvi i time sprečavanje gubitka krvi. Transportni proteini u sastavu ćelijske membrane omogućavaju razmenu materija ćelija sa spoljašnjom sredinom [14].



Slika 2.1: Uloge proteina: Fibrinogen za zgrušavanje krvi, titin u izgradnji mišića, a hemoglobin u crvenim krvnim zrncima. Slike su preuzete iz [1, 3, 6].

Amino acid	Abbreviations	
Alanine	Ala	A
Arginine	Arg	R
Asparagine	Asn	N
Aspartic acid	Asp	D
Cysteine	Cys	C
Glutamine	Gln	Q
Glutamic acid	Glu	E
Glycine	Gly	G
Histidine	His	H
Isoleucine	Ile	I
Leucine	Leu	L
Lysine	Lys	K
Methionine	Met	M
Phenylalanine	Phe	F
Proline	Pro	P
Serine	Ser	S
Threonine	Thr	T
Tryptophan	Trp	W
Tyrosine	Tyr	Y
Valine	Val	V

Slika 2.2: Tabela 20 aminokiselina koje obrazuju proteine živih organizama. Slika je preuzeta iz [27].



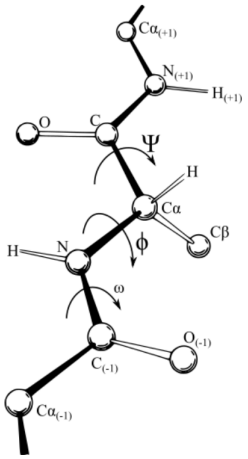
Slika 2.3: Primarna struktura proteina. Slika je preuzeta iz [2].

2.2 Struktura proteina

Proteini su polimeri sačinjeni od aminokiselina [14]. Aminokiseline su međusobno povezane peptidnom vezom između ugljenikovog atoma jedne aminokiseline i atoma azota druge aminokiseline [27]. Postoji 20 različitih aminokiselina koje obrazuju proteine živih organizama [14]. Tabela aminokiselina koje čine živi svet je prikazana na slici 2.2.

Ulančane aminokiseline povezane peptidnom vezom obrazuju polipeptidni lanac. Linearni raspored aminokiselina u polipeptidnom lancu čini njegovu primarnu strukturu (slika 2.3).

Organizacija delova polipeptidnog lanca u prostoru bez vezivanja bočnih nizova predstavlja sekundarnu strukturu proteina. Sekundarna struktura polipeptidnog lanca je uspostavljena vodoničnim vezama između karboksilne i amino grupe [14]. Sekundarna struktura proteina je definisana parom uglova ϕ i ψ za svaku amino-



Slika 2.4: Uglovi rotacije ϕ i ψ u molekulu aminokiseline. Slika je preuzeta iz [27].

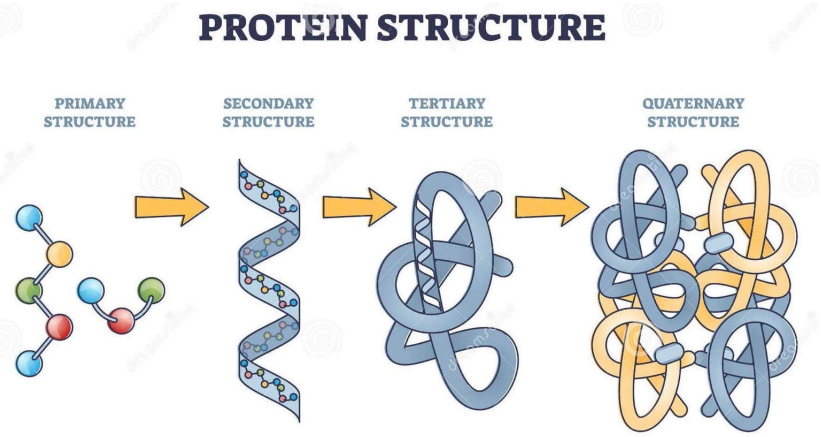
kiselinu. Ugao ϕ predstavlja ugao rotacije oko veze između ugljenikovog α atoma (atoma za koji su u aminokiselini vezane amino grupa i karboksilna grupa) i atoma azota. Drugi ugao ψ predstavlja ugao rotacije oko veze između ugljenikovog α atoma i ugljenikovog atoma u karboksilnoj grupi [27]. Prikaz ova dva ugla je dat na slici 2.4. Dva standardna načina opisivanja sekundarne strukture regiona je sa 3 ili 8 mogućih stanja. Način koji opisuje sekundardnu strukturu regiona sa 3 stanja obuhvata dva regularna stanja α-heliks i β-ravan i jedno nedefinisano stanje (eng. *coil state*) [27].

Tercijerna ili trodimenzionalna struktura se formira uspostavljanjem nekovalentnih veza u okviru osnovnog lanca i bočnih nizova aminokiselinskih ostataka [14]. Vizuelno, linearna struktura lanca se uvija i obrazuju se veze između udaljenih delova lanca. Više polipeptidnih lanaca tercijerne strukture čini njegovu kvaternarnu strukturu [14]. Na slici 2.5 se može videti odnos između različitih nivoa strukture proteina.

2.3 Strukturni alfabet Proteinski blokovi

Strukturni alfabeti

Neophodno je da opišemo trodimenzionalnu strukturu na pogodan način kako bismo mogli da vršimo poravnanja između trodimenzionalnih struktura, da primenjujemo metode istraživanja podataka i razne druge analize pomoću kojih možemo



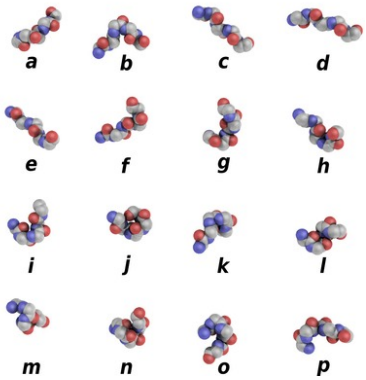
Slika 2.5: Odnos između različitih struktura proteina. Primarnu strukturu čini sam redosled aminokiselina u lancu. Sekundarna struktura vizuelno daje oblik lancu bez poprečnog savijanja i vezivanja udaljenih krajeva. U tercijernoj strukturi se može videti da se lanac savija poprečno i da se međusobno stvaraju veze između njegovih udaljenih delova. Više lanaca tercijerne strukture čini kvaternarnu strukturu proteina. Slika je preuzeta iz [4].

više saznati o međusobno sličnim proteinima i njihovim ulogama.

Često korišćen pristup je predstavljanje strukture sekvencom stanja sekundarne strukture [13]. Za opisivanje bi se koristila reprezentacija od dva regularna stanja sekundarne strukture i jednog nedefinisanog (eng. *coil state*) stanja. Zbog nepostojanja jasnih pravila za određivanje stanja sekundarne strukture različiti algoritmi istoj sekvenci aminokiselina dodeljuju stanja sekundarne strukture koja mogu značajno da se razlikuju [27].

Bolji pristup je opisivanje trodimenzionalne strukture proteina nizom fragmenata od n uzastopnih aminokiselina. Fragmenti dužine n aminokiselina predstavljaju prototipove lokalne strukture proteina. Skup svih prototipova kojima predstavljamo trodimenzionalnu strukturu proteina je strukturni alfabet lokalne strukture proteina.

Razvijeno je više različitih strukturnih alfabeta. Neki od njih su *Building Blocks*, *Structural building blocks* i *I-sites* [27]. Dobijeni su različitim tehnikama istraživanja podataka. Strukturni alfabet *Proteinski blokovi* je jedan od najpoznatijih i najkorišćenijih strukturnih alfabeta [27]. Razvijeni su programi za konverziju iz PDB formata u format nizova proteinskih blokova.



Slika 2.6: Prototipovi lokalne strukture strukturnog alfabeta *Proteinski blokovi*. Slika je preuzeta iz [27].

Proteinski blokovi

Strukturni alfabet *Proteinski blokovi* se sastoji od 16 prototipova od 5 uzastopnih aminokiselina. Svaki blok je određen vrednostima diedralnih uglova ϕ i ψ koji su opisani u odeljku 2.2.

Proteinski blokovi su nastali kao rezultat klasterovanja. Pri klasterovanju su uzimane u obzir sekvencijalne zavisnosti između blokova [9]. Algoritmi koji su korišćeni su zasnovani na Kohonenovom algoritmu (SOM) i skrivenim Markovljevim modelima koji su zasnovani na Bajesovoj teoriji verovatnoće [13]. Svi proteinski blokovi su obeleženi malim slovima engleske abecede od a do p i služe za opis strukture glavnog lanca proteina. Na slici 2.6 se mogu videti proteinski blokovi ovog strukturnog alfabeta.

2.4 Reprezentacija podataka o proteinima u računaru

FASTA format

Standardni i veoma čest format čuvanja bioinformatičkih podataka su *FASTA* datoteke. U FASTA datotekama možemo čuvati podatke o jednom ili više polipeptidnih lanaca, ali i o drugim različitim biološkim sekvencama. Početak prikaza podataka o svakoj proteinskoj sekvenci u datoteci označavamo linijom zaglavlja koja počinje sa '>'. U nastavku linije je obično dat kratak opis ili detaljnije informacije o polipeptidnom lancu. U narednim linijama je data primarna struktura polipeptid-

```
>seq0
FQTWEEFSRAAEKLYLADPMKVRVVLKYRHVDGNLCIKVTDDLVCLVYRTDQAQDVKKIEKF
>seq1
KYRTWEEFTRAAEKLYQADPMKVRVVLKYRHCDGNLCIKVTDDVVCLLYRTDQAQDVKKIEKFHSQLMRLM
LKVTDNKECLKFKTDQAQEAKKMEKLNNIFFTLM
>seq2
EEYQTWEEFARAAEKLYLTDPMKVRVVLKYRHCDGNLCMKVTDDAVCLQYKTDQAQDVKKVEKLHGK
>seq3
MYQVWEEFSRAVEKLYLTDPMKVRVVLKYRHCDGNLCIKVTDNSVCLQYKTDQAQDVK
```

Slika 2.7: Primer jedne FASTA datoteke sa aminokiselinskom sekvencom. Slika je preuzeta iz [15].

nog lanca u kojoj su aminokiseline opisane sa jednim slovom [18]. Primer FASTA datoteke sa aminokiselinskom sekvencom se može videti na slici 2.7.

