

SVEUČILIŠTE U ZAGREBU  
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

**DOKUMENTACIJA**

**Poboljšanje djelomično  
sastavljenog genoma dugim  
očitanjima**

*Bruno Kovač, Tonko Sabolčec, Fabijan Čorak*

*Voditelj: doc. dr. sc. Krešimir Križanović*

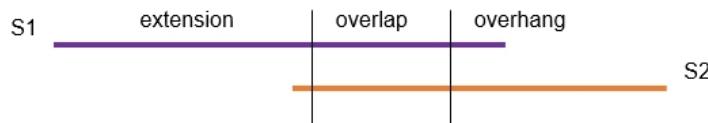
Zagreb, siječanj 2020.

# SADRŽAJ

<b>1. Uvod</b>	<b>1</b>
<b>2. Postupak</b>	<b>2</b>
2.1. Izgradnja grafa . . . . .	2
2.2. Obilazak grafa . . . . .	2
2.3. Obrada putova . . . . .	3
<b>3. Rezultati</b>	<b>6</b>
3.1. Generirani podaci . . . . .	6
3.2. E. coli . . . . .	8
3.3. C. jejuni . . . . .	10
3.4. B. grahamii . . . . .	11
3.5. Vremensko i memorijsko opterećenje . . . . .	14
<b>4. Upute za korištenje</b>	<b>15</b>
4.1. Instalacija . . . . .	15
4.2. Pokretanje . . . . .	15
<b>5. Zaključak</b>	<b>16</b>
<b>6. Literatura</b>	<b>17</b>

# 1. Uvod

Sekvenciranje genoma svodi se na kombiniranje očitanja u jednu cjelinu. Ovaj rad pretpostavlja da su očitanja već sastavljena, ali djelomično - u fragmente. Jedan takav fragment naziva se *contig*. Dakle, zadatak se svodi na što bolje povezivanje *contiga*, što smo učinili postupkom opisanim u [1]. Taj rad definira nekoliko mjera preklopljenosti očitanja koje kombiniraju duljinu područja *overlap* (*OL*), *overhang* (*OH*) i *extension* (*EL*). Mjere su ovdje definirane za dva očitanja  $S_1$  i  $S_2$ ; pripadnost područja određena je indeksom.



**Slika 1.1:** Preklop dvaju očitanja s naznačenim područjima

- *sequence identity (SI)* - omjer ukupnog broja podudarajućih znakova u *overlap* područjima i duljine duljeg od tih dvaju područja

$$SI = \frac{\text{broj\_podudaranja}}{\max(OL_1, OL_2)}$$

- *overlap score (OS)*

$$OS = (OL_1 + OL_2) \frac{SI}{2}$$

- *extension score (ES)* - uz  $S_2$  kao produžetak od  $S_1$

$$ES_2 = OS + \frac{EL_2 - OH_1 - OH_2}{2}$$

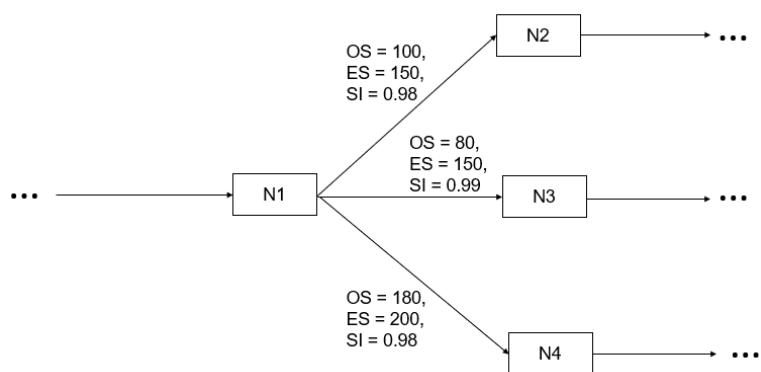
## 2. Postupak

Sastavljanje očitanja u niz modelirano je izgradnjom i obilaskom grafa.

### 2.1. Izgradnja grafa

Svaki *contig* i svako očitanje čine jedan čvor grafa. Dodatno, u skup čvorova dodaje se i reverzna inačica svakog očitanja i *contiga* jer nije unaprijed poznato koja je pogodna orijentacija. Čvor koji predstavlja *contig* zovemo *anchor*. Za svako očitanje Brid postoji između svaka dva čvora čiji je *SI* veći od nekog minimuma. Pritom svaki brid nosi informacije o prekopljenosti čvorova koje povezuje (*SI*, *OS*, *ES*). Te mjerne računaju se na temelju informacija o prekopljenosti dobivenih korištenjem alata *minimap2* opisanog u [3].

### 2.2. Obilazak grafa



**Slika 2.1:** Odabir sljedećeg čvora u obilasku

Slika razmatra grananje u hipotetskom čvoru N1. Prvim postupkom prioritetna lista je: [N4, N2, N3], a drugim postupkom: [N4, N3, N2]. Treći postupak daje vjerojatnosti odabira čvorova {N2: 150/500, N3: 150/500, N4: 200/500}.

