SVEUČILIŠTE U ZAGREBU
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

SEMINAR

HRCM algoritam

Karlo Molnar, Ema Moškatelo
Voditelj: *Mirjana Domazet-Lošo*

Zagreb, svibanj 2025.

SADRŽAJ

1. Uvod

Bioinformatika je interdisciplinarna znanost koja koristi računalne metode za analizu bioloških podataka. S obzirom na to da ljudski genom sadrži oko 3 milijarde nukleotida, kompresija genetičkih podataka postaje ključna za učinkovitu pohranu i analizu. HRCM (Hybrid Referential Compression Method) algoritam je razvijen za učinkovitu kompresiju sekvenci genoma koristeći referencijalnu kompresiju.

Osnovna ideja HRCM algoritma temelji se na činjenici[1] da se genomi različitih ljudi podudaraju u 99,9% sekvenci. Algoritam se sastoji od tri faze: ekstrakcije informacija iz sekvenci, uparivanja informacija i enkodiranja podataka. U prvoj fazi, iz sekvenci se izdvajaju temeljni (bazni) znakovi A, T, C i G, te informacije o malim slovima, znakovima N i specijalnim znakovima. Uparivanje prve razine provodi se između referencijalne i ciljne sekvence, gdje se pronalaze najduži podudarajući nizovi. Ako postoji više ciljnih sekvenci, provodi se i uparivanje druge razine koje dodatno smanjuje veličinu podataka pronalaženjem podudarnosti među ciljanim sekvencama.

Enkodiranje informacija obuhvaća delta enkodiranje pozicija podudarnosti, pohranjivanje duljina podudarnosti i enkodiranje nepodudarnosti. Informacije o malim slovima i specijalnim znakovima također se enkodiraju kako bi se omogućila točna rekonstrukcija izvornih sekvenci tijekom dekompresije. Proces dekompresije uključuje čitanje enkodiranih podataka, dekodiranje informacija te rekonstrukciju izvornih sekvenci.

Implementacijom HRCM algoritma pokazano je da je metoda učinkovita i nadmašuje druge poznate metode kompresije genoma u smanjenju veličine pohranjenih podataka bez gubitka informacija.

2. Seminarski rad

Hibridna referencijalna kompresijska metoda za ulaz uzima, kao što ime kaže, referencijalnu sekvencu i sekvencu za kompresiju (ciljna sekvenca). Sekvence su zapisane u datotekama formata FASTA. Prva linija sadrži znak < i identifikator sekvence. Svaku sekvencu predstavlja niz znakova A, T, C, G koji označuju redom dušične baze adenin, timin, citozin, guanin. Osim navedenih, mogu se pojaviti znakovi N i specijalni znakovi. Prisustvo znaka N nalaže da se ne može sa sigurnošću odrediti što se nalazi na pripadnoj poziciji, dok posebni znakovi označavaju aminokiseline, završetak translacije ili prekid nepoznate duljine.

Za primjer ćemo koristiti ciljnu sekvencu iz datoteke t_seq.fa sadržaja

```
>123
GGCTGXGCCggtttnTTXCNNNaaaTTTcc
```

i referentnu sekvencu iz ref_seq.fa

```
>567
AGCTGGGaaggNNNnTTTCCCAaaaTTTcc
```

2.1. Izvlačenje informacija iz sekvenci

Iz ulaznih sekvenci, osim znakova ACTG moramo izvući sljedeće informacije: identifikacijsku oznaku, male znakove, znakove N, posebne znakove i širinu linije. Pozicija nizova malih slova i znakova N računa se kao udaljenost po slovima A,T,C,G od posljednjeg slova N odnosno malog slova, a specijalnih znakova kao udaljenost po slovima A,T,C,G,N od zadnjeg takvog znaka. Relativne pozicije koristimo zbog uštede memorije. To postižemo na sljedeći način:

1. Zabilježi identifikator iz prve linije
2. Zabilježi broj znakova iz jedne linije kao duljinu linije
3. Promijeni mala slova u velika, zabilježi poziciju i duljinu malih znakova

4. Zabilježi poziciju i duljinu nizova znaka N

5. Ukloni sve znakove osim slova ACGT, tj. izoliraj temeljnu (baznu sekvencu) Rezultati za primjer su sljedeći:

   a) ciljna sekvenca

   bazna sekvenca: GGCTGGCCGGTTTTTCAAATTTCC
   mali znakovi: (9,6), (7,3), (3,2)
   znakovi N: (13,1), (3, 3)
   specijalni znakovi: (5,X), (11,X)

   b) referentna sekvenca

   bazna sekvenca: AGCTGGGAAGGTTTCCCAAAATTTCC
   mali znakovi: (7,4), (3,1), (7,3), (3,2)

   .