Pošto je FASTA uobičajen format čuvanja podataka [18], razvijeni su mnogi programi i biblioteke za parsiranje FASTA datoteka. U okviru biblioteke *Biotite* [29] koja je korišćena u implementaciji postoji potpaket *sequence.io* u okviru kog se nalaze funkcionalnosti za čitanje i pisanje sekvenci u FASTA datoteke.

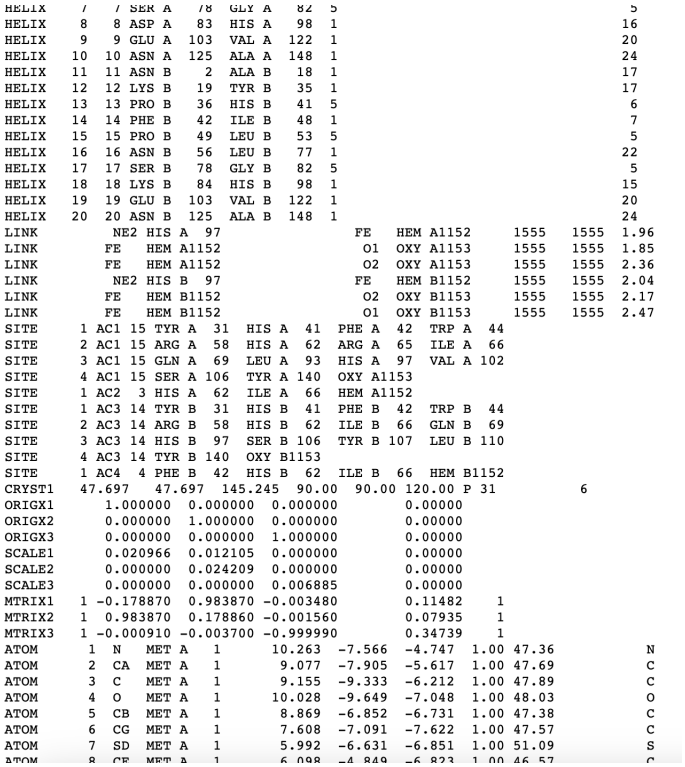
PDB format

Podaci o 3D strukturi proteina se čuvaju u PDB (*Protein Data Bank*) datotekama [8]. U prvom delu datoteke (zaglavlju) se nalaze razne metainformacije o proteinu (ime, izvor uzorka, eksperimentalni metod dobijanja uzorka, autor uzorka, ...). Takođe, može sadržati podatke o sekundarnoj strukturi i informacije o opisanim stanjima sekundarne strukture regiona [18]. U drugom delu do kraja datoteke se u linijama nalaze podaci o koordinatama atoma u prostoru sa informacijama kao što je hemijski element atoma, podaci o aminokiselini kojoj atom pripada i druge [31]. Na slici 2.8 se može videti primer strukture PDB datoteke za čuvanje trodimenzionalne strukture proteina.

PDB baza proteina

Protein Data Bank (PDB) je baza podataka trodimenzionalnih struktura proteina sačuvanih u datotekama PDB formata opisanih u prethodnom odeljku. PDB baza podataka poseduje i različite korisne funkcionalnosti vezanih za pretragu i analizu podataka [18].

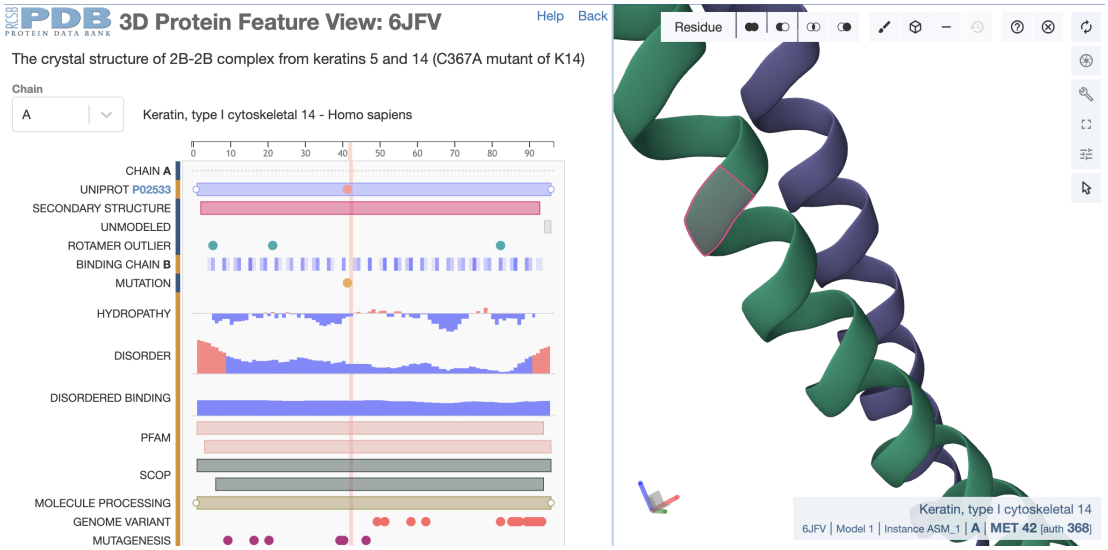
Svakoj instanci u ovoj bazi podataka je dodeljen jedinstveni ID od 4 karaktera koji je naziv odgovarajuće datoteke u PDB bazi podataka i preko koje joj se može direktno pristupati i iz raznih programa i biblioteka.



Slika 2.8: Primer strukture PDB datoteke. Slika je preuzeta iz [20].

Kada se otvori stranica instance u okviru *RCSB.org* sajta [8], pored linka ka PDB datoteci stukture, nalaze se osnovni podaci o sekvenci, umanjena slika trodimenzionalne strukture [18] i još razne vrste drugih podataka i vizuelizacije.

Posebno je interesantan interaktivni 3D pregledač u okviru koga je veran prikaz trodimenzionalne strukture. Na slici 2.9 se nalazi primer prikaza trodimenzionalne strukture keratina.



Slika 2.9: Prikaz trodimenzionalne strukture keratina u 3D interaktivnom pregledaču u okviru PDB baze podataka. Slika je preuzeta iz [20].

Glava 3

Poravnanje i algoritmi za poravnanje sekvenci proteinskih blokova

3.1 Uvod i motivacija

U uobičajenim algoritamskim zadacima se sličnost između dve sekvence poredila egzaktnim merama kao što je Hamingovo rastojanje. Takvim pristupom proverava se da li karakter jedne sekvence na tačno i - toj poziciji odgovara karakteru na tačno i - toj poziciji u drugoj sekvenci. Međutim, biološka jedinjenja kao što su nukleotidne sekvence su podložna različitim mutacijama, među kojima su insercije (umetanje) i delecije (brisanja) nukleotida. Zato se može desiti da simbol na i-toj poziciji jedne sekvence biološki odgovara simbolu na j-toj poziciji druge sekvence [30]. Jasno je da u bioinformatici moramo da imamo drugačiji pristup poređenja sličnosti sekvenci.

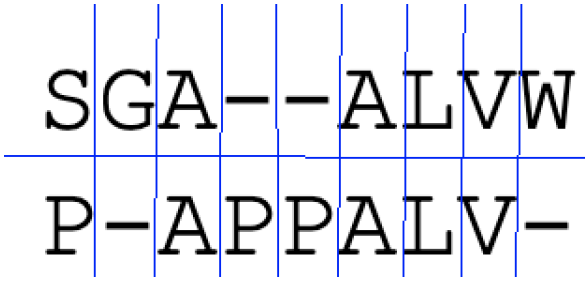
Na slici 3.1 se može videti primer dve sekvence. Po meri Hamingovog rastojanja između sekvenci nema nikakve sličnosti. Međutim, na slici 3.2 se može uočiti da su sekvence zapravo veoma slične.

CCTGACTTC
TGCCTGACT

Slika 3.1: Upoređivanje sekvenci

CCTGATCTTC
TGCCTAGACT

Slika 3.2: Upoređivanje sekvenci.



Slika 3.3: Primer jednog poravnanja dve aminokiselinske sekvence.

Poravnanje između dve sekvence se može posmatrati kao matrica u kojoj se nalaze dva reda koja odgovaraju, redom, prvoj i drugoj sekvenci [25]. U svakom redu se nalaze znakovi odgovarajuće sekvence sa eventualno ubačenim prazninama koje se označavaju specijalnim znakom '-'. Za znakove koji se nalaze u istoj koloni se može reći da su poravnati. Kolone u kojoj se nalaze obe iste vrednosti poravnatih znakova se nazivaju konzerviranim kolonama. Primer jednog poravnanja je dat na slici 3.3.

Pri poravnanju dve sekvence je cilj da bude poklopljen što veći broj istih i sličnih znakova (pojam sličnosti će biti opisan kasnije). U isto vreme je cilj i da u poravnanju bude što manji broj insercija (praznina u prvom redu) i delecija (praznina u drugom redu) [25]. Za insercije i delecije se koristi zajednički naziv indeli [25].

Mere kvaliteta poravnanja

Ukupan skor poravnanja je glavna mera kojom se kvantifikuje kvalitet poravnanja dve sekvence. Postoje različiti načini računanja skora. Ovde ćemo prikazati jedan od najčešće korišćenih [25]. Skor se računa kao zbir vrednosti skorova za svaku kolonu. Ako su poravnata dva potpuno ista znaka vrednost skora je pozitivna i najvećeg intenziteta, a ako su slična, vrednost skora je isto pozitivna. Intenzitet zavisi

od intenziteta sličnosti poravnatih znakova. Ukoliko su znakovi različiti, vrednost skora tog para je negativna. Negativno se boduje i svaki indel. Mogu se sa manjim intenzitetom bodovati indeli koji su već nadovezani u nizu od prvog indela u nizu. Ovakav način kažnjavanja indela je afino kažnjavanje indela [25].

Sada se sa pojmom skora problem poravnanja može preciznije definisati. Optimalno poravnanje je poravnanje kod koga je vrednost skora maksimalna.

