SVEUČILIŠTE U ZAGREBU

**FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA**

**Bioinformatika 1**

**High-speed and high-ratio referential genome compression algorithm**

*Katarina Mišura, Marko Marfat*

Voditelj: *Mirjana Domazet-Lošo*

Zagreb, svibanj, 2022.

**Sadržaj**

[Uvod 1](#_Toc103122008)

[1. Koraci algoritma 1](#_Toc103122009)

[1.1. Pretprocesiranje 1](#_Toc103122010)

[1.2. Pohlepno podudaranje pomoću *hash* tablice 1](#_Toc103122011)

[1.3. Postprocesiranje 1](#_Toc103122012)

[1.4. Dekompresija 1](#_Toc103122013)

[2. Rezultati testiranja 1](#_Toc103122014)

[Zaključak 1](#_Toc103122015)

[Literatura 1](#_Toc103122016)

[Sažetak 2](#_Toc103122017)

# Uvod

U ovome će se projektu, iz predmeta Bioinformatika 1, implementirati algoritam za kompresiju genoma. Puni naziv algoritma je „High-speed and high-ratio referential genome compression“ ili „HiRGC“, a cilj mu je uvesti poboljšane performanse u odnosu na već dostupne algoritme za kompresiju genoma [1].

Motivacija za uvođenje ovakvog algoritma je sve veći broj sekvenciranih genoma koji zahtijevaju velike količine memorijskog prostora za pohranu. Neke od tradicionalnih metoda kompresije poput „gzip-a“ (<http://www.gzip.org/>) i „bzip2-a“ (<http://www.bzip.org>) nisu dovoljno učinkovite u kompresiji genomskih podataka. Razlog tomu je što ovakve metode ne iskorištavaju intrinzična svojstva DNA sekvenci poput male abecede, čestih ponavljanja i palindroma. Preferirani pristup je kompresija bez gubitaka temeljena na pohranjivanju razlike između ciljnog genoma i referentnog genoma. Upravo to je pristup na kojemu se zasniva ovaj algoritam.

# Koraci algoritma

Pri pokretanju program kao argumente prima imena referentnog i ciljnog FASTA dokumenta. Nakon provjere točnosti unesenih argumenata može se započeti daljni proces. Cijeli algoritam možemo podijeliti u četiri cjeline: pretprocesiranje, pohlepno podudaranje pomoću *hash* tasblice, posprocesiranje i dekompresija. Svaka cjelina je opisana u nastavku uz pripadajući primjer.

## **Pretprocesiranje**

Pretprocesiranje je proces pripreme sekvenci referentnog i ciljnog FASTA dokumenta za daljnje uspoređivanje i kompresiju. Cilj pretprcesiranja je izdvojiti znakove A,C,G i T iz sekvenci ciljnog i referentnog FASTA dokumenta. Prilikom uklanjanja preostalih znakova iz ciljne sekvence, potrebno je sačuvati poziciju intervala tih znakova i duljinu intervala. Pri obradi referentne sekvence, potrebno je ukloniti preostale znakove, ali nije potrebno sačuvati informacije o njihovoj poziciji. Očuvane pomoćne informacije o znakovima koji se ne nalaze u skupu znakova A,C,G,T biti će potrebne kasnije pri dekompresiji i ponovnom sastavljanju cjelokupne ciljne sekvence. Pretprocesiranje možemo podijeliti u tri dijela. Prvi dio je procesiranje ciljne sekvence, drugi dio je procesiranje referentne sekvence, a treći dio je spremanje pomoćnih infomracija o ciljnoj sekvenci u dokument.

Funkcija *target\_preprocess()* kao argument prima niz znakova *file\_name* koji predstavlja ime FASTA dokumenta iz kojeg učitavamo ciljnu sekvencu. Funkcija započinje čitanjem redaka iz dokumenta i izdvajanjem prvog redka u kojem se nalaze informacije o sekvenci u niz znakova *id\_tg*. Ostali redci se čitaju jedan po jedan i spremaju se u niz znakova *t\_seq\_L*, a duljina svakog retka se sprema u vektor brojeva *t\_seq\_len*. Nakon učitavanja svih redaka iz dokumenta u niz znakova zatvara se pristup dokumentu. Prvo se pretražuju sva mala slova u sekvenci, sprema se pozicija prvog pronađenog malog slova u intervalu i duljina intervala. Svi znakovi koji su slova se pretvore u velika slova i prepisuju u novi niz znakova *t\_seq\_L1* . Nastavlja se pretraga i spremanje pozicije prvog pronađenog znaka N i duljina intervala. Uklanjaju se znakovi N i preostali znakovi se prepisuju u novi niz znakova *t\_seq\_L2*. Posljednji prolaz kroz seqvencu sprema pozicije prvog pronađenog znaka koji nije A,C,G,T, sprema sve znakove u intervalu i duljinu tog intervala. Prepisivanjem znakova A,C,G,T u novi niz znakova *t\_seq\_L3* dobije se konačna sekvenca. Radi lakše usporedbe i uštede memorije, nad sekvencom se vrši enkodiranje znakova prema pravilu: znak A se zamijenjuje znakom 0, znak C znakom 1, znak G znakom 2 i znak T znakom 3. Konačna sekvenca koja se koristi za daljnju usporedbu nalazi se u nizu znakova *t\_final*.

Funkcija *reference\_preprocess()* kao argument prima niz znakova *file\_name* koji predstavlja ime FASTA dokuemnta iz kojeg učitavamo referentnu sekvencu. Izdvojimo prvi redak sa informacijama o referentnoj sekvenci, a ostale redke spremimo u niz znakova *r\_seq\_L*. Vrši se pretvorba znakova u velike znakove i prepisivanje u novi niz znakova *r\_seq\_L1*. Potrebno je ukloniti sve znakove koji nisu A,C,G ili T, stoga se samo ti znakovi prepisuju u novi niz znakova *r\_seq\_L3*. Konačno se vrši enkodiranje prema istom pravilu za enkodiranje ciljne sekvence. Konačna sekvencakoja se koristi u daljnjoj usporedbi je spremljena u niz znakova *r\_final*.

Funkcija *saveDataToFile()* kao argument prima pokazivač na izlazni tok datoteke *&myfile*, a služi za spremanje dodatnih informacija o ciljnoj sekvenci. U dokument prvo se upisuje dužina intervala malih znakova i njihova duljina. Zatim se u sljedeći red upisuje dužina intervala znakova N i njihova duljina. Par interval i njegova duljina međusobno su odvojeni znakom „-“, a parovi su odvojeni razmakom. Posljednje se upisuje dužina intervala znakova koji nisu A,C,G,T ili N i svi znakovi u intervalu. Par dužina intervala i pripadajući znakovi koji čine interval odvojeni su znakom „-“, a parovi su odvojeni razmakom.

## **Pohlepno podudaranje pomoću *hash* tablice**

## **Postprocesiranje**

## **Dekompresija**

# Rezultati testiranja

# Zaključak

# Literatura

[1] Liu, Y., Peng, H., Wong, L., Li, J. (2017). *High-speed and high-ratio referential genome compression*. Bioinformatics Vol. 33 Iss 21 pp. 3364-3372. Poveznica: <https://doi.org/10.1093/bioinformatics/btx412>; pristupljeno 11. svibnja 2022.

# Sažetak