2.2. Uparivanje informacija iz sekvenci

Iz prethodnog koraka izvukli smo baznu sekvencu i informaciju o malim slovima. Uparivanje prve razine provodi se nad referencijalnom i ciljnom sekvencom. Ukoliko želimo komprimirati više sekvenci, izvršit ćemo i uparivanje druge razine između ciljnih sekvenci. Temeljni princip uparivanja jest da se pronađe najduži niz koji se podudara.

2.2.1. Uparivanje prve razine

Uparivanje započinjemo izradom tablice sažetaka za referentnu baznu sekvencu. Znakove A, T, C, G enkodiramo brojevima 0, 1, 2 i 3. Vrijednosti tablice sažetaka H inicijalno se postavljaju na -1. Kliznim prozorom veličine k iteriramo po sekvenci te računamo sažetak za pripadajući k-mer. Sažetak se računa kao vrijednost k-mera u sustavu s bazom 4.

U mapu H, koja predstavlja tablicu sažetaka, kao ključeve spremamo indeks posljednjeg k-mera nekog sažetka. Kako ne bismo izgubili informaciju o ponavljajućima k-merima, u listi L u trenutni indeks stavljamo vrijednost prethodnog k-mera iste vrijednosti sažetka.

Slijedi uparivanje najdužih podudarajućih nizova između referentne i ciljne sekvence. Računa se sažetak podniza iste veličine k. Ukoliko nije nađen među ključevima mape H, idemo na idući indeks jer u ovome očito nema podudarajućeg niza. U suprotnome, nastavljamo pretraživati jesu li idući znakovi isti te bilježimo duljinu niza. Koristimo listu L kako bismo ispitali sve indekse na kojima je izračunat isti sažetak te naposljetku izdavajmo onaj s najvećom duljinom. Rezultat uparivanja je uređena trojka sastavljena od početnog indeksa podudarnog niza u referentnoj sekvenci (ili -1 ako ne postoji), duljini podudarnog niza i nepodudarnog niza koji slijedi.

Rezultati na našem primjeru oblika (pozicija, duljina, nepodudarajući znakovi) za duljinu k-mera 4 su:

(-1,0,GGCT)
(-1,0,GGCC)
(9,5,TT)
(16,4,T)
(22,4,)

2.2.2. Uparivanje druge razine

Uparivanje druge razine primjenjuje se kada imamo više ciljnih sekvenci koje želimo komprimirati. U ovom koraku koristimo rezultate uparivanja prve razine kako bismo pronašli dodatne podudarnosti među ciljanim sekvencama što dodatno smanjuje količinu podataka za pohranu.

Koraci uparivanja druge razine su sljedeći:

1. Za svaku sekvencu stvorimo tablicu sažetaka koristeći podudarnosti pronađene u prvom koraku.

2. Iteriramo kroz svaku podudarnost iz prve razine te računamo vrijednost sažetaka za k-mere.

3. Koristimo tablice sažetaka za brzo pronalaženje podudarnosti među ciljanim sekvencama.

4. Pronađene podudarnosti pohranjujemo s informacijama o poziciji i duljini.

Razmotrimo primjer ciljne sekvence nakon uparivanja prve razine:

Sekvenca 1: (1,3,C), (5,3,G), (9,3,GT), (15,3,C), (19,4,A)
Sekvenca 2: (1,3,C), (5,3,G), (9,3,GG), (15,3,C), (19,4,A)

Primjenom uparivanja druge razine pronađeni su dodatni podudarajući segmenti:

(1,1,2)
(9,3,GG)
(1,4,2)

Uparivanje druge razine pomaže u smanjenju veličine komprimiranih podataka pronalaženjem dodatnih podudarnosti među ciljanim sekvencama koje dijele slične segmente.

