SVEUČILIŠTE U ZAGREBU
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

Sastavljanje genoma koristeći detekciju tipova očitanja

Luka Požega, Sara Bakić

Zagreb, kolovoz 2020.

Ovaj rad izrađen je u Laboratoriju za bioinformatiku i računalnu biologiju na Fakultetu elektrotehnike i računarstva pod vodstvom prof. dr. sc. Mile Šikića i predan je na natječaj za dodjelu Rektorove nagrade u akademskoj godini 2019./2020

SADRŽAJ

1. Uvod

Bioinformatika (*bios* = *život* + *informatika*) interdisciplinarno je znanstveno područje koje usko povezuje računarstvo s biologijom. Bioinformatika razvija i koristi računalne tehnike pohrane, obrade i analize podataka, a s ciljem brze i točne interpretacije bioloških podataka.

Ubrzani razvoj bioinformatike traje već dva desetljeća, a glavni uzrok iza popularizacije bioinformatike jest pojeftinjenje i sve veća dostupnost tehnologija sekvencioniranja - osnovnih metoda generiranja bioloških podataka.

Sekvencioniranje je proces određivanja slijeda nukleonskih baza - adenina (**A**), citozina (**C**), gvanina (**G**), timina (**T**), uracila (**U**) - u DNA ili RNA lancu. Obzirom da je cjelokupna genetska informacija sadržana u DNA ili RNA lancu, poredak nukleonskih baza u lancu je potpuna informacija o organizmu te se kao takva koristi u bioinformatičkim analizama.

Postoji nekoliko široko korištenih metoda sekvencioniranja koje kao zajedničko obilježje imaju ograničenu sposobnost točnog čitanja nukleotida. Ovisno o korištenoj metodi sekvencioniranja, duljine fragmenata DNA lanca očitanih odjednom kreću se između nekoliko desetaka i nekoliko stotina tisuća nukleotida. Obzirom da se DNA lanci i najsitnijih organizama sastoje od nekoliko milijuna nukleonskih baza, u jednom čitanju je nemoguće očitati cjelovitu sekvencu nukleotida.

Iz tih razloga je nužno proces sekvencioniranja provoditi na način da se DNA lanac očitava fragment po fragment te se potom originalni DNA lanac sastavlja iz prethodno očitanih fragmenata.

Dominantnom metodom sekvencioniranja smatra se takozvano *shotgun* sekvencioniranje [1] gdje se DNA umnaža, potom na slučajan način lomi u mnogo sitnih fragmenata koji se neovisno jedan o drugome sekvencioniraju. Logika iza ovakve metodologije je vrlo jednostavna. Osnovna ideja sekvencioniranja je generiranje cjelovite DNA sekvence. Obzirom da, zbog ograničenja metoda sekvencioniranja, možemo očitavati samo dijelove sekvence odjednom, bez umnažanja DNA lanca bilo bi nemoguće odrediti ispravan poredak očitanih fragmenata te konačno, nemoguće odrediti origi-

nalni poredak nukleonskih baza što kao posljedicu ima gubitak genetske informacije o organizmu.

Problem ispravne i potpune rekonstrukcije genoma iz sekvencioniranih očitanja jedan je od najkompleksnijih problema kojima se bavi bioinformatika. U procesu de novo sastavljanja genoma nailazi se na mnoštvo problema, uzrokovanih, kako biološkim specifičnostima, tako i greškama u procesu sekvencioniranja

Ovaj rad fokusira se na specifične vrste očitanja koja predstavljaju probleme u procesu de novo sastavljanja genoma vođenom referentnom sekvencom te njihov utjecaj na gornju granicu u sastavljenosti genoma. S ciljem povećanja kvalitete sastavljenog genoma, razvijene su metodologije za detekciju i upravljanje problematičnim očitanjima. Razvijen je i alat za računanje gornje granice u sastavljanju genoma koji je korišten za istraživanje uspješnosti sastavljenosti genoma uz korištenje informacija o različitim tipovima očitanja.

U drugom poglavlju su prikazani i pobliže objašnjeni formati podataka koji se koriste u bioinformatici, a koji su korišteni i u ovom radu.

Treće poglavlje sadrži teoretsku pozadinu problema sastavljanja genoma te definira i pobliže objašnjava tipove očitanja koja predstavljaju problem za ispravno i potpuno sastavljanje genoma.

U četvrtom poglavlju su opisani algoritmi i metodologije korištene za detekciju i korištenje problematičnih očitanja te računanje gornje granice u sastavljanju genoma.

Peto poglavlje sadrži implementacijske pojedinosti i upute za korištenje alata.

U šestom poglavlju su prikazani rezultati testiranja alata u cjelosti.

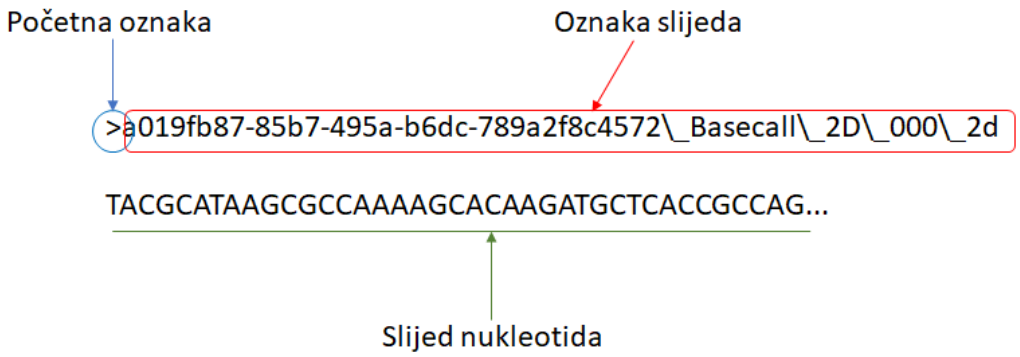
2. Podaci

Bioinformatički podaci uobičajeno se pojavljuju u nekoliko standardnih formata.

2.1. FASTA format

Najčešći oblik bioinformatičkih podataka jest FASTA [2] tekstualni format. FASTA tekstualni format prikazuje sljedove nukleotida ili nukleonskih kiselina pri čemu je svaki nukleotid ili nukleonska kiselina prikazan jednim slovom.

Svaka sekvenca u FASTA formatu sastoji se od dvije linije. Prva linija započinje znakom ">" nakon kojega slijedi identifikator sekvence. Važno je napomenuti da između znaka ">" i oznake slijeda ne smije biti razmak. U drugoj liniji zapisan je sami slijed.



Slika 2.1: Primjer sekvence u FASTA formatu

Primjer zapisa jedne sekvence u FASTA formatu prikazan je na slici 2.1.

U FASTA datotekama se pojavljuju i komentari koji počinju znakom ",", ali su vrlo rijetki.

Kako bismo znali interpretirati sadržaj FASTA datoteke nužno je znati koji se kodovi mogu pojaviti u FASTA datotekama te što oni znače.

U tablici 2.1 prikazani su kodovi koji su pojavljuju u FASTA datotekama te njihovo značenje.

Tablica 2.1: Tablica kodova u FASTA formatu

Slovo reprezentant	Značenje
A	Adenin
C	Citozin
G	Gvanin
T	Timin
U	Uracil
R	Adenin ili gvanin
Y	Citozin, timin ili gvanin
K	Gvanin, timin ili uracil
M	Adenin ili gvanin
S	Citozin ili gvanin
W	Adenin, timin ili uracil
B	Ne adenin
D	Ne citozin
H	Ne gvanin
V	Niti timin niti uracil
N	Adenin, citozin, gvanin, timin, uracil
X	Maskiranje
-	Procjep neodređene duljine

Osim u FASTA formatu, bioinformatički podaci se često pojavljuju i u FASTQ formatu.

2.2. FASTQ format

FASTQ[2] format koristi se za prikaz podataka na izlazu uređaja za sekvencioniranja. Iako vrlo sličan FASTA formatu, FASTQ format dodatno enkapsulira podatke o kvaliteti svakog očitanja.

Svaki zapis u FASTQ datoteci sastoji se od četiri linije. Prva linija započinje znakom „@“ nakon koje slijedi identifikator slijeda. U drugoj liniji nalazi se sami slijed. Treća linija započinje znakom „+“ nakon kojeg opcionalno slijedi identifikator slijeda i u četvrtoj liniji nalazi se vrijednost kvalitete slijeda iz druge linije.

```
@SEQ_ID
GATTTGGGGTTCAAAGCAGTATCGATCAAATAGTAAATCCATTTGTTCAACTCACAGTTT
+
!''*((((***+))%%%++)(%%%%).1***-+*''))**55CCF>>>>>>CCCCCCC65
```

Slika 2.2: Primjer sekvence u FASTQ format

Na slici 2.2 prikazan je primjer sekvence u FASTQ formatu. Dok je veći dio zapisa ekvivalentan FASTA formatu, u zadnjoj liniji nalazi se, kao što je već spomenuto, vrijednost kvaliteta slijeda. Vrijednost kvalitete je dugačka koliko i slijed, a kvaliteta svake baze je označena jednim bajtom i poprima vrijednosti između 33 što odgovara ’!’ u ASCII kodu i 126 što odgovara ’~’ u ASCII kodu.

Vrijednosti u rasponu od 33 do 126 predstavljaju vrijednost Sangerovog formata kvalitete definiranog kao:

$$Q_{Sanger} = -10 \log_{10} p \tag{2.1}$$

gdje je p vjerojatnost pogrešnog očitanja.

Vrijednosti Sangerovog formata kvalitete inače poprimaju vrijednosti u rasponu između 0 i 93, ali prilikom preslikavanja u ASCII kod raspon pomičemo kako bismo izbjegli kontrolne znakove.

Posljednji format datoteka koji ne sadrži originalne bioinformatičke podatke, ali je izuzetno važan u kontekstu ovoga rada jest PAF format.

2.3. PAF format

PAF[3] format tekstualni je format koji opisuje aproksimativne pozicije mapiranja između dva seta sekvenci. Sastoji se od ovih informacija odijeljenih tabulatorom:

Tablica 2.2: Opis PAF formata

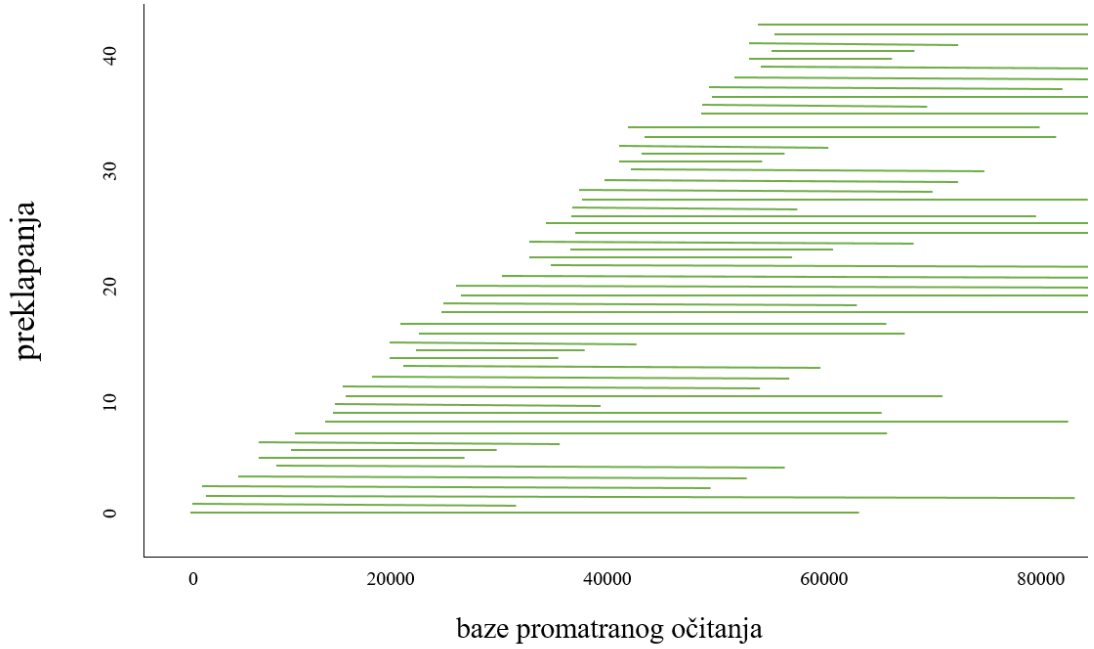
Stupac	Tip	Opis
1	string	Naziv očitanja
2	int	Duljina očitanja
3	int	Početna pozicija mapiranja očitanja
4	int	Krajnja pozicija mapiranja očitanja
5	char	Relativni slijed "+" ili "-"
6	string	Naziv reference
7	int	Duljina reference
8	int	Početna pozicija mapiranja očitanja na referenci
9	int	Krajnja pozicija mapiranja očitanja na referenci
10	int	Broj poklapanja očitanja i reference
11	int	Ukupan broj poklapanja, promašaja, umetanja i brisanja u poravnanju
12	int	Kvaliteta mapiranja

PAF format moguće je koristiti za različite potrebe, ali u kontekstu našeg rada korišten je za ekstrakciju i korištenje informacija o pozicijama i broju mapiranja očitanja na referentni genom.

