SVEUČILIŠTE U ZAGREBU
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

ZAVRŠNI RAD br. 4586

Poravnanje RNA očitanja na poznate gene

Ivan Krpelnik

Zagreb, lipanj 2016.

Zahvaljujem mentoru Mili Šikiću na pruženoj prilici, pomoći i brzim odgovorima. Zahvaljujem i Krešimiru Križanoviću i Ivanu Soviću na pomoći prilikom ispitivanja izrađenog alata i dodatnim materijalima.

SADRŽAJ

1. Uvod

Tehnološkim napretkom, zadnjih je godina cijena sekvenciranja DNA drastično smanjena i sekvenciranje pojednostavljeno što je omogućilo da istraživanja mogu raditi i manji timovi i manje institucije, a ne samo veliki centri za istraživanja genoma [1]. Skupinu novih, modernih tehnologija za sekvenciranje, kao što su *Illumina*, *Roche454* i drugi, nazivamo *Next-Generation Sequencing (NGS)*). RNA sekvenciranje je nova metodologija koja se temelji upravo na sekvenciranju nove generacije.

U svakom živom biću DNA sadrži informacije koje određuju sva svojstva i funkcionalnosti svake stanice. Stanice dohvaćaju iz DNA pojedine gene na način da se dijelovi gena prepisuju u RNA molekule koje se zatim mogu koristiti za stvaranje proteina ili kontrolu ekspresivnosti gena[1]. Skupina RNA molekula opisuju neko stanje stanice iz čega se mogu otkriti patološki mehanizmi bolesti. U praksi, sekvenciranjem dobivamo mnogo kraćih nizova koja nazivamo očitanjima iz nasumičnih pozicija ulaznog skupa RNA molekula. Ta očitanja zatim treba poravnati na referentni genom gdje broj poravnatih molekula RNA na neki gen pokazuje ekspresivnost tog gena [2].

Ovaj rad daje uvid u metode RNA sekvenciranja te opisuje nadogradnju postojećeg alata *GraphMap* [3] kako bi se omogućilo poravnanje RNA očitanja na poznate gene. U poglavlju 2, opisane su različite metode RNA sekvenciranja, a u poglavlju 3 je pobliže opisano sekvenciranje nanoporama i podaci koji se dobivaju istim. U poglavlju 4 su prikazane metode korištene u ovom radu za poravnanje RNA očitanja na poznate gene, odnosno opis alata *GraphMap* i njegove nadogradnje. Poglavlje 5 opisuje podatke i formate datoteka korištenih za opis gena i poravnanja te spremanje sekvenci. U istom je poglavlju opisan je alat *pbsim* korišten za generiranje simuliranih očitanja. Konačno, poglavlje 6 daje pregled rezultata poravnanja opisanim metodama.

[1] Ekspresivnost gena — sposobnost gena da jače ili slabije izrazi neku osobinu

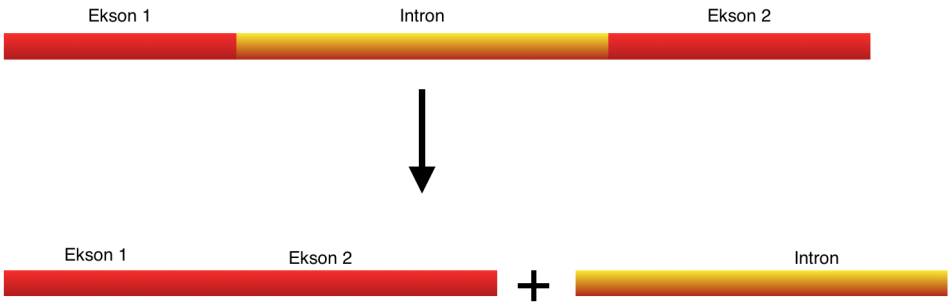
2. RNA sekvenciranje

RNA sekvenciranje (*RNAseq*) je primjena bilo koje od novih metoda sekvenciranja (*NGS*) za proučavanje RNA. Samo sekvenciranje je generalno jednako i za RNA i DNA, ono što se razlikuje je analiza podataka u koju za RNA sekvenciranje spada sastavljanje poznatih transkriptoma, alternativno prekrajanje transkripata, otkrivanje novih transkripata i drugo.
Prilikom sekvenciranja RNA, nakon što su dobiveni dovoljno veliki RNA lanci, stvara se komplementarna DNA (*cDNA*) iz molekula glasničke RNA (*mRNA*). Rezultat su sekvence koje su reverzni komplement RNA molekula, ali s bazama koje ima DNA (*ACTG*) [4]. Dva su razloga pretvaranja RNA molekula u cDNA — RNA molekule su nestabilne i većina metoda sekvenciranja čita samo DNA molekule.

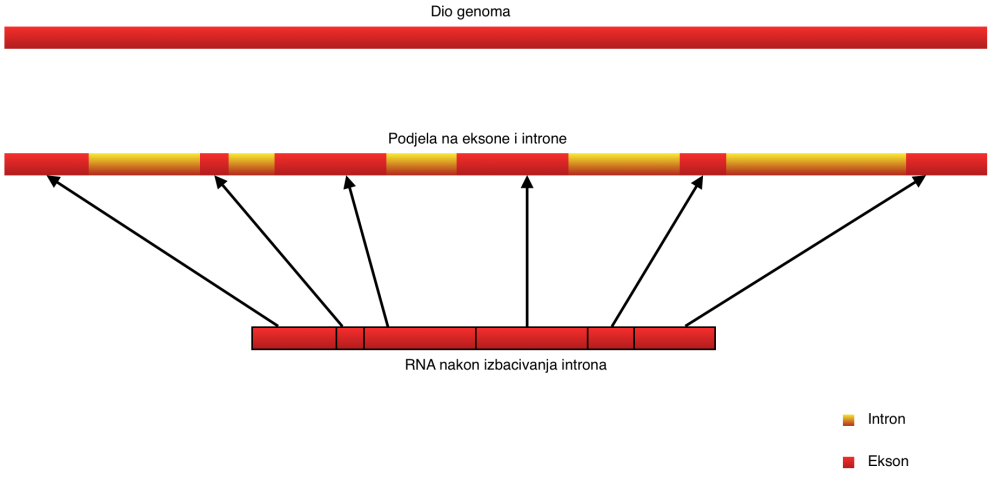
2.1. mRNA

RNA molekule dobivaju se postupkom transkripcije DNA molekula. Osim što se događa prepisivanje DNA molekula, ono što je posebno zanimljivo za poravnanje RNA očitanja je prekrajanje (engl. *splicing*). RNA molekula prije prekrajanja sastavljena je od više regija koje nazivamo intronima i eksonima. One regije koje tvore konačnu mRNA, nakon prekrajanja, nazivamo eksonima. Prilikom prekrajanja, ne moraju se nužno spojiti svi eksoni već može i samo neki podskup, što nazivamo alternativnim prekrajanjem.