Pored skora, u postojećim alatima i bibliotekama su obuhvaćene i dodatne vrednosti koje mogu pomoći pri kvantifikovanju kvaliteta poravnanja:

- broj insercija i broj delecija ili ukupan broj indela [18] (što su manji, to je poravnanje bolje);
- broj poklopljenih znakova (eng. *match*);
- mera identičnosti (eng. *identity*) koja predstavlja udeo poklopljenih znakova u poravnanju - količnik broja poklopljenih znakova i ukupne dužine poravnatih sekvenci [18].

Matrica supstitucije

Za svaki par znakova iz azbuke znakova u sekvencama koji se pojavi u istoj koloni u poravnanju je potrebno da bude definisan skor. Za tu svrhu se definiše matrica supstitucije (matrica skora). Ako se u azbuci nalazi n znakova (npr. u azbuci aminokiselina se nalazi $n = 20$ znakova), matrica supstitucije sadrži n^2 vrednosti. Znaku na i-toj poziciji u azbuci odgovaraju i- ti red i i- ta kolona. Vrednost u i-tom redu i j-toj koloni predstavlja skor između i-tog i j-tog znaka azbuke.

Matrica supstitucije je ključan deo algoritama za poravnanje sekvenci i zato im se posvećuje velika pažnja. Postojeće matrice skora koje su u širokoj upotrebi su konstruisane empirijski iz prethodno rađenih poravnanja. Na primer, jedna od najpoznatijih matrica skora za aminokiseline je BLOSUM62, konstruisana 1992. godine. Biološko objašnjenje skora između dve aminokiseline je u činjenici da što su dve aminokiseline sličnije, tj. imaju veći skor, one su više menjale jedna drugu tokom evolucije [18].

Lokalno poravnanje

Do sada su razmatrana globalna poravnanja, poravnanja na nivou čitavih sekvenci. Ipak, često nije u interesu pronaći optimalno poravnanje između dve cele

Slika 3.4: Primeri višestrukog poravnanja. Slika preuzeta iz [30].

sekvence, nego pronaći najsličnije zajedničke regione [18]. Ovaj problem je pronalaženje lokalnog optimalnog poravnanja. Za lokalno poravnanje je pogodnije koristiti afino kažnjavanje indela jer bi linearno kažnjavanje svakog indela bilo preoštro [25].

Višestruko poravnanje

Između vise bioloških sekvenci mogu postojati suptilne sličnosti koje ne mogu da budu otkrivene metodama dvostrukog poravnanja, zbog čega se koriste višestruka poravnanja koja mogu uočiti prikrivene sličnosti između sekvenci [25].

Analogno kao kod problema dvostrukog poravnanja, poravnanje n ($n > 2$) sekvenci se može posmatrati kao matrica od n redova koju čine simboli iz sekvenci sa eventualnim prazninama, označenih sa '-'. Na slici 3.4 se mogu videti primeri višestrukog poravnanja. Pored konzerviranih kolona (kolona koje čine iste vrednosti), značajne su i kolone kod kojih vrednost nije ista u svim redovima, ali je ista u dva ili više redova [30].

3.2 Opšte metode za poravnanje sekvenci

U nastavku će biti dat kratak pregled opštih metoda za poravnanje sekvenci. Pored izlaganja metoda biće prikazani i primeri onlajn dostupnih alata koji primenjuju te metode.

Globalno poravnanje dve sekvence

Za globalno poravnanje sekvenci je razvijen algoritam *Needlman-Wunsch* [28]. Ideja ovog algoritma je zasnovana na dinamičkom programiranju.

Za dve sekvence dužina, redom, m i n se konstruiše graf poravnanja koji sadrži $(m+1)\cdot(n+1)$ čvorova koji su raspoređeni u pravougaonu mrežu sa po $n+1$ čvorova

u svakom redu i $m+1$ čvorova u svakoj koloni. Svaki čvor po dužini odgovara znaku, a jednoj poziciji u prvoj sekvenci, izuzev prvog čvora koji odgovara praznoj sekvenci. Analogno je i za svaki čvor po širini. Svaki čvor je povezan usmerenom granom sa sledećim čvorom u koloni, redu i dijagonalno. Graf je težinski. Težina svake grane ka narednom čvoru u redu i koloni je jednak kazni za indele, a prelazak dijagonalnom granom zavisi od vrednosti u matrici supstitucije.

Prelazak u naredni čvor u istom redu ili koloni predstavlja dodavanje indela u jednoj sekvenci i dodavanje narednog karaktera u drugoj sekvenci, a prelazak na naredni čvor na dijagonali predstavlja dodavanje narednog karaktera u obe sekvence. Težina dijagonalne grane je jednaka vrednosti u matrici skora koja odgovara paru karaktera koji se dodaje u sekvence.

Pronalaženje optimalnog poravnanja se svodi na pronalaženje optimalne putanje od čvora u gornjem levom uglu (čvora u prvom redu i prvoj koloni) do čvora u donjem desnom uglu (čvora u poslednjem redu i poslednjoj koloni) u grafu poravnanja. Time se rešavanje problema dvostrukog poravnanja svodi na rešavanje problema turiste na Menhetnu [25]. Problem turiste na Menhetnu se rešava dinamičkim programiranjem tako što za svaki čvor u grafu pronalazi optimalna putanja do njega. Pamćenjem optimalnih putanja do prethodnih čvorova na kraju se dolazi do optimalne putanje do ciljnog čvora.

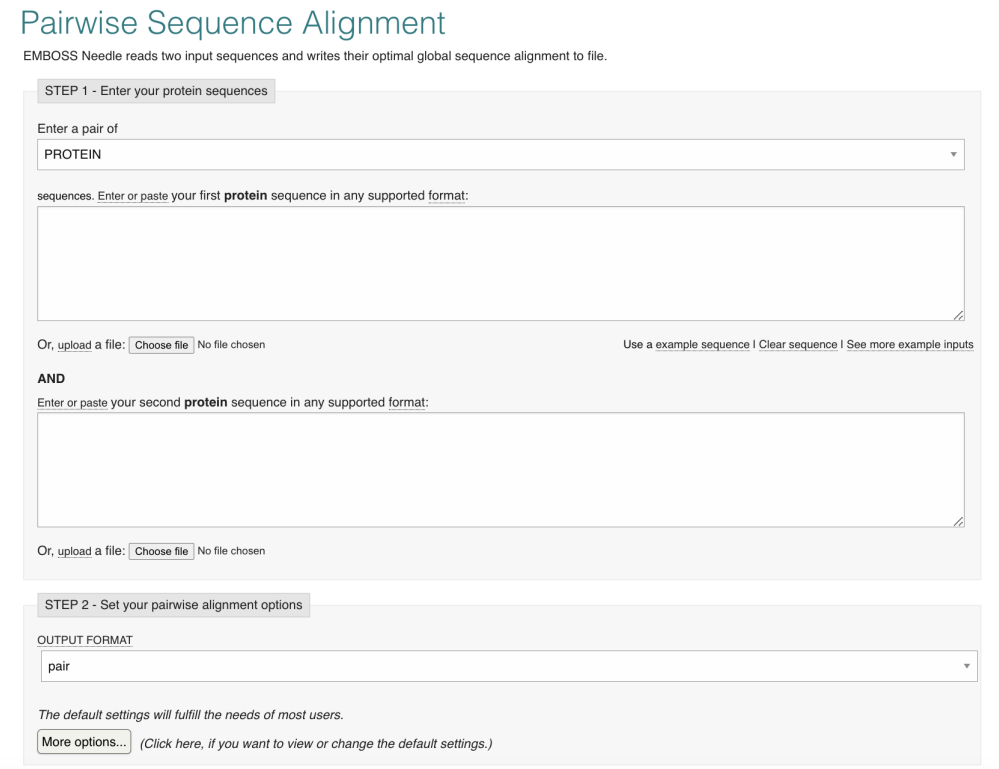
Pri primeni principa afinog kažnjavanja indela se modifikuje algoritam. Umesto jednog Menhetn grafa se izgrađuje Menhetn graf na tri nivoa [25].

Onlajn dostupan alat

Na veb sajtu Evropskog bioinformatičkog instituta je dostupan alat *Emboss Needle* [22] koji primenjuje algoritam *Needleman-Wunsch* za globalno poravnanje dve nuklteotidne ili aminokiselinske sekvence. Izgled korisničkog interfejsa se može videti na slici 3.5. Na početku se učitavaju sekvence koje se poravnavaju. Moguće je i dodatno podešavanje kojim se bira format izveštaja poravnanja. Primer standardnog tekstualnog izveštaja poravnanja se može videti na slici 3.6.

Lokalno poravnanje dve sekvence

Za lokalno poravnanje dve sekvence je razvijen algoritam *Smith-Waterman* [34]. Osnovni princip ovog algoritma je isti kao kod algoritma *Needlman-Wunsch*. Na isti način se konstruiše težinski graf poravnanja. Jedina je razlika u tome što se kod



Slika 3.5: Izgled interfejsa *Emboss Needle* onlajn alata. Slika je preuzeta iz [22].

algoritma *Smith-Waterman* ne kažnjava niz praznina od početka sekvenci do regiona obuhvaćenog lokalnim poravnanjem, ni niz praznina od kraja regiona obuhvaćenog lokalnim poravnanjem do kraja sekvenci.

Onlajn dostupan alat

Takođe je na veb sajtu Evropskog bioinformatičkog instituta dostupan alat *Emboss Water* [23] za lokalno poravnanje sekvenci algoritmom *Smith-Waterman*. Pošto su korisnički interfejs i izgled izveštaja potpuno identični kao kod alata *Emboss Needle*, njihov izgled neće biti prikazan.

