SVEUČILIŠTE U ZAGREBU
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

SEMINAR

HRCM algoritam

Paulo Erak, Lara Grgurić
Voditelj: *Mirjana Domazet-Lošo*

Zagreb, svibanj 2023.

SADRŽAJ

1. Uvod

Jedan od bitnih problema u području bioinformatike je sažimanje i dekompresija genomskih podataka. Ovaj je problem aktualan jer je spremanje podataka skupo, a s razvojem tehnologija povećava se važnost i količina podataka. Jedna od metoda kompresije velikih genomskih podataka je metoda hibridne referentne kompresije koju smo odlučili implementirati. Cilj je projekta napraviti vlastitu implementaciju metode hibridne referentne kompresije po uzoru na rad OBLAK te testirati na sintetski stvorenim podacima i podacima korištenima u sklopu spomenutog rada.

2. Algoritam

Algoritam koji implementiramo opisan je u radu [REFERENCA] te ćemo dati kratki osvrt na njega. Metoda hibridne referentne kompresije podijeljena je na procese kompresije i dekompresije. Proces kompresije podijeljen je na tri dijela ovim redoslijedom izvođenja: vađenje, podudaranje i kodiranje podataka. Ulaz u proces kompresije su jedna datoteka referentnog genoma te jedna ili više datoteka genoma koje treba sažeti.

2.1. Vađenje podataka

Ulaz dobiven u ovaj dio kompresije su datoteke koje sadrže genome. Vađenje podataka sastoji se od nekoliko koraka, međutim koraci potrebni za vađenje podataka iz referentnog genoma su samo oni koji vraćaju ACGT sekvencu. Koraci redom su:

(i) Izvođenje prvog reda kao identifikatora sekvence

(ii) Zapis duljine pročitanog reda

(iii) Pretvorba malih u velika slova i bilježenje pozicija i duljina dijelova genoma zabilježenih malim slovima

(iv) Spremanje informacija o pozicijama i duljinama segmenata sačinjenih od N znakova te informacija o pozicijama i duljinama segmenata sačinjenih od specijalnih znakova

(v) Izbacivanje svih znakova iz sekvence osim ACGT znakova

2.2. Podudaranje podataka

Proces podudaranja podataka obavlja se kroz dva dijela: podudaranje ACGT sekvenci i podudaranje informacija o malim znakovima. Podudaranje ACGT sekvenci temelji se na podudaranju k-mera sekvence koja se sažima s k-merima referentne sekvence te se izvodi na dvije razine.

2.2.1. Podudaranje prve razine

Za podudaranje prve razine potrebni su referentna sekvenca i sekvenca koju želimo sažeti. Prvi korak izrada je hash tablice na temelju k-mera referentne sekvence. Pri tome koristimo dvije podatkovne strukture: H i L. U strukturu H upisujemo vrijednosti oblika: $H[vrijednost_i] = i$. U slučaju pojave k-mera iste hash vrijednosti, vrijednost zapisana u $H[vrijednost_i]$ zapisuje se na poziciju trenutačnoog indeksa u strukturu L. Znači da su nam za ostvarenje ovog podudaranja potrebne dvije formule: $L[i] = H[vrijednost_i]$ i $H[vrijednost_i] = i$. Na taj način stvaramo lanac indeksa pozicija k-mera identične hash vrijednosti. Drugi je korak pretraga sekvence koju želimo sažeti za najduljim podnizom koji se poklapa s referentnom sekvencom. To izvodimo izračunavanjem hash vrijednosti k-mera te pretraživanjem lanaca stvorenih strukturama H i L s ciljem pronalaska najduljeg niza uzastopnih vrijednosti indeksa tih k-mera u referentnoj sekvenci. Pri tom pretraživanju dok se pronalaze podudaranja, bilježi se duljina niza, a pri pronalasku nepodudaranja, bilježi se jedan ili više znakova nepodudaranja. Time dobivamo tuple vrijednost (pozicija, duljina) koji služi kao zamjena za pronađeni segment sekvence.