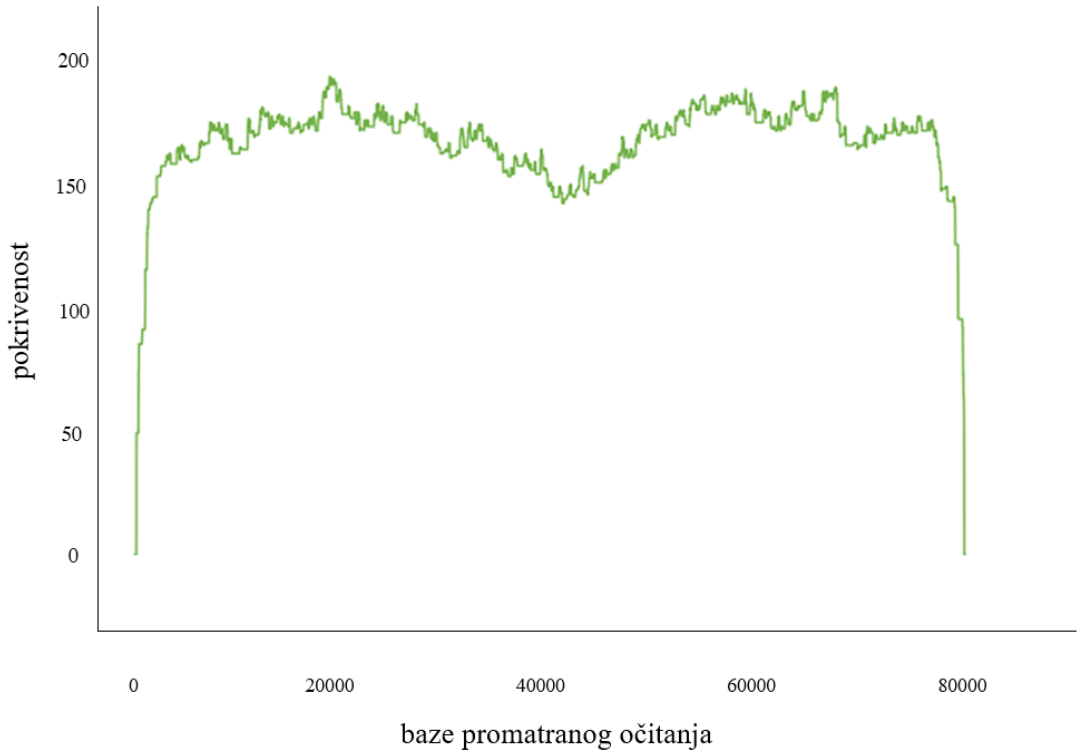
3. Sastavljanje genoma

Sastavljanje genoma smatra se jednim od najsloženijih problema u području bioinformatike. Pripada skupu NP teških problema i zbog toga se u rješavanju koriste brojne heurističke metode. Bioinformatički podaci su kompleksni, a dodatnu otežavajuću okolnost prilikom sastavljanja genoma predstavljaju specifična, neuobičajena i nejednoznačna očitanja. S obzirom da se najčešće koristi *shotgun* sekvencioniranje opisano u uvodu ovog rada, nastaju razne greške na očitanjima. S obzirom na vrstu greške ili sveukupnu kvalitetu očitanja, dijelimo ih na nekoliko skupina: ponavljajuća, kimerna, nekvalitetna i pravilna očitanja. Svaka od ovih vrsta opisana je u zasebnom poglavlju. Identificiranje ovih pogrešaka, odnosno vrste očitanja od velike je važnosti i jedan je od primarnih ciljeva alata opisanog u ovom radu.

Kod razlučivanja vrste očitanja od velike su koristi takozvani grafovi pokrivenosti (eng. *pile-o-gram* [4]). Graf pokrivenosti je sredstvo koje daje bolji uvid u problem sastavljanja genoma. Izrađuju se tako da se nad jednim očitanjem naslažu sva očitanja koja se s njim preklapaju, odnosno koja s referentim očitanjem dijele značajan dio baza u nizu, na cijelom ili samo dijelu očitanja. Ovaj dio prikazuje slika 3.1. Tako se nad svakom bazom očitanja nalazi broj drugih očitanja koja se preko nje preklapaju. Zbrajanjem svih očitanja koja se nalaze iznad te baze se dobiva pokrivenost baze (eng. *coverage*). Tako nastaje dvodimenzionalni signal ovisno o pokrivenosti svake baze prikazan na slici 3.2. Izrada i analiza grafova pokrivenosti vremenski je veoma skupa, ali korisna za upoznavanje s očitanjima. Ovisno o vrsti očitanja, graf pokrivenosti će imati specifičan izgled, što je objašnjeno u poglavljima za svaku vrstu očitanja.



Slika 3.1: Naslagivanje preklapajućih očitanja nad referentnim očitanjem

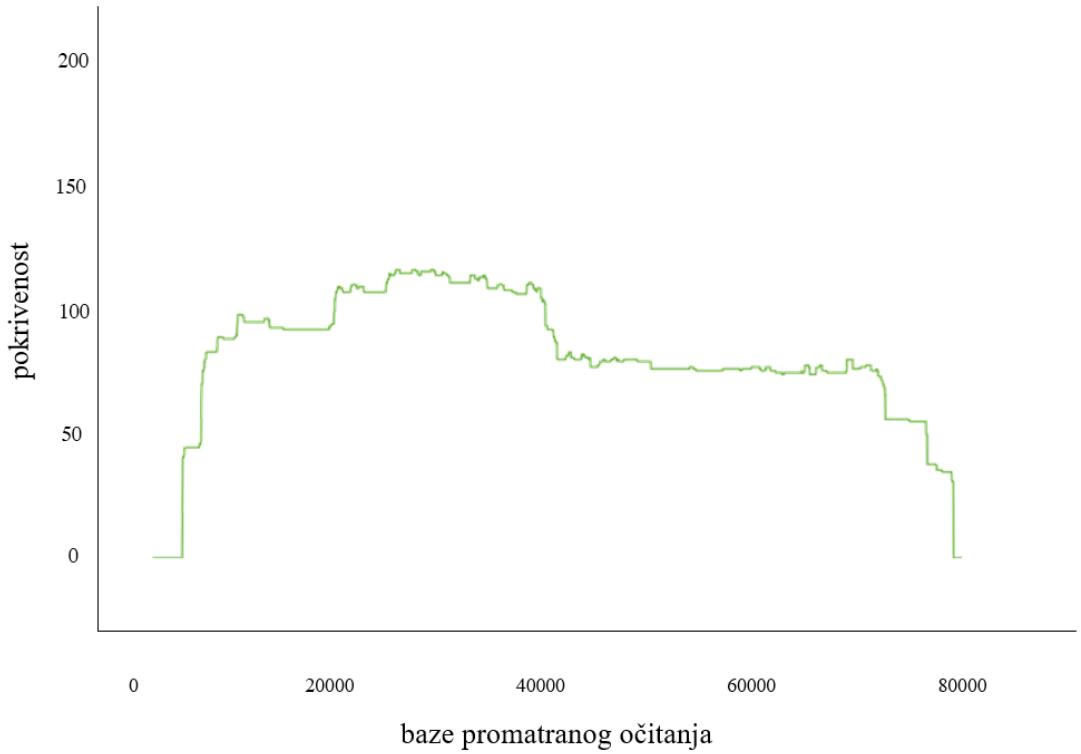


Slika 3.2: Primjer grafa pokrivenosti

3.1. Ponavljajuća očitanja

Ponavljajućim očitanjima smatramo ona očitanja koja se mogu mapirati na više mjesta u referentnom genomu zbog postojanja dugačkih istih ili vrlo sličnih sljedova u genomu.

Graf pokrivenosti ponavljajućeg očitanja ima neke karakteristike po kojima ga je moguće izdvojiti od ostalih očitanja.

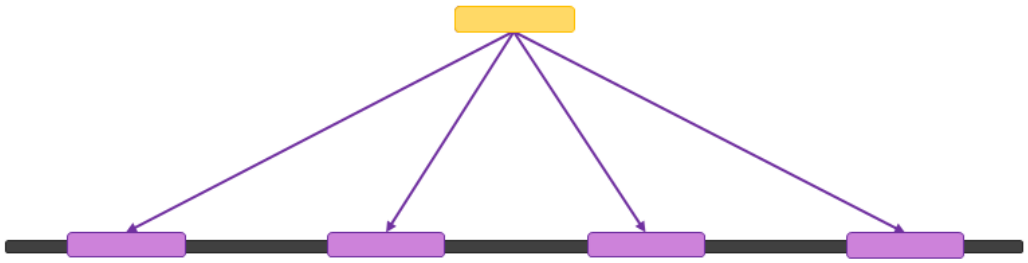


Slika 3.3: Primjer grafa pokrivenosti ponavljajućeg očitanja

Na slici 3.3 je prikazan primjer grafa pokrivenosti koji je karakterističan za ponavljajuće očitanje. Ono što ga čini takvim jest činjenica da distribucija pokrivenosti baza nije uniformna, odnosno postoje područja sa značajno većom pokrivenosti od ostatka očitanja. Znajući da ponavljajuće regije očitanja imaju višestruka mapiranja na referencu, posljedično slijedi da će na ponavljajuću regiju očitanja biti moguće mapirati puno više ostalih očitanja u odnosu na pravilni dio očitanja što je vidljivo i iz grafa pokrivenosti. U slučaju očitanja za koji je izrađen graf pokrivenosti na slici 3.3, postoje jedno područje s vidljivo većom pokrivenosti od ostatka očitanja, a proteže se između 20 000. i 40 000. baze na očitanju.

Regije na referenci na koje je moguće mapirati više očitanja nazivaju se ponavljajuće regije, a dvije su osnovne vrste ponavljajućih regija.

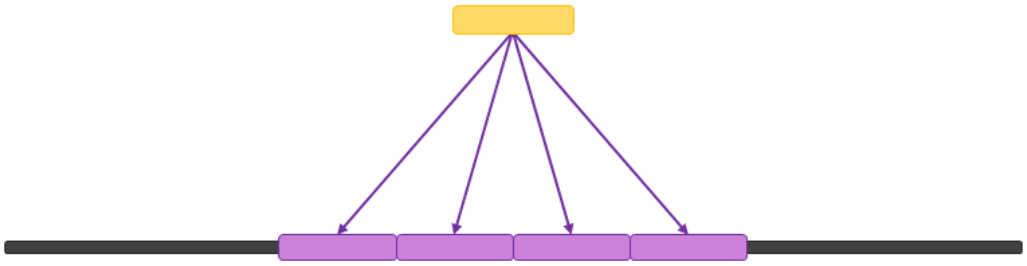
Prva vrsta ponavljajućih regija su raspršene (en. *interspersed*) regije koje se nalaze na udaljenim područjima u referentnom genomu te nisu ni na koji način susjedne jedna drugoj. Raspršene ponavljajuće regije najčešće nastaju kao posljedica takozvanih *transposona*, odnosno dijelova DNA koji, uz pomoć enzima, imaju mogućnost umetanja na određena mjesta u kromosomu.