Višestruko poravnanje

Moguće je pravolinijsko uopštavanje dvodimenzionalnog grafa poravnanja na k-dimenzioni graf poravnanja n sekvenci i svođenje problema višestrukog poravnanja na problem pronalaženja puta u k-dimenzionom grafu poravnanja. Za računanje skora bi se koristila k-dimenziona matrica supstistucije. Međutim, vremenska slo-

Results for job emboss_needle-I20230925-063932-0898-80975366-p1m

Alignment | Submission Details

View Alignment File

```
########################################
# Program: needle
# Rundate: Mon 25 Sep 2023 06:39:43
# Commandline: needle
#    -auto
#    -stdout
#    -asequence emboss_needle-I20230925-063932-0898-80975366-p1m.asequence
#    -bsequence emboss_needle-I20230925-063932-0898-80975366-p1m.bsequence
#    -datafile EBLOSUM62
#    -gapopen 10.0
#    -gapextend 0.5
#    -endopen 10.0
#    -endextend 0.5
#    -aformat3 pair
#    -sprotein1
#    -sprotein2
# Align_format: pair
# Report_file: stdout
########################################

#=======================================
#
# Aligned_sequences: 2
# 1: HBA_HUMAN
# 2: HBA_MOUSE
# Matrix: EBLOSUM62
# Gap_penalty: 10.0
# Extend_penalty: 0.5
#
# Length: 142
# Identity:     122/142 (85.9%)
# Similarity:   131/142 (92.3%)
# Gaps:           0/142 ( 0.0%)
# Score: 648.0
#
#
#=======================================

HBA_HUMAN           1 MVLSPADKTNVKAAWGKVGAHAGEYGAEALERMFLSFPTTKTYFPHFDLS     50
                      ||||..||:|:||||||:|.|..||||||||||.|||||||||||||:|
HBA_MOUSE           1 MVLSGEDKSNIKAAWGKIGGHGAEYGAEALERMFASFPTTKTYFPHFDVS     50

HBA_HUMAN          51 HGSAQVKGHGKKVADALTNAVAHVDDMPNALSALSDLHAHKLRVDPVNFK    100
                      ||||||||||||||||.:|..|:||:|.|||||||||||||||||||||
HBA_MOUSE          51 HGSAQVKGHGKKVADALASAAGHLDDLPGALSALSDLHAHKLRVDPVNFK    100

HBA_HUMAN         101 LLSHCLLVTLAAHLPAEFTPAVHASLDKFLASVSTVLTSKYR    142
                      ||||||||||:|.||:|||||||||||||||||||||||||
HBA_MOUSE         101 LLSHCLLVTLASHHPADFTPAVHASLDKFLASVSTVLTSKYR    142


#---------------------------------------
#---------------------------------------
```

Slika 3.6: Primer standardnog tekstualnog izveštaja poravnanja alata *Emboss Needle*. Slika je preuzeta iz [22].

ženost ovog algoritma je $O(n^k \cdot 2^k)$ [30]. Za velike vrednosti k je ova vremenska složenost neprihvatljiva, pa je potrebno da se primeni drugačiji pristup. Metod progresivnog poravnanja se zasniva na računanju sličnosti parova sekvenci dvostrukim poravnanjem.

Ideja metoda je u računanju sličnosti parova sekvenci dvostrukim poravnanjem. Rezultati poravnanja se čuvaju u matrici rastojanja između sekvenci. Na osnovu dobijene matrice rastojanja se uzimaju redom najsličnije sekvence i izgrađuje poravnanje. Takođe, izgrađuje se i filogenetsko stablo [16].

Onlajn dostupan alat

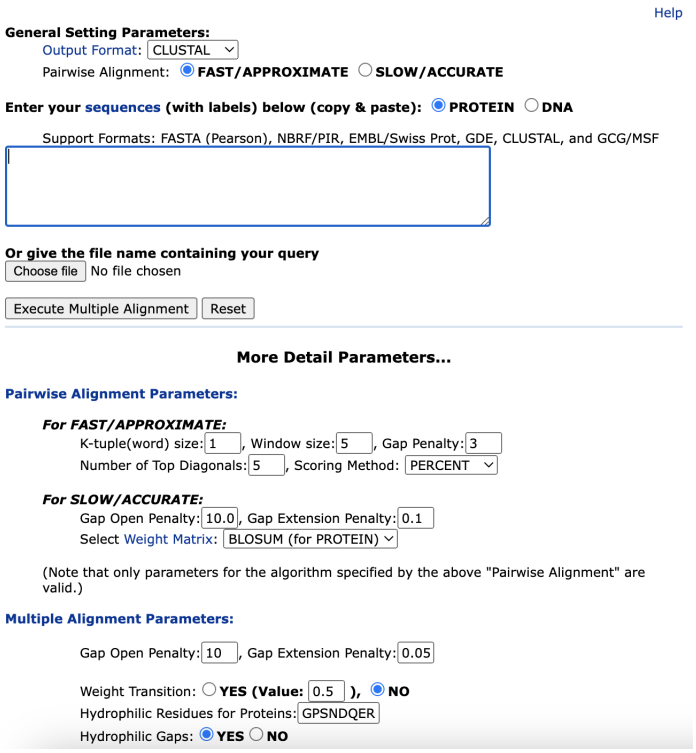
Alat *ClustalW* [7] primenjuje metod progresivnog poravnanja. Primer korisničkog interfejsa implementacije *ClustalW* sa *GenomeNet* portala [5] je prikazan na slici 3.7. Prikazanom implementacijom je moguće poravnanje aminokiselinskih ili nukleotidnih sekvenci. Na početku se zadaju sekvence. Kao i kod prethodno prikazanih alata, moguća su i dodatna podešavanja, kao što je zadavanje matrice supstitucije, podešavanja intenziteta kazni za indele i druga. Primer dobijenog izveštaja se može videti na slici 3.8.

3.3 Metode za poravnanje PB sekvenci

Zahvaljujući činjenici da je moguće trodimenzionalnu strukturu glavnog lanca proteina svesti na jednodimenzioni niz proteinskih blokova [13] poravnanje proteina se može svesti na poravnanje sekvenci. U ovom delu će biti opisana dva metoda za poravnanje sekvenci proteinskih blokova: *iPBA* [13] za poravnanje dve sekvence i *mulPBA* [33] za poravnanje više sekvenci. Pored opisa samih metoda biće prikazani i postojeći onlajn alati u okviru kojih su primenjene te metode.

iPBA

Prva verzija metoda je imala naziv *PBALIGN* [13]. Ona je primenjivala algoritam *Needlman-Wunsch* [28] za globalno poravnanje i algoritam *Smith-Waterman* [34] za lokalno poravnanje [11]. Matrica supsitucije za proteinske blokove je bila generisana na osnovu uparenih proteinskih blokova proteina u okviru strukturnih poravnanja iz PALI skupa podataka [32, 36]. Korišćeno je linearno kažnjavanje pra-



Slika 3.7: Korisnički interfejs *ClustalW* sa portala *GenomeNet*. Slika je preuzeta iz [7].

znina, sa vrednošću -3.0 za svaku prazninu u globalnom, odnosno vrednošću -5.0 za svaku prazninu u lokalnom poravnanju.

Daljim unapređivanjem metode *PBALIGN* je dobijena metoda *iPBA* (*improved the Protein Block Alignment methodology*) [13]. Rađeno je na unapređivanju dva aspekta metoda: profinjavanju matrice supstitucije i modifikaciji algoritma dinamičkog programiranja. Matrica supstitucije je generisana na osnovu novije verzije *PALI* baze podataka, izvršena je normalizacija vrednosti u matrici i vršena je selekcija poravnatih parova na osnovu kojih je generisana [13].

Izmenjena verzija dinamičkog programiranja (*anchor based dynamic programming algorithm*) polazi od skupa lokalnih poravnanja (eng. *anchors*) dobijenih SIM algoritmom [11, 13]. Segmenti između lokalnih poravnanja (eng. *linkers*) su poravnati klasičnim algoritmom *Needlman-Wunsch*. I kod lokalnih poravnanja i kod poravnanja segmenata je korišćen afin metod kažnjavanja indela [11].

Unapređena verzija metoda *PBALIGN* u opštem slučaju daje bolje rezultate od većine dostupnih alata za strukturna poravnanja [13].

CLUSTALW Result

[clustalw.aln][clustalw.dnd][readme]

Select tree menu | Exec

```
 CLUSTAL 2.1 Multiple Sequence Alignments


Sequence type explicitly set to Protein
Sequence format is Pearson
Sequence 1: sp|P69905|HBA_HUMAN    142 aa
Sequence 2: sp|P01942|HBA_MOUSE    142 aa
Sequence 3: sp|P13786|HBAZ_CAPHI   142 aa
Start of Pairwise alignments
Aligning...


Sequences (1:2) Aligned. Score: 85.9155
Sequences (1:3) Aligned. Score: 56.338
Sequences (2:3) Aligned. Score: 56.338
Guide tree file created:   [clustalw.dnd]

There are 2 groups
Start of Multiple Alignment

Aligning...
Group 1: Sequences:   2      Score:2179
Group 2: Sequences:   3      Score:1216
Alignment Score 1764

CLUSTAL-Alignment file created  [clustalw.aln]
```

clustalw.aln

```
CLUSTAL 2.1 multiple sequence alignment


sp|P69905|HBA_HUMAN       MVLSPADKTNVKAAWGKVGAHAGEYGAEALERMFLSFPTTKTYFPHFDLSHGSAQVKGHG
sp|P01942|HBA_MOUSE       MVLSGEDKSNIKAAWGKIGGHGAEYGAEALERMFASFPTTKTYFPHFDVSHGSAQVKGHG
sp|P13786|HBAZ_CAPHI      MSLTRTERTIILSLWSKISTQADVIGTETLERLFSCYPQAKTYFPHFDLHSGSAQLRAHG
                          * *:  ::: : : *.*:. :.   *:*:***:* .:* :********:  ****::.**

sp|P69905|HBA_HUMAN       KKVADALTNAVAHVDDMPNALSALSDLHAHKLRVDPVNFKLLSHCLLVTLAAHLPAEFTP
sp|P01942|HBA_MOUSE       KKVADALASAAGHLDDLPGALSALSDLHAHKLRVDPVNFKLLSHCLLVTLASHHPADFTP
sp|P13786|HBAZ_CAPHI      SKVVAAVGDAVKSIDNVTSALSKLSELHAYVLRVDPVNFKFLSHCLLVTLASHFPADFTA
                          .**. *: .*.  :*::..*** **:***: **********:********** :* **:**.

sp|P69905|HBA_HUMAN       AVHASLDKFLASVSTVLTSKYR
sp|P01942|HBA_MOUSE       AVHASLDKFLASVSTVLTSKYR
sp|P13786|HBAZ_CAPHI      DAHAAWDKFLSIVSGVLTEKYR
                           .**: ****: ** ***.***
```

