

SVEUČILIŠTE U ZAGREBU  
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

DIPLOMSKI RAD br. 2350

# **Učinkovito sažimanje genoma korištenjem referentnog genoma**

Juraj Radanović

Zagreb, travanj 2021.

**DIPLOMSKI ZADATAK br. 2530**

Pristupnik: **Juraj Radanović (0036503660)**  
Studij: Računarstvo  
Profil: Računarska znanost  
Mentor: izv. prof. dr. sc. Mirjana Domazet-Lošo

Zadatak: **Učinkovito sažimanje genoma korištenjem referentnog genoma**

Opis zadatka:

U okviru ovoga diplomskog rada potrebno je proučiti problem sažimanja skupa genoma korištenjem referentnog genoma te postojeće metode koje se koriste za taj problem. U sklopu rada potrebno je proučiti i implementirati algoritam SCCG te vlastitu i originalnu implementaciju primijeniti na kompresiju i dekompresiju skupa genoma.

Rok za predaju rada: 28. lipnja 2021.

*Zahvaljujem se mentorici doc. dr. sc. Mirjani Domazet-Lošo na vremenu i pomoći  
u izradi rada*

# SADRŽAJ

<b>1. Uvod</b>	<b>1</b>
<b>2. Kompresija genoma</b>	<b>2</b>
2.1. SCCG algoritam . . . . .	3
2.1.1. Lokalno pretraživanje . . . . .	3
<b>3. Zaključak</b>	<b>5</b>
<b>Literatura</b>	<b>6</b>

# 1. Uvod

Određivanje DNK organizma, tj. sekvenciranje je postalo jeftinije i brže nego prije, zahvaljujući novim tehnologijama sekvenciranja. Jedna od svrha sekvenciranja bi bila detekcija abnormalnih DNK u organizmu koja mogu upućivati na bolest. Sekvenciranje je jedan proces, koji dovodi do pohranjivanja DNK sekvence, prijenosa i procesiranja iste. Svaki od ovih procesa zahtjeva specifične implementacije kako bi svaki proces bio što više optimiziran, memorijski i vremenski. U ovome radu se bavim pohranjivanjem genoma, konkretnije njihovom kompresijom. Što se više genoma sekvencira, to je više podataka za pohranu i treba ih moći pohraniti na efikasne načine u smislu memorije i smanjenja gubitaka podataka tijekom dekompresije. Zapis ljudskog genoma je u redovima  $10^9$  okteta i držanje takvih datoteka je nemoguće. Stoga je potrebno napraviti algoritam koji će efikasno pohraniti taj genom. Postoje općeniti algoritmi za kompresiju, kao npr. 7zip, ali oni su neučinkoviti za pohranu genoma iz razloga što su genomi srodnih vrsta veoma slični. Konkretno, ljudski DNK su 99.9% slični, što znači da postoje puno redundantnih podataka. U ovom radu ću implementirati algoritam koji koristi referentni genom uz ciljani genom koji se komprimira, kako bi se iskoristila neka inherentna svojstva genoma i time ostvario veći stupanj kompresije.

## 2. Kompresija genoma

Kompresiju uz pomoć referentnog genoma su prvi predložili Brandon et al (2009). Prije toga su se genomi sažimali ne koristeći druge podatke. Kako bi takvi algoritmi bili bolji, često su se koristila statistička svojstva i teorija informacije za bolju kompresiju. BioCompress2 (Grumbach, Tahi, 1994.) koristi LZ77 (Lempel, Ziv, 1977.) algoritam za kompresiju bez gubitaka podataka i statističko kodiranje za detekciju i kodiranje palindroma. DNACompress (Chen et al., 2002.) pronalazi približno slična ponavljanja u sekvenci i kodira ih koristeći LZ77. Ovo su neki od primjera kompresijskih algoritama za genome koji ne koriste referentni genom. Posljedično tome, oni ignoriraju visoku sličnost homolognih organizama što dovodi do toga da nisu efikasni za kompresiju velikih sekvenci na razini gigabajta.

Algoritmi koji koriste referentni genom iskorištavaju tu sličnost, na način da se podniz ciljne sekvence pozicionira na referentnu sekvencu koristeći sekvencijsko podudaranje (uparivanje dvaju podnizova). Time se dugački nizovi u ciljnom genomu mogu sažeto napisati pomoću samo dva broja koji odgovaraju početnoj poziciji uparenih podnizova u referentnom genomu i duljini tog podniza. Jedan od takvih algoritama je GRS (Wang, Zhang, 2011.) koji koristi Unix diff program za traženje sličnosti između dvije sekvence, te se onda na najdužem zajedničkom podnizu koristi Huffmanovo kodiranje. Algoritam radi dobro samo ako su ciljni i referentni genom (sekvence) vrlo slični. GReEn (Pinho et al., 2010.) koristi globalnu hash tablicu (slično kao i u algoritmu u radu) za podnizove sekvence fiksne duljine referentnog genoma. Tada se podnizovi sekvenca ciljnog genoma hashiraju i traže u globalnoj tablici. Ako ciljni podniz postoji u globalnoj tablici, tada se kopiraju samo pozicije i duljine iz referentnog genoma, inače se koristi neki od standardnih statističkih alata za kodiranje. RLZ (Kuruppu et al., 2010.) koristi poboljšana sufiksna polja za traženje najduljeg prefiksa ciljne sekvence u referentnoj, te LZ77 za kompresiju tog prefiksa. ERGC (Saha, Rajasekaran, 2015.) dijeli ciljne i referentne sekvence na podnizove fiksne duljine i stvara hash tablicu za sve referentne segmente. Tada se traži podudaranje segmenta podniza ciljnog i podniza referentnog genoma, engl. *local matching*. Podudarni nizovi

se spremaju u datoteku koja se sažima koristeći delta kodiranje.