Kroz graf se traže putovi čije su krajnje točke *anchor* čvorovi. Za to se koriste tri načina obilaska, prilikom kojih je postavljena maksimalna dubina pretraživanja i pamte se obiđeni čvorovi:

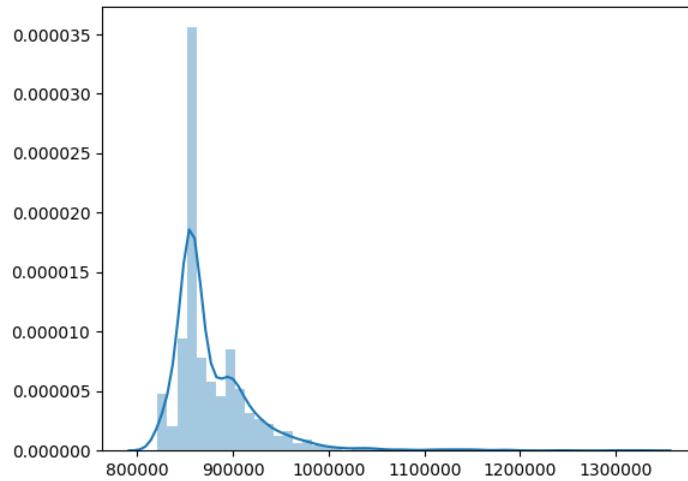
1. Iz *anchor* čvora pretraga se nastavlja u sve susjedne čvorove. Iz svakog sljedećeg čvora, pretraga se nastavlja u onaj susjed s kojim je najveći *overlap score*, a kojim se u konačnici dolazi do *anchor* čvora. Ako je *overlap score* jednak, gleda se *sequence identity*. Ako je pak i ta mjera jednaka, gleda se duljina očitanja.
2. Kao i prethodni način, ali umjesto mjere *overlap score* gleda se *extension score*.
3. U svakom čvoru susjed se odabire probabilistički - s vjerojatnošću odabira proporcionalnom mjeri *extension score*, sve dok se ne dosegne *anchor*. Postupak se pokreće iz svakog *anchor* čvora proizvoljan broj puta. Ovo je tzv. Monte Carlo metoda.

### 2.3. Obrada putova

Nakon agregacije putova dobivenih opisanim postupcima, odbacuju se duplikati i obrađuju se putovi između svaka dva *anchor* čvora. Putovi se sortiraju uzlazno prema duljini i razmatraju se prozori fiksne širine  $W$ . Pritom  $i$ -ti prozor obuhvaća sve puteve čija je duljina  $L$ :  $(i-1)W < L \leq iW$  za dobro definirane  $i$ . Koliko je koji prozor frekventan vidljivo je na primjeru povezivanja dvaju *contiga* na slici 2.2. Prepostavka je da dominacija nekog prozora ukazuje na to da su u njemu putovi koji su izgledni kandidati za povezivanje dvaju *anchor* čvorova između kojih se nalaze. Problem je što to ne mora biti istina, tj. moguće je dominiranje nekog prozora, a da ta dva *anchor* čvora uopće nisu uzastopna.

Za svaki par *anchor* čvorova računa se konsenzus - struktura podataka koja sadrži reprezentativni put te broj valjanih putova između tih dvaju čvorova. Kao reprezentant odabire se proizvoljni čvor maksimalne duljine. Na slici 2.2 to je jedan od putova duljine cca 850000 (najviši stupac). Dodatno, iz susjedstva svakog *anchor* čvora uklanjuju se oni *anchor* čvorovi do kojih je broj putova ispod neke određene granice. Drugim riječima, takvi susjadi vjerojatno ne trebaju biti povezani.