clustalw.dnd

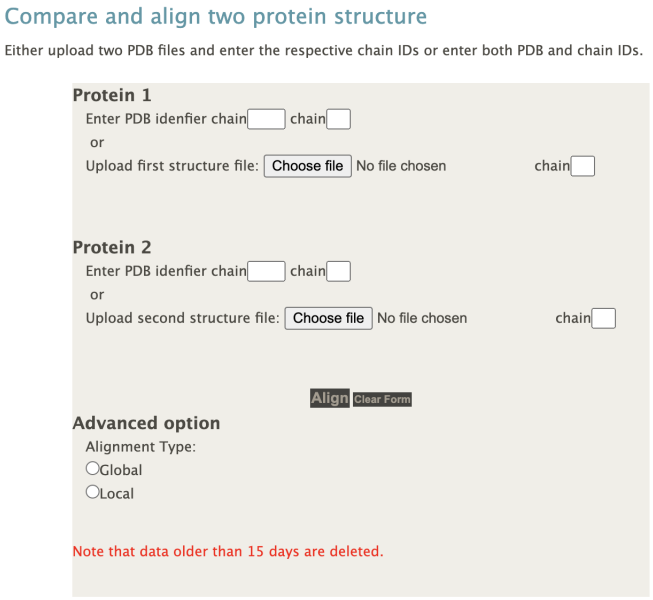
```
(
sp|P69905|HBA_HUMAN:0.07042,
sp|P01942|HBA_MOUSE:0.07042,
sp|P13786|HBAZ_CAPHI:0.36620);
```

Slika 3.8: Primer izveštaja dobijenog u prikazanoj implementaciji *ClustalW* alata. Slika je preuzeta iz [7].

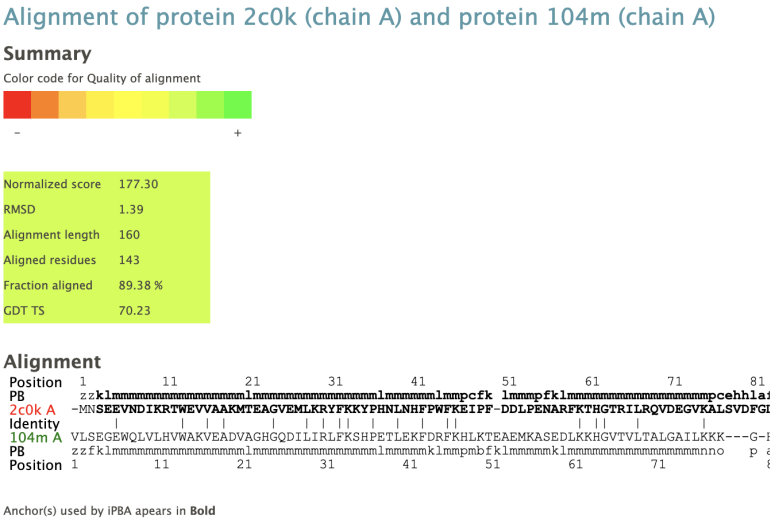
Onlajn dostupan alat

U okviru veb sajta *DSIMB* (*Dynamics of Structures and Interactions of Macromolecules in Biology*) [17] je dostupan alat *IPBA web server* [11]. Za svaki od dva proteina su moguća dva načina zadavanja ulaza: unos PDB ID-a ili prilaganje odgovarajuće strukturne datoteke. Za svaki od dva ulaza se zadaje lanac koji učestvuje u poravnanju. Ukoliko lanac nije izabran, podrazumevano je izabran A lanac. Na kraju se zadaje željeni tip poravnanja. Podrazumevano je izabrano globalno poravnanje. Na slici 3.9 se može videti forma za zadavanje ulaza za poravnanje dva proteina. Primer izveštaja poravnanja je dat na slici 3.10.

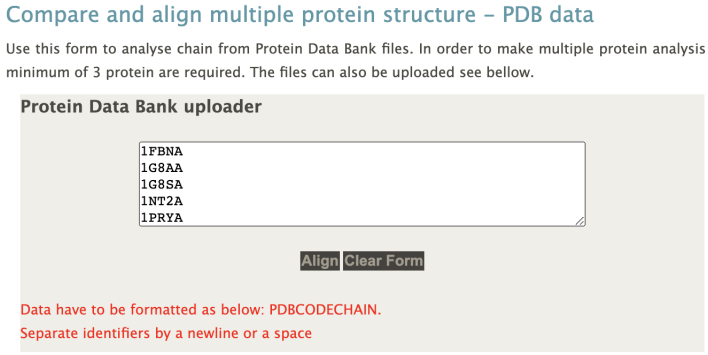
Pored poravnanja zadate dve sekvence, ovim alatom se mogu pretraživati slični lanci proteina unutar strukturne SCOP baze podataka [10]. Kao parametar se može zadati mera sličnosti, kao i tip poravnanja.



Slika 3.9: Zadavanje ulaza za poravnanje dva proteina u okviru onlajn alata *iPBA web server*. Slika je preuzeta iz [11].



Slika 3.10: Primer izveštaja dobijenog alatom *iPBA web server*. Slika je preuzeta iz [11].



Slika 3.11: Zadavanje ulaza u formatu <PDB ID><lanac> u okviru alata *mulPBA web server*. Slika je preuzeta iz [12].

mulPBA

Metod *mulPBA* [33] je zasnovan na sličnoj strategiji progresivnog poravnanja kao kod alata *ClustalW*. Za svaka dva proteina se vrši poravnanje metodom *iPBA* i time utvrđuje sličnost [12]. Na osnovu dobijenih sličnosti između proteina se izgrađuje stablo na osnovu koga se formira višestruko poravnanje.

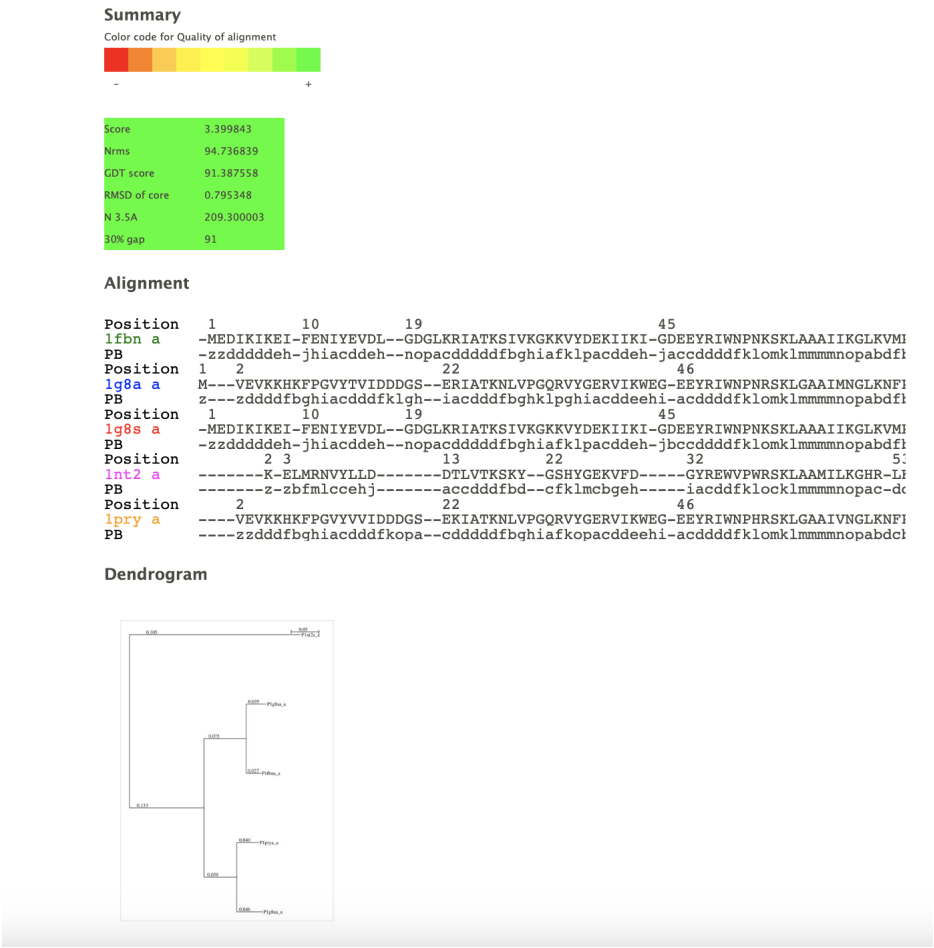
Gledajući mere kvaliteta poravnanja i efikasnost pri prepoznavanju strukturno sličnih proteina ovaj metod se pokazao prilično uspešnim poredeći ga sa velikim brojem dostupnih metoda [33].

Onlajn dostupan alat

Takođe se u okviru veb sajta *DSIMB* [17] može naći onlajn dostupan alat koji je zasnovan na metodu *mulPBA*. Ime alata je *mulPBA web server* [12].

U ovom alatu je nekoliko mogućih načina zadavanja ulaza. Moguće je zadavanje redom PDB ID-a za svaki protein sa oznakom željenog lanca (slika 3.11) i sukcesivno prilaganje PDB datoteka sa zadavanjem oznake lanca za svaku od njih. Takođe se može priložiti i kompresovana arhiva u okviru koje se nalaze PDB datoteke proteina. Pored poravnanja, može se i pristupiti izveštajima već izvršenih poravnanja zadavanjem ID-a procesa ukoliko nisu izvršena pre više od 7 dana.

Primer izveštaja se može videti na slici 3.12.



Slika 3.12: Primer izveštaja dobijenog u okviru alata *mulPBA web server*. Slika je preuzeta iz [12].