2.2.2. Podudaranje druge razine

Podudaranje druge razine kompleksnije je od podudaranja prve razine zbog više razloga. Umjesto korištenja referentne sekvence, koriste se isključivo rezultati podudaranja prve razine. Kako bi se to ostvarilo, potrebno je odabrati postotak svih sekvenci za sažimanje od kojih će svaka imati ulogu referentne sekvence za sve sekvence koje slijede iza nje. Postupak također započinje izgradnjom hash tablice, ali u ovom je koraku izračun hash vrijednosti kompleksniji jer u obzir uzima poziciju, duljinu i hash vrijednost znaka nepodudaranja. Podudaranje hash vrijednosti ne znači nužno da su ta dva tuplea identična, što znači da je potrebna usporedba svakog elementa kako bi se potvrdilo podudaranje. Za svaku sekvencu potrebno je usporediti podudaranja sa svim referentnim sekvencama te odabrati najdulje podudaranje. U slučaju podudaranja, zapisuje se tuple (id sekvence, pozicija i duljina), pri čemu vrijednost id sekvence označava indeks referentne sekvence. U slučaju nepodudaranja, tuple kojemu ne možemo pronaći podudaranje prepisujemo.

2.2.3. Podudaranje malih znakova

Tuplovi malih slova pretražuju se...

2.3. Kodiranje podataka

Ovim procesom kodiraju se sve informacije. Identifikator sekvence i duljine redova kodiraju se RLE (run-length encoding) metodom, dok se posebni znakovi kodiraju statičnim entropijskim kodiranjem. Vrijednosti pozicije kodiraju se prediktivnim inkrementalnim kodiranjem tako da se pozicija sljedećeg entiteta predviđa na temelju vrijednosti pozicije i duljine trenutačnog entiteta. Sve kodirane informacije kodiraju se PPMD koderom koristeći alat 7-zip(https://www.7-zip.org/).

2.4. Dekompresija

Dekompresija obavlja se obrnutim postupkom od kompresije. Komprimirana datoteka dekomprimira se alatom 7-zip te RLE metodom, statičnim entropijskim kodiranjem i prediktivnim inkrementalnim kodiranjem. Sljedeće što je potrebno primijeniti je dekompresija dobivena procesom podudaranja. Kako bi se to ostvarilo, prva zapisana sekvenca dekodira se iščitavanjem podnizova u referentnom genomu koristeći informacije o pozicijama, duljinama i znakovima nepodudaranja. Sve ostale sekvence moraju se najprije dekomprimirati koristeći zapis rezultata podudaranja prve razine dobivenih za sve prethodeće sekvence. To se izvodi koristeći zapisane podatke o identifikatoru sekvence, pozicije i duljine podudarajućeg segmenta. Rezultat te dekompresije je rezultat podudaranja prve razine, koji se dekomprimira već opisanim postupkom. Naposlijetku, informacije o malim slovima, N znakovima i posebnim znakovima koriste se za sastavljanje originalnog zapisa sekvence.

3. Implementacija

Za implementaciju opisanog algoritma koristili smo programski jezik C++ te razvojno okruženje Visual Studio Code.

3.1. Vađenje podataka

3.2. Podudaranje podataka

Za implementaciju podudaranja bilo je potrebno odabrati način na koji izračunati hash vrijednost ACGT niza za podudaranje prve razine te kombinacije pozicije, duljine i nepodudarajućih znakova za podudaranje druge razine. Metoda izračuna hash vrijednosti koju smo odabrali je polinomska rekurzivna hash funkcija. Koristili smo gotovi algoritam kao osnovu za implementaciju te funkcije (https://www.geeksforgeeks.org/string-hashing-using-polynomial-rolling-hash-function/) te smo joj dodali funkciju za kodiranje znakova A,C,G i T u brojeve 1,2,3 i 4. Za ostvarenje hash tablice koristili smo strukturu *unordered map* zbog konstantog vremena pretraživanja, dok smo za zapis vrijednosti pozicija i duljina segmenata, kao i nepodudarajućih znakova koristili vektore zbog mogućnosti mijenjanja veličine te manje prostorne složenosti od strukture *list*.

3.3. Kodiranje podataka

4. Zaključak

Zaključak.

5. Sažetak

Sažetak.