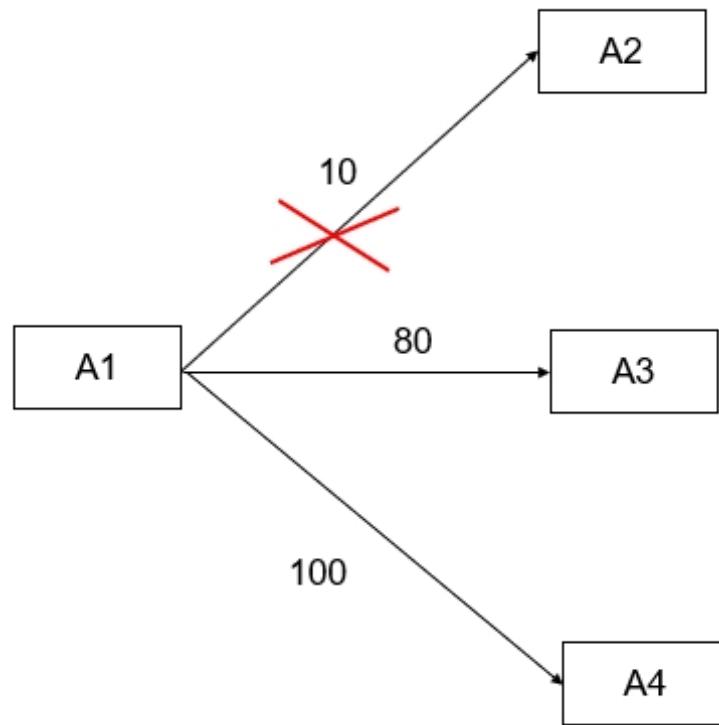
U konačnici se konstruiraju poboljšani sljedovi *contiga* i za svako poboljšanje generira se izlazna datoteka. Poboljšanjem se smatra utvrđeni niz od barem dva *contiga* tj. *anchor* čvora s odgovarajućim očitanjima koja popunjavaju praznine. Podržana su dva načina konstrukcije konačnih poboljšanih nizova na temelju filtriranih konsenzusa:



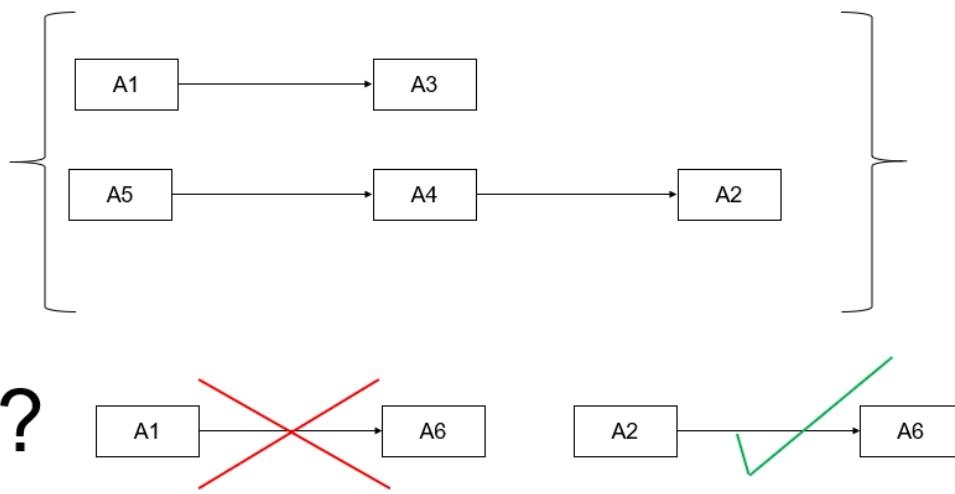
**Slika 2.2:** Dstribucija putova po prozorima fiksne širine

Distribucija putova između dva *anchor* čvora. Na x-osi označene su širine prozora, a na y-osi frekventnost (udio brojnosti putova prozora u ukupnom broju putova). Najviši stupac dat će i reprezentant povezanosti dvaju čvorova.

1. Skup poboljšanja inicijalno je prazan. Iterira se po silazno sortiranim konsenzusima i putanja svakog pokuša se dodati. Kao kriterij sortiranja uzima se broj putova između pripadnih *anchor* čvorova. Dodavanje je moguće ako su krajnji čvorovi puta koji se dodaje slobodni u skupu poboljšanja. Ilustracija je dana na slici 2.4.
2. Ovaj način koristi se indeksom konflikta. Neka iz nekog *anchor* čvora postoji po više putanja do drugih *anchor* čvorova. Ako je  $D_1$  najveći broj putova koji postoji do drugog *anchor* čvora, a  $D_2$  drugi najveći takav broj, tada se indeks konflikta ( $CI$ ) tog čvora definira kao  $CI = \frac{D_2}{D_1}$ . Primjerice,  $CI$  čvora A1 sa slike 2.3 iznosi 0.8. Dodatno, kažemo da *anchor* čvor ima konfliktne veze ako njegov  $CI$  premašuje prethodno definiranu vrijednost. Ako čvor nije konfliktan, u skup poboljšanja uvrštava se njegov put do onog *anchor* čvora do kojeg vodi najveći broj putova. U primjeru sa slike 2.3 to bi bio čvor A4.



**Slika 2.3:** Otpadanje jedne grane u slučaju kada je granica za odbacivanje 12.5% od najvećeg broja valjanih putova (100).