Glava 4

Razvijena aplikacija za poravnanje sekvenci

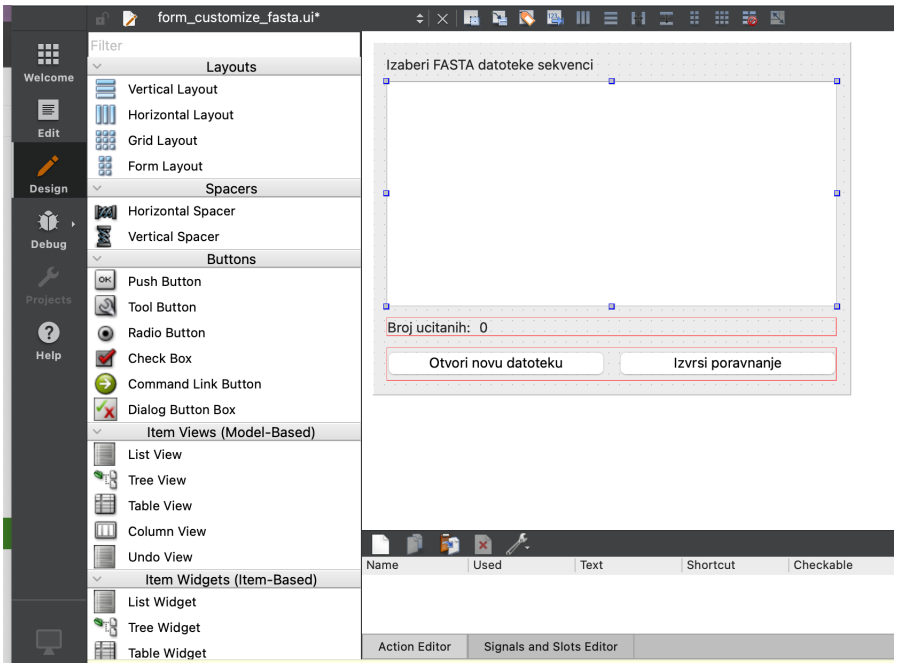
4.1 Implementacija

Alat *Lotos* [26] je implementiran u programskom jeziku *Python* u radnom okviru *Qt* [35]. Izgled celog grafičkog korisničkog interfejsa je inicijalno napravljen u grafičkom alatu *Qt Designer* u okviru razvojnog okruženja *Qt Creator* [35]. Grafički korisnički interfejs je formiran prevlačenjem i dodavanjem elemenata, kao što je prikazano na slici 4.1. Izgled prozora je sačuvan u okviru *.xml* datoteka sa ekstenzijom *.ui*. Datoteke u okviru kojih je sačuvan izgled prozora su prevedene u odgovarajuće *PySide* klase čije su kontrole povezivane sa odgovarajućim slotovima (metodama). Ti slotovi u zavisnosti od akcija korisnika pozivaju funkcije koje na osnovu ulaza učitavaju podatke, izvršavaju poravnanje i generišu izveštaje i vizuelizacije. Na slici 4.2 je prikazan uvodni prozor ove aplikacije.

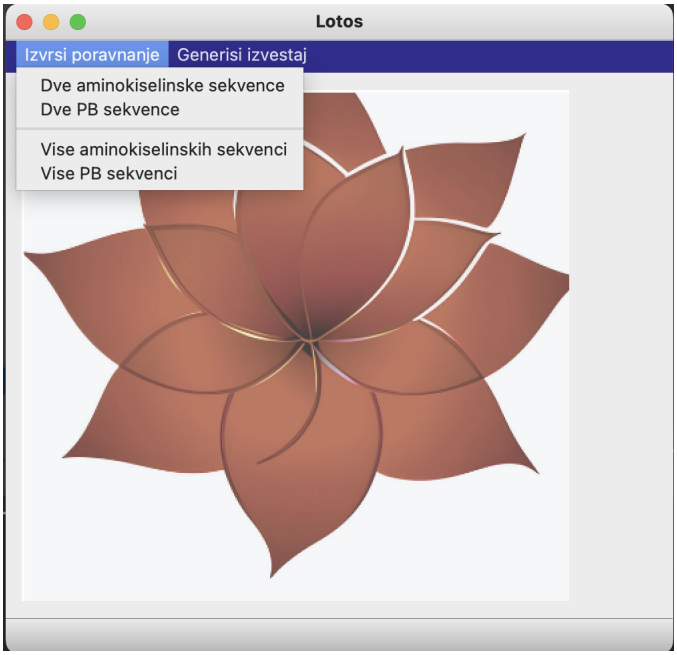
Biblioteka *Biotite*

Centralna biblioteka koja je korišćena pri implementaciji ovog alata je biblioteka *Biotite* [29] za rad sa bioinformatičkim podacima. Iz te biblioteke su korićeni moduli *sequences* za rad sa biološkim sekvencama, *align* za poravnanje sekvenci i *graphics* za generisanje vizuelizacija. Ostale biblioteke koje su korišćene pri izradi ovog alata za vizuelizaciju i podršku vizuelizaciji su *Matplotlib* i *Seaborn* [21, 37].

Metode modula *align* koje su korišćene za poravnanje sekvenci su *align_ optimal()* za poravnanje dve sekvence i *align_ multiple()* za višestruko poravnanje. Funkcija



Slika 4.1: Pravljenje izgleda grafičkog korisničkog interfejsa u okviru alata *Qt Designer*. Slika je kreirana u okviru okruženja *Qt Creator* [35].



Slika 4.2: Uvodni prozor aplikacije *Lotos*. Napomena: Slika unutar prozora je generisana uz pomoć veštačke inteligencije.

align_ optimal() koristi *Needleman-Wunsch* algoritam za globalno i *Smith-Waterman* algoritam za lokalno poravnanje, a funkcija *align_ multiple()* koristi metod progresivnog poravnanja. Osim za poravnanje nukleotidnih i aminokiselinskih sekvenci za koje postoje predefinisana podešavanja alfabeta, ove metode se mogu koristiti za poravnanje sekvenci simbola iz proizvoljne azbuke. U tom slučaju je potrebno definisati skup mogućih simbola kao i matricu supstitucije tih simbola. Na taj način je i urađeno sa strukturnim alfabetom *Proteinski blokovi*. Matrica supstitucije koja je korišćena pri poravnanju je preuzeta iz implementacije alata *PBXplore* [24].

Kratak pregled funkcionalnosti aplikacije

Učitavanje sekvenci

Moguća su tri načina zadavanja ulaza koji se mogu međusobno kombinovati. Jedan način je da korisnik unese PDB ID željenog proteina preko prozora. Ukoliko je odgovarajuća PDB datoteka već na mašini, alatom *PBXplore* [24] se vrši konverzija u format proteinskih blokova koji se čuva u FASTA datoteci. U suprotnom, najpre se iz PDB baze podataka preuzima PDB datoteka. Drugi način zadavanja ulaza je da korisnik izabere PDB datoteku sa računara na osnovu koje će biti generisana sekvenca proteinskih blokova. Treći način zadavanja ulaza je biranje FASTA datoteke sa sekvencama proteinskih blokova i njihovo direktno učitavanje u aplikaciju.

Za učitavanje sekvenci aminokiselina korisnik u prozoru pregledača datoteka bira odgovarajuće FASTA datoteke iz kojih se sekvence direktno učitavaju u program.

Poravnanje sekvenci

Obezbeđeno je poravnanje dve sekvence proteinskih blokova (lokalno i globalno), dve aminokiselinske sekvence (lokalno i globalno), više sekvenci proteinskih blokova i više aminokiselinskih sekvenci. Za svako od ovih poravnanja se prikazuje tekstualni izveštaj sa rezultatima u novom prozoru. Ukoliko korisnik želi, može da bira i način prikaza rezultata u grafičkom obliku.

Vizuelizacije

Za svaku vrstu poravnanja korisnik može da bira jedan ili više različitih načina prikaza rezultata u grafičkom obliku koje želi da mu se prikaže. Više o mogućim oblicima grafickog prikaza rezultata će biti reči u delu 4.3.

```
Izvestaj lokalnog poravnanja izmedju dve PB sekvence:
*****************
Duzina poravnanja: 203
Broj poklopljenih: 99
Broj insercija: 56
Broj delecija: 45
Mera identity: 0.4876847290640394
Score poravnanja: 30535

0                                                  49
klmmmmmm----------------------------------m---mmmm---
|||||||                                   |||
klmmmmmnopafbdcddfkopacehjhiafbfcfkbcfknojlmmmnopa
50                                                   99
--------mmmmmlmmmmmmmmmmmmmmmlmmmmm-mlmmpmbfklmmmp
             |||||||||||||||| |||||  ||| |    ||||
ekhpacfk-----lmmmmmmmmmmmmmmm-mmmmmk-lmm-m----mmmp
100                                                149
--mklmmmmmmmmmmmmmmmnopam-lmmmmmm-mmmmmmnop-afkl
   ||||||||||||      | |  | ||||||| ||||| |||   ||
cf-klmmmmmmmmmm------n-o--mmlmmmmmmlmmmmm-nopm--kl
150                                                199
```

Slika 4.3: Tekstualni izveštaj poravnanja između dve sekvence proteinskih blokova.

4.2 Tekstualni izveštaji

Posle poravnanja će se korisniku prikazati tekstualni izveštaj koji može da sačuva za ponovljena gledanja i dalje moguće analize. Izveštaji su pravljeni po ugledu na tekstualne izveštaje postojećih alata *ClustalW* [7] i *EMBOSS Needle* [22]. U svakom izveštaju su obuhvaćeni grafički prikaz poravnanja između sekvenci i osnovne mere kvaliteta.

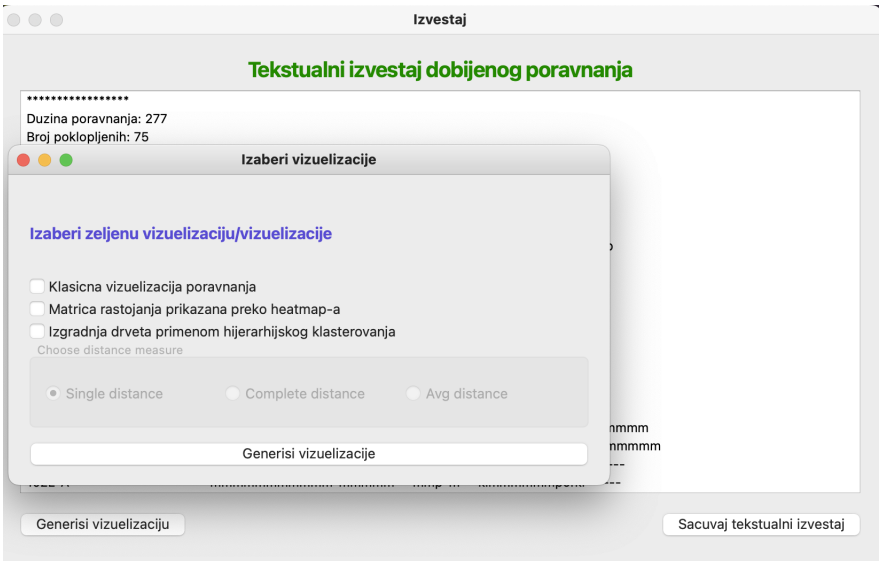
Poravnanje između dve sekvence

Na slici 4.3 možemo da vidimo primer tekstualnog izveštaja dobijenog lokalnim poravnanjem između dve sekvence. U zaglavlju su date ukupne dužine celog poravnanja i koliko je proteinskih blokova poklopljeno, broj insercija i broj delecija i mere kvaliteta identiteta i skor poravnanja.

