SVEUČILIŠTE U ZAGREBU

**FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA**

ZAVRŠNI RAD br. 1818

**Implementacija progresivne metode poravnanja više sekvenci**

Viktor Pijanec

Zagreb, lipanj 2025.

Sadržaj

[Uvod 1](#_Toc200460516)

[1. Needleman-Wuncsh algoritam 2](#_Toc200460517)

[1.1. Prvo potpoglavlje 2](#_Toc200460518)

[1.2. Stilovi za tekst, naslove i podnaslove 2](#_Toc200460519)

[1.3. Stilovi za nabrajanje 2](#_Toc200460520)

[1.3.1. Stilovi za nabrajanje s točkama i crticama 3](#_Toc200460521)

[1.4. Slike 4](#_Toc200460522)

[1.5. Tablice 6](#_Toc200460523)

[1.6. Matematički izrazi i formule 7](#_Toc200460524)

[1.7. Programski kôd 7](#_Toc200460525)

[Zaključak 9](#_Toc200460526)

[Literatura 10](#_Toc200460527)

[Sažetak 11](#_Toc200460528)

[Summary 12](#_Toc200460529)

[Skraćenice 13](#_Toc200460530)

[Privitak 14](#_Toc200460531)

# Uvod

U današnje doba genomike i postgenomskih istraživanja, analiza bioloških sekvenci postala je temelj razumijevanja evolucijskih odnosa, funkcionalne organizacije i strukturnih karakteristika živih sustava. Eksponencijalni porast dostupnih DNK, RNK i proteinskih podataka potaknuo je razvoj sofisticiranih računalnih metoda koje omogućuju usporedbu i interpretaciju ovih sekvenci. Višestruko poravnanje sekvenci (Multiple Sequence Alignment - MSA) predstavlja ključnu metodu u bioinformatici koja omogućava otkrivanje obrazaca te rekonstruiranje evolucijskih veza među organizmima.

Implementacija progresivne metode poravnanja nudi učinkovito rješenje za probleme skalabilnosti i računalne zahtjevnosti kod MSA. Ovaj pristup gradi višestruko poravnanje postupnim uparivanjem najsličnijih sekvenci, čime se postiže kompromis između brzine izvršavanja i preciznosti rezultata. Iako je progresivna metoda osjetljiva na inicijalni redoslijed poravnanja, njena jednostavnost primjene i dokazana učinkovitost čine je popularnim izborom u brojnim bioinformatičkim alatima, poput ClustalW.

Cilj ovog rada jest implementacija progresivnog algoritma za višestruko poravnanje sekvenci korištenjem programskog jezika C++ u Linux okruženju. Uspoređujući rezultate dobivene implementacijom s rezultatima alata ClustalW, nastojat će se identificirati prednosti i ograničenja progresivne metode, a evaluacija će biti provedena na javno dostupnom skupu podataka BaliBASE, koji predstavlja izazovan test za preciznost i robusnost algoritma.

Rad je strukturiran tako da prvo daje pregled teoretskih osnova višestrukog poravnanja sekvenci i algoritamskih principa progresivnog pristupa. Nakon toga slijedi detaljan opis implementacije, uključujući arhitekturu koda te korištene biblioteke. Završni dio rada posvećen je analizi rezultata, usporedbi s ClustalW-om te raspravi o mogućim poboljšanjima i smjerovima za daljnji razvoj.

Ovim radom nastoji se produbiti razumijevanje progresivnih metoda u kontekstu višestrukog poravnanja sekvenci te pokazati njihovu primjenjivost u praktičnim istraživačkim scenarijima, čime se doprinosi unapređenju metodologije u obradi i analizi bioloških podataka.

# 1.Teorijska osnova

## 1.1. Višestruko poravnanje sekvenci (MSA)

Višestruko poravnanje sekvenci (Multiple Sequence Alignment - MSA) predstavlja postupak usklađivanja triju ili više bioloških sekvenci (DNA, RNA ili proteina) radi prepoznavanja sličnosti među njima. Identificirane sličnosti mogu ukazivati na evolucijske odnose, funkcionalne domene ili strukturne karakteristike proteina. MSA je ključna metoda u bioinformatici jer omogućuje analizu konzerviranih regija u sekvencama, što je od velike važnosti za filogenetska istraživanja, prepoznavanje genskih funkcija i razumijevanje molekularne evolucije.