**Slika 2.4:** Prvi način izgradnje poboljšanja. Pokušaj dodavanja dvaju putova u skup poboljšanja.

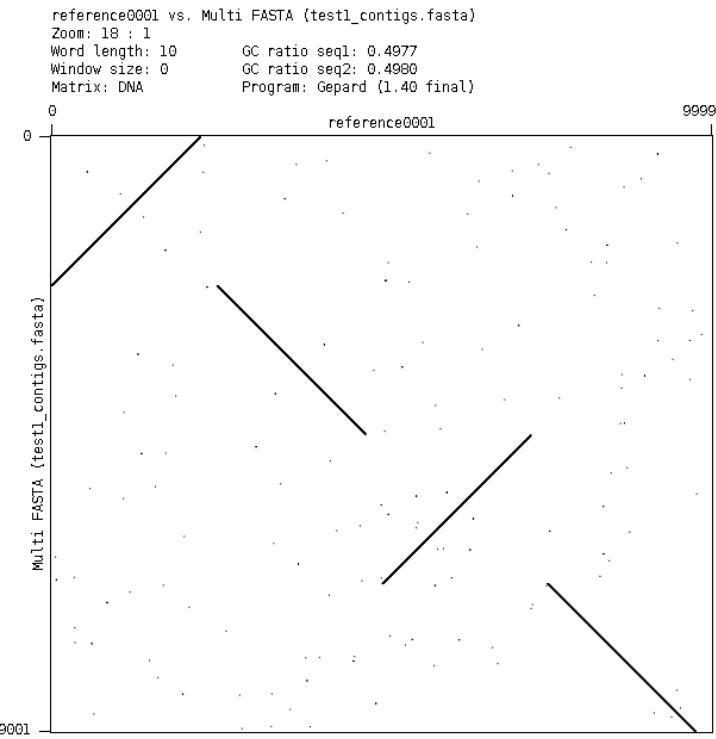
Gornji dio slike prikazuje dosadašnji skup poboljšanja. Put 1, 6 nije moguće dodati jer *anchor* 1 nije slobodan. Put 2, 6 moguće je dodati i njime nastaje put 5, 4, 2, 6. Kada bi postupak izgradnje poboljšanih nizova završio nakon ovog dodavanja, konačni bi rezultat bila dva poboljšanja: 1, 3 i 5, 4, 2, 6.

# 3. Rezultati

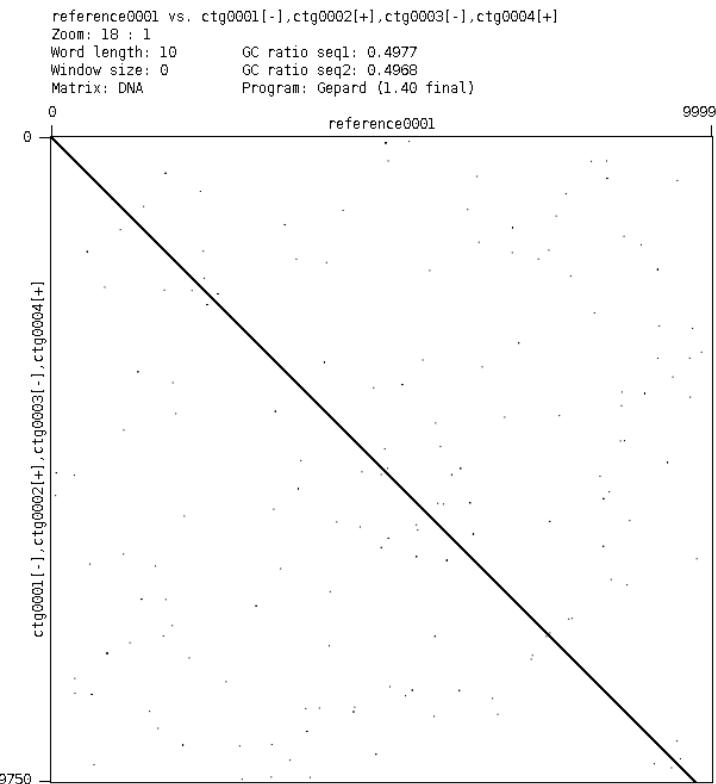
Implementacija je ispitana na generiranom skupu te na skupovima *E. coli*, *C. jejuni* i *B. grahamii*. Posljednja tri skupa podataka temelje se na genomima reda veličine 1 Mb. Priložene su matrice preklapanja između referentnih skupova te prvo neobrađenih, a odmah zatim i obrađenih *contiga*. Za svaki skup podataka prikazana su moguća poboljšanja sastavljanja *contiga*. Sve matrice dobivene su alatom *Gepard* opisanim u [2]. Ishodište je u gornjem lijevom kutu, x-os predstavlja referencu, a y-os predstavlja *contige*. Crne točke označavaju podudaranja. Predznak u opisima matrica označava korištenu orijentaciju *contiga*: + je izvorna, a - reverzna. Npr. +5, -2 znači da je pronađeno povezivanje za izvorno orijentirani *contig* 5 i reverzni 2.

## 3.1. Generirani podaci

Implementacija je ispitana na umjetno generiranom skupu podataka s duljinama reda veličine 10 kb.