Na slici 2.1 prikazano je prekrajanje dva eksona, odnosno izbacivanje jednog introna, a na slici 2.2 prikazan je odnos neke konačne mRNA i početnog dijela genoma iz kojeg je prepisana. Sada je jasnija razlika poravnanja DNA očitanja i RNA očitanja — prilikom poravnanja RNA očitanja na referentni genom, treba uzeti u obzir regije introna koje su izbačene. To otežava poravnavanje jer duljina introna može biti dugačka od svega desetak baza do nekoliko tisuća.



Slika 2.1: RNA prekrajanje



Slika 2.2: Usporedba RNA i dijela genoma

2.2. Metode RNA sekvenciranja

Podaci dobiveni različitim RNA sekvenciranjem jako variraju. Te varijacije mogu bitno utjecati na eksperimente koji se kasnije vrše pa je bitno unaprijed poznavati iste i računati s njima. Većina modernih metoda RNA sekvenciranja bazira se na sekvenciranju sintezom (engl. *sequencing by synthesis*) s DNA polimerazom ili ligazom kao osnovnom komponentom. Neki alati koji koriste DNA polimerazu su Roche 454, Illumina, Helicos i PacBio, dok SOLiD i Complete Genomics koriste DNA ligazu. Te alate možemo podjeliti i po sekvenciranju jedne molekule — Helicos i PacBio te sekvenciranju više molekula — Illumina i SOLiD [4].

Alati koji koriste jednu molekulu za sekvenciranje obično imaju veći postotak pogrešaka, oko 5%. Zbog toga je očitanja takvim sekvenciranjem teže poravnati na referentni genom. Kod takvih očitanja najčešće greške su umetanje (engl. *insertion*) i

brisanje (engl. *deletion*) baza. Alati koji koriste više molekula imaju manje greške, do oko 1% i najčešće su te greške neslaganja (engl. *mismatch*). Bitna razlika je i duljina očitanja koja proizvode alati. Duža očitanja bi trebalo biti lakše poravnati, ali duža očitanja dolaze i s više grešaka zbog čega se ipak ne poravnavaju tako lako. Illumina očitanja su duljine do par stotina bp [1], dok su PacBio očitanja u prosjeku veličine od 5000 do 8000 bp. Očitanja u sekvenciranju nanoporama mogu biti dugačka do čak 50 000 bp, ali imaju veliki postotak grešaka. Sekvenciranje nanoporama i podaci dobiveni istim biti će opisani detaljnije u trećem poglavlju.

2.3. Alati za poravnanje RNA očitanja

Alate za poravnanje RNA očitanja se dijele na one koji poravnavaju očitanja na genom bez praznina, odnosno dijelova koji se preskaču (engl. *Unspliced aligners*) i na one koji poravnavaju sa prazninama *Spliced aligners* [5].
Unspliced alati se obično dijele u dvije grupe [6]:

1. Alati temeljeni na Burrows-Wheeler transformaciji. Npr. *Bowtie* i *BWA*
2. Alati temeljeni na seed metodi. Npr. *Stampy*.

Alati za poravnanje s prazninama dijele se na alate s anotiranim regijama koje se preskaču i *De novo* alate koji ne trebaju anotirane regije. Najpoznatiji alati koji koriste anotirane regije su RUM i SpliceSeq, a neki poznati *De novo* alati su Tophat, BBMap, STAR i drugi.

[1] Base pair — parova baza

3. Sekvenciranje Nanoporama

Sekvenciranje nanoporama čini se kao tehnologija koja bi mogla zadovoljiti standarde određene tzv. načelom *"$1000 Genome"* [7]. To načelo predviđa da će u budućnosti sekvenciranje ljudskog genoma koštati otprilike 1000 USD [1]. U zadnjih 10 do 15 godina, cijena sekvenciranja ljudskog genoma pala je s više stotina tisuća dolara na svega nekoliko tisuća [8]. Osim toga, jedan od bitnih ciljeva je moći primjenjivati takve uređaje u svakodnevnim bolničkim pregledima gdje bi i cijena od 1000 USD bila prevelika.

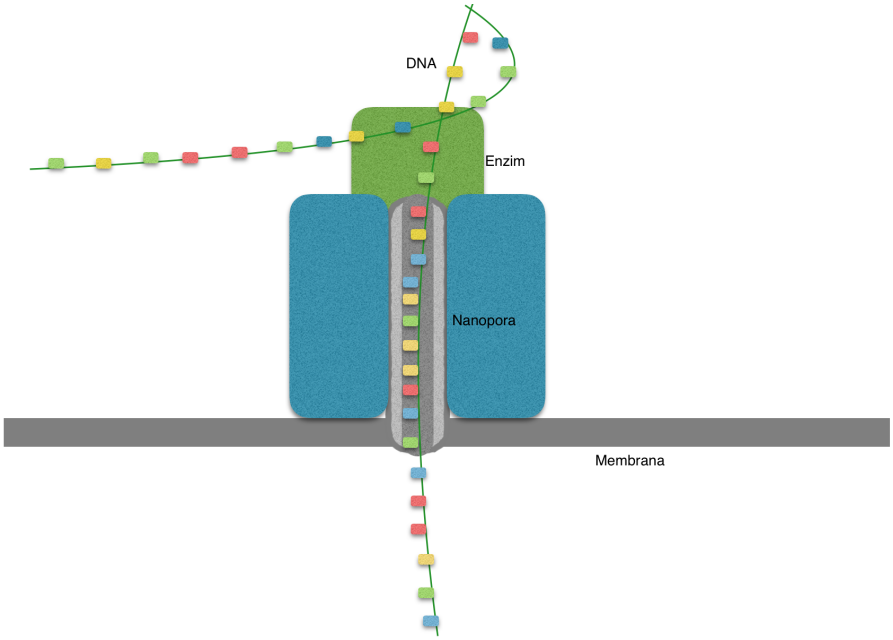
Nanopore su pore malih dimenzija (u nanometrima) stvorene na membrani. Kod sekvenciranja nanoporama, nanopore su na membrani koja ne propušta struju. Stvaranjem napona, struja počinje teći kroz poru. Ovisno o tome koja molekula ulazi kroz nanoporu, stvaraju se karakteristične smetnje za tu molekulu iz kojih se može zaključiti o kojoj se molekuli radi.