Slika 3.4: Primjer raspršene ponavljajuće regije

Na slici 3.4 je ilustriran primjer raspršene ponavljajuće regije. Kao što je vidljivo, na referenci postoji nekoliko nesusjednih ponavljajućih područja na koje je moguće mapirati jedno očitanje što čini takvo očitanje ponavljajućim očitanjem, a ta područja raspršenim ponavljajućim regijama.

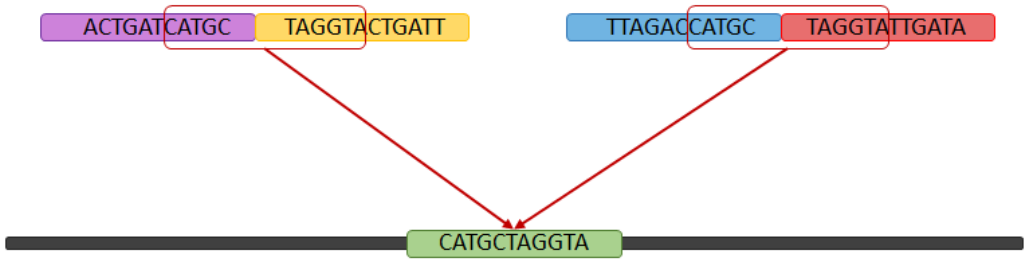
S druge strane, postoje ponavljajuće regije koje se na referenci pojavljuju jedna za drugom, grupirane na jednoj većoj regiji. Takve ponavljajuće regije nazivaju se *tandem* [5] ponavljajuće regije, a nastaju kao posljedica raznih bioloških mehanizama te se pojavljuju u različitim duljinama i brojnostima unutar genoma.



Slika 3.5: Primjer tandem ponavljajuće regije

Primjer tandem ponavljajuće regije prikazan je na slici 3.5 gdje je vidljiva jedna velika regija na referenci koja se sastoji od uzastopnih regija na koje se mapira isto očitanje što čini to očitanje ponavljajućim, a regiju tandem ponavljajućom.

Gore objašnjen *shotgun* pristup sekvenciranju cijelog genoma u kombinaciji s ponavljajućim regijama genoma, posebice onima duljima, stvara potencijalne probleme u sastavljanju genoma. Ako se prilikom *shotgun* sekvencioniranja generira ponavljajuća regija koju nije moguće jednoznačno premostiti sekvencioniranim očitanjima, odnosno kada ponavljajuća regija nije dovoljno malena da ju neko očitanje u potpunosti sadrži, nastaje nejednoznačnost u procesu sastavljanja genoma.



Slika 3.6: Primjer nepremostive ponavljajuće regije

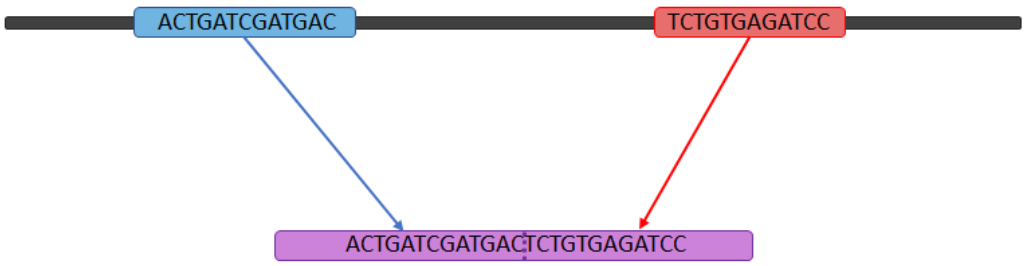
Primjer ponavljajuće regije koja je nepremostiva prikazan je na slici 3.6. Vidimo da je na regiju na referenci moguće mapirati više kombinacija očitanja što u procesu sastavljanja genoma rezultira različitim finalnim genomima.

U nekim slučajevima su ponavljajuće regije toliko velike da u potpunosti sadrže neka očitanja pa možemo zamisliti da se između krajnjih očitanja ponavljajuće regije na slici 3.6 nalazi još manjih očitanja što dodatno otežava proces sastavljanja genoma.

Iz tih razloga vrlo je bitno kvalitetno detektirati ponavljajuća očitanja.

3.2. Kimerna očitanja

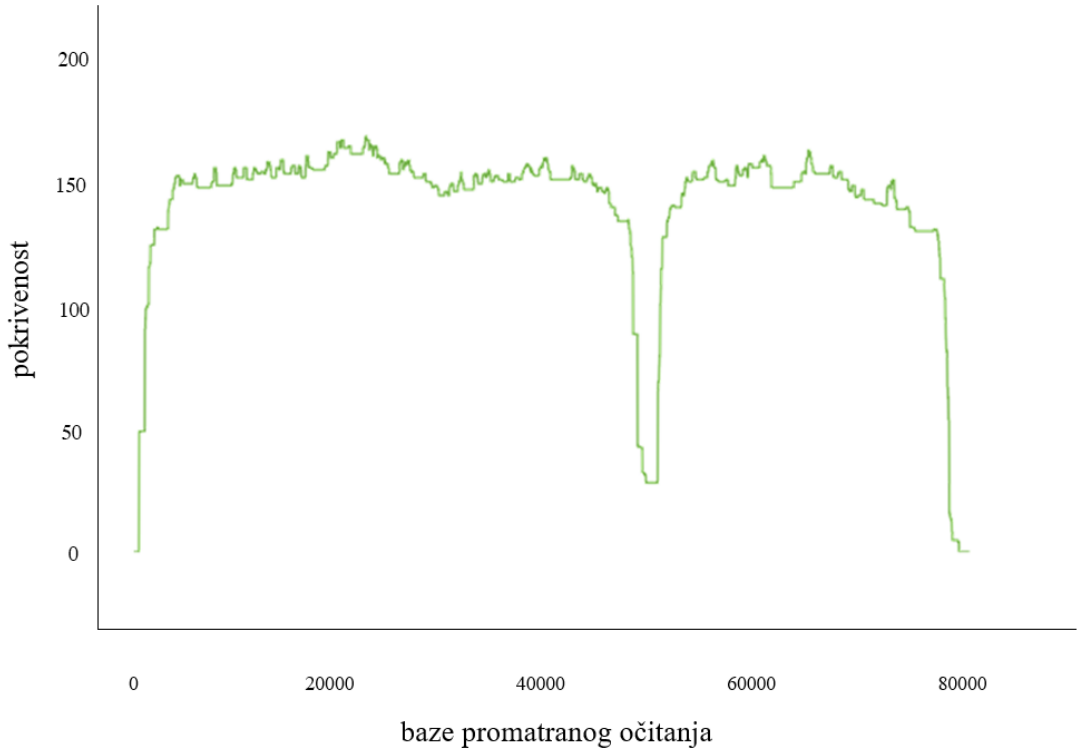
Kimernim očitanjima (*kimera – jedinka sastavljena od dijelova različitih rasa*) smatramo ona očitanja koja nastaju pogrešnim spajanjem dva očitanja koja predstavljaju u prirodi veoma udaljene dijelove genoma. Oni nastaju kao posljedica grešaka prilikom identificiranja nukleotida u očitavanjima. Zbog brzine kojom uređaji novijih generacija to rade događaju se greške u očitanjima zbog kojih se pojavljuju lažna preklapanja između prirodno vrlo udaljenih očitanja te se kao posljedica ta očitanja spajaju u jedinstveno očitanje koje bi trebalo predstavljati jedan dio genoma, ali zapravo predstavlja dvije vrlo udaljene regije genoma.



Slika 3.7: Primjer kimernog očitanja

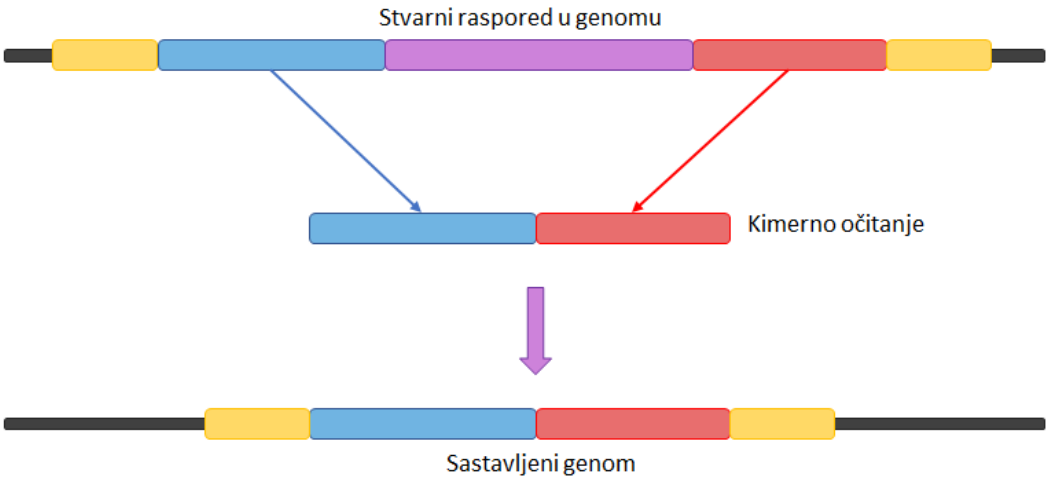
Na slici 3.7 prikazan je primjer kimernog očitanja koje originalno kao takvo ne postoji već se sastoji od dva odvojena očitanja mapirana na različitim regijama referentnom genoma, ali se uslijed pogreške prilikom sekvencioniranja čini kao jedinstveno očitanje. Takvo očitanje stvara značajne probleme u procesu izgradnje reprezentativnog genoma.

I kimerno očitanja, kao i ponavljajuće, ima karakteričan graf pokrivenosti po kojem ga je moguće izvodvojiti od ostalih očitanja.



Slika 3.8: Primjer grafa pokrivenosti kimernog očitanja

Primjer grafa pokrivenosti za kimerno očitanje prikazan je na slici 3.8. Ono što je karakteristično za kimerno očitanje, a što se manifestira i na pripadajućem grafu pokrivenosti jest činjenica da je kimerno očitanje sintetičko očitanje nastalo iz dva pravilna očitanja. Iz tog razloga, analizirajući pokrivenost baza očitanja, nailazimo na područja koja imaju značajno manju pokrivenost u odnosu na ostatak očitanja. Na tom području nalazi se prekid u očitanju, odnosno na tom području nalazi se granica između pravilnih očitanja enkapsuliranih u kimernom očitanju na koju je vrlo teško mapirati značajan broj očitanja jer je slijed baza nekarakterističan za pripadajući genom. Očitanje za koje je izrađen graf pokrivenosti prikazan na slici 3.8 ima prekid oko pozicije 50 000 i sastoji se od dva očitanja duljina 50 000 baza i 30 000 baza.