**Slika 3.1:** Matrica preklapanja neobrađenih umjetno generiranih *contiga* i točne reference.

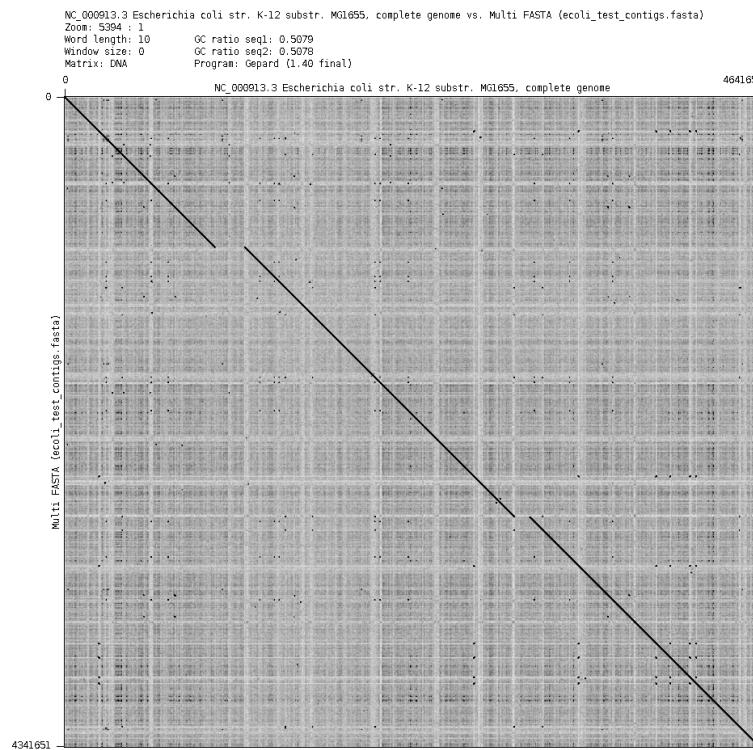


**Slika 3.2:** Poboljšanje generiranog testnog slučaja.

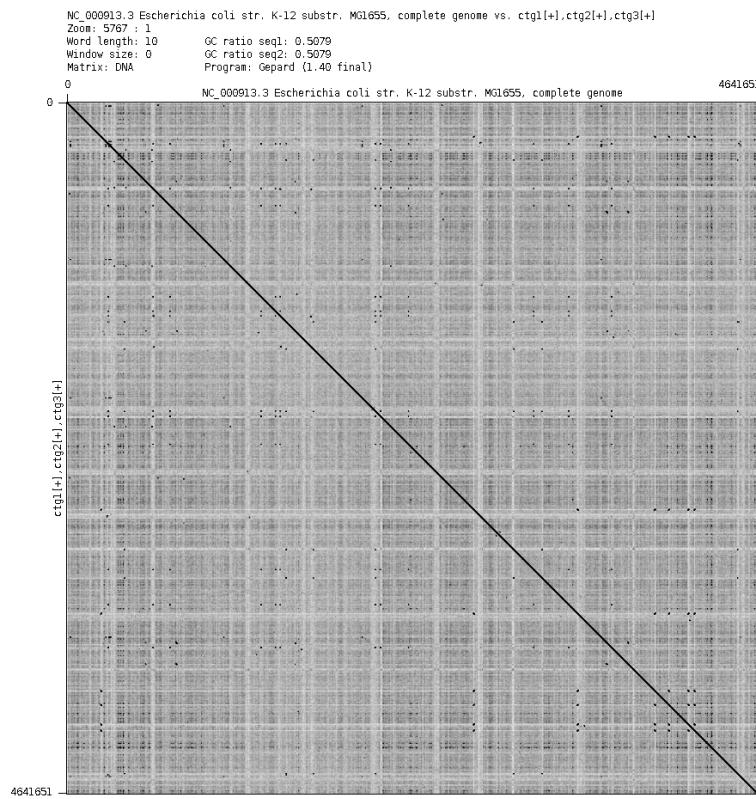
Pronađeni redoslijed *contiga* je -1, +2, -3, +4.