U nastavku izveštaja je dat grafički prikaz ovog lokalnog poravnanja. Kao i kod postojećih poznatih alata, između poklopljenih proteinskih blokova je prikazan znak '|'.

Poravnanje između više sekvenci

U zaglavlju tekstualnog izveštaja su date ukupna dužina poravnanja, broj poklopljenih proteinskih blokova (kod svih poravnatih sekvenci na istoj poziciji) i ukupan skor poravnanja. Skor poravnanja više sekvenci se računa kao zbir skorova između svih parova sekvenci.



Slika 4.4: Izbor vizuelizacija za prikaz rezultata poravnanja više sekvenci proteinskih blokova.

U nastavku izveštaja je dat grafički prikaz poravnanja između više sekvenci. Ispod kolone u kojoj su svi poklopljeni proteinski blokovi je prikazan znak '*'.

4.3 Vizuelizacije u aplikaciji

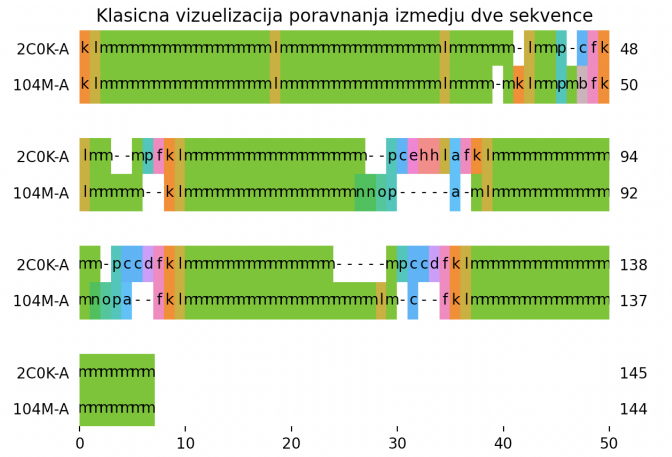
U ovoj sekciji će biti prikazani mogući izbori za grafički prikaz rezultata. Naravno, mogući izbor zavisi od toga da li je izvršeno poravnanje dve ili više sekvenci.

Kada se izvrši poravnanje i kada se dobije tekstualni izveštaj, korisnik može pritisnuti dugme *Generisi vizuelizaciju* nakon čega se otvara prozor *Izaberi vizuelizacije* (slika 4.4) na kom može da bira na koji način želi da se rezultat prikaže u grafičkom obliku. Posle pritiska na dugme *Generisi vizuelizacije* će biti prikazani svi izabrani oblici grafičkog prikaza rezultata.

Vizuelizacije rezultata poravnanja dve sekvence

Klasična vizuelizacija poravnanja

Na slici 4.5 je prikazana klasična vizelizacija dve sekvence proteinskih blokova (eng. *Plot alignment type based*). Isti prototipovi su označeni istom bojom. Insercije i delecije nemaju svoje boje i izgledaju kao - na slici. Na osnovu ove vizuelizacije



Slika 4.5: Primer klasične vizuelizacije poravnanja dve sekvence proteinskih blokova.

se jasno može videti da je ovo poravnanje sa visokim skorom i velikim udelom poklapanja proteinskih blokova.

Vizuelizacija poravnanja zasnovana na sličnosti

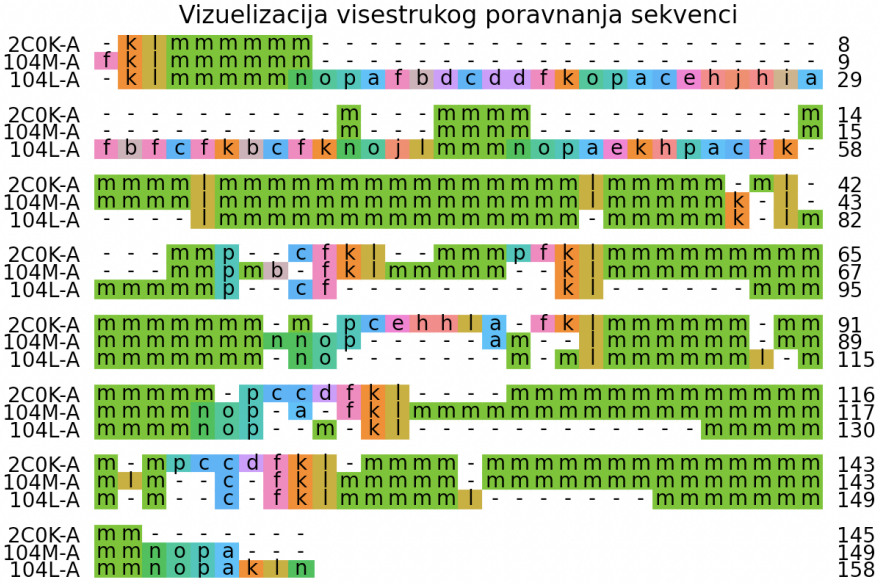
Na grafičkom prikazu poravnanja baziranom na sličnosti (eng. *Plot alignment similarity based*) uparenih proteinskih blokova, upareni proteinski blokovi se boje na osnovu njihove međusobne sličnosti. Intenzitet međusobne sličnosti je definisan vrednošću u matrici supstitucije koja odgovara paru uparenih proteinskih blokova. Što su upareni proteinski blokovi međusobno sličniji, nijansa boje je intenzivnija. Ako su poravnati proteinski blokovi isti, boja je najtamnija. Insercije i delecije, kao i protenski blokovi upareni sa njima ostaju neobojeni. Na slici 4.6 se može videti primer jedne takve vizuelizacije.

Vizuelizacije rezultata poravnanja više sekvenci

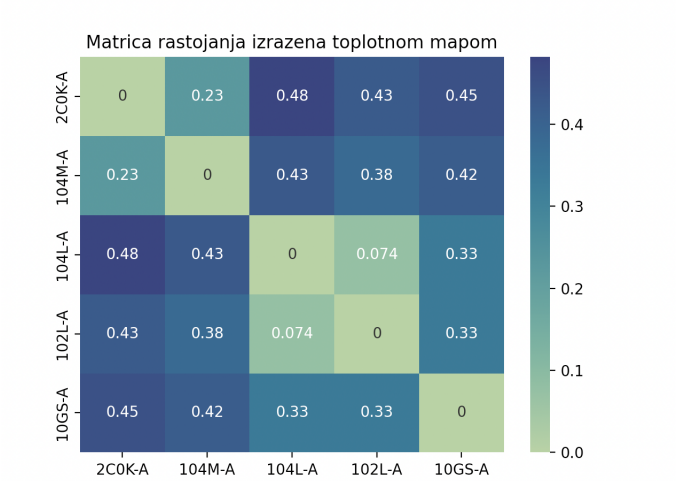
Klasična vizuelizacija poravnanja (*Plot alignment type based*)

Ovaj tip vizuelizacije je isti kao i kod klasične vizuelizacije poravnanja dve sekvence. Kao i kod poravnanja dve sekvence i ovde se sa slike očigledno može videti koliko je poravnanje kvalitetno. Na slici 4.7 je prikazan primer klasične vizuelizacije za više sekvenci.

Slika 4.6: Primer vizuelizacije poravnanja dve sekvence proteinskih blokova zasnovane na sličnosti uparenih proteinskih blokova.



Slika 4.7: Primer klasične vizuelizacije poravnanja između više sekvenci proteinskih blokova.



Slika 4.8: Toplotna mapa na osnovu matrice rastojanja dobijene višestrukim poravnanjem sekvenci proteinskih blokova.

Vizuelizacija rastojanja između sekvenci toplotnom mapom

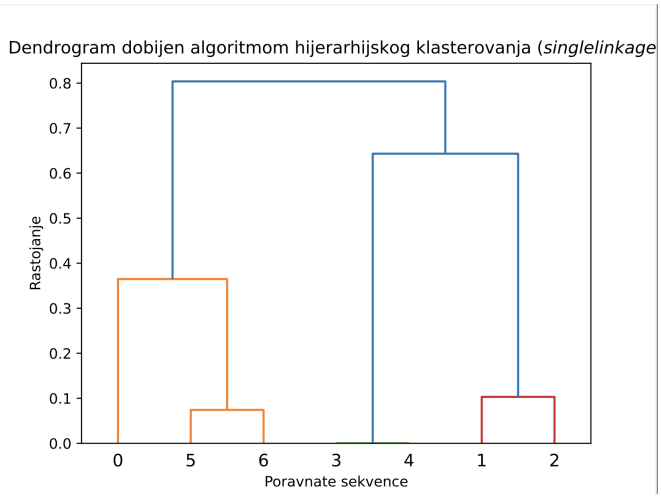
Matrica rastojanja koju dobijamo algoritmom poravnanja više sekvenci se može prikazati toplotnom mapom (eng. *heat map*) (slika 4.8). Takvim prikazom se stiče uvid u međusobnu sličnost poravnatih sekvenci. Što je boja na mapi tamnija, to su sekvence manje slične. Toplotne mape su generisane korišćenjem biblioteke *Seaborn* [37].