Sekvenciranje nanoporama radi tako da se lanac DNA propusti kroz poru iz čega se poremećajima u toku stroje zaključuje o kojim se bazama radi. Na DNA se prije sekvenciranja na jedan kraj stavlja enzim koji propušta jednu po jednu bazu jedne strane DNA lanca što prikazuje slika 3.1. Brzina sekvenciranja se može prilagoditi mijenjanjem korištenog enzima. Sekvenciranje radi neovisno o duljini lanca pa su duljine očitanja sukladne s time. Umjesto da se generiraju unaprijed određene duljine očitanja, očitanja su duljine danog lanca. Za razliku od drugih metoda sekvenciranja koje očitanja određenih duljina daju po završetku obrade, kod ove metode sekvenciranja, podaci su dostupni tijekom obrade. Tijek rada zamišljen je tako da se podaci obrađuju kako dolaze od sekvenciranja i da se sekvenciranje završava onda kad je odlučeno da je prikupljeno dovoljno podataka. Takav način rada nije moguć s drugim tehnologijama gdje je potrebno čekati određeno vrijeme na podatke iz sekvenciranja, odnosno gdje podaci nisu dostupni odmah tijekom sekvenciranja.

Očitanja dolaze u dva tipa — *1D* i *2D*. *2D* očitanja zahtjevaju više vremena pripreme i obrade, ali daju pouzdanije rezultate s obzirom da se temelje na oba lanca.

[1] United States Dollar

Prilikom provlačenja i razdvajanja DNA lanca kroz nanoporu, ako se radi o *2D* očitanjima, biti će provućen i drugi dio DNA.



Slika 3.1: Skica nanopore kroz koju prolazi DNA

3.1. Oxford Nanopore MinION

GraphMap je napravljen kako bi rukovao očitanjima dobivenim uređajem *Oxford Nanopore MinION*. MinION je uređaj za sekvenciranje nanoporama malih dimenzija (veličine dlana) koji radi na prethodno opisan način. Taj relativno jeftin uređaj može se spojiti na računalo USB priključkom i na taj način šalje podatke u realnom vremenu tijekom sekvenciranja. To je korak prema budućnosti u kojoj će bilo tko, bilo gdje moći sekvencirati što god želi[2]. Ovaj uređaj radi brzinom od oko 250 bps[3] po pori. Trenutni cilj je preći brzinu od 1000 bps i to će biti moguće kada se razvije bolji sustav hlađenja jer bi se trenutno osjetile smetnje u očitanjima zbog istog pri većim brzinama.

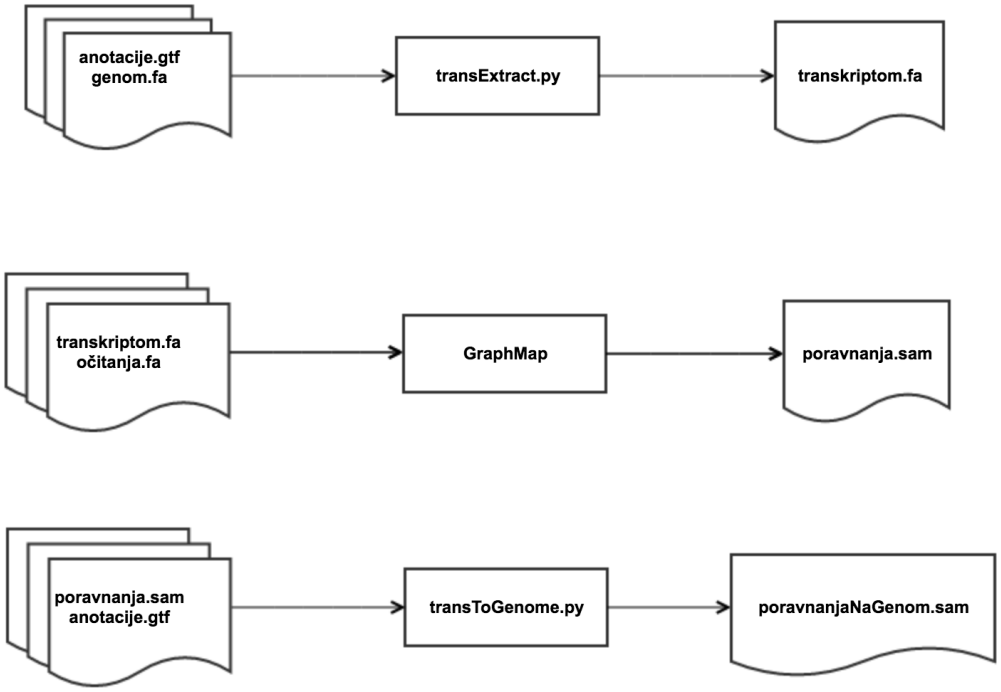
[2]Inside the SkunkWorx: Clive Brown, CTO
[3]Base per second — baze po sekundi

4. Metode

U ovom poglavlju opisane su metode korištene za nadogradnju alata *GraphMap* u svrhu poravnanja RNA očitanja. Priloženi su pseudokodovi korištenih algoritama. Izvorni kodovi dostupni su u repozitorijima na githubu: `https://github.com/Krpa/RnaSeq` i `https://github.com/isovic/graphmap`

Prva verzija nadogradnje, koja služi za testiranje i dokazivanje koncepta, napravljena je kao dvije skripte u Pythonu. Zamišljen rad s te dvije skripte i *GraphMapom* prikazan je na slici 4.1.

Kao prvi korak, potrebno je generirati transkriptom. Za to su potrebni podaci o genima koji se nalaze u datoteci formata GTF, na dijagramu označena imenom *anotacije.gtf* i referentni genom koji se nalazi u FASTA datoteci *genom.fa*. Ti podaci predaju se skripti *transExtract.py* koja u datoteku *transkriptom.fa* zapisuje generirani transkriptom. Ta se datoteka zatim predaje *GraphMapu* zajedno s datotekom *očitanja.fa* koja sadrži RNA očitanja koja je potrebno poravnati na transkriptom. Izlaz iz *GraphMapa* sadrži zapise o poravnanjima RNA očitanja u SAM datoteci *poravnanja.sam*. Te zapise još treba prilagoditi da odgovaraju početnom genomu umjesto generiranom transkriptomu. To radi skripta *transToGenome.py*. Ona iz zadanih poravnanja u SAM datoteci i anotacija u GTF datoteci prepravlja poravnanja na način da odgovaraju genomu. Time smo postigli mapiranje RNA očitanja u 3 zasebna koraka koje možemo i zasebno ispitati. Implementirane skripte mogu biti i same po sebi korisne, bez ostalih koraka i nude neke dodatne mogućnosti.



Slika 4.1: Dijagram komponenti

Implementacija se može naći na github repozitoriju na linku `https://github.com/Krpa/RnaSeq`. Nakon što su dobiveni željeni rezultati sa opisanim skriptama, krajnja implementacija integrirana je u alat *GraphMap* u jeziku C++ koji je uz jezik C isto popularan u području bioinformatike. C++ i C, iako manje intuitivni od Pythona, često su korišteni zbog brzine izvođenja programa. Prilikom ispitivanja C++ implementacije, mogu se koristiti napisane Python skripte kao referentni rezultati.