## 3.2. E. coli

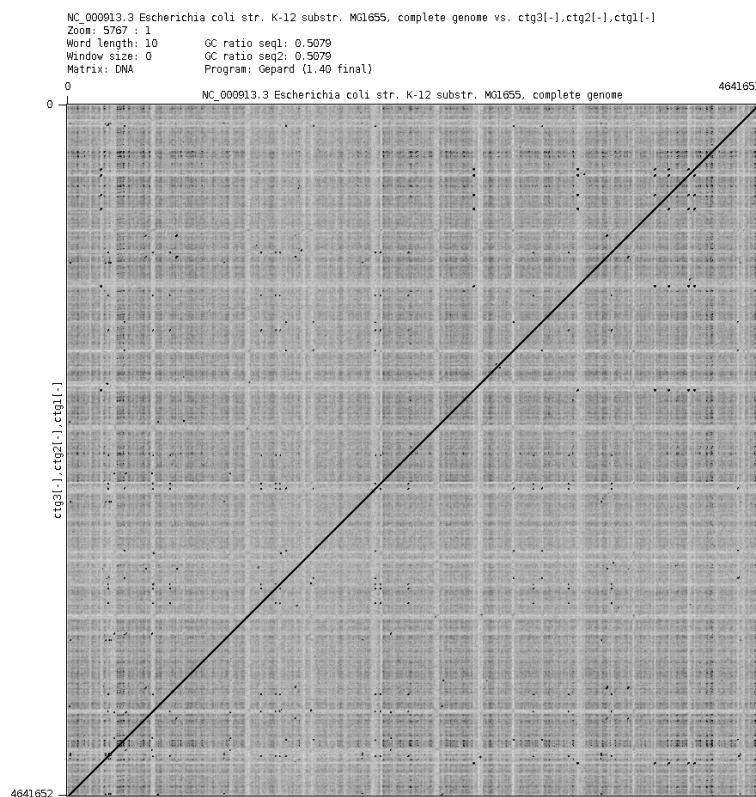
Ovo je sintetski skup podataka, shodno čemu su i ostvareni rezultati bolji u usporedbi s narednim dvama, realnim skupovima. Rezultirajući izlazi programa zapravo su inverzi jedan drugog. Pripadne matrice preklapanja priložene su na slikama 3.3, 3.4 i 3.5.



**Slika 3.3:** Matrica preklapanja neobrađenih *contiga* E. coli i referentnog skupa.



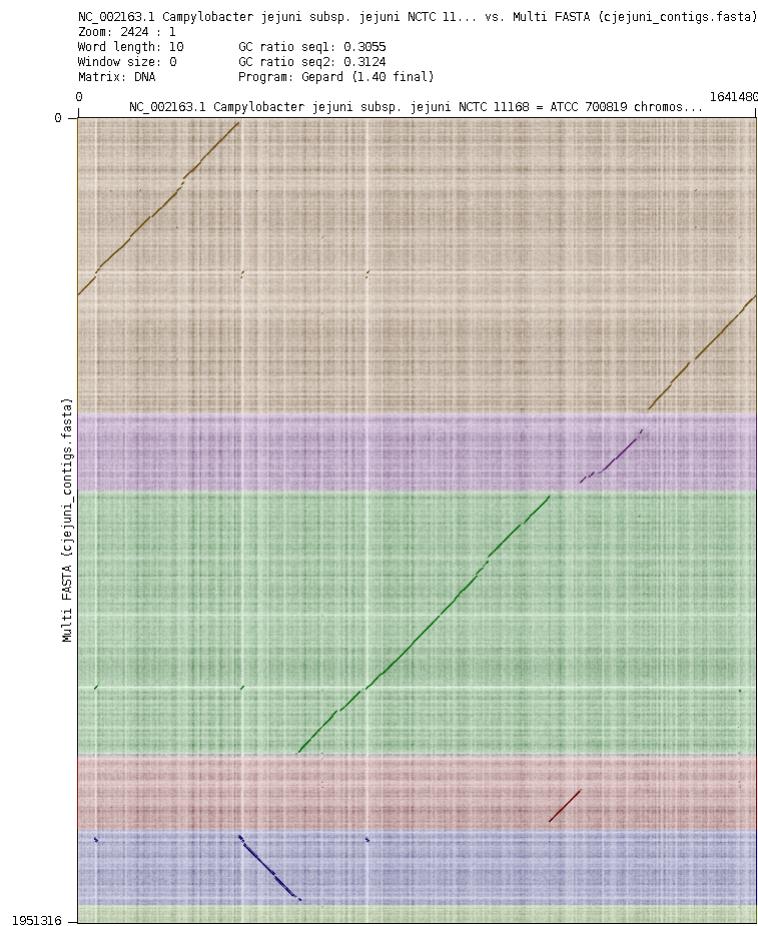
**Slika 3.4:** Prvo poboljšanje: niz *contiga* +1, +2, +3.



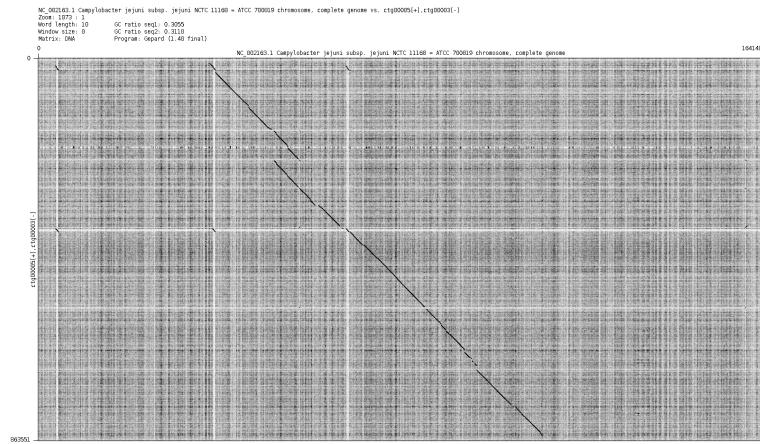
**Slika 3.5:** Drugo poboljšanje: niz *contiga* -3, -2, -1.