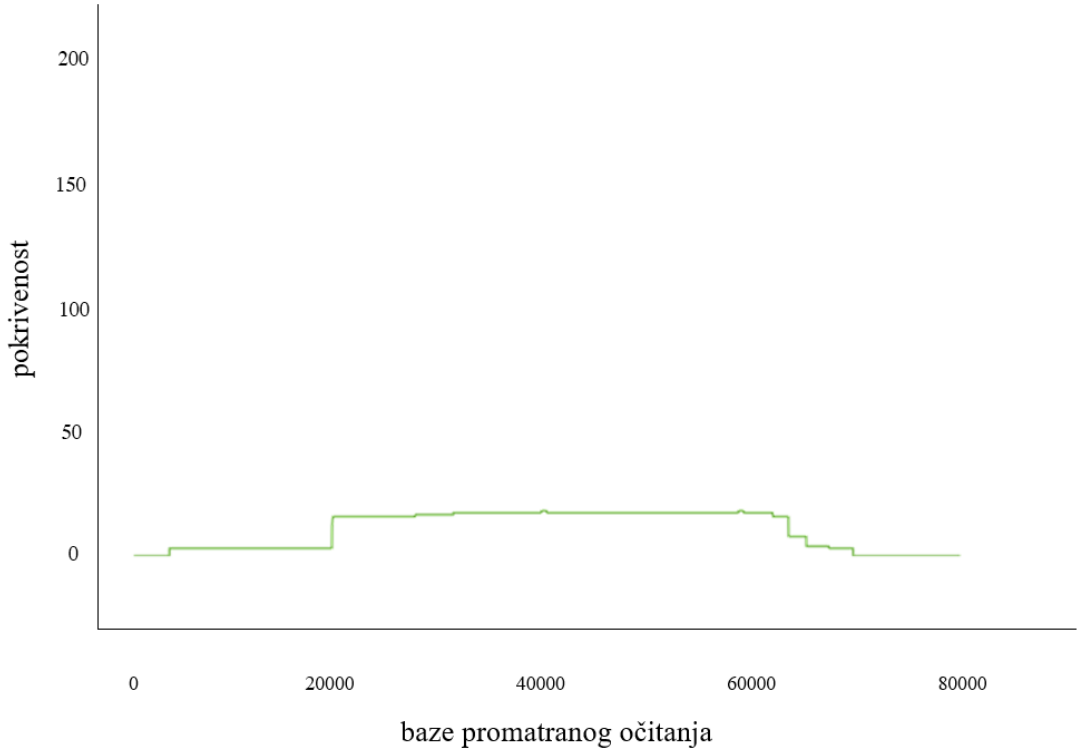


Slika 3.9: Pogrešno sastavljanje genoma s kimernim očitanjem

Na slici 3.9 prikazan je utjecaj kimernog očitanja na konačni rezultat sastavljanja genoma. Obzirom da se genom sastavlja na način da se redom traže očitanja koja se preklapaju (kraj prethodnog očitanja se preklapa s početkom idućeg očitanja) regija genoma koja se nalazi između stvarnih očitanja koja su spojena u jedno kimerno očitanje će biti u potpunosti preskočena, a očitanja koja se mapiraju na toj regiji će biti ili u potpunosti preskočena ili greškom mapirana na nekoj drugoj regiji genoma. Tako sastavljen genom uvelike će se razlikovati i po duljini i po sadržanoj informaciji od originalnog genoma stoga je vrlo važno uspješno detektirati i kimerna očitanja.

3.3. Nekvalitetna očitanja

Očitanja koja sadrže dosta pogrešaka većim dijelom svoje dužine nazivamo nekvalitetnim očitanjima (eng. *junk*). Budući da je dosta baza takvog očitanja zapravno neispravno, neće imati preklapanja s drugim očitanjima. Stoga će pokrivenost gotovo cijelog očitanja biti dosta niska, baš kako prikazuje slika 3.10. Identificiranje ove vrste od velike je važnosti kako se ne bi koristila u rekreiranju genoma.



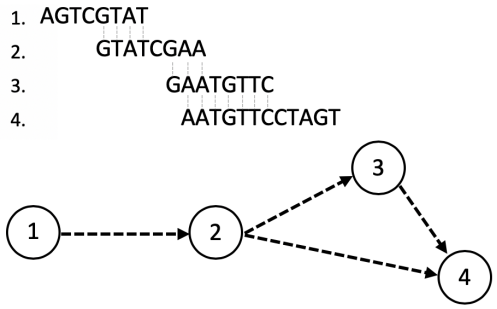
Slika 3.10: Primjer grafa pokrivenosti nekvalitetnog očitanja

3.4. Pravilna očitanja

Kada bi očitanja genoma bila potpuno ispravna onda bi ona dijelila iste nizove baza na nekim svojim dijelovima. Točnije jedno očitanje bi se preklapalo s velikim brojem drugih očitanja. Također svaka baza bi imala veoma sličnu i visoku pokrivenost. Očitanja visoke i generalno jednolične pokrivenosti na cijeloj svojoj dužini nazivamo pravilnim očitanjima (eng. *regular*). Graf pokrivenosti pravilnog očitanja prikazan je na slici 3.2. U idelanom slučaju sva bi očitanja bila ove vrsta i s njima bi se najkvalitetnije rekreirao željeni genom.

3.5. Graf sastavljanja

Dva niza znakova (u našem slučaju baza) mogu se preklapati na različite načine. Ako niz n može biti mapiran na podniz niza m, tada kažemo da niz m sadrži niz n. Kada se kraj niza n mapira na početak niza m, tada kažemo da se nizovi m i n preklapaju. Ovaj slučaj ćemo dalje razmatrati te koristiti u kreiranju grafa. Čvorove grafa će predstavljati očitanja genoma, a usmjereni brid između dva čvora će postojati ako se ta dva očitanja preklapaju. Broj znakova u preklapanju predstavljati će duljinu brida. Jednostavan primjer grafa preklapanja prikazuje slika 3.11.

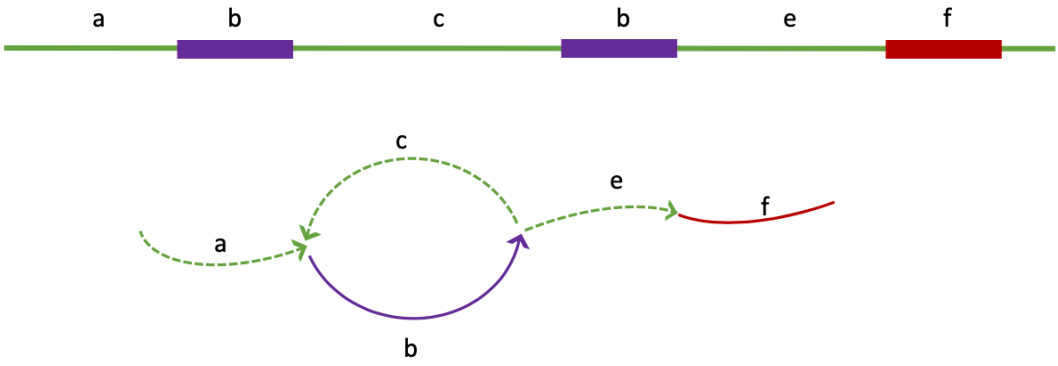


Slika 3.11: Jednostavan primjer grafa preklapanja

Možemo kreirati usmjereni graf $G = (V, E, l)$ koji ne sadrži višestruke bridove, gdje je V skup svih očitanja genoma, E su preklapanja očitanja, a l skup duljina preklapanja. Kako bi sastavljanje genoma pomoću ovog grafa bilo moguće mora biti zadovoljen jedan uvjet: niti jedno očitanje ne smije biti sadržano u drugome. Kraj grafa se definira čvor iz kojeg ne izlaze bridovi, u kojeg je usmjeren barem jedan brid. Početak je suprotno, čvor iz kojeg samo izlaze bridovi. Sastavljanje genoma nad ovako definiranim grafom je postupak u kojem je potrebno naći put od početka do kraja grafa, te će taj put predstavljati sastavljeni genom. Budući da ne mora postojati izlazni brid iz svakog čvora, tj. ne mora postojati očitanje kojim će se spojiti druga dva očitanja, moguće je da u grafu nastane više od jedne komponente. Genom tada neće biti moguće u potpunosti sastaviti, nego će biti sastavljena njegova dva ili više dijelova. Mjera koju uzimamo pri sastavljanju je pokrivenost, tj. koliki dio genoma je moguće sastaviti od jedne komponente. Za gornju granicu sastavljanja jednog kromosoma će se uzeti ona komponenta koja pokriva najveći dio kromosoma.

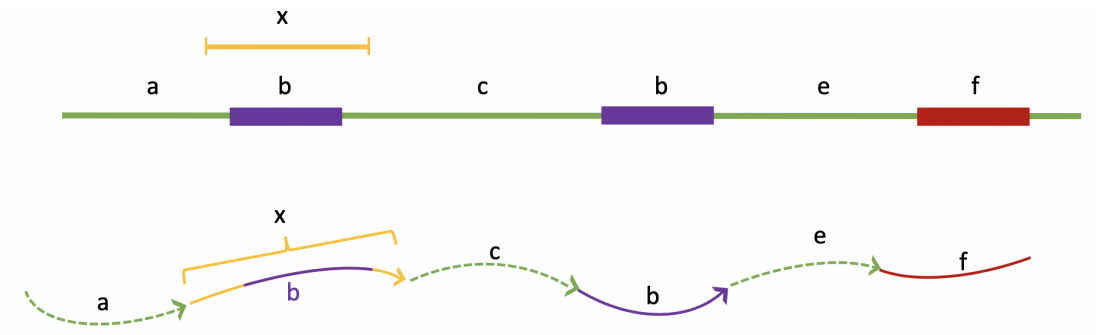
Problemi kod sastavljanja nastaju kada se u genomu pojavljuju regije jednakih nizova. Ponavljajuća očitanja opisana su u poglavlju 3.1. Ponavljajuće regije u genomu će se na grafu odraziti kao ciklusi u kojima pravi put ne može biti određen. Stoga se početak ciklusa tretira kao prekid u sastavljanju genoma, a sastavljanje se nastavlja

nakon ciklusa, baš kao da očitanja na tom dijelu nikad nije ni bilo. Iz teorije grafova, Hamiltonov ciklus je put koji prolazi svim vrhovima točno jednom. Kako bi sastavili genom mora postojati jedinstveni Hamiltonov ciklus kroz graf sastavljanja. Ako postoje ponavljajuće regije, takav ciklus ne postoji tj. nije jedinstven.



Slika 3.12: Repetitivne regije u genomu i njihov prikaz u grafu

Rješenje ovog problema postiže se premošćivanjem. Potrebno je naći očitanje koje će se na genom mapirati prije ponavljajuće regije, a završiti poslije nje. Dakle, ako je ponavljajuća regija duljine l, kako bi ona bila premošćena mora postojati očitanje koje je duljine m, tako da je $m > l$ i da očitanje sadrži tu ponavljajuću regiju. Ako postoji takvo očitanje onda je moguće sa sigurnošću tvrditi koji put u grafu je potrebno odabrati u njegovom obilasku, kako je prikazano na slici 3.13. Nije potrebno premostiti sve ponavljajuće regije nego samo dok ne ostane najviše jedna neke vrste tj. ponavljajuća regija ne smije imati svog para drugdje na genomu koji nije premošten.



Slika 3.13: Premošćivanje ponavljajuće regije

4. Metode

4.1. Priprema podataka alatom minimap2

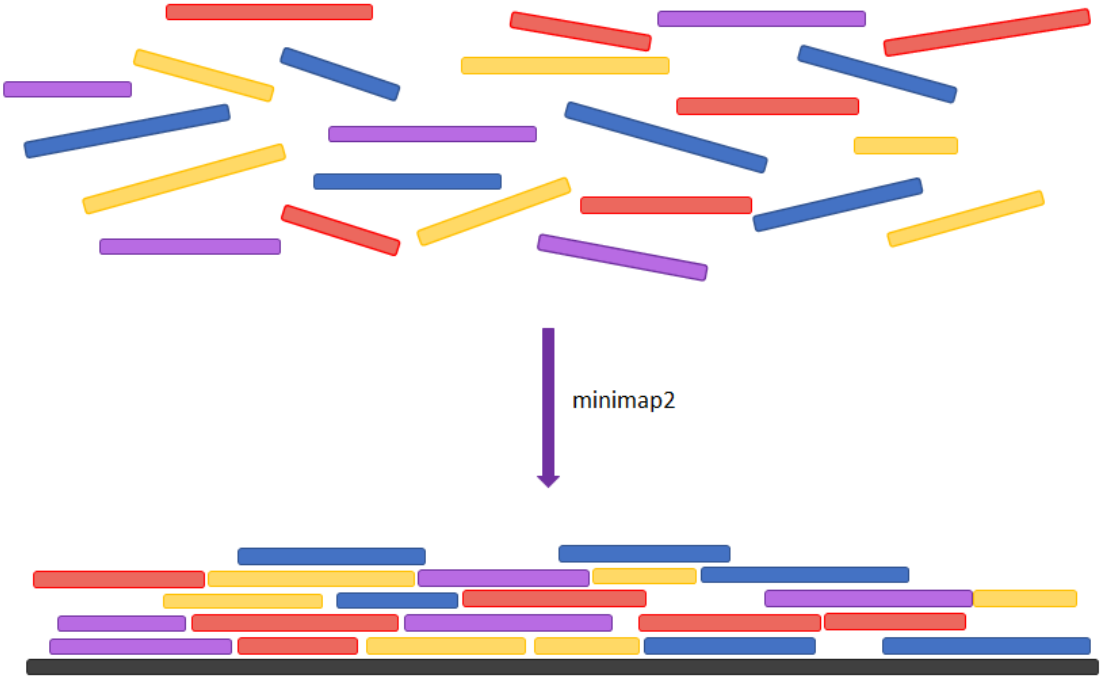
Minimap2 svestrani je alat korišten za mapiranje DNA ili RNA sekvenci na referentnu bazu podataka.