4.1. Generiranje transkriptoma

Prvi problem koji treba rješiti da bi se *GraphMap* mogao iskoristiti za poravnanje RNA očitanja na poznate gene jest izgraditi referencu na koju će se vršiti poravnanje. Referenca je u ovom slučaju transkriptom. Da bi se transkriptom izgradio, potrebni su podaci o referentnom genomu i anotirane eksonske regije. Anotirane eksonske regije sadrže informaciju o pozicijama eksona na genomu kao i o lancu na kojem se nalaze ($+$ ili $-$) te kojem transkriptu pripadaju. Za svaki transkript treba spojiti eksone u jednu sekvencu. Algoritam koji generira transkripte za neku sekvencu koja je dio genoma dan je algoritmom 4.1.

Algoritam 1 Generiranje transkripata za neku sekvencu

Ulaz: seq – podaci o sekvenci, $transcripts$ – imena transkripata
Izlaz: Transkripti koji još nisu rješeni
toWrite := []
for $(i := 0; i < length(transcripts); inc(i))$ **do**
  if $seqName(transcripts_i) = seq.name$ **then**
    add($toWrite, transcripts_i$)
  end if
end for
for $(i := 0; i < length(toWrite); inc(i))$ **do**
  exonList := exons($toWrite_i$)
  str := strand($toWrite_i$)
  transcript := makeTranscript($seq, exonList, str$)
  writeTranscript($transcript$)
end for
return $transcripts \setminus toWrite$

Na početku je potrebno stvoriti listu svih imena transkripata koji se nalaze na zadanoj sekvenci iz genoma. Za svaki transkript, poziva se funkcija *makeTranscript* koja kao argumente prima sekvencu iz koje je potrebno izvaditi dijelove, u ovom slučaju eksone, listu eksona koji čine taj transkript te oznaku $+$ ili $-$ lanca DNA s kojeg je prepisan transkript. Ta funkcija u slučaju $+$ lanca uzima dijelove sekvence zadane listom eksona i spaja ih u sekvencu koja predstavlja transkript. U slučaju $-$ lanca, od tako generiranog transkripta potrebno je načiniti reverzni komplement, odnosno sekvencu je potrebno preokrenuti (pročitati od kraja prema početku) te svaku bazu komplementirati. Taj je postupak prikazano algoritmom 2. Konačno, takav se transkript zapisuje pozivom funkcije *writeTranscript*. Povratna vrijednost algoritma je lista koja sadrži sve transkripte zadane u parametru *transcripts* koji ne pripadaju zadanoj sekvenci. Ono što se dobiva time je da se, u sljedećem generiranju transkripata za neku drugu sekvencu, smanjuje prostor (u ovom slučaju lista) u kojem se pretražuju transkripti.

Algoritam 2 Funkcija makeTranscript

Ulaz: seq – podaci o sekvenci, $exons$ – lista eksona, $strand$ – oznaka $+$ ili $-$
Izlaz: Generirani transkript
transcript := []
for ($i := 0; i < length(exons); inc(i)$) **do**
  for ($j := start(exons_i); j <= end(exons_i; inc(j))$) **do**
    base := getBase(seq, j)
    add($transcript, base$)
  end for
end for
if $strand = -$ **then**
  reverse($transcript$)
  complement($transcript$)
end if
return transcript

Dodatno je implementirana funkcija *altSplicing* koja radi alternativna prekrajanja eksona. Ona trenutno nije važna u ovom načinu poravnanja jer je cilj ovog rada poravnanje RNA očitanja na poznate gene što znači da su poznati transkripti odnosno transkriptom na koji se poravnavaju očitanja pa nema potrebe za stvaranjem transkripata koji nisu zadani anotiranim eksonskim regijama.
Zamišljen način generiranja alternativnih prekrajanja je generiranje svih pravih podskupova (izuzevši prazan skup) zadanog skupa eksonskih regija. S obzirom da eksonske regije moraju zadržati svoj početni poredak, nije potrebno dodatno permutirati te podskupove. Algoritam počiva na činjenici da se prolaskom kroz sve brojeve od 0 do $2^n - 1$ uključujući, gdje je n broj elemenata u skupu, zapravo dobivaju predlošci za sve podskupove zadanog skupa ako pogledamo binarni zapis tih brojeva. Način na koji se interpretira svaki bit u nekom broju koji zovemo bit maskom je sljedeći:
Ako je bit na j-tom mjestu u bit masci postavljen, tada u podskup, opisanim tom bitmaskom, treba staviti j-ti element iz originalnog skupa. U slučaju da j-ti bit nije postavljen, j-ti element se preskače.
Tako će npr. 0 predstavljati prazan skup, dok će $2^n - 1$ predstavljati skup identičan početnom skupu jer takav broj ima postavljene sve bitove. Broj bitova koji treba provjeriti je n. Pseudokod je dan u nastavku.

Algoritam 3 Alternativna prekrajanja

Ulaz: $regions$ – niz regija za koje treba napraviti alternativna prekrajanja
Izlaz: skup alternativnih prekrajanja
m := length($regions$)
n := $2^m - 1$
alternatives := []
for ($bitMask := 1; bitMask < n; inc(bitMask)$) **do**
  nextSplicing := []
  for ($j := 0; j < m; inc(j)$) **do**
    if $isBitSet(bitMask, j)$ **then**
      add($nextSplicing, regions_j$)
    end if
  end for
  add($alternatives, nextSplicing$)
end for
return alternatives

4.2. Poravnanje alatom GraphMap

GraphMap je alat namijenjen za poravnanje dugih DNA očitanja sklonih pogreškama poput onih dobivenim iz Oxford Nanopore MinION-a. Otvorenog je koda i nalazi se na github repozitoriju `https://github.com/isovic/graphmap`. Poravnanje radi u nekoliko koraka:

1. Stvaranje indeks strukture: Potrebno je stvoriti strukturu nad referentnim genomom koja efikasno pronalazi sve lokacije na kojima se nalazi neka zadana podsekvenca koja ima *don't care* pozicije u smislu da se te pozicije preskaču prilikom pretraživanja i uspoređivanja. Takvu strukturu ne treba uvijek stvarati, već se jednom stvorena struktura za neki genom, može spremiti na disk i učitati s diska.

2. Selekcija regija: Referentni genom podijeli se na niz nepreklapajućih regija veličine $D/3$ gdje je D duljina očitanja. Cilj ovog koraka je smanjenje prostora pretraživanja za sljedeće korake. Za neko očitanje se pomičnim prozorom odrede manji podnizovi. Za svaki takav podniz nađu se sve pozicije na kojima se on nalazi pomoću strukture iz prošlog koraka. Za svaku poziciju se odredi regija kojoj ta pozicija pripada i uveća se brojač u toj regiji. Regije se zatim sortiraju

po tim brojačima i obrađuju slijedno.