### 3.3. C. jejuni

Prvi realni skup podataka predstavlja očitanja nad genomom bakterije C. jejuni. Pripadne matrice preklapanja priložene su na slikama 3.6 i 3.7.



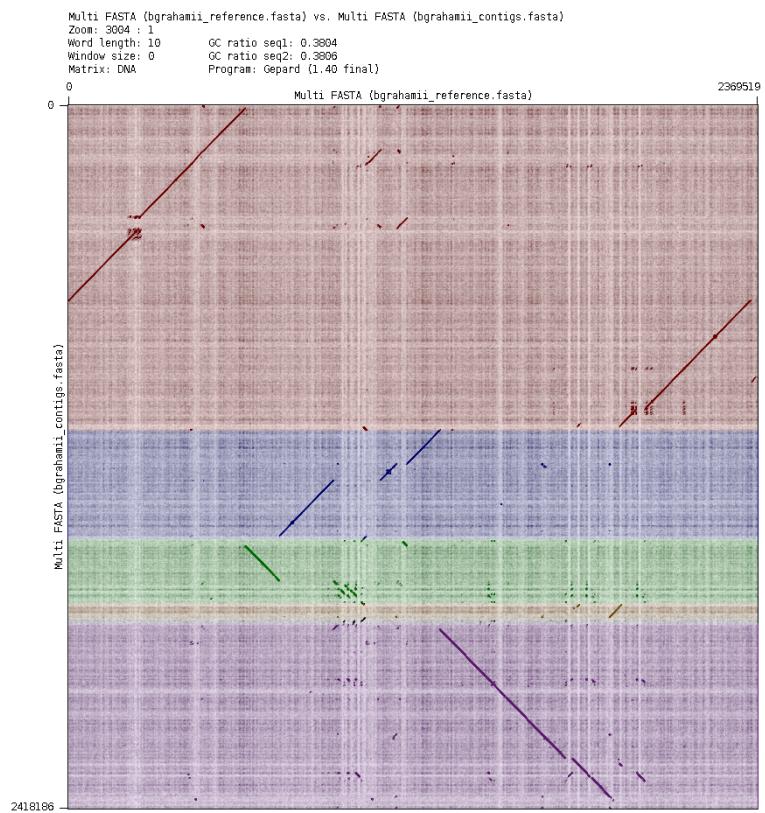
**Slika 3.6:** Matrica preklapanja neobrađenih *contiga* C. jejuni i referentnog skupa. Svaki contig predstavljen je jednom bojom. *Contig 1* (njegozini) već je dakle u ulaznoj datoteci definiran *izlomljeno*, tj. obuhvaća očitanja sa stvarnog početka i kraja.



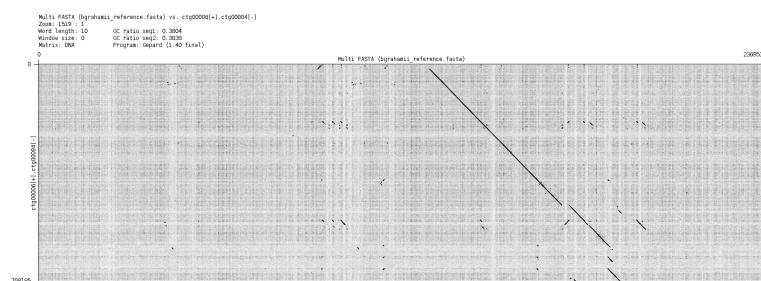
**Slika 3.7:** Poboljšanje: niz *contiga* +5, -3.