Nekoliko je najčešćih tipova korištenja alata minimap2:

- mapiranje PacBio ili Nanopore očitanja na ljudski genom
- traženje preklapanja između dugačkih očitanja
- *splice-aware* poravnanje na referentni genom
- poravnanje jednostrukih ili uparenih Illumina očitanja
- *assembly-to-assembly* poravnanje
- poravnanje cijelih genoma srodnih vrsta

U kontekstu ovog rada, minimap2 se koristi za mapiranje kratkih očitanja u FASTA formatu na referentnu bazu u FASTQ formatu. Izlaz alata minimap2 jest index mapiranja u PAF formatu iz kojeg je moguće iščitati koje se očitanje mapira na koji dio reference te pomoću informacija o mapiranjima razlikovati ponavljajuća, kimerna, nekvalitetna i pravilna očitanja.

Sekvencionirana očitanja možemo zamisliti kao skup elemenata koji moramo pozicionirati u nekom prostoru. Prostor u kojem pozicioniramo naša očitanja jest referentni genom, a minimap2 je algoritam kojim određujem koordinate određenog očitanja u referentnom prostoru. Ilustracija mapiranja očitanja na referentni genom prikazana je na slici 4.1.



Slika 4.1: Mapiranje očitanja alatom minimap2

Kako bi korištenje rezultata minimap2 u svrhu detekcije kimernih i ponavljajućih očitanja bilo što razumljivije, ovo je kratak pregled algoritma minimap2:

1. za **I** referentnih baza se pronađu takozvani *minimizeri* - specifični podnizovi preddefinirane duljine **k** koji služe za detekciju sličnih regija među dvjema sekvencama - i spremaju u indeks u obliku tablice raspršenog adresiranja

2. čita se **K** baza sekvenci i za svaku se ponavljaju koraci 3 do 7

3. svaki minimizer očitanja provjerava se u indeksu minimizera referentne baze i ako se ne nalazi među preddefiniranih **f** najčešćih, skupljaju se njegova pojavljivanja u referenci te se ona nazivaju *seeds*

4. *seeds* sortiraju se po poziciji na referenci te se dinamički povezuju u lance

5. za svaki lanac, počevši od najboljeg po njegovoj lančanoj vrijednosti, provjerava se preklapanje s lancima u postojećem (početno praznom) setu primarnih mapiranja; ako je preklapanje manje od preddefinirane vrijednosti **mask-level** dodaje se u set primarnih mapiranja, inače se deklarira kao sekundarno mapiranje lancu s kojim ima preklapanje veće od zadane vrijednosti

6. uzimaju se sva primarna mapiranja i **N** najboljih sekundarnih očitanja koja imaju lančanu vrijednost veću od **p** posto odgovarajućeg primarnog mapiranja

7. filtriraju se oni *seedovi* koji vode do velikih umetanja i brisanja prilikom poravnanja te se nad preostalima vrši globalno poravnanje; lanci se dijele na manje ako vrijednost poravnanje padne za **z**, zanemarujući dugačke praznine te se lanci i njihova mapiranja ispisuju u odgovarajućem formatu (u našem slučaju PAF formatu)

8. ako postoji više sekvenci vraća se na korak 2 dok god postoji još neobrađenih sekvenci

9. ako postoji više referentnih sekvenci ponavlja se cijeli postupak dok god sve referentne sekvence nisu obrađene

4.2. Genetski algoritam

Genetski algoritam je metoda optimiranja koja imitira prirodni evolucijski proces. S obzirom na to da je potrebno podesiti parametre pomoću kojih se određuje vrsta očitanja, iskorišten je genetski algoritam kako bi se dobili najbolji rezultati. Potencijalno rješenje problema je predstavljeno pomoću jedinke koju nazivamo kromosom. Kromosom se sastoji od gena, gdje je gen jedan parametar koji je potrebno optimirati. Stoga je kromosom zapravo niz parametara koji su moguće rješenje. Svako rješenje ima svoju kvalitetu koju nazivamo dobrota. Koliko dobro kromosom, tj. niz parametara rješava problem, njegova dobrota će biti veća. Skup kromosoma naziva se populacija. Najbolji kromosomi iz populacije se odabiru te se iz njih križanjem stvaraju nove jedinke i nova populacija. Konačno se primjenjuje mutacija kako bi se uvela raznovrsnost među populacijom. Pseudokod je prikazan algoritmom 1.

Genetski algoritam odvija se u nekoliko faza. Prva faza je kreiranje populacije. Populaciju čini skup jedinki tj. kromosoma, u ovom slučaju njih osam. Svaka jedinka nastaje tako da mu se za svaki parametar pridruži nasumična vrijednost. Jednom tako nastale jedinke čine početnu populaciju. S obzirom na to da se kromosomi međusobno razlikuju, ovaj način se naziva neuniformno generiranje. U slučaju kada su sve jedinke na početku iste, a poslije nastaju nove uslijed mutacije, radi se o uniformnom generiranju.

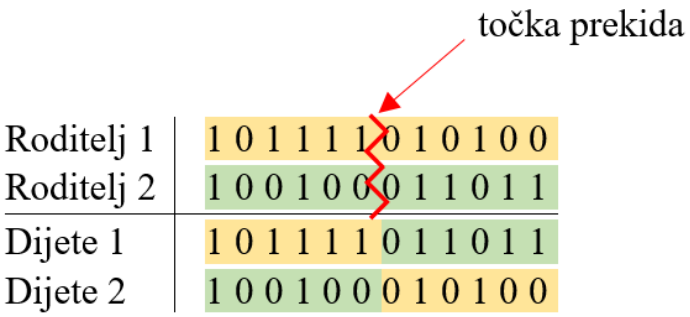
Sljedeći korak genetskog algoritma je selekcija (eng. *selection*). U ovom koraku se odabiru jedinke trenutne populacije iz kojih će se kreirati nove jedinke koje će činiti

Algorithm 1 Genetski algoritam

$P := generirajPopulaciju()$
$m := brojIteracija$
while $i < m$ **do**
  Odaberi jedinke iz populacije P – **selekcija**
  Krizaj odabrane jedinke – **križanje**
  Mutiraj nastalu djecu – **mutacija**
  U P zamijeni jedinke novonastalom djecom
end while
return najbolja jedinka u populaciji P

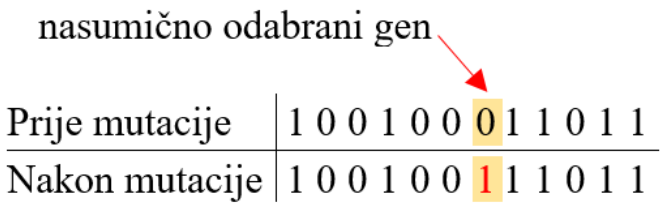
sljedeću generaciju populacije. Postoji nekoliko načina selekcije. Ovdje je korištena takozvana K-turnirska selekcija. To je rangirajuća vrsta selekcije koja za generiranje nove populacije uzima k jedinki iz stare populacije, odabire po dobroti najbolje jedinke od tih k te njih stavlja u bazen za reprodukciju. U slučaju ovog rada se odabiralo 3 nasumične jedinke koje su zatim bile sortirane po dobroti kako bi dvije najbolje bile korištene za reprodukciju.

Križanje (eng. *crossover*) je sljedeća faza algoritma. Iz napravljenog reprodukcijskog bazena se uzimaju dvije jedinke te se njihovi geni miješaju. Odabire se nasumična točka prekida nakon nekog gena u kromosomu, geni prije te točke se uzimaju od jedne jedinke, a geni nakon te točke od druge jedinke. Tako nastaje dvoje nove djece koji imaju dio gena od jednog roditelja, a dio od drugog roditelja. Dakle, iz dvije trenutne jedinke nastaju dvije nove. Ovaj postupak prikazuje slika 4.2.



Slika 4.2: Križanje roditelja kako bi nastala djeca s novim svojstvima

Konačan korak je mutacija. U jedinki se nasumično odabere jedan gen te mu se vrijednosti promijeni. Šansa za mutacijom nije stopostotna, nego se uzme neka vjerojatnost da će se ona dogoditi. U ovom konkretnom slučaju vjerojatnost da dođe do mutacije je 70%. Mutacijom se unosi novo svojstvo u populaciju ili se obnavlja izgubljeno svojstvo prilikom pretraživanja. Postupak je prikazan slikom 4.3.



Slika 4.3: Mutacija

Jednom kada su stvorene nove jedinke križanjem, a nad njima je primijenjena eventualna mutacija one mijenjanju trenutne jedinke u populaciji. Iz trenutne populacije se sačuva određeni broj najboljih kromosoma, a sv ostali se zamijene novim. Tako je nastala nova populacija s istim brojem jedinki koje se ponovno evaluiraju. U ovom alatu samo najbolja jedinka ulazi u sljedeću populaciju a ostale se zamijene s novima. Proces se nastavlja do određenog broja iteracija ili dok trenutna populacija ne zadovolji određenu točnost.

4.3. Detekcija ponavljajućih i kimernih očitanja

Ponavljajuća očitanja, kao što je već spomenuto, karakteristična su po tome što ih je moguće mapirati na više mjesta u referentnom genomu. Alat minimap2 prilikom izgradnje indeksa mapiranja detektira sva višestruka mapiranja jednog očitanja.

Očitanje može biti ponavljajuće u dva smisla. U prvom slučaju očitanje ima jedinstveno mapiranje na referentni genom cijelom svojom duljinom (misli se na barem 90% duljine očitanja), ali sadrži i dio koji je ponavljajući i koji je moguće mapirati na više pozicija u referentnom genomu. U drugom slučaju je cjelokupno očitanje ponavljajuće, odnosno očitanje nema jedinstveno mapiranje cijelom svojom duljinom nego je cjelokupno očitanje moguće mapirati na više mjesta u referentnom genomu. Problem je rekurzivne prirode pa u svakoj ponavljajućoj regiji očitanje može postojati dodatna ponavljajuća regija čiji je skup mogućih mapiranja na referentnom genomu strogo veći od skupa mapiranja ponavljajuće nadregije.

Kako i u kojoj mjeri će ponavljajuće regije očitanja biti detektirane uvelike ovisi o koracima 5 i 6 gore objašnjenog algoritma alata minimap2. Naime, sekundarna mapiranja odgovarajućeg primarnog mapiranja nekog očitanja odgovaraju ponavljajućim regijama tog očitanja. Kada se u koraku 5 odrede sva primarna i sekundarna mapiranja očitanja, ovisno o hiperparametrima, selektiraju se sva primarna i određen broj najboljih sekundarnih mapiranja te se time na izlaz algoritma minimap2 proslijeđuje informacija o očitanjima za koja su detektirana višestruka mapiranja i koja su potencijalno ponavljajuća očitanja.