3. Mapiranje na graf: U ovom se koraku stvaraju tzv. sidra iz podataka iz prethodnog koraka. Stvara se graf u kojem su čvorovi mali podnizovi očitanja. U ovom koraku se koristi nova struktura slična prvoj indeks strukturi. Šetnjom po tom grafu, određuju se različite pozicije poravnanja na referenci, odnosno na regijama odabranim u prethodnom koraku.

4. LCSk: zbog ponavljanja u sekvencama, pozicije sidra ne rastu monotono po koordinatama u očitanjima i referenci na koju se mapiraju. Zato se pronalazi najdulji niz tih pozicija koji monotono raste i na očitanjima i na referenci. Taj problem se može svesti na LCSk, odnosno na problem pronalaženja najvećeg broja podnizova duljine k, koji se nalaze i na očitanju i na referenci, čuvajući njihov poredak.

5. Filtriranje i grupiranje: filtriranje se vrši tako da se izbacuju sidra koja previše odskaču od pravca nagiba $45°$. Nakon filtriranja, u slučaju da se na nekim mjestima nalazi velika rupa, sidra se grupiraju u manje, linearne podgrupe.

6. Konačno poravnanje: nakon što su sve regije obrađene, one se sortiraju po kvaliteti i ona sa najvećom kvalitetom, izabrana je kao konačno poravnanje.

Detaljnije o *GraphMapu*, može se pronaći u [3]. *GraphMap* za ulazni transkriptom i RNA očitanja može primjeniti isti algoritam za poravnanje kao i za DNA očitanja i genom. Generiranjem transkriptoma izbačene su problematične praznine (regije introna) spomenute u poglavlju 2 tako da se algoritam za poravnanje ne mora brinuti o tome. Ostaje još rješiti problem translatiranja dobivenih poravnanja u prostor genoma.

4.3. Translacija u prostor genoma

Dobivene pozicije poravnanja su pozicije u transkriptomu i treba ih prebaciti u prostor genoma odnosno u poravnanja treba ubaciti regije introna. Za translaciju, osim podataka o poravnanjima, potrebni su isti podaci o anotiranim eksonskim regijama kao i kod generiranja transkriptoma.

Za početak, nalazimo početnu poziciju početka poravnanja na genomu. Pozicija se dobiva kao pozicija početka eksona na kojem poravnanje počinje zbrojena sa pomakom unutar tog eksona. To je prikazano algoritmom 4.

Algoritam 4 Pronalazak pozicije na genomu

Ulaz: $regions$ – niz regija eksona, pos – pozicija očitanja na transkriptu
Izlaz: pozicija na referenci
size := 0
i := 0
while $size + last(regions_i) - first(regions_i) + 1 < pos$ **do**
  size := $size + last(regions_i) - first(regions_i) + 1$
  $inc(i)$
end while
return $regions_i + pos - size - 1$

Nakon pronalaska pozicije na genomu, ključan dio je umetanje introna u niz znakova koji opisuju operacije u poravnanju. Taj niz znakova naziva se *Cigar string* i opisan je detaljnije u poglavlju o podacima u sekciji o SAM formatu. Pseudokod tog postupka dan je algoritmom 5. Algoritam prolazi kroz niz operacija poravnanja na način da taj niz tretira kao stog. Ako se radi o operaciji umetanja ili podrezivanja, nije potrebno pomaknuti kazaljku po referenci jer se te operacije događaju u odnosu na referencu, odnosno na očitanju. Svaki put kad se zbrajanjem duljine operacija dolazi do kraja neke eksonske regije, umeće se intron duljine koja je jednaka razlici početne pozicije sljedećeg introna i krajnje pozicije trenutnog eksona umanjenoj za 1 jer su obje pozicije uključene u eksone. Osim toga, treba trenutnu operaciju podjeliti na dvije, tako da se prva operacija ponavlja onoliko puta koliko stane do kraja trenutne eksonske regije, a druga onoliko kolika je preostala razlika trenutne operacije i prve novonastale operacije. Ta druga operacija koja je nastala dijeljenjem stavlja se na stog kao sljedeća za obradu. U slučaju da nije kraj neke eksonske regije, uzima se cijela operacija.
Po završetku te petlje, u slučaju podrezivanja očitanja, potrebno je još pridodati preostale operacije. Nakon toga, novi niz operacija koji odgovara poravnanju na genom nalazi se u varijabli *newCigar*. Time je rješena translacija dviju ključnih stvari, pozicije i operacija poravnanja, iz prostora transkriptoma u prostor genoma.

Algoritam 5 Umetanje introna

Ulaz: $cigar$ – stog koji predstavlja Cigar string, pos – pozicija na transkriptu, $size$ – suma duljina eksona prije eksona na kojem je trenutna pozicija

Izlaz: novi Cigar string s umetnutim intronima

```
newCigar := []
while ¬ isEmpty(cigar) do
    count := pop(cigar)
    op := pop(cigar)
    if op =′ I′ ∨ op =′ S′ ∨ op =′ H′ then
        add(newCigar, count)
        add(newCigar, op)
    else if count + pos > regionSize(regInd)+size then
        take := regionSize(regInd) +size − pos
        add(newCigar, take)
        add(newCigar, op)
        push(cigar, op)
        push(cigar, count − take)
        add(newCigar, regionSize(regInd + 1) − regionSize(regInd) − 1)
        add(newCigar, intronOp())
        last := last + regionSize(regInd)
        pos := pos + take
        inc(regInd)
    else
        pos := pos + count
        add(newCigar, count)
        add(newCigar, op)
    end if
end while
return newCigar
```

5. Podaci

5.1. GTF format

GTF (engl. *Gene transfer format*)[1] format koristi se za pohranu informacija o genima. Vrlo je sličan formatu GFF, ali sadrži dodatnu strukturu zbog koje zaslužuje biti zaseban format. Datoteka koja sadrži opise eksona i iz koje smo generirali transkriptom je upravo ovog formata. Sadrži sljedećih 10 polja odvojenih znakom *TAB*:

seqname: ime sekvence kojoj pripada svojstvo opisano tom linijom.

source: izvor iz kojeg su došli podaci o ovoj liniji.

feature: ime svojstva opisanog ovom linijom.

start: indeks početka svojstva na sekvenci — brojanje indeksa počinje od 1.

end: indeks kraja svojstva na sekvenci — baza na indeksu kraja je uključen u svojstvo.

strand: definiran znakovima *'+'* i *'-'*, a označava smjer lanca DNA na kojem se nalazi svojstvo.

score: broj koji opisuje sigurnost postajanja ovog svojstva.

frame: označuje koja baza je prva baza kodona.

attributes: atributi opisani parovima ključ - vrijednost. Različiti atributi su odvojeni znakom *';'*. Obavezni atributi su *gene_id value;* i *transcript_id value;*.

comments: ovo polje nije obavezno, a služi za komentare. Komentari započinju znakom *'#'* i protežu se do kraja linije.