### 3.4. **B. grahamii**

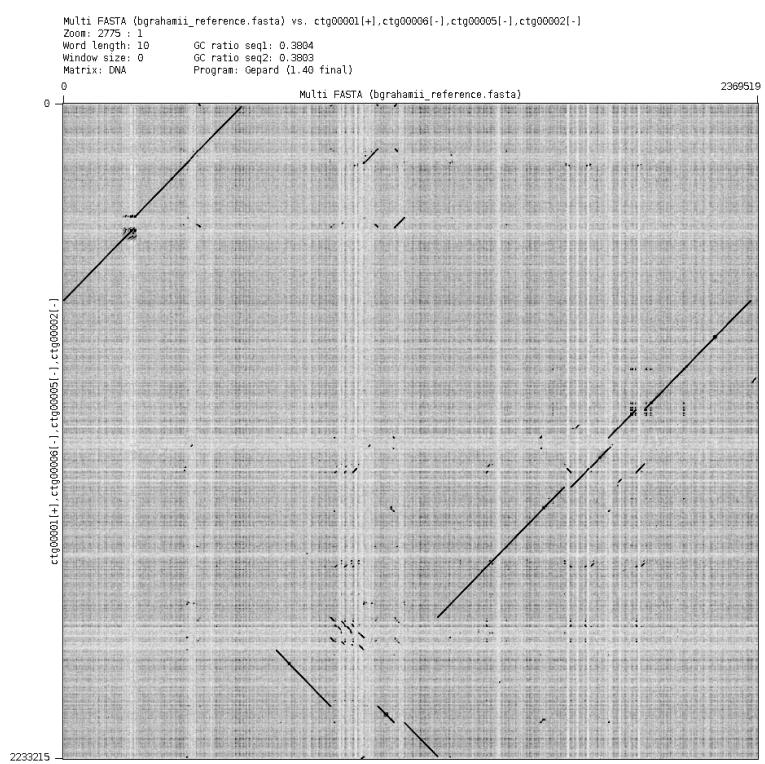
Drugi realni skup podataka čine očitanja genoma bakterije *B. grahamii*. Pripadne matrice preklapanja priložene su na slkama 3.8, 3.9 i 3.10.



**Slika 3.8:** Matrica preklapanja neobrađenih *contiga* *B. grahamii* i referentnog skupa. Svaki contig predstavljen je jednom bojom. Ponovno su vidljive izlomljenosti u ulaznim podacima.



**Slika 3.9:** Poboljšanje: niz *contiga* +6, -4.



**Slika 3.10:** Poboljšanje: niz *contiga* +1, -6, -5, -2.

### 3.5. Vremensko i memorjsko opterećenje

U tablici 3.1 priloženi su podaci o vremenu i memoriji jednoretvenog izvođenja procesorom Intel Core i7-8565U nepojačane frekvencije.

skup podataka	vrijeme (s)	memorija (MB)
E. coli	9.396	1498
C. jejuni	26.675	1756
B. grahamii	58.714	3923

**Tablica 3.1:** Vremensko i memorjsko opterećenje izvođenja.

# 4. Upute za korištenje

## 4.1. Instalacija

Za početak je potrebno skinuti kod s GitHub repozitorija pokretanjem naredbe:

```
git clone https://github.com/brunokovac/BIOINF
```

Za prevođenje Java paketa unutar direktorija *Kodovi/PROJEKT\_8* pokrenite naredbu:

```
javac -d bin -sourcepath src src/hr/fer/bioinf/Main.java
```

## 4.2. Pokretanje

Osnovni oblik pokretanja programa zadan je sljedećom naredbom:

```
java -cp bin/ hr.fer.bioinf.Main \
--reads-path=reads.fasta \
--contigs-path=contigs.fasta \
--reads-overlaps-path=reads_reads_overlaps.paf \
--contigs-reads-overlaps-path=reads_contigs_overlaps.paf \
--output-folder=output_dir/
```

Pritom je potrebno postaviti argumente u naredbenom retku `--reads-path` i `--contigs-path` na datoteke u *fasta* formatu. Opcije `--reads-overlaps-path` i `--contigs-reads-overlaps-path` potrebno postaviti na lokacije datoteka koje se prethodno dobivaju korištenjem alata Minimap. Konačno, `--output-path` potrebno je postaviti na direktorij u koji će se spremiti generirani DNA slijedovi. Ovo su samo neke od više mogućih opcija dostupnih u glavnom programu. Pokretanjem programa s opcijom `--help` moguće je vidjeti popis ostalih opcija (i njihovim pretpostavljenim vrijednostima) pomoću kojih je moguće kontrolirati izvođenje algoritma.

## 5. Zaključak

Implementacija uspijeva pronaći poboljšane nizove *contiga*, s većom uspješnošću za sintetski primjer E. coli koji nema dubioznih *contiga*. Vremensko i memorijsko opterećenje implementacije unutar je granica izvedivosti na prosječnom suvremenom kućnom računalu.

## 6. Literatura

- [1] Huilong Du i Chengzhi Liang. Assembly of chromosome-scale contigs by efficiently resolving repetitive sequences with long reads. *bioRxiv*, 2018. doi: 10.1101/345983. URL <https://www.biorxiv.org/content/early/2018/06/13/345983>.
- [2] Jan Krumsiek, Roland Arnold, i Thomas Rattei. Gepard: a rapid and sensitive tool for creating dotplots on genome scale. *Bioinformatics*, 23(8):1026–1028, 2007.
- [3] Heng Li. Minimap2: pairwise alignment for nucleotide sequences. *Bioinformatics*, 34(18):3094–3100, 05 2018. ISSN 1367-4803. doi: 10.1093/bioinformatics/bty191. URL <https://doi.org/10.1093/bioinformatics/bty191>.