Dok ponavljajuća očitanja možemo zamisliti kao očitanja u kojem se učahurena nalaze manja očitanja, osnovna karakteristika kimernih očitanja je ta da između pravilnih dijelova kimernog očitanja nema "preklapanja". Obzirom da su kimerna očitanja nastala pogrešnim spajanjem dva odvojena očitanja u jedno, ona nemaju niti jedno potpuno mapiranje na referentni genom već se pravilni dijelovi kimernog očitanja mapiraju na potpuno drugačije dijelove reference. Za detekciju kimernih očitanja posebno je važan korak 7 gore objašnjenog algoritma alata minimap2 gdje se vrši korekcija lošijih mapiranja na način da se mapiranja s velikim prazninama između najboljeg poravnanja razdvajaju na više manjih mapiranja. Nakon tog koraka više neće postojati jedinstveno mapiranje kimernog očitanja u njegovoj punoj duljini nego će kimerno očitanje biti mapirano "isjeckano".

Sve informacije o mapiranjima očitanja dostupne su u gore objašnjenom PAF formatu koji je na izlazu alata minimap2. Dakle, na samom početku nužno je parsirati PAF format kako bismo izvukli i pohranili informacije koje nam govore je li neko očitanje potencijalno kimerno ili ponavljajuće. Sva dostupna mapiranja spremljena u PAF formatu spremaju se u mapu kojoj je ključ identifikator očitanja, a vrijednosti pridružene ključu su sva pronađena mapiranja tog očitanja.

Nakon što su sva mapiranja svakog očitanja spremljena u mapu, traže se potencijalna ponavljajuća i kimerna očitanja, a to su ona koja u mapi mapiranja imaju više od jednog pronađenog mapiranja, odnosno ona očitanja koja je minimap2 mapirao na referencu na nejednoznačan način.

Uzimajući u obzir osnovne karakteristike ponavljajućih i kimernih očitanja te analizirajući početne i krajnje pozicije mapiranja očitanja, moguće je razdvojiti potencijalno ponavljajuća od potencijalno kimernih očitanja. Kao što je već pojašnjeno, ponavljajuće regije očitanja su podskupovi većih regija očitanja dok pravilni dijelovi kimernih očitanja nemaju presjeke mapiranja.

Uzmimo za primjer šesterostruko mapiranje jednog očitanja prikazano u tablici 4.1. Vidljivo je da očitanje ima jedno primarno mapiranje koje navedeno prvo u tablici. Ostala mapiranja su sekundarna i odnose se na istu regiju očitanja s početkom na poziciji 7822 i završetkom na poziciji 8577. Na osnovu pozicija mapiranja ovog očitanja lako je zaključiti da je riječ o ponavljajućem očitanju.

Tablica 4.1: Primjer mapiranja ponavljajuće sekvence

Početna pozicija na očitanju	Krajnja pozicija na očitanju	Početna pozicija na referenci	Krajnja pozicija na referenci
85	16679	1865475	1881092
7822	8577	1978527	1979239
7822	8577	257932	258644
7822	8577	19820	20532
7822	8577	3583458	3584170
7822	8577	1049808	1050520

S druge strane, možemo za primjer uzeti dvostruko mapiranje očitanje prikazano u tablici 4.2 koje je po mnogočemu drugačije od višestrukog mapiranja u tablici 4.1. Naime, ako pogledamo pozicije mapiranja na očitanju, vidimo da prvo i drugo mapiranje nemaju presjek. Osim toga pozicije mapiranja na referenci su jako udaljena što ukazuje na to da je riječ o kimernom očitanju.

Tablica 4.2: Primjer mapiranja ponavljajuće sekvence

Početna pozicija na očitanju	Krajnja pozicija na očitanju	Početna pozicija na referenci	Krajnja pozicija na referenci
17	3842	756995	760771
3884	15141	3320818	3331885

4.4. Potencijalni problemi prilikom detekcije kimernih i ponavljajućih očitanja

Nekoliko je slučajeva u kojima nije odmah vidljivo je li riječ o ponavljajućem, kimernom, nepravilnom ili pravilnom očitanju.

Prvi takav slučaj ima očitanje kao što je prikazano u tablici 4.3. Očitanje ima sedmerostruko mapiranje i to na način da nije u cjelosti mapirano na jednom dijelu reference nego je na dijelu od pozicije 177 do pozicije 2445 mapiran na jednom dijelu reference, a od pozicije 2531 do pozicije 3306 ima višestruka mapiranja po referenci. Dakle, ovo očitanje ima karakteristike i kimernog i ponavljajućeg očitanja zbog čega na njega treba obratiti posebnu pozornost. Obzirom da je ovo očitanje sintetički nastalo spajanjem dva pravilna očitanja i da kao takvo zapravo ne postoji, ovakvo očitanje klasificiramo kao kimerno očitanje bez obzira na ponavljajuću regiju. Ponavljajuća regija očitanja je zapravo samostalno očitanje koje ima karakteristike ponavljajućeg očitanja.

Tablica 4.3: Primjer mapiranja kimernog očitanja s ponavljajućom regijom

Početna pozicija na očitanju	Krajnja pozicija na očitanju	Početna pozicija na referenci	Krajnja pozicija na referenci
170	2445	1870705	1872977
2531	3306	3583433	3584182
2531	3306	257920	258669
2531	3306	19808	20557
2531	3306	1978515	1979264
2531	3306	1049783	1050532
2531	3306	279175	279924

Drugi slučaj vezan je uz prokariotske kromosome koji su po prirodi cirkularni. Obzirom da cirkularni kromosomi nemaju jednoznačan "početak" i "kraj" određuje se nasumična pozicija na kromosomu i kromosom se na toj poziciji sintetički "prereže". Očitanja koja se prirodno mapiraju na taj dio reference čine se kao kimerna očitanja mapirana na krajnje udaljene dijelove reference što je isključivo posljedica sintetičkog određivanja granica kromosoma te su ta očitanja u stvarnosti pravilna.

Primjer takvog mapiranja prikazan je u tablici 4.4. Vidimo da očitanje ima nepreklapajuća mapiranja i to na krajnje pozicije reference, ali u ovom slučaju je riječ o cirkularnoj referenci te je ovo očitanje lažno kimerno.

Tablica 4.4: Mapiranje očitanja na "rubove" cirkularnog kromosoma

Početna pozicija na očitanju	Krajnja pozicija na očitanju	Početna pozicija na referenci	Krajnja pozicija na referenci	Duljina reference
46	11545	4630203	4641628	4641652
11575	19354	6	7676	4641652

I treći problematičan slučaj odnosi se na kimerna očitanja. Postoje očitanja koja imaju višestruka nepreklapajuća očitanja te na prvi pogled zadovoljavaju uvjete da ih se proglasi kimernim očitanjima. Međutim, ako se obrati pozornost na pozicije mapiranja na referenci i razmake između različitih mapiranja može se uočiti da je razmak između mapiranja na očitanju otprilike jednak razmaku mapiranja na referenci. U tom slučaju očitanje zapravo nije kimerno već regularno, a njegovo mapiranje nije jedinstveno zbog nekakve pogreške, primjerice u procesu sekvencioniranje, zbog koje se manji dio očitanje lošije mapira na referencu.

I na kraju, postoje očitanja koja kao rezultat minimap2-ovog algoritma imaju više mapiranja koja djeluju kao kimerna, ali među tim mapiranjima postoji najveće mapiranje koje sadrži barem 90% duljine cijelog očitanja. Iako se takvo očitanje u početku čini kao kimerno, ono je zapravo regularno očitanje s manjom greškom zbog koje se sitan dio očitanja mapira na drugi dio reference.

4.5. Klasificiranje očitanja

Klasificiranje očitanja pomoću grafova pokrivenosti relativno je jednostavan postupak, ali isto tako i vremenski skup. Ovaj alat nastoji što bolje klasificirati očitanja bez uporabe grafova pokrivenosti koristeći podatke o mapiranju očitanja na referencu. Od velike pomoći su informacije dostupne iz PAF formata koji je opisan u poglavlju 2.3. Koristeći te informacije, ovaj način trebao bi biti značajno brži uz prihvatljivu točnost.

Nekvalitetna očitanja najteže je odrediti promatrajući samo mapiranja na referencu. Potrebno je najviše nagađanja. Očitanje je svrstano u ovu grupu ako postoji samo jedno mapiranje na genom i ako je zadovoljen barem jedan od sljedeća dva uvjeta. Prvi uvjet je da je broj baza korišten za mapiranje na genom manji od 50% duljine cijelog

očitanja. Točnije, broj baza između početne i krajnje pozicije mapiranja očitanja mora biti manji od polovice svih baza očitanja. Drugi uvjet je da je udio preklapanja u poravnanju očitanja manji od 27%.

Za klasificiranje kimernih očitanja prvo su sva mapiranja sortirana po početnoj poziciji mapiranja očitanja, a zatim po daljoj krajnjoj. Zatim se promatraju dva uzastopna mapiranja. Računaju se dva podatka, broj baza između početnih pozicija ta dva mapiranja na očitanju i razlika između početnih pozicija mapiranja na referenci. Ako je manja od ove dvije razlike manja od 88% veće razlike, očitanje je označeno kao kimerno.

Ponavljajuće očitanje određeno je tako da se ponovno promatraju sortirana mapiranja, kao i za kimerna očitanja. Ako postoje dva mapiranja takva da jednom krajnja pozicija na očitanju završava prije početne pozicije sljedećeg, to očitanje je kandidat za ponavljajuće očitanje. Prvo od ta dva mapiranja mora imati mapiranih barem 500 baza i zadovoljiti barem jedan od sljedećih uvjeta. Početak mapiranja na očitanju mora biti u prvih 5% očitanja, ili krajnja pozicija mora biti u zadnjih 5% očitanja ili pak mora biti duljine 1500 baza. Ako je zadovoljen i jedan od ova tri uvjeta, i duljina mapiranja je veća od 500 baza, očitanje je označeno kao ponavljajuće.

Ovdje navedeni parametri, poput duljine očitanja, broja poklapanja itd. nisu odabrani nasumično nego su pronađeni genetskim algoritmom opisanim u poglavlju 4.2.

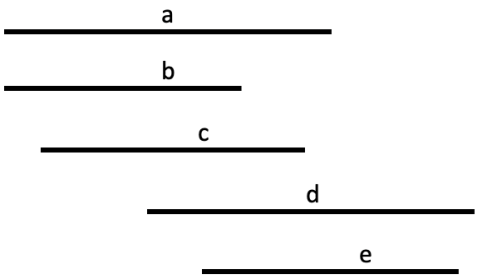
4.6. *Sweep line algoritam*

Da bi bilo moguće izgraditi graf sastavljanja, svako očitanje tj. vrh ne smije biti sadržan u drugom očitanju. Budući da se cijeli genom čita nekoliko puta takav slučaj se često pojavljuje u ulaznom skupu podataka. Stoga je jedan od koraka ovog alata da se ulazni skup pročisti od takvih očitanja. Prilikom mapiranja svakom očitanju je određen početak i kraj na referentnom genomu.

Naivni pristup ovom problemu bi bio da se svaka dva očitanja uspoređuju. Neka je l i r par očitanja za koje treba provjeriti je li jedno sadržano u drugom. Ako l počinje prije r te završava poslije r, očitanje r je sadržano u l i valjalo bi ga izbaciti iz skupa podataka. Ovaj pristup je ispravan, ali uz kvadratnu vremensku složenost, $O(n^2)$, koja na velikom skupu podataka može biti znatno osjetna na performansama alata.

Problem je moguće riješiti i drugim pristupom, tako zvanim algoritmom *Sweep line*. Algoritam se općenito koristi kada je potrebno naći elemente skupa koji su sadržani jedni u drugima. Njegova složenost je $O(nlogn)$. Prvo je potrebno sortirati sva očitanja po početnoj poziciji, a zatim po daljoj krajnjoj poziciji. Nakon toga se

za svako očitanje obilaze sva sljedeća dok god je krajnja pozicija sljedećeg očitanja manja od krajnje pozicije trenutnog. Sva tako posjećena očitanja je potrebno izbaciti. Kada se završi postupak za jedno očitanje, samo se prelazi na ono sljedeće po početnoj poziciji.