Na sljedećoj slici nalazi se početak GTF datoteke korištene za testiranje. Datoteka sadrži svojstva genoma organizma *Saccharomyces cerevisiae*. Retci koji su nama posebno zanimljivi su oni koji predstavljaju svojstva *exon*.

[1] `http://mblab.wustl.edu/GTF22.html`

```
chrI    sacCer3_sgdGene start_codon     130799  130801  0.000000        +       .       gene_id "YAL012W"; transcript_id "YAL012W";
chrI    sacCer3_sgdGene CDS     130799  131980  0.000000        +       0       gene_id "YAL012W"; transcript_id "YAL012W";
chrI    sacCer3_sgdGene stop_codon      131981  131983  0.000000        +       .       gene_id "YAL012W"; transcript_id "YAL012W";
chrI    sacCer3_sgdGene exon    130799  131983  0.000000        +       .       gene_id "YAL012W"; transcript_id "YAL012W";
chrI    sacCer3_sgdGene start_codon     335     337     0.000000        +       .       gene_id "YAL069W"; transcript_id "YAL069W";
chrI    sacCer3_sgdGene CDS     335     646     0.000000        +       0       gene_id "YAL069W"; transcript_id "YAL069W";
chrI    sacCer3_sgdGene stop_codon      647     649     0.000000        +       .       gene_id "YAL069W"; transcript_id "YAL069W";
chrI    sacCer3_sgdGene exon    335     649     0.000000        +       .       gene_id "YAL069W"; transcript_id "YAL069W";
chrI    sacCer3_sgdGene start_codon     538     540     0.000000        +       .       gene_id "YAL068W-A"; transcript_id "YAL068W-A";
chrI    sacCer3_sgdGene CDS     538     789     0.000000        +       0       gene_id "YAL068W-A"; transcript_id "YAL068W-A";
```

Slika 5.1: Sadržaj GTF datoteke

5.2. FASTA format

FASTA format[2] koristi se za pohranu nukleinskih sekvenci i amino kiselina. Format je vrlo jednostavan za parsiranje i manipuliranje. Svaka sekvenca je opisana zaglavljem — linijom koja započinje znakom '>' i daje ime i opis sekvenci, iza koje slijedi jedna ili više linija koje opisuju baze sadržane u toj sekvenci. Prazne linije nisu dozvoljene i linije ne bi trebale biti duže od 120 znakova (običaj je da se uzima do 80). Ovaj format korišten je za pohranu genoma i generiranog transkriptoma.

Na sljedećoj slici prikazan je zapis jednog generiranog transkripta. U prvoj liniji je njegovo ime i oznaka *'A'* koju smo dogovorom odredili kao oznaku za alternativno prekrajanje. Sufiks u imenu nakon znaka '_' služi kao oznaka podskupa eksona koji je uzet u tom alternativnom prekrajanju.

```
>YAL001C_10 A
TTCCCTTATTTGAAGCAATTTTATCAGACACTATTTGTACGAGTTCGTCAGGATAAATCG
TCAGTACCAT
```

Slika 5.2: Transkript u generiranoj FASTA datoteci

5.3. SAM format

SAM format koristi se za prikaz poravnanja. Prije linija koje opisuju poravnanje, mogu se pojaviti linije zaglavlja koje započinju znakom '@'. Svaka linija poravnanja ima 11 obaveznih polja [9]. Kratak opis obaveznih polja dan je u nastavku:

QNAME: ime očitanja

FLAG: zastavica — cijeli broj veličine 12 bita. Neki bit je postavljen ako vrijedi svojstvo koje je označeno tim bitom. Slijedi značenje pojedinog bita[3]:

[2] `https://en.wikipedia.org/wiki/FASTA_format`

[3] Pojedini bitovi naznačeni su potencijom broja 2 u heksadekadskom brojevnom sustavu.

- *0x1*: postoji više segmenata u sekvenciranju.
- *0x2*: svaki segment je pravilno poravnan.
- *0x4*: sekvenca nije mapirana.
- *0x8*: sljedeći segment ove sekvence nije mapiran.
- *0x10*: polje *SEQ* je reverzni komplement.
- *0x20*: polje *SEQ* sljedećeg segmenta je reverzni komplement.
- *0x40*: prvi segment u sekvenci.
- *0x80*: zadnji segment u sekvenci.
- *0x100*: sekundarno poravnanje.
- *0x200*: ne prolazi provjeru kvalitete.
- *0x400*: duplicirano poravnanje.
- *0x800*: dopunsko poravnanje.

RNAME: ime reference na koju je poravnato očitanje. Ako je vrijednost ovog polja '*' očitanje nije poravnato.

POS: pozicija prve baze poravnanja na referenci. Pozicije počinju od 1, a ako je ovo polje 0, tada se radi o očitanju bez koordinate koje nije poravnato.

MAPQ: Kvaliteta mapiranja koja se računa prema formuli:
$-10 \log_{10} P\{Pozicija\ mapiranja\ je\ kriva\}$
gdje je *P* oznaka za vjerojatnost, a konačna vrijednost se zaokružuje na najbliži cijeli broj.

CIGAR: niz znakova sastavljen od operacija i broja ponavljanja operacije. Broj ponavljanja prethodi oznaci operacija, a između operacija nema nikakvog posebnog razdjelnika. Valjane operacije su sljedeće:

- *M*: slaganje poravnanja (može biti ista baza ili neka druga).
- *I*: umetanje u odnosu na referencu.
- *D*: brisanje u odnosu na referencu.
- *N*: preskočena regija u referenci.
- *S*: engl. *soft clipping* — podrezivanje očitanja s time da je podrezani dio prisutan u polju *SEQ*.
- *H*: engl. *hard clipping* — podrezivanje očitanja s time da podrezani dio nije prisutan u polju *SEQ*.

– *P*: engl. *padding* — punjenje koje se može protumačiti kao tiho brisanje iz reference.
– =: baze su jednake u očitanju i referenci.
– *X*: baze nisu jednake u očitanju i referenci.

RNEXT: ime reference primarnog poravnanja sljedećeg očitanja u istom predlošku.

PNEXT: pozicija primarnog poravnanje sljedećeg očitanja u istom predlošku.

TLEN: ukupna duljina mapiranih segmenata u istom predlošku. Prvi segment u predlošku ima predznak +, a zadnji . Za segmente između nije defeniran predznak. Ako predložak ima jedan segment, vrijednost ovog polja je 0. Isto vrijedi ako ova informacija nije dostupna.

SEQ: sekvenca koja predstavlja očitanje. Ako sekvenca nije pohranjena, vrijednost polja je *'*'*, inače duljina sekvence mora biti suma duljina operacija *M/I/S/=/X* operacija u polju *CIGAR*.