2.2.3. Uparivanje informacija o malim znakovima

U ovom koraku, informacije o malim slovima (koja označavaju specifične regije u sekvencama) također se podudaraju između referentne i ciljnih sekvenci. Postupak je sličan uparivanju osnovnih sekvenci:

1. Izgradimo tablicu sažetaka za regije s malim slovima u referentnoj sekvenci.
2. Iteriramo kroz ciljne sekvence i tražimo podudarnosti s regijama malih slova u referentnoj sekvenci.
3. Pohranjujemo pozicije i duljine podudarnosti, kao i informacije o nepodudarnostima.

Za ciljnu sekvencu s malim znakovima:

GGCTGXGCCggtttnTTXCNNNaaaTTTcc

i referentnu sekvencu s malim znakovima:

AGCTGGgaaggTTTCCNNNaaaTTTcc

Informacije o malim znakovima nakon uparivanja su:

Ciljna sekvenca: (9,6), (7,3), (3,3)
Referentna sekvenca: (6,5), (8,3), (3,3)

Podudarnosti omogućuju kompresiju specifičnih regija s malim slovima.

Uparivanje informacija o malim znakovima omogućava preciznije komprimiranje sekvenci zadržavanjem informacija o specifičnim regijama.

2.3. Enkodiranje informacija o sekvencama

Nakon što su podudarnosti pronađene, informacije moramo enkodirati kako bismo ih mogli pohraniti u komprimirani format. U radu smo vršili kompresiju u binarnu datoteku. Enkodiranje uključuje:

1. Delta enkodiranje pozicija podudarnosti.
2. Pohranjivanje duljina podudarnosti.
3. Enkodiranje informacija o nepodudarnostima.

Delta enkodiranje smanjuje veličinu pohranjenih podataka pohranjivanjem razlika između uzastopnih pozicija umjesto apsolutnih pozicija.

2.3.1. Enkodiranje informacija o malim znakovima

Slično osnovnim sekvencama, informacije o malim slovima također se enkodiraju:

1. Delta enkodiranje pozicija regija malih slova.
2. Pohranjivanje duljina regija malih slova.

Enkodirane informacije se pohranjuju zajedno s osnovnim sekvencama kako bi se omogućila točna dekompresija.

2.3.2. Enkodiranje specijalnih znakova

Specijalni znakovi (npr. 'N' ili ostali simboli) enkodiraju se pohranjivanjem njihovih pozicija i znakova:

1. Pohranjivanje pozicija specijalnih znakova.
2. Pohranjivanje znakova.

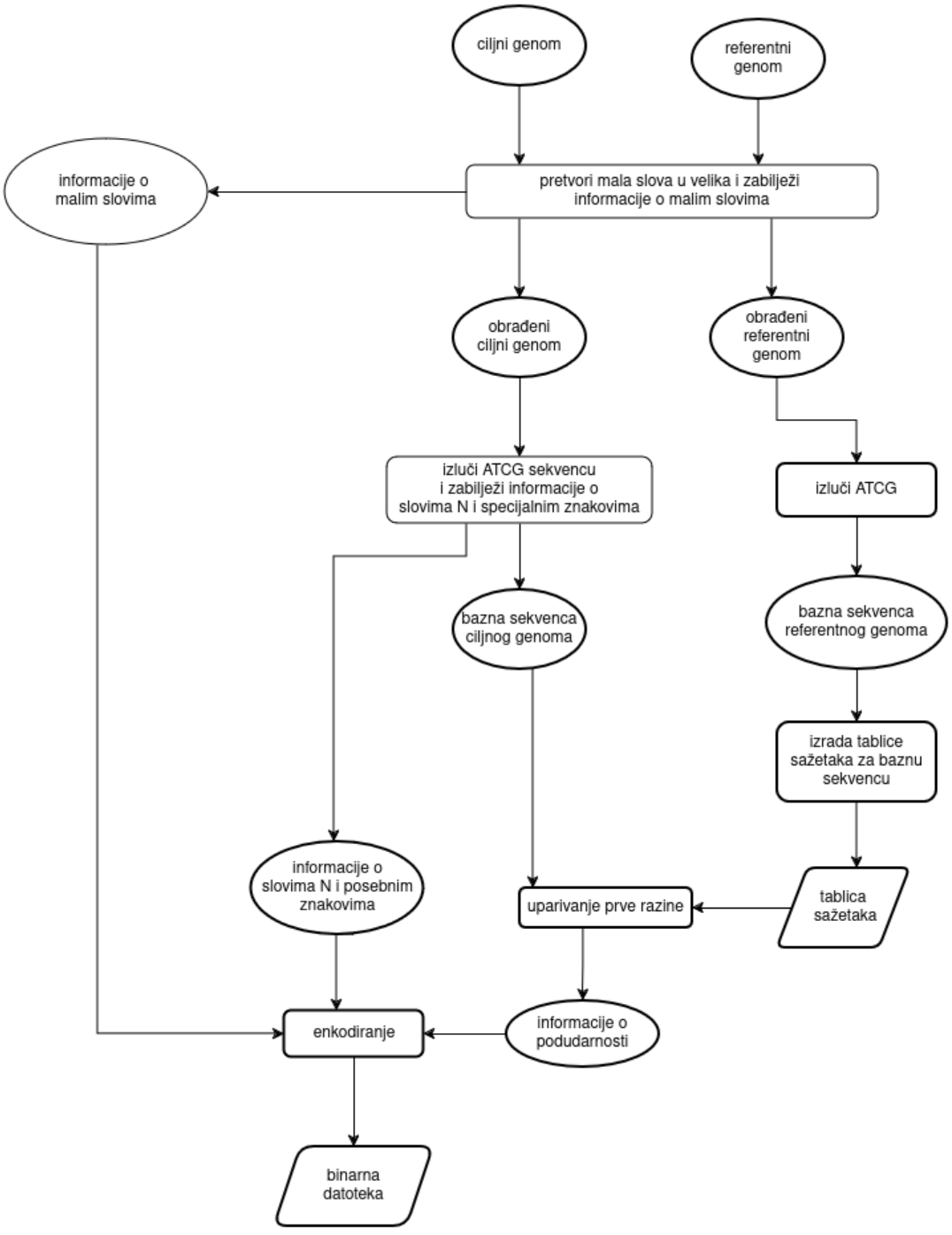
Ove informacije omogućuju točnu rekonstrukciju izvornih sekvenci tijekom dekompresije.

2.4. Dekompresija

Proces dekompresije uključuje rekonstrukciju izvornih sekvenci iz komprimiranih podataka. Koraci dekompresije su:

1. Čitanje enkodiranih podataka iz datoteke.
2. Dekodiranje informacija o sekvencama koristeći referentne sekvence i pohranjene podudarnosti.
3. Rekonstrukcija regija malih slova i specijalnih znakova na temelju pohranjenih informacija.

Dekompresija osigurava točnu rekonstrukciju izvornih sekvenci, uključujući sve specifične informacije kao što su mala slova i specijalni znakovi.



Slika 2.1: Dijagram toka algoritma kompresije

3. Analiza i usporedba rezultata

Rezultati u usporedbi sa radom[3] imaju veće vrijeme kompresije, a jednako ili manje vrijeme dekompresije. Na performanse utječe i dodatno postavljanje parametara kompresije. Svi su rezultati uprosječeni

3.1. Ispitni podaci

Datoteka	**Link**
hg13	`ftp://hgdownload.soe.ucsc.edu/goldenPath/hg13/chromosomes/`
hg14	`ftp://hgdownload.soe.ucsc.edu/goldenPath/hg14/chromosomes/`
hg16	`ftp://hgdownload.soe.ucsc.edu/goldenPath/hg16/chromosomes/`
hg17	`ftp://hgdownload.soe.ucsc.edu/goldenPath/hg17/chromosomes/`
hg18	`ftp://hgdownload.soe.ucsc.edu/goldenPath/hg18/chromosomes/`
hg19	`ftp://hgdownload.soe.ucsc.edu/goldenPath/hg19/chromosomes/`
ce6	`ftp://hgdownload.soe.ucsc.edu/goldenPath/ce6/chromosomes/`
ce10	`ftp://hgdownload.soe.ucsc.edu/goldenPath/ce10/chromosomes/`
ce11	`ftp://hgdownload.soe.ucsc.edu/goldenPath/ce11/chromosomes/`