Drvo spajanja na osnovu rastojanja između sekvenci

Na osnovu dobijene matrice rastojanja se može izvršiti hijerarhijsko klasterovanje sekvenci. U ovoj aplikaciji je takvo generisanje stabla (dendrograma) omogućeno izvršavanjem hijerarhijskog sakupljajućeg klasterovanja funkcijom biblioteke *scikit-learn* [19]. Korisnik može da bira da li će tip veze biti *single* (podrazumevano), *complete* ili *average*. Primer ovako dobijenog drveta spajanja je prikazan na slici 4.9.

Vizuelizacije poravnanja aminokiselinskih sekvenci

Identične funkcionalnosti koje su omogućene za sekvence strukturnih prototipova se koriste i za vizuelizaciju poravnanja aminokiselinskih sekvenci. Pošto primarna struktura proteina nije fokus ovog rada, neće biti prikazane.



Slika 4.9: Dendrogram dobijen sakupljajućim klasterovanjem (tip veze *single-linkage*).

4.4 Uporedni rezultati sa postojećim alatom

U ovom poglavlju je izvršeno upoređivanje rezultata lokalnog poravnanja parova sekvenci dobijene alatom *Lotos* sa rezultatima postojećeg onlajn alata *iPBA web server* [11] koji je zasnovan na *iPBA* metodi poravnanja sekvenci proteinskih blokova [13]. Upoređene su vrednosti identičnosti poravnanja i dužine poravnanja.

Za svaki par sekvenci proteinskih blokova je pokrenut alat *Lotos* za kombinacije vrednosti parametara $\sigma = 1000$ i $\epsilon = 200$ i $\sigma = 1000$ i $\epsilon = 400$ (parametar σ označava afinu kaznu za otvaranje praznine, a parametar ϵ označava afinu kaznu za produžavanje praznine) i alat *iPBA web server*. Rezultati se mogu videti u tabeli 4.1. Za odgovarajuće vrednosti parametara je ponekad davao rezultate sa boljom vrednosti identičnosti od onlajn dostupnog alata. Može se primetiti da je za većinu parova sekvenci izračunavao veće poravnanje od alata *iPBA web server*. Prema dobijenim rezultatima na osnovu test podataka za lokalna poravnanja za iste dužine dobijenih poravnanja se dobija slična identičnost. Alat *iPBA web server* preferira kraće dužine sa većom identičnošću.

Parovi proteina	*Lotos*, $\sigma = 1000, \epsilon = 200$		*Lotos*, $\sigma = 1000, \epsilon = 400$		*iPBA* web server	
	identičnost	dužina	identičnost	dužina	identičnost	dužina
10GS-A 117E-A	0.5849	53	0.7105	38	1.0	5
183L-A 117E-A	0.6205	166	0.7223	90	0.68	25
1B80-A 2PTH-A	0.5616	146	0.5872	109	0.7576	33
2C0K-A 104M-A	0.8243	148	0.8243	148	0.8231	147
104M-A 104L-A	0.74	100	0.7396	96	0.7228	100

Tabela 4.1: Rezultati lokalnog poravnanja pomoću alata *Lotos* i alata *iPBA web server*.

Glava 5

Zaključak

U okviru ovog rada je rađeno na razvoju alata koji omogućava poravnanje trodimenzionalnih struktura glavnog lanca proteina predstavljenih kao sekvence proteinskih blokova. U aplikaciji se poravnanje dve sekvence vrši opštim pristupom dinamičkog programiranja, algoritmima *Needlman-Wunsch* i *Smith-Waterman*. Postoje metode koje su posebno pravljene za poravnanje sekvenci proteinskih blokova i onlajn alati koji primenjuju te metode. Prednosti razvijenog alata uključuju veći broj načina zadavanja ulaza i mogućnost direktnog učitavanja sekvenci proteinskih blokova. Omogućen je i veći broj mogućih vizuelizacija rezultata algoritma. Takođe, za upotrebu ove aplikacije nije neophodna mrežna konekcija.

Mogući dalji pravci unapređivanja razvijene aplikacije bi obuhvatili kombinovanje zadavanja aminokiselinskih sekvenci i PDB datoteka ili sekvenci proteinskih blokova, kao i dodatne vizuelizacije. Takođe, postoji prostor i za unapređivanje metode poravnanja.

Bibliografija

[1] AZoNano.com. https://www.azonano.com/.

[2] Biology LibreTexts. https://bio.libretexts.org/.

[3] Differencebetween.net. http://www.differencebetween.net/.

[4] Dreamstime stock photography. https://www.dreamstime.com/.

[5] GenomeNet portal network of database and computational services for genome research and related research areas in biomedical sciences. https://www.genome.jp/.

[6] MsdManuals.com. https://www.msdmanuals.com/.

[7] Multiple Sequence Alignment by CLUSTALW. https://www.genome.jp/tools-bin/clustalw, 2023.

[8] „RCSB PDB website". https://www.rcsb.org/, 2023.

[9] A. G. de Brevern, C. Etchebest, S. Hazout. Bayesian probabilistic approach for predicting backbone structures in terms of protein blocks. *Proteins: Structure, Function, and Bioinformatics*, 41(3):271–287, 2000.

[10] A. G. Murzin, S. E. Brenner, T. Hubbard, C. Chothia. Scop: a structural classification of proteins database for the investigation of sequences and structures. *Journal of molecular biology*, 247(4):536–540, 1995.

[11] A. P. Joseph, J. C. Gelly, A. G. de Brevern. iPBA Web Server. https://www.dsimb.inserm.fr/dsimb_tools/ipba/index.php/.

[12] A. P. Joseph, J. C. Gelly, A. G. de Brevern. mulPBA Web Server. https://www.dsimb.inserm.fr/dsimb_tools/mulpba/index.php/.

[13] A. P. Joseph, N. Srinivasan, A. G. de Brevern. Improvement of protein structure comparison using a structural alphabet. *Biochimie*, 93(9):1434–1445, 2011.

[14] B. Dobrković, J. Stošić, J. Popović. *Biologija 3M: za treći razred Matematičke gimnazije: (odabrana poglavlja)*. Krug Beograd, 2021.

[15] The UniProt Consortium. UniProt: the Universal Protein Knowledgebase in 2023. *Nucleic Acids Research*, 51(D1):D523–D531, 11 2022.

[16] D. F. Feng, R. F. Doolittle. Progressive sequence alignment as a prerequisitetto correct phylogenetic trees. *Journal of molecular evolution*, 25:351–360, 1987.

[17] DSIMB. Website of DSIMB Bioinformatics team. `https://www.dsimb.inserm.fr`.

[18] F. Pazos, M. Chagoyen. *Practical Protein Bioinformatics*. Springer, 01 2015.

[19] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer , R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[20] H. M. Berman, J. Westbrook, Z. Feng, G. Gilliland, T.N. Bhat, H. Weissig, I.N. Shindyalov, P. E. Bourne. The Protein Data Bank. *Nucleic Acids Research*, 28(1):235–242, 2000.

[21] J. D. Hunter. Matplotlib: A 2d graphics environment. *Computing in Science & Engineering*, 9(3):90–95, 2007.

[22] EMBL's European Bioinformatics Institute. Emboss Needle. `https://www.ebi.ac.uk/Tools/psa/emboss_needle/`.

[23] EMBL's European Bioinformatics Institute. Emboss Water. `https://www.ebi.ac.uk/Tools/psa/emboss_water/`.

[24] J. Barnoud, H. Santuz, P. Craveur , A. P. Joseph, V. Jallu, A. G. de Brevern, P. Poulain. PBXplore: a tool to analyze local protein structure and deformability with Protein Blocks. `https://https://github.com/pierrepo/PBxplore`, 2017.

[25] J. Kovačević. Materijali za predmet Uvod u bioinformatiku, Matematički fakultet, Univerzitet u Beogradu. `http://www.bioinformatika.matf.bg.ac.rs/`, 2023.

[26] M. Stojić. Lotos PB Alignment tool. `https://github.com/kate-97/Lotos_Alignment_Application-`, 2023.

[27] M. Maljković. *Prediction of alphabets of local protein structures using data mining methods*. Phd thesis, Faculty Of Mathematics, University of Belgrade, 2021.

[28] S. B. Needleman. Needleman-wunsch algorithm for sequence similarity searches. *J Mol Biol*, 48:443–453, 1970.

[29] P. Kunzmann, K. Hamacher. Biotite: a unifying open source computational biology framework in Python. *BMC Bioinformatics*, page 346, 10 2018.

[30] P. Pevzner, P. Compeau. *Bioinformatics Algorithms: An Active Learning Approach*. Active Learning Publishers, 2018.

[31] I. Ruczinski. Introduction to Protein Data Bank Format, Course of Protein Bioinformatics. `https://www.biostat.jhsph.edu/~iruczins/teaching/260.655/`.

[32] S. Balaji, S. Sujatha, S. Sai Chetan Kumar, N. Srinivasan. Pali—a database of phylogeny and alignment of homologous protein structures. *Nucleic Acids Research*, 29(1):61–65, 2001.

[33] S. Léonard, A. P. Joseph, N. Srinivasan, J. C. Gelly, A. G. de Brevern. Mulpba: An efficient multiple protein structure alignment method based on a structural alphabet. *Journal of biomolecular structure dynamics*, 32, 05 2013.

[34] T. F. Smith and M. S. Waterman. Identification of common molecular subsequences. *Journal of molecular biology*, 147, 1981.

[35] The Qt Company. Qt Framework. `https://www.qt.io/`.

[36] V. S. Gowri, S. B. Pandit, P. S. Karthik, N. Srinivasan, S. Balaji. Integration of related sequences with protein three-dimensional structural families in an updated version of pali database. *Nucleic acids research*, 31(1):486–488, 2003.

[37] M. L. Waskom. Seaborn: statistical data visualization. *Journal of Open Source Software*, 6(60):3021, 2021.

Biografija autora

Milena Stojić (*Beograd, 11. avgust 1997.*) je 2016. završila Matematičku gimnaziju u Beogradu i upisala osnovne studije na smeru *Informatika* na Matematičkom fakultetu Univerziteta u Beogradu. Osnovne studije je završila u septembru 2021. godine sa prosečnom ocenom $9,7$ i u oktobru iste godine upisala master studije. Zaposlena je kao saradnik u nastavi na Katedri za računarstvo i informatiku na Matematičkom fakultetu i takođe je angažovana u softverskoj kompaniji *Syrmia* u okviru koje radi u oblasti sistemskog softvera.