QUAL: kvaliteta očitanja. Vrijednost za svaku bazu je dana sljedećom formulom:
$ASCII(-10\log_{10} P\{Baza\ je\ pogresna\} + 33)$
gdje $P\{Baza\ je\ pogresna\}$ označuje vjerojatnost da je baza pogrešna, a $ASCII(broj)$ funkcija koja zaokružuje $broj$ na najbliži cijeli broj i pretvara ga u znak po *ASCII* tablici [4].

Na sljedećoj je slici primjer SAM retka dobivenog alatom *GraphMap*. Prikazuje očitanje imena *Sim1_S1390_4* koje je uspješno poravnato kao reverzni komplement na referencu imena chrXVI, od pozicije 528985. Linija je razlomljena u više redaka zbog veličine prozora u kojem je ispisana.

```
Sim1_S1390_4 16 chrXVI 528985 40 4S1D5M1I4M2D6M1D8M1D8M1I11M1I44M1I9M1I26M1I36M1I1
4M1I29M1D4M1I17M1D1M1D15M1D26M1I28M1I26M1I6M1D17M * 0 0 TCACTCACTATTGGTTCCATCTTTCA
AAACAACTTTACCTATTCACAAGTGCTGAACTTACTAGTGCCCCATGCATCATTTAATTTACCAGCCCTTTCGTAAACAGTT
TAGCAACTGTTTCGCTAGGTTTATGCTCCCATACCAATCCAATATATTGTTCTATCAAACACCAAGCTTCACGTTCTTTAAT
AGGGACAGCCCCCCCCTTCAATCTCTTCTTGGTCACTGGGCTGGTCCGCCACGTAGGTTTTAGTACCTCGATCGGTTTAGAA
TCCCCTCGCCTTCCGGGAAATAGCCACTACGATTATCATACAGCTGACTAATTTTTCTTCGCAGATCAGCTAAACTTAGTAG
TA ./-&#&&,#'"+%-..,/-,,'++++%..,...,.'."-*-.--).."'.,-..$+),'*.*%$%/)++.-.-...-.)
.-...-.)(-...+/.,*()-&..&#.+.--/,+.++...+-.-%.-.,.(%*,+..&-."&.,...--.,+.*$(,-'...
+,-.-,.&-.-+-,-,%-(-#/*,-%,./-.+$.+"&)'.,.,%-&...,.&,+-*,/#,.'&*.-++&.%.-/+.,*,(.-
--.-.+.+&..-(.,-,/.-)..--++#.-.-&(,,.,&"*"..%..-.-&#,+$/.,+-.*-,.#..,#&&+%!,.'$".-
+-.,--)))-....'(---.&.',.#-.+.,
```

Slika 5.3: Linija SAM datoteke

[4] American Standard Code for Information Interchange

6. Rezultati

Svi rezultati dobiveni su prilikom ispitivanja sa genomom organizma *Saccharomyces cerevisiae*. Taj organizam se često koristi prilikom ispitivanja jer se lagano uzgaja, razdvaja se mejozom i s obzirom da je eukariot sadrži strukturu stanice sličnu životinjama i biljkama bez velikog postotka nekodirajućih regija DNA.

6.1. Analiza transkriptoma

Kako je već spomenuto, transkriptom je generiran iz genoma organizma S. cerevisiae. Genom tog organizma sastavljen je od 16 kromosoma. Mjerene duljine, dane su tablicom 6.1. Ukupna duljina transkriptoma i genoma je dovoljno mala da se može analizirati u relativno kratkom vremenu na današnjem prosječnom računalu. Za usporedbu, duljina ljudskog genoma je oko 3234.83 Mb[1] što je otprilike 300 puta dulje od genoma S. cerevisiae.

Tablica 6.1: Transkriptom

Sekvenca	Duljina (bp)
Genom	11123260
Transkriptom	9024021
Najdulji transkript	14733
Najkraći transkript	51
Prosječni transkript	1354
Medijan transkripata	1080

[1] Mega-basepairs

Prema tablici 6.2 od ukupno 6664 transkripata, otprilike pola je na + lancu, a pola na − lancu genoma S. cerevisiae.

Tablica 6.2: Transkripti

Svi transkripti	6664
Transkripti na + lancu	3357
Transkripti na − lancu	3307

6.2. Analiza očitanja

Očitanja su dobivena simulacijom alatom *PBSIM*. PBSIM je namijenjen za simuliranje očitanja dobivenih PacBio sekvenciranjem. Napravljena su 3 skupa podataka. Za generiranje svakog skupa nasumično je uzeto 20% transkriptoma. Skupovi se razlikuju po pokrivenosti svakog očitanja.

Duljina simuliranih očitanja određena je sljedećim parametrima i njihovim vrijednostima:

length-mean: srednja duljina očitanja — 9753.

length-sd: standardna devijacija duljine — 4260.

length-min: minimalna duljina — 5.

length-max: maksimalna duljina očitanja — 100000.

Točnost simuliranih očitanja određena je sljedećim parametrima i njihovim vrijednostima:

accuracy-mean: srednja točnost — 0.9.

accuracy-sd: standardna devijacija točnosti — 0.05.

accuracy-min: minimalna točnost — 0.7.

difference-ratio: omjer grešaka koje se pojavljuju — 50:30:20. Tri su tipa grešaka (poredak odgovara poretku u prethodnom omjeru):

- Zamjena baza
- Umetanje u odnosu na referencu
- Brisanje u odnosu na referencu

Na taj su način generirani *Skup1*, *Skup2* i *Skup3* iz tablice 6.3.

Tablica 6.3: Očitanja

Ime	Pokrivenost	Broj očitanja
Skup1	5	8340
Skup2	50	69203
Skup3	20	26636

6.3. Analiza rezultata poravnanja

Broj poravnatih očitanja u odnosu na ukupni broj očitanja za prethodno navedene skupove dan je tablicom 6.4. Postotak poravnatih očitanja za sva 3 skupa je iznad 90%. Za svaki skup izvršeno je više mjerenja u kojima se mijenjao parametar koji

Tablica 6.4: Odnos poravnatog i ukupnog broja očitanja

Ime	Broj očitanja	Poravnata očitanja	Kvocijent
Skup1	8340	7706	0,924
Skup2	69203	67860	0,981
Skup3	26636	26325	0,988

dopušta odstupanja od početne pozicije i krajnje pozicije poravnanja na referenci. Prilikom ispitivanja, ustanovljeno je da skripte koje ocjenjuju rezultat poravnanja imaju neke greške u računanju pozicija pa se mijenjanjem odstupanja dobiva realnija slika o točnosti poravnanja. Korišteno odstupanje je iz skupa $0, 1, 3, 5$ tako da poravnata očitanja uhvaćena tim odstupanjem zbilja pogađaju dobar gen, odnosno regiju na referenci. Tablice za svaki skup koje prikazuju broj uspješno poravnatih i broj krivo poravnatih očitanja za različita odstupanja, dane su u nastavku.