Performanse, tj. stupanj kompresije ovih algoritma znatno ovisi o izboru referentnog genoma. Što su ciljani i referentni genom sličniji, to će kompresija biti bolja. Local matching povezuje segmente ciljnog sa segmentima referentnog genoma. Ako sekvence nisu poravnane dovoljno dobro, moguće je da ciljni podnizovi ne će naći svoj par u referentnom segmentu. Algoritam HiRGC (Liu et al., 2017.) je riješio ovaj problem koristeći globalnu hash tablicu za cijelu referentnu sekvencu. U ovome radu ću implementirati SCCG algoritam (Shi et al., 2018.) za kompresiju genoma bez gubitaka u jeziku C++. Algoritam koristi neka svojstva već prethodno navedenih algoritma.

## 2.1. SCCG algoritam

Ulaz u algoritam su ciljni genom odnosno sekvenca i referentni genom odnosno sekvenca. Implementirani algoritam sažima datoteke koje su u FASTA formatu. FASTA format je jednostavni format datoteke koji sadrži nukleotidne sekvence (baze A, C, G, T) ili aminokiseline zapisane jednim slovom za svaku bazu. Prvi redak svake FASTA datoteke sadrži opis sekvence, a dalje se nižu redovi baza gdje je svaki red tipično duljine 80 slova. Jednostavni format FASTA datoteke čini ju laganom za parsirati za korištenje u nekom programskom jeziku. Abeceda koja se koristi za zapis sekvence je  $\Sigma$ , a veličina te abecede je  $|\Sigma|$ . Generalno govoreći, algoritmi za kompresiju su efikasniji, što je  $|\Sigma|$  manja, stoga je prvi korak ovog algoritma pretvaranje svih malih slova u velika, čime se efektivno smanji veličina abecede. Ta mala slova su potrebna prilikom dekompresije, kako ne bi bilo gubitaka podataka, iz tog razloga se u privremenu datoteku spremaju pozicije i duljine svih podnizova malih slova, točnije pamti se početna i završna pozicija podniza u cijelom nizu. Algoritam koristi lokalnu hash tablicu za pretraživanje segmenata, kao i globalnu hash tablicu, ukoliko je lokalno pretraživanje loše.

### 2.1.1. Lokalno pretraživanje

Lokalno pretraživanje uključuje segmentiranje ciljnog genoma  $T$  i referentnog segmenta  $R$ , izgradnja lokalne hash tablice  $H_L$ , te traženje istih segmenata. Prvi korak je segmentacija obiju sekvenca  $R$  i  $T$  na segmente  $r_1, r_2, \dots, r_i$  i  $t_1, t_2, \dots, t_j$  gdje su svi segmenti jednake duljine  $L$ . Za zadnje segmente  $r_i$  i  $t_j$  vrijedi da su manji ili jednaki  $L$ . Slijedeći korak je izgradnja hash tablice  $H_L$  za sve referentne segmente  $r$ . Prilikom pretraživanja segmenta, lokalna hash tablica se izgradi za svaki segment, pri čemu se

svaki referentni segment dijeli na još manje podnizove zvane *k-merovima*, podnizove segmenta duljine  $k$ . Za svaki *k-mer* izračuna se njegov hash koristeći `std::hash<>()` standardne biblioteke te se *k-mer* i njegova početna pozicija u segmentu  $r$  spremaju pod tim hashom. Moguće je da postoji više istih *k-mera*, pri čemu se svi spremaju pod istim hashom jer imaju različite pozicije unutar segmenta. Važno je napomenuti kako se ovdje koristim strukturu *unordered\_map* koja odgovara hash mapi sa konstantnim vremenom pretraživanja, unosa i brisanja elementa. Ovime završava izgradnja lokalne hash mape  $H_L$ .



### **3. Zaključak**

Zaključak.

# LITERATURA

Delta encoding. URL [https://en.wikipedia.org/wiki/Delta\\_encoding](https://en.wikipedia.org/wiki/Delta_encoding). Pristup: 21.4.2021.

Fasta format. URL [https://en.wikipedia.org/wiki/FASTA\\_format](https://en.wikipedia.org/wiki/FASTA_format). Pristup: 24.4.2021.

Compression of genomic sequencing data. URL [https://en.wikipedia.org/wiki/Compression\\_of\\_Genomic\\_Sequencing\\_Data](https://en.wikipedia.org/wiki/Compression_of_Genomic_Sequencing_Data). Pristup: 20.4.2021.

M. C. Brandon, D. C. Wallace, i P. Baldi. Data structures and compression algorithms for genomic sequence data. 25(14):1731–1738, 2009.

Dmitir Pavlichin i Tsachy Weissman. The Desperate Quest for Genomic Compression Algorithms, 2018.

Wei Shi, Jianhua Chen, Mao Luo, i Min Chen. High efficiency referential genome compression algorithm. 2018.

## **Učinkovito sažimanje genoma korištenjem referentnog genoma**

### **Sažetak**

Sažetak na hrvatskom jeziku.

**Ključne riječi:** Ključne riječi, odvojene zarezima.

### **Title**

### **Abstract**

Abstract.

**Keywords:** Keywords.