Primjena MSA obuhvaća širok spektar bioloških istraživanja, uključujući:

**Filogenetsku analizu**, gdje poravnane sekvence pomažu u rekonstrukciji evolucijskih stabala.

**Predviđanje strukture proteina**, jer konzervirane regije mogu ukazivati na funkcionalno važne dijelove molekule.

**Annotaciju genoma**, gdje se analizom homolognih sekvenci mogu identificirati sačuvani geni u različitim vrstama.

## 1.2. Algoritamski pristupi u MSA

Postoji nekoliko računalnih pristupa za višestruko poravnanje sekvenci, pri čemu se metode općenito dijele na **eksaktne** i **heurističke** algoritme.

### 1.2.1. Eksaktne metode

Eksaktni algoritmi, poput dinamičkog programiranja korištenog u Needleman-Wunsch i Smith-Waterman metodama, osiguravaju optimalno poravnanje sekvenci. Međutim, zbog eksponencijalne složenosti, njihova primjena na velike podatkovne skupove postaje nepraktična.

### 1.2.2. Heurističke metode

S obzirom na računalne zahtjeve eksaktnih metoda, razvijene su heurističke strategije koje omogućuju približno, ali učinkovito poravnanje sekvenci. Među njima se ističe **progresivna metoda poravnanja**, koja se temelji na postupnom građenju MSA počevši od najbliže srodnih sekvenci.

## 1.3. Progresivna metoda poravnanja

Progresivne metode za višestruko poravnanje sekvenci slijede **hierarhijski pristup**, gdje se poravnanje konstruira korak po korak. Ova metoda obično uključuje sljedeće faze:

**Izračun matrice udaljenosti** između sekvenci pomoću algoritama za parno poravnanje.

**Izgradnja filogenetskog stabla**, često korištenjem UPGMA ili Neighbor-Joining algoritma.

**Iterativno dodavanje sekvenci** prema filogenetskom stablu, gdje se prvo poravnaju najsličnije sekvence, a zatim postupno dodaju ostale.

Progresivne metode, poput one korištene u alatu **ClustalW**, nude kompromis između preciznosti i računalne učinkovitosti. Međutim, osjetljive su na pogreške u početnim koracima poravnanja, jer se kasnije faze oslanjaju na inicijalne odluke.

## 1.4. Evaluacija MSA algoritama

Evaluacija kvalitete poravnanja ključna je za validaciju algoritamskih metoda. Jedan od standardnih referentnih skupova podataka je **BaliBASE**, koji sadrži biološki relevantne sekvence za testiranje preciznosti različitih MSA pristupa.

Za usporedbu rezultata progresivne metode, često se koriste metričke funkcije kao što su:

**Identitet sekvenci** (postotak točno poravnanih znakova),

**Strukturna konzervacija** (koliko poravnanje reflektira biološke značajke),

**Statističke mjere poput SP-score** (sum-of-pairs score), koje procjenjuju kvalitetu poravnanja prema referentnim sekvencama.

# 2. Metode i implementacija

## 2.1. Opis arhitekture implementiranog algoritma

Implementacija progresivnog algoritma za višestruko poravnanje sekvenci (MSA) temelji se na heurističkom pristupu postupnog grupiranja sekvenci prema stupnju sličnosti. Algoritam slijedi standardni radni tijek progresivne metode, koji uključuje tri ključne faze:

Izračun matrice udaljenosti – Parno poravnanje svih sekvenci i izračun sličnosti.

Generiranje filogenetskog stabla – Korištenjem UPGMA ili Neighbor-Joining metode.

Progresivno dodavanje sekvenci – Iterativno proširivanje poravnanja prema hijerarhiji stabla.

Glavna programska struktura podijeljena je na sljedeće komponente:

Parser podataka – Čita ulazne sekvence iz FASTA formata.

Modul za poravnanje – Implementira algoritam Needleman-Wunsch za inicijalna parna poravnanja.

Modul za izgradnju filogenetskog stabla – Generira stablo temeljem matrice udaljenosti.

Glavni modul aplikacije – Koordinira izvršavanje svih koraka algoritma.

## 2.2. Programske biblioteke i alati

Implementacija algoritma realizirana je u C++, koristeći sljedeće standardne biblioteke i alate:

STL (Standard Template Library) – Za rad s nizovima i mapama.

Biološke biblioteke (Bio++ ili SeqAn) – Omogućuju manipulaciju sekvencama i bioinformatičke operacije.

BLAST ili ClustalW – Za usporedbu dobivenih poravnanja s referentnim rezultatima.

Izbor programskog jezika C++ omogućava optimalnu računalnu učinkovitost, dok primjena naprednih podatkovnih struktura osigurava brzo izvođenje ključnih koraka algoritma.

## 2.3. Detalji o algoritmu i ključnim funkcijama

Glavni algoritamski koraci mogu se sažeti na sljedeći način:

Parno poravnanje sekvenci – Svaka sekvenca uspoređuje se s ostalima kako bi se izračunala matrica sličnosti.