Tablica 6.5: Poravnanje očitanja za Skup1

Dopušteno odstupanje	Točno poravnatih	Krivo poravnatih
0	5005	2646
1	6320	1331
3	6789	862
5	6975	676

Tablica 6.6: Poravnanje očitanja za Skup2

Dopušteno odstupanje	Točno poravnatih	Krivo poravnatih
0	43508	23907
1	54970	12445
3	59236	8179
5	61029	6386

Tablica 6.7: Poravnanje očitanja za Skup3

Dopušteno odstupanje	Točno poravnatih	Krivo poravnatih
0	16988	9154
1	21397	4745
3	23056	3086
5	23768	2374

Tablica 6.8 prikazuje udio dobro poravnatih očitanja u ukupnom broju očitanja za pojedine skupove i dopuštena odstupanja. Sve vrijednosti su postoci. Točnost je već za manja dopuštena odstupanja iznad 80%. To je dokaz da bi ovakva metoda mogla biti dobar način poravnanje RNA očitanja na poznate gene. Metode ispitivanja bi trebalo poboljšati kako bi se dobila još bolje slika o točnosti ove metode.

Tablica 6.8: Točnost poravnatih očitanja za različita odstupanja u postocima

Ime skupa	0	1	3	5
Skup1	60	76	81	84
Skup2	63	79	86	88
Skup3	64	80	87	89

7. Zaključak

Bioinformatika je područje u kojem ima prostora za veliki napredak. Još uvijek nisu nađeni lijekovi za razne bolesti, a u tome nam mogu pomoći upravo metode korištene u bioinformatici. RNA sekvenciranjem dobivamo uvid u stanje stanica u trenutku sekvenciranja. Stanja u različitim trenucima mogu se usporediti i naći se uvjeti koji su odgovorni za razvijanje nekih bolesti ili mehanizmi koji su odgovorni za obranu od tih bolesti.

Veliki problem sekvenciranja općenito je cijena koja je zadnjih godina drastično smanjena. No, smanjenjem cijene dolaze i veći postoci pogrešaka. Neki očekivani cilj je da će u budućnosti bilo tko moći sekvencirati bilo što u kratkom vremenu po pristupačnoj cijeni. Tada bi se sekvenciranje moglo koristiti na dnevnoj bazi u bolničkim ustanovama za postavljenje bržih i boljih dijagnoza.

Dolaskom novih metoda sekvenciranja, poput Oxford Nanopore MinIONa, cijena sekvenciranja pala je na nekoliko tisuća američkih dolara. To omogućuje brži razvoj istraživanja koja se temelje na sekvenciranju jer sekvenciranje postaje dostupno manjim ustanovama i timovima koji prije nisu imali mogućnosti ili sredstva za tako nešto. Da bi te nove metode sekvenciranja bile iskoristive, potrebno je i razviti nove metode poravnanja. Jedan takav alat je *GraphMap* koji je posebno napravljen kako bi mogao poravnati podatke dobivene Oxford Nanopore MinIONom s velikom točnošću. Ovaj rad dokazuje kako bi metoda koja koristi *GraphMap* za poravnanje RNA očitanja na transkriptom i zatim translatira te rezultate na genom, mogla biti jedno rješenje poravnanja RNA očitanja na poznate gene. U budućnosti bi još trebalo razviti bolju metodu ispitivanja, odnosno poboljšati trenutno korištenu.

LITERATURA

[1] Ji Hanlee Shendure Jay. Next-generation dna sequencing. *Nat Biotech*, 26, 2008.

[2] Francesca Finotello and Barbara Di Camillo. Measuring differential gene expression with rna-seq: challenges and strategies for data analysis. *Briefings in Functional Genomics*, 14(2):130–142, 2015. doi: 10.1093/bfgp/elu035. URL `http://bfg.oxfordjournals.org/content/14/2/130.abstract`.

[3] Ivan Sovic, Mile Sikic, Andreas Wilm, Shannon Nicole Fenlon, Swaine Chen, and Niranjan Nagarajan. Fast and sensitive mapping of error-prone nanopore sequencing reads with graphmap. *bioRxiv*, 2015. doi: 10.1101/020719. URL `http://biorxiv.org/content/early/2015/06/10/020719`.

[4] Corey David R. Chu Yongjun. Rna sequencing: Platform selection, experimental design, and data interpretation. *Nucleic Acid Therapeutics*, 22, 2012.

[5] Jerković Igor. Rna-seq mapper. Master's thesis, Faculty of Electrical Engineering and Computing, University of Zagreb, 2014.

[6] Ashlee M. Benjamin, Marshall Nichols, Thomas W. Burke, Geoffrey S. Ginsburg, and Joseph E. Lucas. Comparing reference-based rna-seq mapping methods for non-human primate data. *BMC Genomics*, 15(1):1–14, 2014. ISSN 1471-2164. doi: 10.1186/1471-2164-15-570. URL `http://dx.doi.org/10.1186/1471-2164-15-570`.

[7] Wang Zhimin Wang Yue, Yang Qiuping. The evolution of nanopore sequencing. *Frontiers in Genetics*, 5, 2014.

[8] Hayden Erika Check. Technology: The $1000 genome. *Nature*, 507, 2014.

[9] *Sequence Alignment/Map Format Specification*. SAMtools, 2015. URL `https://samtools.github.io/hts-specs/SAMv1.pdf`.

Poravnanje RNA očitanja na poznate gene

Sažetak

RNA sekvenciranje je metoda za analizu transkriptoma. Analizom transkriptoma u nekom trenutku dobivamo uvid u stanje organizma i mogućnost identificiranja regija genoma koje su odgovorne za različite funkcije. U ovom radu je prikazano proširenje alata *GraphMap* u svrhu poravnanja RNA očitanja na poznate gene. Osim toga, opisani su osnovni mehanizmi u sekvenciranju nanoporama, te neke druge tehnologije korištene za sekvenciranje i poravnanje. Dan je i pregled formata datoteka koje su korištene za spremanje anotacija eksonski regija, spremanje genoma i transkriptoma i spremanje podataka o poravnanjima.

Ključne riječi: RNA, poravnanje, GraphMap, Geni, Genom, Transkriptom, Nanopore, SAM, GTF, FASTA, bioinformatika

RNA-seq alignment using existing gene annotation

Abstract

RNA sequencing is a method used for transcriptome analysis. Transcriptome analysis provides us with insight into the state of cells. There is a possibility of identifying regions of a genome in order to determine which regions are responsible for which functions and which regions are active in a state of disease. This thesis describes an upgrade to *GraphMap* alignment tool that should make alignment of RNA reads using existing gene annotation possible. Besides the mentioned upgrade, thesis describes different technologies used in sequencing, other alignment tools and file formats used to store information about genome and transcriptome.

Keywords: RNA, alignment, GraphMap, Genes, Genome, Transcriptome, Nanopore, SAM, GTF, FASTA, bioinformatics