Tablica 3.1: Izvori podataka

Vrsta	Datoteka	Broj kromosoma
H.sapiens	hg13	24
H.sapiens	hg14	24
H.sapiens	hg16	24
H.sapiens	hg17	24
H.sapiens	hg18	24
H.sapiens	hg19	24
Caenorhabditis elegans	ce6	7
Caenorhabditis elegans	ce10	7
Caenorhabditis elegans	ce11	7

Tablica 3.2: Vrsta i broj kromosoma

Datoteka	Veličina (MB)	Utrošak memorije (GB)	Vrijeme(s)
chr13	108.26	2.99	433.45
chr14	98.5	2.97	370.22
chr16	88.53	2.95	302.65
chr17	82.03	3.13	303.97
chr18	75.2	2.83	254.83
chr19	57.62	2.82	180.7
hg13_chr22	46.54	1.35	123.83
hg17_chr22	48.5	1.07	121.92
hg18_chr22	48.3	1.33	123.08

Tablica 3.3: Detalji kompresije datoteka(chr13 korištena kao referentna datoteka)

Datoteka	Veličina (MB)	Utrošak memorije (GB)	Vrijeme(s)
chr13	108.26	0.987	23.60
chr14	98.5	0.883	23.51
chr16	88.53	0.88	21.29
chr17	82.03	0.788	19.86
chr18	75.2	0.712	20.47
chr19	57.62	0.595	14.60
hg13_chr22	46.54	0.11	2.60
hg17_chr22	48.5	0.13	2.67
hg18_chr22	48.3	0.11	2.55

Tablica 3.4: Detalji dekompresije datoteka (chr13 korištena kao referentna datoteka)

Referentna	Ciljne datoteke	Veličina(MB)	Kompresirano(MB)	Vrijeme (s)	Memorija(GB)
chr13	chr17, chr18	157.23	6.8	388.80	3.23
chr13	chr19, hg13	104.16	5.5	205.13	2.52
chr13	hg17, hg18	96.8	5.3	200.75	2.57
chr14	chr17, chr18	157.23	6.8	370.50	2.90
chr14	chr19, hg17	106.12	5.6	205.22	2.41
chr14	hg18, hg13	94.84	5.3	198.43	2.33
hg13	chr16, chr17	170.56	7.0	392.89	3.03
hg13	chr18, chr19	132.82	6.0	351.53	2.62
hg13	hg17, hg18	95.04	5.3	258.90	2.32
hg17	chr16, chr17	170.56	7.0	377.86	3.01
hg17	chr18, chr19	132.82	6.0	368.76	2.60
hg17	hg13, hg18	95.04	5.3	294.36	2.38
hg18	chr16, chr17	170.56	7.0	395.47	3.08
hg18	chr18, chr19	132.82	6.0	365.91	2.65
hg18	hg13, hg17	95.04	5.3	203.09	2.36

Tablica 3.5: Detalji kompresije više datoteka (Second level matching)

Referentna	Ciljne datoteke	Veličina(MB)	Kompresirano(MB)	Vrijeme (s)	Memorija(GB)
chr13	chr17, chr18	157.23	6.8	31.32	0.98
chr13	chr19, hg13	104.16	5.5	21.05	0.99
chr13	hg17, hg18	96.8	5.3	19.22	0.75
chr14	chr17, chr18	157.23	6.8	28.37	0.79
chr14	chr19, hg17	106.12	5.6	20.70	0.64
chr14	hg18, hg13	94.84	5.3	20.57	0.57
hg13	chr16, chr17	170.56	7.0	29.86	0.88
hg13	chr18, chr19	132.82	6.0	23.83	0.87
hg13	hg17, hg18	95.04	5.3	19.01	0.78
hg17	chr16, chr17	170.56	7.0	30.79	0.90
hg17	chr18, chr19	132.82	6.0	29.20	0.91
hg17	hg13, hg18	95.04	5.3	20.03	0.65
hg18	chr16, chr17	170.56	7.0	27.32	0.96
hg18	chr18, chr19	132.82	6.0	22.80	0.87
hg18	hg13, hg17	95.04	5.3	17.32	0.50

Tablica 3.6: Detalji dekompresije više datoteka (Second level matching)

3.2. Umjetno generirani podatci

Podatci su generirani nasumičnim ispisom slova A, T, C, G, X i N. Veličine su redom 10^3, 10^4, 10^5, 10^6, 10^7 znakova te su duljine linije 34. Referentna je datoteka file_0.fa veličine 10^3 znakova.