Slika 4.4: Prikaz sortiranih očitanja prije sweep line algoritma

Slika 4.4 prikazuje skup podataka nakon što je sortiran prvo po manjoj početnoj poziciji, a zatim po većoj daljnjoj poziciji. Prateći algoritam, prvo ćemo uzeti očitanje a i naći sva očitanje koja imaju kraj manji od njega. Takva očitanja će biti sadržana u a i treba ih izbaciti. Za ovaj slučaj to su očitanja b i c. Očitanje d ima kraj veći od a te tu treba stati i prijeći na iduće očitanje, koje je upravo d. Ono u sebi sadrži e pa ćemo i njega izbaciti. Algoritam 2 prikazuje pseudokod. Funkcija $inc(x)$ vraća sljedbenika elementa x u nizu kojem x pripada ako on postoji. U najgorem slučaju složenost je $O(nlogn)$ što je znatno bolje od kvadratne složenosti.

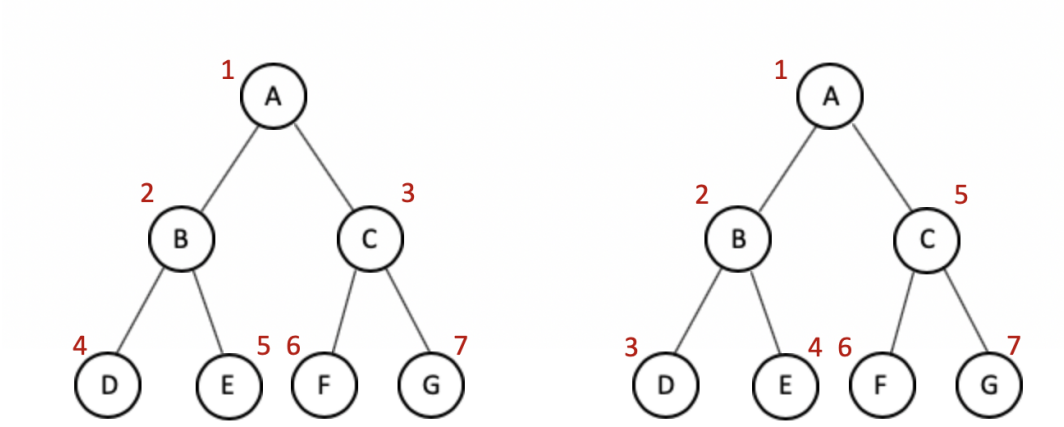
Algorithm 2 Sweep line algoritam

Ulaz: A – Skup očitanja.
Izlaz: Skup bez očitanja sadržanih u drugim očitanjima A.
$sort(A)$ *// sortiraj po početnoj, a zatim po krajnjoj poziciji*
for (i in A) **do**
  $j := inc(i)$ *// j = element koji slijedi i*
  while $j.end <= i.end$ **do**
    $to\_remove = j$
    $j := inc(j)$
    $A.remove(to\_remove)$
  end while
end for

4.7. Pretraživanje u dubinu

Graf sastavljanje vrlo se intuitivno može prevesti u stablo. Svako očitanje će predstavljati jedan čvor, a njegova djeca će biti sva očitanja u koja je taj čvor usmjeren, tj. sva očitanja koja se nastavljaju na odabrano očitanje. Sastavljanje genoma u ovako definiranom problemu zapravo se svodi na pretraživanje stabla od početnog čvora do onog konačnog.

Postoji nekoliko načina pretraživanja stabla, od kojih su vjerojatno najpoznatiji pretraživanje u širinu (eng. *Breadth-first search, BFS*) i pretraživanje u dubinu (eng. *Depth-first search, DFS*). Ideja pretraživanja u širinu je da se čvorovi obilaze razinu po razinu, s lijeva na desno. Tek kada su posjećeni svi čvorovi razine prelazi se na sljedeću razinu. Pretraživanje u dubinu je također jednostavno, prvo se obilazi najljevije neposjećeno dijete svakog čvora, te tako zapravo pretražujemo kroz sve razine dok ne dođemo do one zadnje.



Slika 4.5: Pretraživanje u širinu (lijevo) i pretraživanje u dubinu (desno)

Slika 4.5 prikazuje primjer pretraživanja za dva navedena algoritma. Redoslijed obilaženja čvorova kod pretraživanja u širinu na danom primjeru bi bio: ABCDEFG, a redoslijed kod pretraživanja u dubinu: ABDECFG. Faktor grananja kod stabla je broj djece koje čvorovi u stablu imaju i označava se sa b. Ako sa m označimo maksimalnu dubinu stabla, a sa d dubinu na kojoj prvi puta nalazimo rješenje, tada je vremenska složenost pretraživanja u dubinu $O(b^m)$. Vremenska složenost pretraživanja u širinu je $O(b^{d+1})$.

Za problem sastavljanja genoma pretraživanje u dubinu će nam biti sasvim dovoljno odnosno davat će i najbolje rezultate. Ova pretpostavka vrijedi jer ako je moguće naći put kroz graf onda ga je sigurno moguće naći uzimajući najdulja očitanja.

Uvijek će sljedeći čvor za obilazak biti ono dijete kojemu se kraj mapira najdalje na genomu. Tako će nam u stablu najljevije dijete biti zapravo ono najdalje na genomu koje slijedi trenutni čvor. Na ovaj način ćemo obići samo ona nužna očitanja i njih će biti minimalno. Jednom kada je nađen završni čvor, moguće se vratiti na početak prateći roditelje svakog posjećenog čvora do početnog stanja.

Algoritam 3 predstavlja pseudokod pretraživanja u dubinu. Funkcija $expand$ vraća svu djecu predanog čvora. Struktura koju koristi pretraživanje u dubinu je stog. Jedina razlika kod pretraživanja u širinu je što se u toj liniji inicijalizira red, a ne stog.

Algorithm 3 Pretraživanje u dubinu

Ulaz: S – Početni čvor u stablu.
Ulaz: G – Ciljni čvor u stablu.
$L := Stack()$
$L.push(S)$
while $lenght(L) > 0$ **do**
  $n := L.pop()$
  if $(goal(n))$ **then**
    return n
  end if
  for $(i \in expand(n))$ **do**
    $L.push(i)$
  end for
end while
return $fail$

5. Implementacija

Opisane metode implementirane su u programskom jeziku C++, a integracija algoritama i upute za korištenje alata opisane su u ovom poglavlju.

5.1. Integracija algoritama

Strukturu alata moguće je podijeliti u dva glavna dijela: označavanje vrste predanih očitanja i rekonstruiranje genoma. Na samom početku potrebno je pročitati predane datoteke u FASTA ili FASTQ formatu za što se koristi alat po imenu *Bioparser* [6]. Pomoću njega se svako očitanje ili dio reference sprema u strukturu koju je kasnije lako koristiti kako bi se dobio naziv, duljina ili niz baza tog očitanja. Nakon toga, korak koji je potreban u oba dijela jest mapiranje očitanja na referencu. To je postignuto alatom *Ram* [7]. *Ram* koristi *Minimap2* [8] kako bi poravnao svako očitanje na referencu tj. genom. Kao povratnu vrijednost vrati skup mapiranja za svako očitanje i podatke poput početka i kraja mapiranja, kvalitete mapiranja, broja poklapanja i tako dalje. Ove statistike su objedinjene u strukturi koja nalikuje PAF formatu.

Označavanje očitanja je prvi korak i provodi se po algoritmima opisanim u poglavlju 4.3. Redoslijed detekcije vrste očitanja ide ovim redoslijedom. Prvo se pretražuju sva nekvalitetna očitanja. Svako očitanje koje zadovoljava kriterij nekvalitetnog očitanja je označeno kao takvo i više ne prolazi kroj kriterije za ostale vrste. Nakon ovog koraka u skupu podataka su ostala samo kimerna, ponavljajuća i pravilna očitanja. U drugom koraka označavaju se sva kimerna očitanja koja se također nakon označavanja uklanjaju iz skupa podataka. Konačno se za preostala očitanja provjerava jesu li ponavljajuća, a sva ona koja nisu svrstana su u skupinu pravilnih očitanja. Tako je svakom očitanju pridružena samo jedna oznaka vrste. Ako očitanje ne zadovoljava niti jedan kriteriji opisan u poglavlju 4.3, ono je označeno kao pravilno.

Kod rekonstruiranja genoma korištena su dva algoritma. Kako bi se mogao napraviti graf sastavljanja potrebno je ukloniti očitanja sadržana u drugim očitanja. Najbrži način za to je *Sweep line algoritam* koji je stoga ovdje i korišten, a opisan je u po-

glavlju 4.6. Obzirom da je uvjet za graf sastavljanja zadovoljen kreće se u njegovo stvaranje nadovezujući jedno očitanje na drugo uzimajući u obzir njihova mapiranja na genom. Za svako očitanje se gleda samo najbolje mapiranje, tj. ono koje se mapira najvećom duljinom. Jednom napravljeni graf sastavljanja obilazi se pretraživanjem u dubinu. Tako obiđeni graf omogućuje uvid u statistiku pokrivenosti genoma tj. koliki dio genoma se može rekonstruirati. Moguće je da postoji točka prekida, dio genoma koji nije moguće rekonstruirati pa se za pokrivenost uzima komponenta najveće duljine. Ovaj dio omogućuje i ispis odnosa očitanja, kako se jedno očitanje nadovezuje na druga, što je kasnije moguće grafički prikazati.

5.2. Korištenje i izlaz programa

Alat i dodatne upute za korištenje su dostupne na poveznici `https://github.com/lbcb-sci/ratlesnake`. Za njegovo korištenje potreban je alat *cmake* pomoću kojeg se prevodi. U sljedećim koracima je prikazano kako se koristi na UNIX baziranim operacijskim sustavima, a preuzimanje je postignuto pomoću sustava *git*.

```
$ git clone --recursive
https://github.com/lbcb-sci/ratlesnake.git
$ cd ratlesnake
$ mkdir build
$ cd build
$ cmake -DCMAKE_BUILD_TYPE=Release ..
$ make
```

Nakon ovih naredbi u direktoriju *bin* će se nalaziti izvršna datoteka. Kao dva argumenta je potrebno predati datoteku s očitanjima i datoteku s cijelim genomom. Obje datoteke mogu biti FASTA ili FASTQ formata.

Po završetku izvođenja u ljusci je ispisano koliki dio predanog genoma se može sastaviti iz očitanja. Također je ispisano i koliko je koje vrste očitanja pronađeno u ulaznom skupu podataka. Stvorene su i četiri nove datoteke FASTA formata. U svakoj datoteci se nalaze očitanja određene vrste: ponavljajuća, kimerna, regularna i nekvalitetna. Alat kreira i datoteku GFA formata [9] koja omogućava grafički prikaz grafa sastavljanja.

6. Rezultati

Alat je testiran na genomu bakterije *Bacillus subtilis*. Ova bakterija ima oko 4.2 milijuna baza. Skup očitanja koji je korišten sadrži 220 tisuća očitanja. Među njima je 8200 očitanja ručno pregledano i označeno kao pravilno, nepravilno, ponavljajuće ili kimerno očitanje. To je postignuto promatranjem grafova pokrivenosti očitanja te njihovih mapiranja na genom bakterije. Označena očitanja su proslijeđena alatu te su rezultati detekcije vrste očitanja evaluirani.