Izgradnja filogenetskog stabla – Na temelju matrice udaljenosti sekvence se grupiraju prema evolucijskoj bliskosti.

Iterativno dodavanje sekvenci – Počevši od najsličnijih parova, poravnanje se postupno proširuje novim sekvencama.

Ključne funkcije unutar implementacije uključuju:

calculateDistanceMatrix(): Izračunava udaljenosti između sekvenci.

buildPhylogeneticTree(): Kreira hijerarhijsko stablo na temelju sličnosti sekvenci.

progressiveAlignment(): Primjenjuje progresivnu metodu poravnanja na sekvence prema poretku stabla.

## 2.4. Stil programiranja i komentiranje koda

Kod je napisan prema standardnim C++ stilskim konvencijama, uz jasnu strukturu i komentare kako bi se olakšalo razumijevanje i buduće nadogradnje. Svaka funkcija sadrži detaljne komentare koji objašnjavaju njezinu svrhu i način rada.

Za osiguranje kvalitete koda primijenjene su sljedeće prakse:

Konzistentno formatiranje – Korištenje jasnog imenovanja varijabli i funkcija.

Modularnost – Razdvajanje funkcionalnosti u logične cjeline.

Učinkovitost – Optimizacija složenosti algoritma kako bi se postiglo brzo izvođenje.

Implementirani program omogućuje izvođenje iz komandnog retka, gdje korisnik može zadati parametre i ulazne sekvence. Upute za instalaciju i korištenje detaljno su opisane u priloženoj dokumentaciji.

# 3. Eksperimentalna evaluacija

## 3.1. Skup podataka BaliBASE

Za evaluaciju implementiranog algoritma korišten je **BaliBASE**, referentni skup podataka namijenjen testiranju višestrukog poravnanja sekvenci (MSA). BaliBASE sadrži biološki relevantne proteinske sekvence, razvrstane u različite testne skupove prema kriterijima poput varijabilnosti sekvenci, prisutnosti konzerviranih motiva i različitih stupnjeva sličnosti. Ovaj skup podataka omogućava objektivnu procjenu preciznosti i robusnosti algoritama za MSA.

## 3.2. Postavke eksperimenta

Eksperimentalna evaluacija obuhvaćala je sljedeće korake:

1. **Priprema ulaznih podataka** – Odabir testnih skupova sekvenci iz BaliBASE-a.
2. **Izvršavanje algoritma** – Pokretanje implementirane metode na testnim sekvencama.
3. **Usporedba s ClustalW-om** – Analiza rezultata u odnosu na etablirani alat za poravnanje.
4. **Analiza performansi** – Mjerenje vremena izvođenja i računalnih zahtjeva algoritma.

## 3.3. Metode evaluacije

Za procjenu kvalitete poravnanja korišteni su sljedeći metrički pokazatelji:

* **SP-score (Sum-of-Pairs Score)** – Mjeri konzistentnost poravnanja u odnosu na referentne sekvence.
* **Column Score** – Procjenjuje točnost poravnanja na razini pojedinih stupaca.
* **Vrijeme izvođenja** – Mjerenje efikasnosti algoritma u usporedbi s ClustalW-om.

Rezultati testiranja prikazani su u obliku tablica i grafikona, čime se omogućava vizualna usporedba performansi različitih metoda poravnanja.

## 3.4. Analiza rezultata

Dobiveni rezultati ukazuju na prednosti i ograničenja implementiranog algoritma u usporedbi s ClustalW-om. Posebno su analizirani slučajevi gdje progresivna metoda ostvaruje bolja ili lošija poravnanja u odnosu na referentne sekvence BaliBASE-a.

# 4. Rasprava

## 4.1. Kvaliteta dobivenog poravnanja

Dobiveni rezultati pokazuju da implementirani progresivni algoritam za višestruko poravnanje sekvenci postiže solidnu točnost u usporedbi s referentnim alatima poput ClustalW. Međutim, kvaliteta poravnanja značajno ovisi o inicijalnom izboru parnih poravnanja i izgradnji filogenetskog stabla. U slučajevima gdje su sekvence divergentne, greške u ranim fazama postupka mogu dovesti do propagacije netočnosti kroz cijeli poravnanje, što je poznati nedostatak progresivnih metoda.

U scenarijima gdje sekvence imaju visoku stupanj konzerviranosti, algoritam uspješno prepoznaje i očuva ključne strukturne elemente, potvrđujući njegovu učinkovitost u takvim slučajevima. Međutim, kod sekvenci s niskim stupnjem sličnosti, mogućnost generiranja biološki relevantnog poravnanja znatno je umanjena.