Datoteka	Veličina (KB)	Utrošak memorije (kb)	Vrijeme(s)
file_1	4	3840	0.01
file_6	4	3968	0.02
file_2	12	3840	0.03
file_7	12	4096	0.01
file_3	104	4656	0.03
file_8	104	4796	0.05
file_4	1008	14236	0.19
file_9	1008	14360	0.19
file_5	10056	113268	1.51
file_10	10056	113892	1.53

Tablica 3.7: Detalji kompresije datoteka

Datoteka	Veličina (KB)	Utrošak memorije (kb)	Vrijeme(s)
file_1	4	3960	0.00
file_6	4	3968	0.01
file_2	12	4070	0.00
file_7	12	4096	0.00
file_3	104	4536	0.00
file_8	104	4532	0.03
file_4	1008	13444	0.02
file_9	1008	13050	0.05
file_5	10056	30023	0.14
file_10	10056	99008	0.10

Tablica 3.8: Detalji dekompresije datoteka

4. Zaključak

Naša implementacija HRCM algoritma pokazala je veće vrijeme kompresije, dok dekompresija pokazuje jednako ili manje vrijeme izvođenja. Unutar kompresije, najviše vremena odlazi na izradu tablica sažetaka i uparivanje prve razine. Optimizacije uključuju korištenje 'rolling-hash' algoritma za sažetke i bitovne aritmetike za brži pristup podacima. Naša implementacija ne koristi globalne varijable te dijeli kod na manje funkcije, čime se poboljšava čitljivost i održavanje koda.

5. Literatura

[1] Zia Ahmed, Shariq Zeeshan, Dhaval Mendhe, i Xin Dong. Human gene and disease associations for clinical-genomics and precision medicine research. *Clin Transl Med*, 10(1):297–318, Siječanj 2020. doi: 10.1002/ctm2.28.

[2] Reid J. Robinson. How big is the human genome?, 2020. URL `https://medium.com/precision-medicine/how-big-is-the-human-genome-e90caa3409b0`. Accessed: 2024-05-30.

[3] Haichang Yao, Yimu Ji, Kui Li, Shangdong Liu, Jing He, i Ruchuan Wang. Hrcm: An efficient hybrid referential compression method for genomic big data. `https://doi.org/10.1155/2019/3108950`, 2019. BioMed Research International, vol. 2019, Article ID 3108950, 13 pages.

6. Sažetak

Bioinformatika je relativno mlada znanstvena disciplina unutar koje se primjenjuju znanja iz računalne znanosti kako bi se analizirali podatci iz bioloških istraživanja i mjerenja. Sekvenciranje genoma tehnika je kojom se utvrđuju vrsta i redoslijed pripadnih nukleotida. Genomi živih bića veliki su skupovi podataka. Primjerice, ljudski genom sadrži oko 3 milijardi nukleotida što zahtjeva otprilike 725 megabajta prostora za pohranu[2]. Stoga, potrebno je razvijati algoritme koji će osigurati pohranu genoma u smanjenoj veličini bez gubitka podataka. Drugim riječima, potrebno je izvesti kompresiju bez gubitka informacije.

HRCM(Hybrid Referential Compression Method) algoritam je koji sažima jednu ili više sekvenci. Osnovna ideja referencijalne kompresije počiva na činjenici[1] da se svi ljudi podudaraju u 99,9% genoma. Algoritam se odvija u tri faze: ekstrakcija informacija iz sekvenci, uparivanje informacija iz sekvenci i enkodiranje informacija iz sekvenci. U ovom projektu, implementirali smo HRCM algoritam te usporedili njegovu izvedbu s originalnim radom[3]. Naša implementacija omogućava brzu dekompresiju, dok je kompresija nešto sporija zbog detaljnih uparivanja i enkodiranja podataka.