Tablica 6.1: Matrica konfuzije

Oznaka alata	Stvarna oznaka					Preciznost
	kimerna	nekvalitetna	pravilna	ponavljajuća	kim+pon	
kimerna	316	100	41	9	31	63.6%
nekvalitetna	36	2872	353	0	1	88.0%
pravilna	109	166	3386	47	8	91.1%
ponavljajuća	4	170	20	553	40	70.3%
Odziv	68.0%	86.8%	89.1%	90.8%	-	86.7%

Za potrebe evaluacije alata uzete su oznake očitanja koje je alat pridružio svakom od 8200 ručno označenih očitanja te su potom te oznake uspoređene s poznatim oznakama očitanja. Svako očitanje može biti ispravno označeno (ako se oznaka alata i poznata oznaka poklapaju) ili pogrešno označeno (ako se oznaka alata i poznata oznaka ne poklapaju). Posebna pozornost obraćena je na očitanja koja imaju karakteristike i kimernih i ponavljajućih očitanja, a koja su opisana u poglavlju 4.3. Takva očitanja su izuzetno specifična jer na neki način pripadaju objema vrstama očitanja, ali se na izlazu alata klasificiraju kao samo jedna od te dvije vrste. Rezultati su prikazani u tablici 6.1 koja predstavlja matricu konfuzije. Matrica konfuzije je često korišten format evaluacije klasifikacijskih alata. Svaki red u matrici konfuzije predstavlja instance očitanja kako su označena alatom dok svaki stupac predstavlja instance očitanja po stvarnim klasama. Na "glavnoj dijagonali" matrice nalaze se ispravno označeni podaci dok se

na ostalim pozicijama nalaze pogrešno označeni primjeri. Primjerice na presjeku retka kimerna i stupca nekvalitetna u tablici 6.1 nalazi se 100 primjera koji su alatom označeni kao kimerna očitanja, a u stvarnosti su nekvalitetna. Konfuzijska matrica koristi se za izračun brojnih klasifikacijskih mjera uspješnosti, primjerice preciznosti, odziva i točnosti. U matrici konfuzije u tablici 6.1 prikazane su i vrijednosti odziva i preciznosti po razredu te ukupna točnost alata.

Odziv se računa po formuli:

$$odziv = \frac{tp}{tp + fn} \tag{6.1}$$

Preciznost se računa po formuli:

$$preciznost = \frac{tp}{tp + fp} \tag{6.2}$$

Točnost se računa po formuli:

$$\textit{točnost} = \frac{\sum tp}{\sum(tp + fn + fp)} \tag{6.3}$$

U formulama 6.1, 6.2 i 6.3 oznake imaju ovo značenje:

- tp = točno predviđena očitanja
- fn = za točnu klasu X, očitanja koja nisu označena kao X
- fp = očitanja označena kao klasa X, a ne pripadaju toj klasi

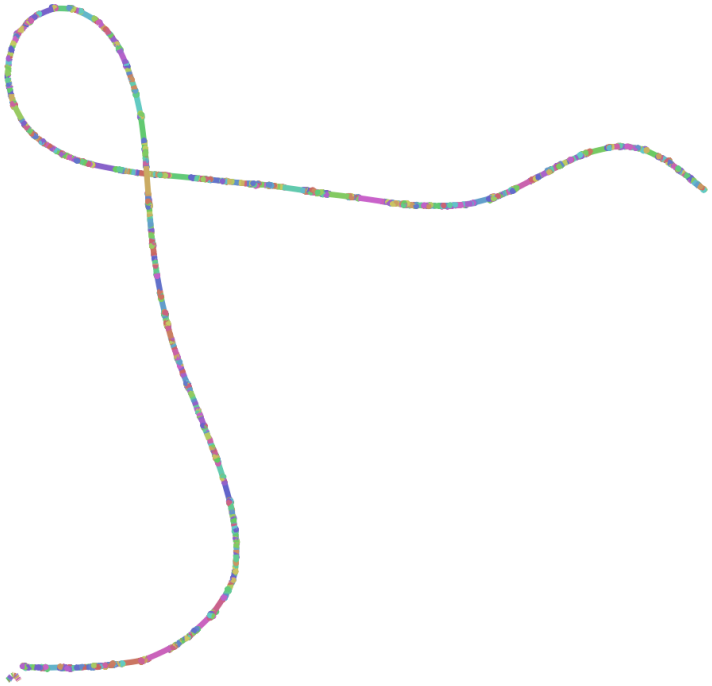
Iz konfuzijske matrice vidimo da su najlošiji rezultati postignuti za kimerna očitanja gdje je od svih očitanja koja su kimerna samo 68% označeno kao kimerno, a među onim očitanjima koja su označena kao kimerna samo je 63.6% bilo zaista kimerno. Ostale vrste očitanja postižu značajno bolji odziv dok je preciznost nešto slabija u slučaju ponavljajućih očitanja zbog 170 nekvalitetnih očitanja koja su označena kao ponavljajuća. Ukupna točnost alata je 86.7%. Očitanja koja imaju karakteristike ponavljajućih i kimernih očitanja su u najvećem broju slučajeva označena kao ponavljajuća i kimerna očitanja dok je nekoliko očitanja označeno kao nekvalitetno ili nepravilno očitanje.

Bakterija *Bacillus subtilis* ima dva kromosoma koji su zapisani u FASTA datoteci kao dva odvojena slijeda. Svaki od dva kromosoma ima i svoj identifikator. U slučaju bakterija koje imaju dva kromosoma, jedan je takozvani plazmid - manji kromosmi koje bakterije lako razmjenjuju. Nakon označavanja očitanja testirana je gornja granica sastavljanja genoma. Svaki od dva kromosoma je sastavljen koristeći metode objašnjene u poglavlju 4.3. U tablici 6.2 prikazani su rezultati sastavljanja svakog kromosoma pojedinačno u formi postotka sastavljenosti. Iz tablice je vidljivo da je

većinski dio genoma bakterije uspješno sastavljen (u slučaju prvog fragmenta gotovo 100%) što pokazuje da je, koristeći označena očitanja, metodama sastavljanja genoma moguće vrlo kvalitetno rekonstruirati originalni genom.

Tablica 6.2: Rezultati alata za *Bacillus subtilis*

	Oznaka	Pokrivenost genoma
1.	$fragment_1$	99.8%
2.	$fragment_2$	94.3%



Slika 6.1: Prikaz rezultata sastavljanje bakterije *Bacillus subtilis* pomoću Bandage alata

Na izlazu alata je i, kao što je već spomenuto, datoteka u GFA formatu koja se koristi za grafički prikaz grafa sastavljanja. Ta datoteka se proslijeđuje alatu *Bandage* [10] koji prikazuje kako se očitanja nadovezuju jedno na drugo, ali omogućuje i interakciju s očitanjima za bolji uvid u graf sastavaljanja. Graf sastavaljanja bakterije *Bacillus subtilis* prikazan je na slici 6.1. Svako očitanje na slici prikazano je kao pravokutnik nasumične boje, a ako postoji preklapanje između očitanje, pravokutnici su povezani. Iz grafa sastavljanja vidljivo je da se bakterija sastoji od jednog velikog kromosoma i jednog manjeg, plazmida, koji se na grafu nalazi u donjem lijevom kutu.

7. Zaključak

Bioinformatika kao interdisciplinarno područje okuplja širok spektar znanja i stručnjaka koji provode kompleksna istraživanja s ciljem pronalaska što točnijih i konkretnijih informacija vezanih uz bioznanost. Bioznanost kao vrlo osjetljivo područje zahtijeva podatke koji su visoke kvalitete i preciznosti.

S padom cijena istraživanja počeo je ubrzani razvoj s novim metodama sekvenciranja. To je omogućilo veći broj istraživanja, ali je uzrokovalo i veću stopu pogrešaka u procesu sekvenciranja. Problemi odabira odgovarajućih očitanja na ponavljajućim područjima genoma te izbjegavanje pogrešnog sastavljanja genoma uslijed kreiranja "umjetnih" očitanja, problemi su koji nisu izbjegnuti u novoj generaciji sekvenciranja. Problem sastavljanja genoma je jedan od najtežih problema prisutnih u području bioinformatike, a kao dodatni otežavajući faktor za kvalitetno sastavljenih genoma, a samim time i kvalitetu istraživanja kojima se podvrgavaju tako dobiveni genomi, nameću se specifična, atipična očitanja kao što su kimerna i ponavljajuća očitanja. Čak i najnovije metodologije korištene za sastavljanje genoma imaju problema s lažnim preklapanjima uzrokovanim specifičnim očitanjima i regijama na genomu.

Razvojem metoda za detekciju problematičnih očitanja te korištenjem informacija o tim očitanjima tijekom procesa sastavljanja metoda otvaraju se nove mogućnosti za poboljšanje kvalitete sastavljenih genoma. U aktualnim pokušajima poboljšanja metoda de novo sastavljanja genoma, glavni fokus stavlja se na primjenu metoda strojnog učenja na detekciju tipova očitanja i poboljšanje kvalitete sastavljenosti genoma. Ono što je zajedničko svim metodama strojnog učenja jest potreba za kvalitetnim i balansiranim skupovima za učenje. Alat predstavljen u ovom radu s ukupnom točnosti od 86.7% može biti od značajne pri generiranju kvalitetnih skupova podataka koji mogu biti korišteni za metode polu-nadziranog učenja.

Trenutni rezultati pokazuju da najveći problem u procesu detekcije predstavljaju kimerna očitanja koja su u velikom broju slučajeva detektirana kao pravilna. Obzirom da su kimerna očitanja umjetno stvorena očitanja i da kao takva predstavljaju značajan problem u procesu sastavljanja genoma, kako bi budući rezultati u procesu sastavljanja

genoma bili što uspješniji, nužno je fokus budućeg rada staviti na dodatni razvoj metoda za kvalitetnu detekciju kimernih očitanje uz uvjet da se kvaliteta detekcije ostalih tipova očitanja zadrži na trenutnoj razini ili također poboljša.

LITERATURA

[1] J. Craig Venter Institute. Genome sequencing, 2003.

[2] Mile Sikic and Mirjana Domazet-Loso. Bioinformatika, 12 2013.

[3] Heng Li. Paf: a pairwise mapping format, 2018.

[4] Robert Vaser and Mile Sikic. *Yet another de novo genome assembler*, 05 2019.

[5] Amy M. Hauth and Deborah A. Joseph. Beyond tandem repeats: complex pattern structures and distant regions of similarity, 2002.

[6] Vaser R. Bioparser, 2019.

[7] Laboratory for Bioinformatics, Faculty of Electrical Engineering Computational Biology University of Zagreb, and Computing. Ram, 2019.

[8] Heng Li. *Minimap2: pairwise alignment for nucleotide sequences*, 05 2018.

[9] GFA format. `https://github.com/GFA-spec/GFA-spec/blob/master/GFA2.md`.

[10] Ryan R. Wick, Mark B. Schultz, Justin Zobel, and Kathryn E. Holt. *Bandage: interactive visualization of de novo genome assemblies*, 2015.

Sastavljanje genoma koristeći detekciju tipova očitanja

Luka Požega, Sara Bakić

Sažetak

De novo sastavljanje genoma jedan je od najkompleksijih problema u bioinformatici. Brojni su problemi prilikom sastavljanja genoma, uzrokovani, kako biološkim specifičnostima, tako i greškama u procesu sekvenciranja. Probleme stvaraju takozvana ponavljajuća, kimerna i nekvalitetna očitanja. U ovom radu detaljno su objašnjena problematična očitanja. Prikazani su potencijalni problemi u procesu sastavljanja genoma uzrokovani specifičnim kimernim i ponavljajućim očitanjima, razvijene su metode za detekciju takvih očitanja te su razvijene metode za sastavljanje genoma uz korištenje znanja o problematičnim očitanjima.

Ključne riječi: bioinformatika, sekvencioniranje, kimerno očitanje, ponavljajuće očitanje, sastavljanje genoma

Genome assembly with the help of detection of types of reads

Luka Požega, Sara Bakić

Abstract

De novo genome assembly is one of the most complex problems in Bioinformatics. There are many problems, either caused by biological specificities or mistakes made during sequencing process, that complicate de novo genome assembly process. The problems are mainly caused by chimeric and repeating reads. This thesis thoroughly describes problematic reads. Potential problems occurring during the assembly process that are caused by specific chimeric and repeating reads are thoroughly described, along with methods developed for detection of such reads and methods developed for genome assembly that use the knowledge collected on problematic reads and with that knowledge try to assemble the target genome as precise and complete as possible.

Keywords: Bioinformatics, sequencing, chimeric reads, repeating reads, genome assembly