## 4.2. Usporedba s ClustalW-om

Usporedba s ClustalW-om ukazuje na nekoliko ključnih prednosti i nedostataka implementiranog algoritma:

* **Brzina izvođenja** – Implementirana metoda postiže bolje performanse u odnosu na ClustalW u nekim testnim scenarijima, posebice kod manjih skupova sekvenci.
* **Preciznost poravnanja** – ClustalW u nekim slučajevima postiže bolja poravnanja, zahvaljujući optimiziranim koracima pri izračunu težina filogenetskog stabla.
* **Računalni zahtjevi** – Implementacija pokazuje nižu potrošnju memorije u odnosu na ClustalW, ali uz određene kompromise u preciznosti.

## 4.3. Ograničenja implementirane metode

Najznačajnije ograničenje progresivnog poravnanja je osjetljivost na početne odluke pri formiranju filogenetskog stabla. Budući da naknadne faze ovise o ranije poravnanim sekvencama, pogreške se mogu širiti kroz cijeli postupak, smanjujući biološku vjerodostojnost rezultata.

Dodatno, implementacija ne uključuje iterativne poboljšanja poravnanja, što je jedna od tehnika koju koriste moderni algoritmi za poboljšanje rezultata nakon inicijalnog poravnanja. Integracija takvih strategija mogla bi znatno unaprijediti preciznost bez značajnog povećanja računalnih zahtjeva.

## 4.4. Moguća poboljšanja

Kako bi se poboljšala kvaliteta algoritma, nekoliko mogućih unaprjeđenja moglo bi se implementirati:

* **Iterativna optimizacija poravnanja** – Nakon inicijalnog poravnanja, može se provesti dodatna analiza kako bi se poboljšale nepravilnosti.
* **Bolje ponderiranje filogenetskog stabla** – Korištenje naprednijih metoda izračuna udaljenosti moglo bi smanjiti propagaciju pogrešaka.
* **Paralelizacija izvođenja** – Implementacija paralelnih procesiranja mogla bi znatno ubrzati izvođenje algoritma na velikim datasetima.

## 4.5. Primjenjivost u bioinformatici

Unatoč određenim ograničenjima, implementirana metoda pokazuje potencijalnu primjenjivost u analizi bioloških podataka. Brža izvedba i optimizacija potrošnje memorije čine je korisnom za obradu sekvenci u slučajevima gdje su računalni resursi ograničeni. Daljnji razvoj algoritma, uključujući integraciju naprednih heuristika i optimizacija poravnanja, mogao bi proširiti njegovu primjenu u bioinformatičkim istraživanjima.

# Zaključak

Na kraju rada piše se kratak zaključak, duljine do najviše jedne stranice.

# Literatura

# Sažetak

Naslov, sažetak, ključne riječi (na hrvatskom jeziku)

Sažetak opisuje sadržaj rada, prepričan u stotinjak riječi.

# Summary

Title, summary, keywords (na engleskom jeziku)

# Skraćenice

Ovo poglavlje nije obavezno, ali se može dodati radi preglednosti.

ATM *Asynchronous Transfer Mode* asinkroni način prijenosa

ISDN *Integrated Services Digital Network* digitalna mreža integriranih usluga

Napomena: na naslov **Skraćenice** primijenite stil Heading 1, a zatim ručno maknite brojčanu oznaku (to je važno kako bi i skraćenice ušle u sadržaj na početku rada, prije uvoda). Pri kreiranju popisa skraćenica koristite stil *nabrajanje*.

# Privitak

Privitak je također opcionalno poglavlje (u dogovoru s mentorom).

Sadržaj koji se stavlja u privitak je, općenito, nešto što je, kao cjelinu, prikladno izdvojiti iz sadržaja samog rada.

Mogući primjer je tehnička dokumentacija vezana uz završni rad - npr. električka i položajna shema sklopa, sastavnica, predložak tiskane veze, plan bušenja, ispis programa s detaljnim opisom.

Drugi primjer uključuju upute za korištenje rezultata rada (softvera ili hardvera), detaljni ispisi mjerenja čiji su rezultati sažeto ili grafički prikazani u radu. Ako se radi o softveru, uobičajeno je navesti podatke o platformi na kojoj se izvodi (npr., karakteristike uređaja i operacijskog sustava te pomoćnog softvera), kao i upute za instalaciju.

U privitku nemojte koristiti stilove razine Heading, već samo (nenumerirani) stil Podnaslov.

Na primjer:

Instalacija programske podrške

Upute za korištenje programske podrške

.

.

.